Machine Learning

Amin Golzari Oskouei

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

https://github.com/Amin-Golzari-Oskouei

Azarbaijan Shahid Madani University
2023

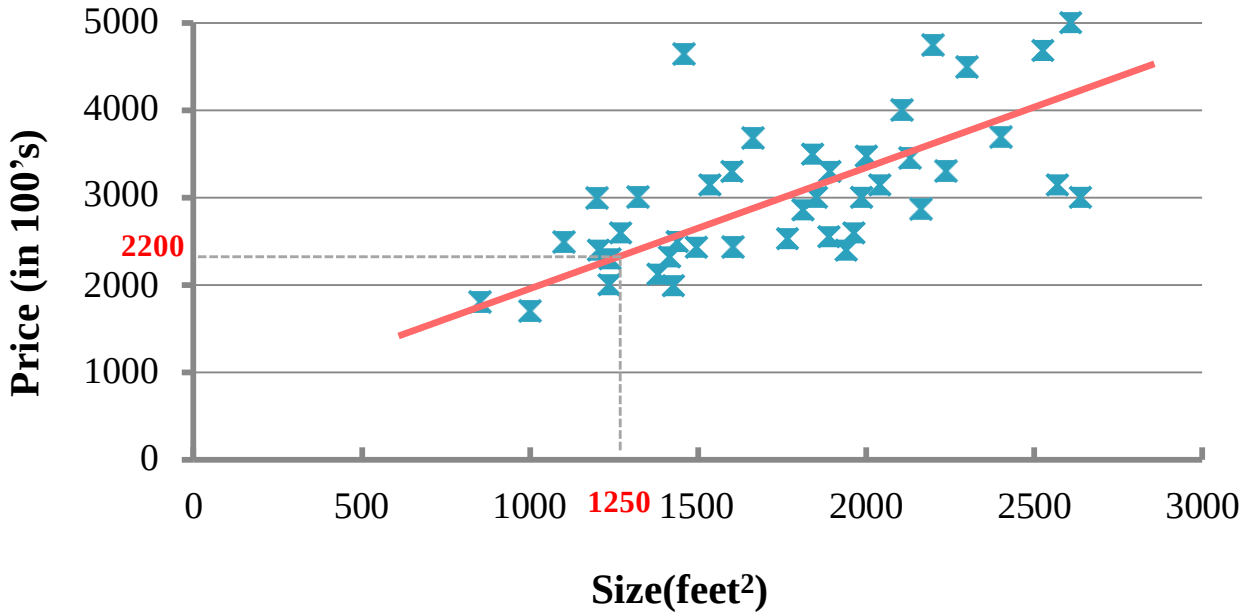
Supervised learning: Regression

Table of Contents

- Regression
  - Univariate and multivariate linear regression
- Gradient descent
- Normal equation
- Regression with local weighting
- Probabilistic interpretation of regression
- Maximum likelihood estimation.

Univariate linear regression

Home pricing



☐Supervised learning

For each Train set example, the correct answer is given (having Lable)

☐Regression

Predicting quantities with continuous values (such as the price of a house)

symbolization

Train Set

$m = 47$

Area (square feet) (x)	Price (in 1000 dollars) (y)
2104	460
1416	232
1534	315
852	178
….	….

Symbols

m = Number of training samples

x = Input variable, features

y = Output variable, target variable

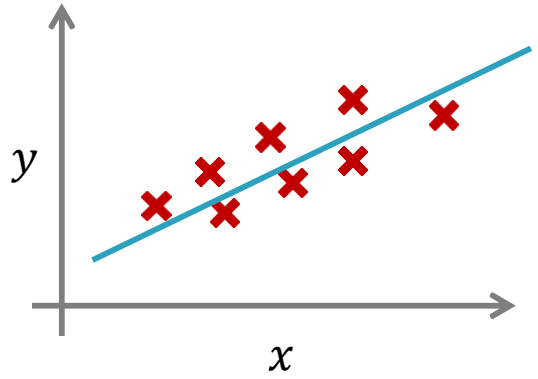
(x,y): an trainset sample

(x^{i}, y^{i}): i-th sample of train set

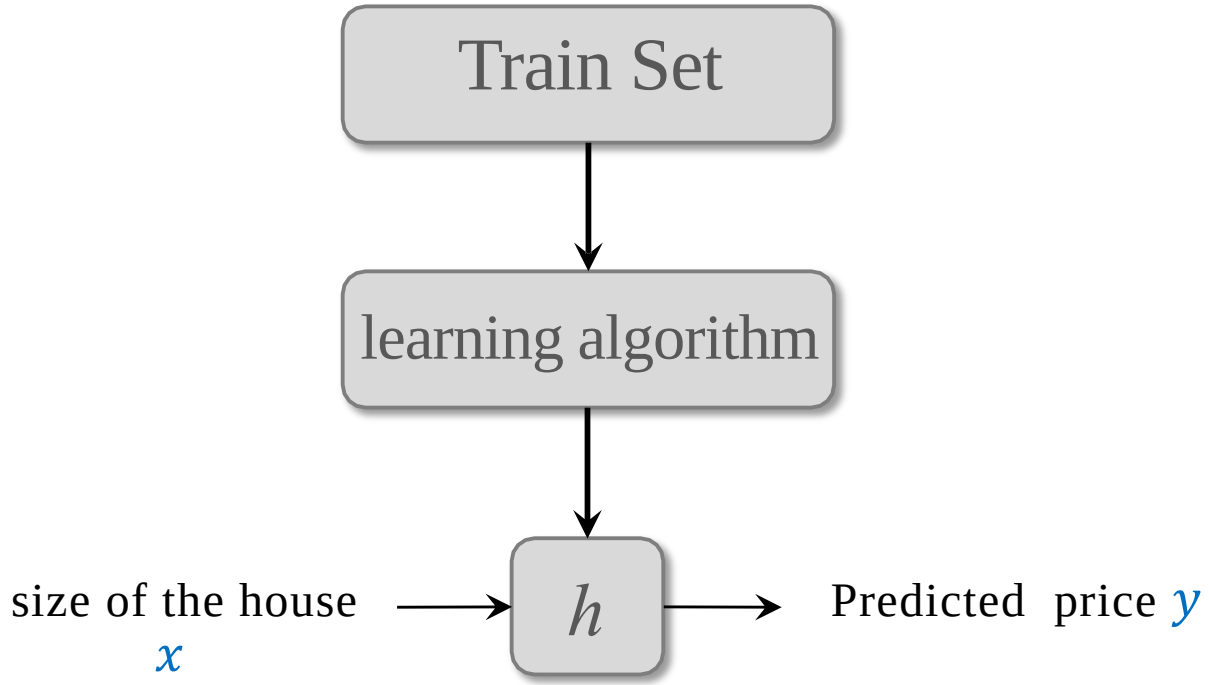
Model representation

Show hypothesis h

$$h_\theta(x) = \theta_0 + \theta_1 x$$



Univariate linear regression



h : $x \rightarrow y$

Cost function

Hypothesis evaluation

Train Set

$m = 47$

Area (square feet) (x)	Price (in 1000 dollars) (y)
2104	460
1416	232
1534	315
852	178
….	….

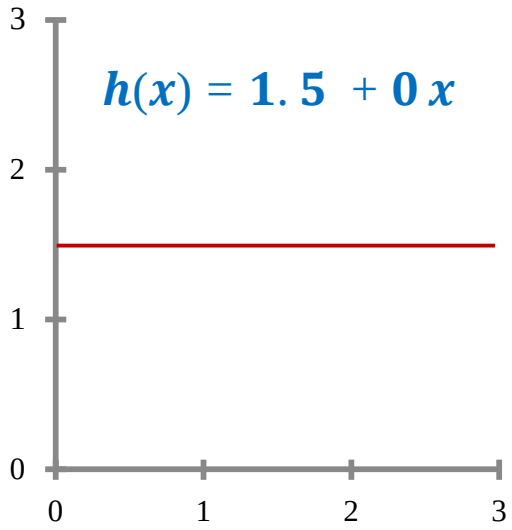
Hypothesis : $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters : (θ_0 , θ_1)

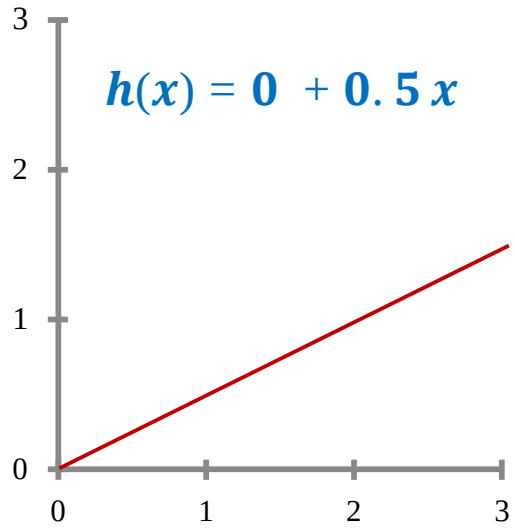
Question: How to choose the value of the parameters?

Hypothesis evaluation

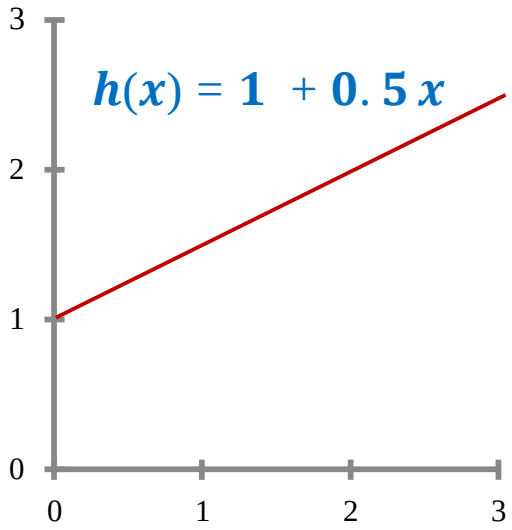
$$h_\theta(x) = \theta_0 + \theta_1 x$$



$\theta_0 = 1.5$
$\theta_1 = 0$



$\theta_0 = 0$
$\theta_1 = 0.5$

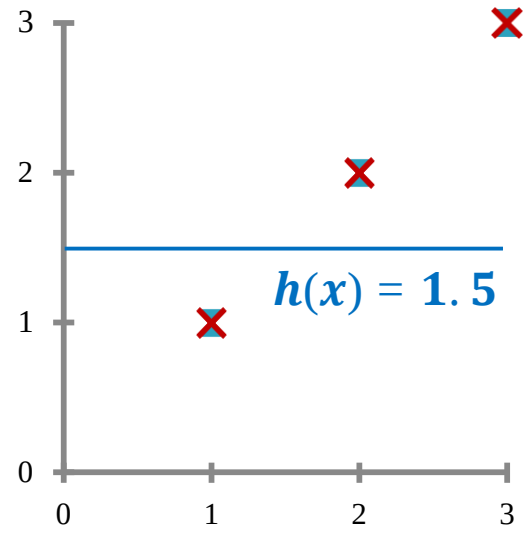


$\theta_0 = 1$
$\theta_1 = 0.5$

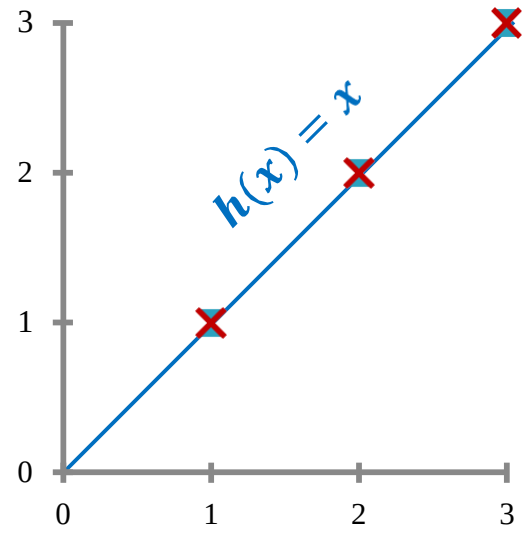
Hypothesis evaluation

Question: Which hypothesis is better?

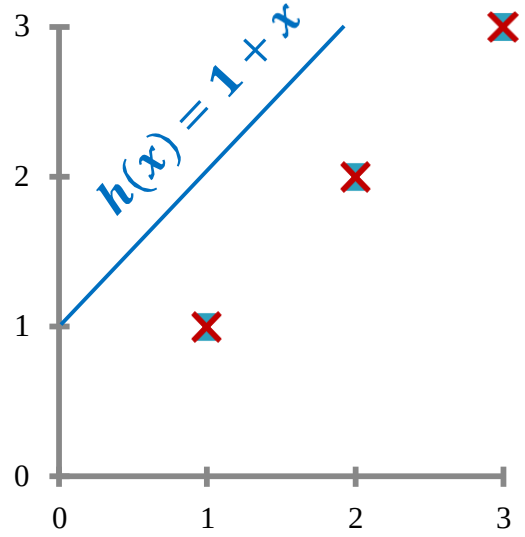
$$h_\theta(x) = \theta_0 + \theta_1 x$$



$\theta_0 = 1.5$
$\theta_1 = 0$



$\theta_0 = 0$
$\theta_1 = 1$



$\theta_0 = 1$
$\theta_1 = 1$

Cost function

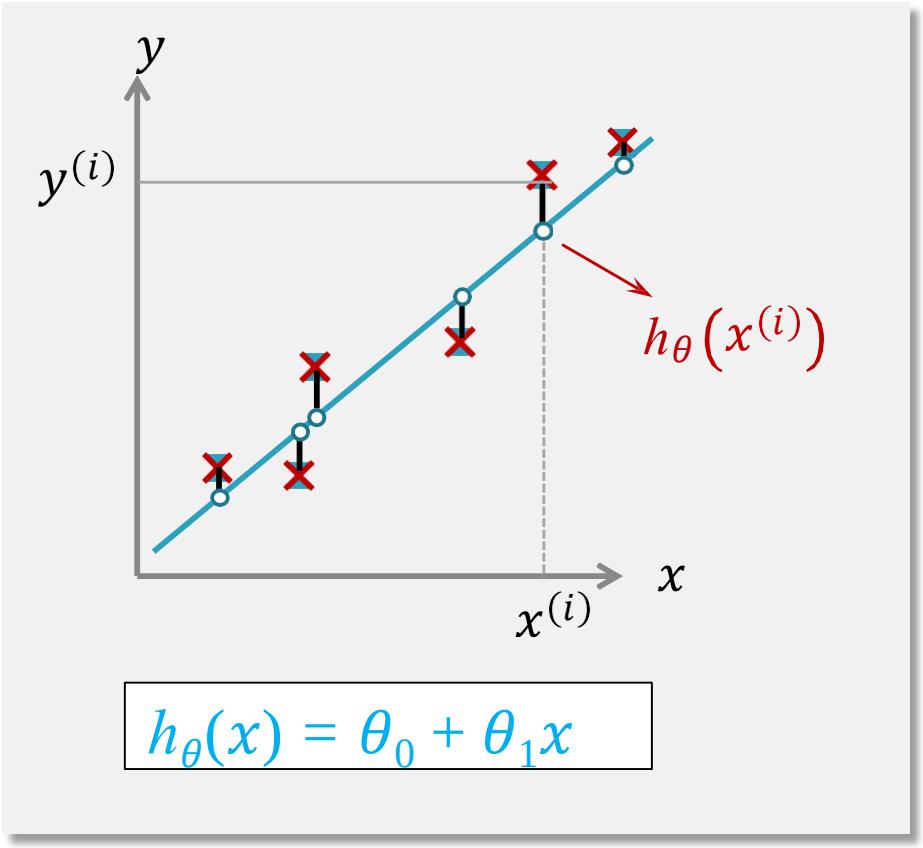
☐Ideas. Choosing the parameters so that for each training sample such as x,y, the value of $h_\theta(x)$ is as close as possible to the value of y .

☐cost function, sum of squared error

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

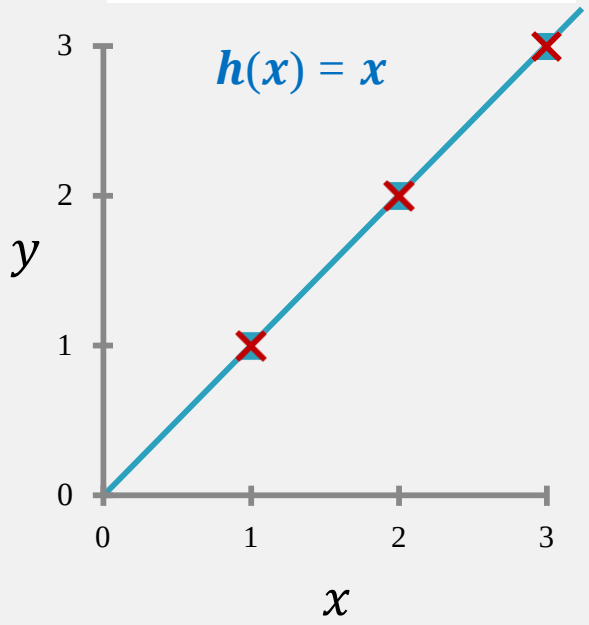
☐purpose

$$\underset{\theta_0,\theta_1}{\text{minimize}} \quad J(\theta_0, \theta_1)$$



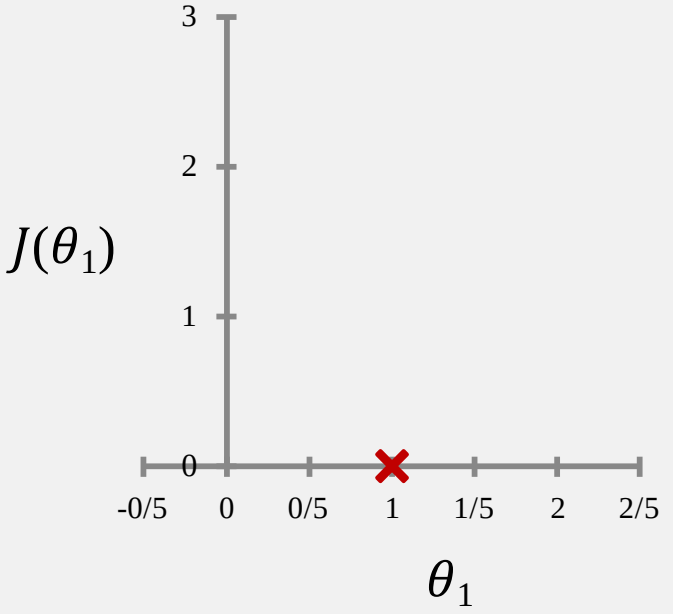
Simplified cost function

$h_\theta(x) = \theta_1 x$



$\theta_1 = 1$

$J(\theta_1)$



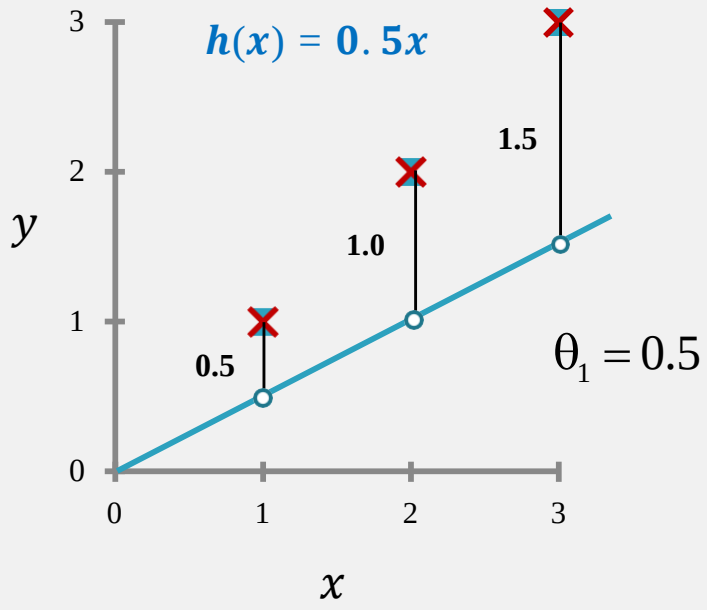
$$J(\theta_0, \theta_1) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}(x^{(i)} - y^{(i)})^2 = \frac{1}{2}(0^2 + 0^2 + 0^2) = 0$$

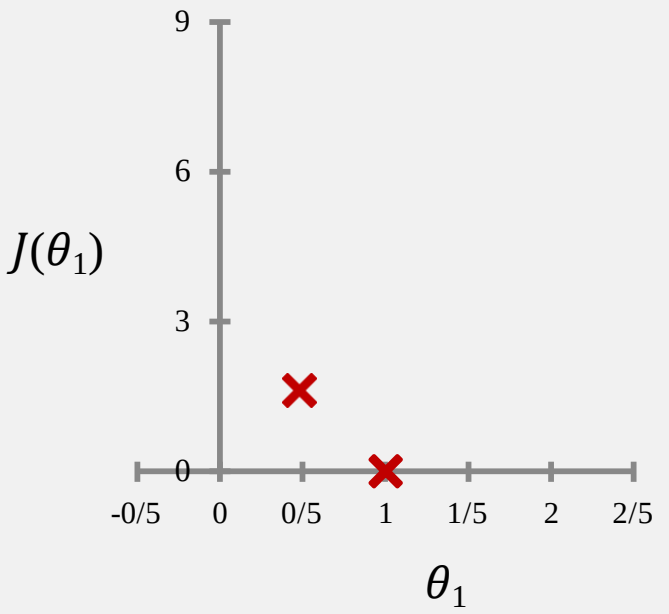
$J(1) = 0$

Cost function

$h_\theta(x) = \theta_1 x$



$J(\theta_1)$

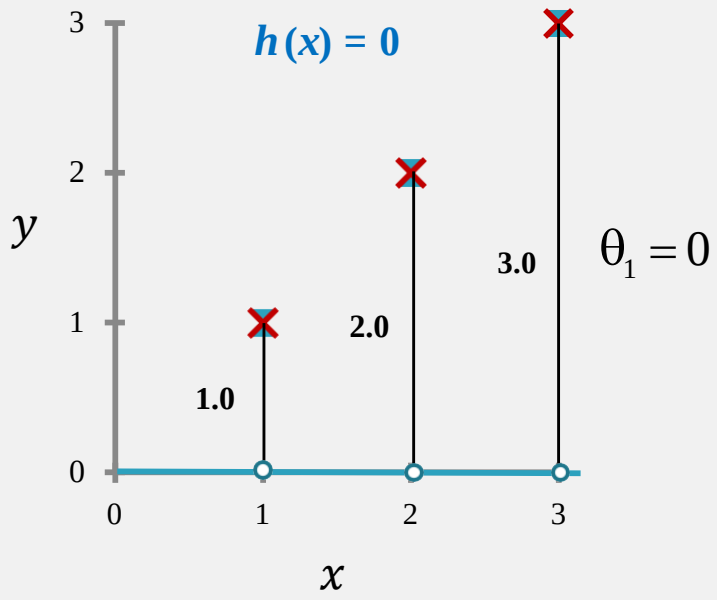


$J(0.5) = \frac{1}{2}(0.5^2 + 1^2 + 1.5^2) = \frac{1}{2}(3.5) = 1.75$

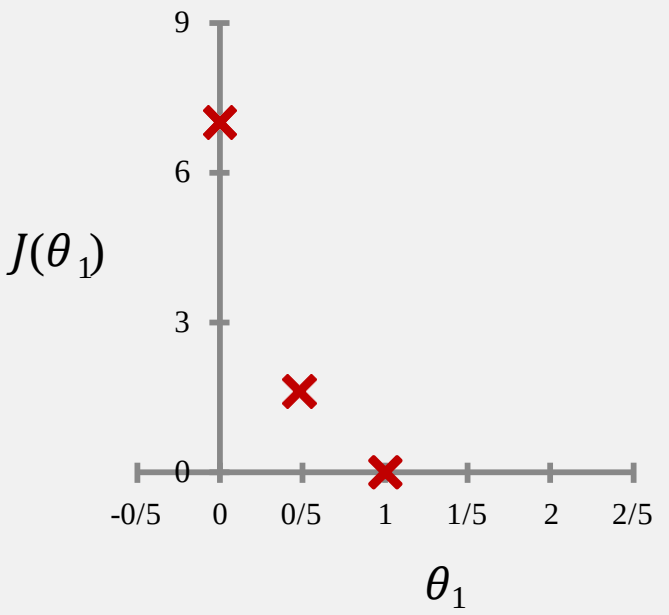
$J(0.5) = 1.75$

Cost function

$h_\theta(x) = \theta_1 x$



$J(\theta_1)$

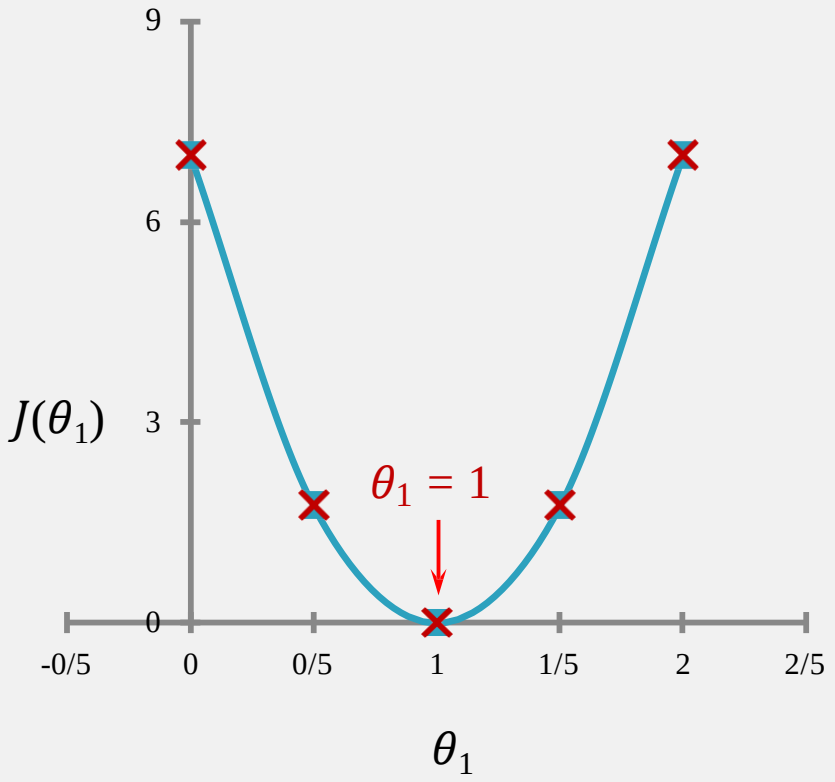


$J(0) = \frac{1}{2}(1^2 + 2^2 + 3^2) = \frac{1}{2}(14) = 7.0$

$J(0) = 7.0$

Cost function

$$\underset{\theta_1}{\text{minimize}} \quad J(\theta_1)$$



Univariate linear regression

- hypothesis

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- parameters

$$\theta_0, \theta_1$$

- Cost functions

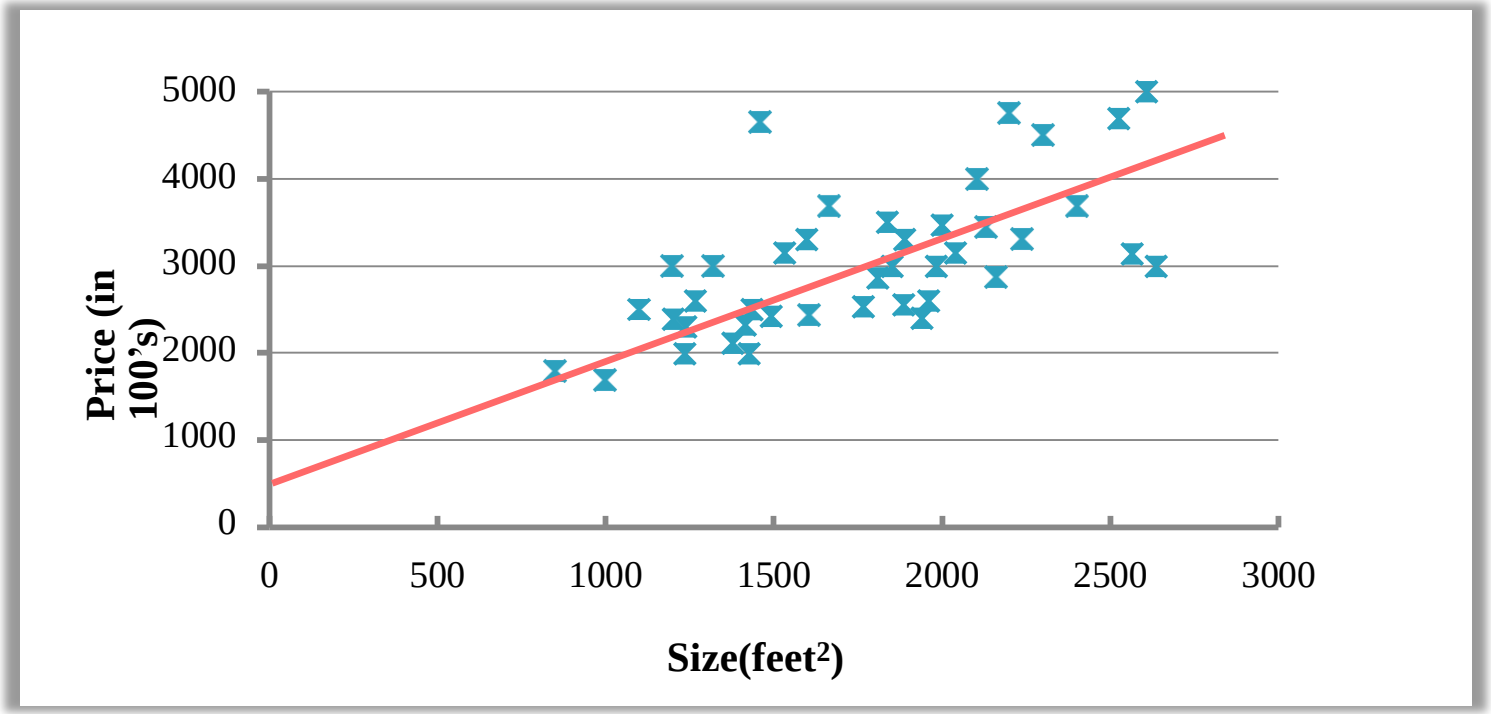
$$J(\theta_0, \theta_1) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

- purpose

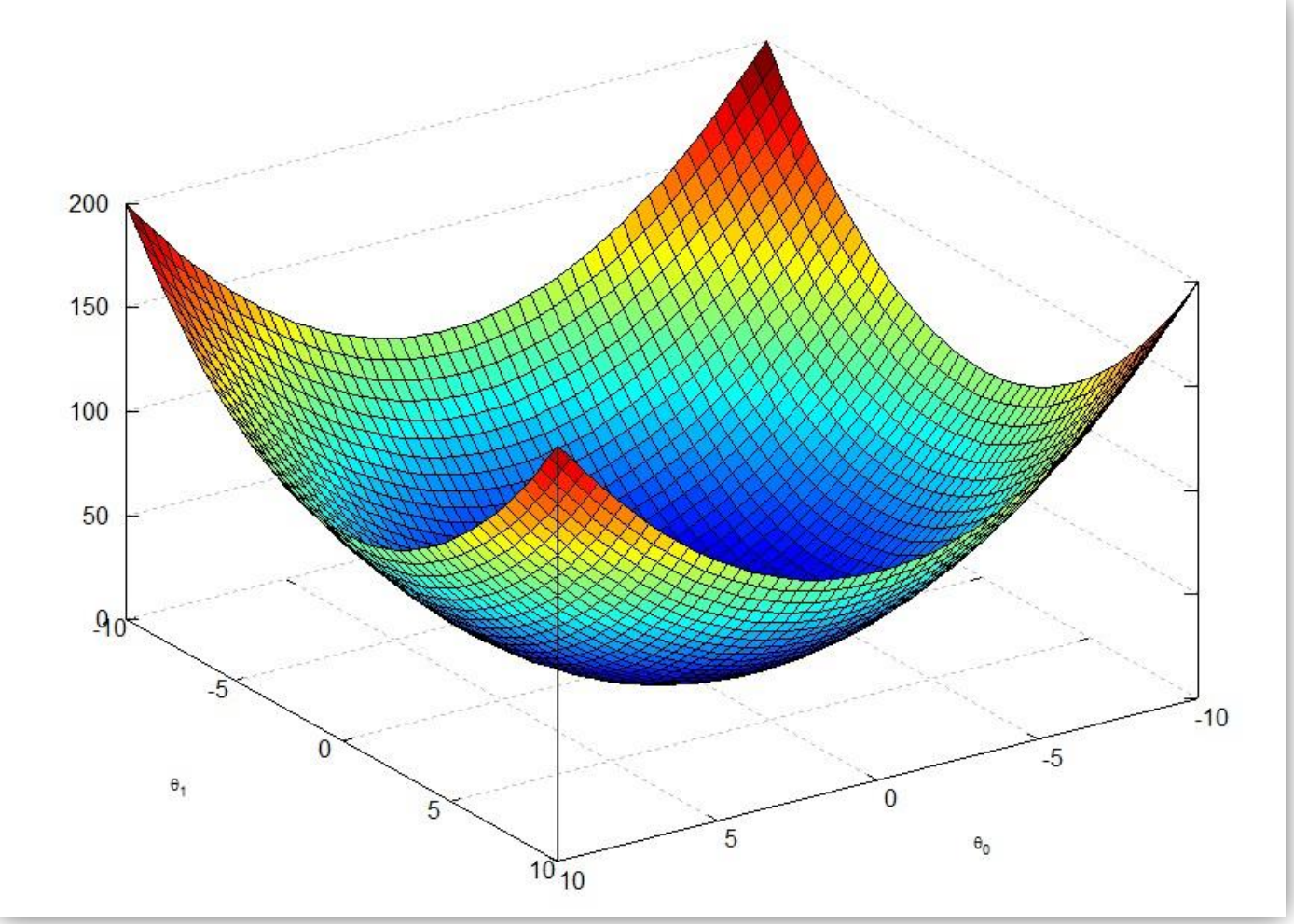
$$\underset{\theta_0,\theta_1}{\text{minimize}} \quad J(\theta_0, \theta_1)$$

Example: home pricing

$$h_\theta(x) = \theta_0 + \theta_1 x$$



Cost function

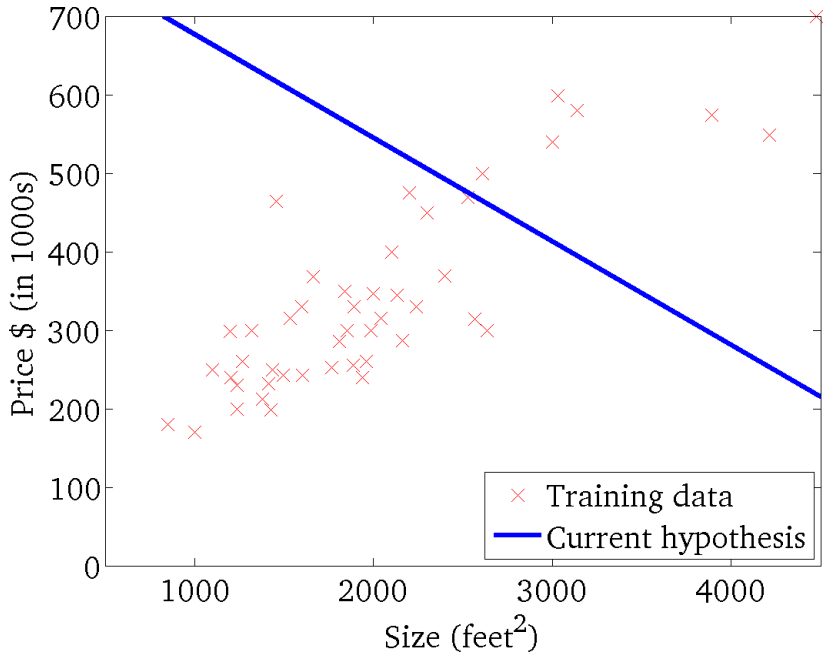


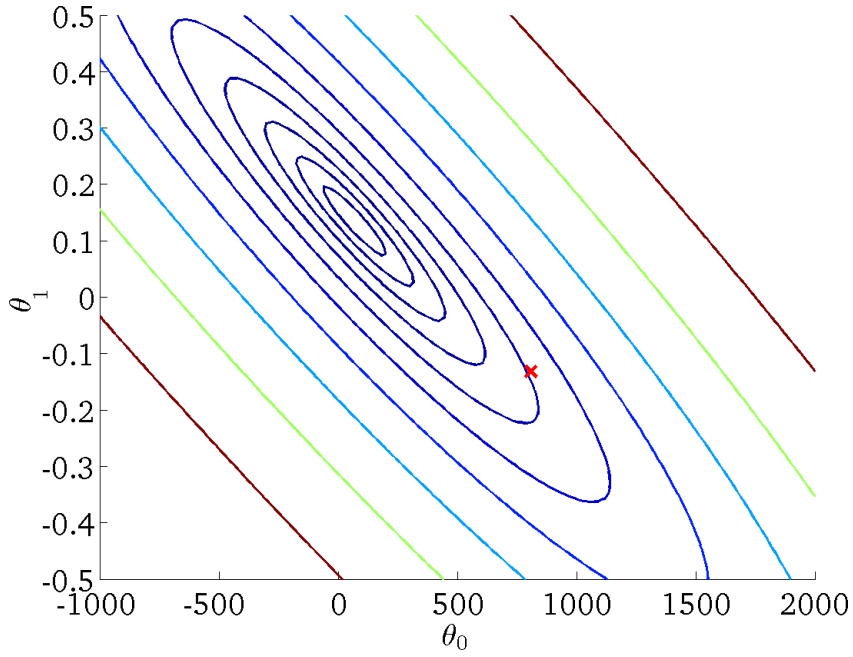
$$x = -10:0.5:10;$$
$$y = -10:0.5:10;$$
$$[X\,Y] = meshgrid(x, y);$$
$$Z = X.^2 + Y.^2;$$
$$surf(X, Y, Z);$$

Cost function: contour curve

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

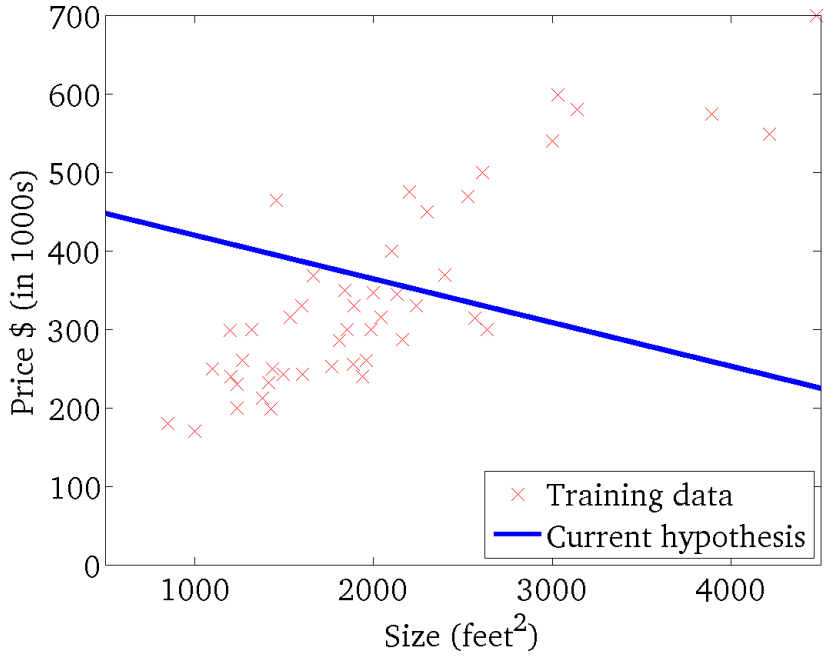


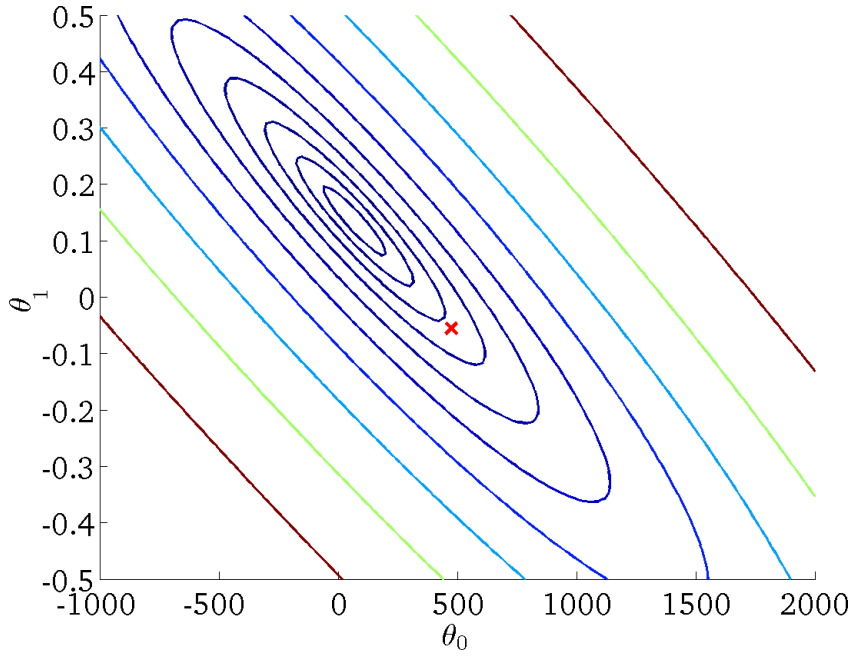


Cost function: contour curve

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

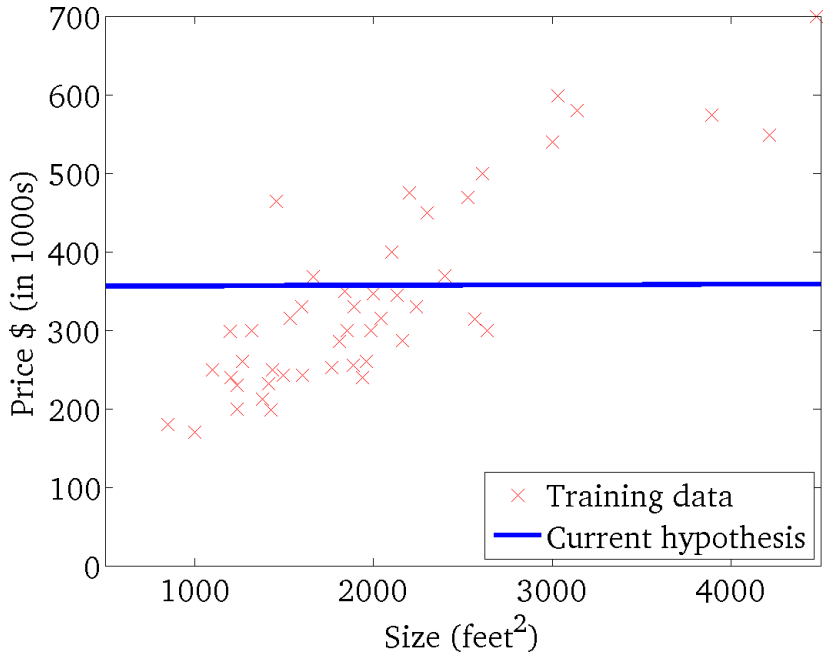


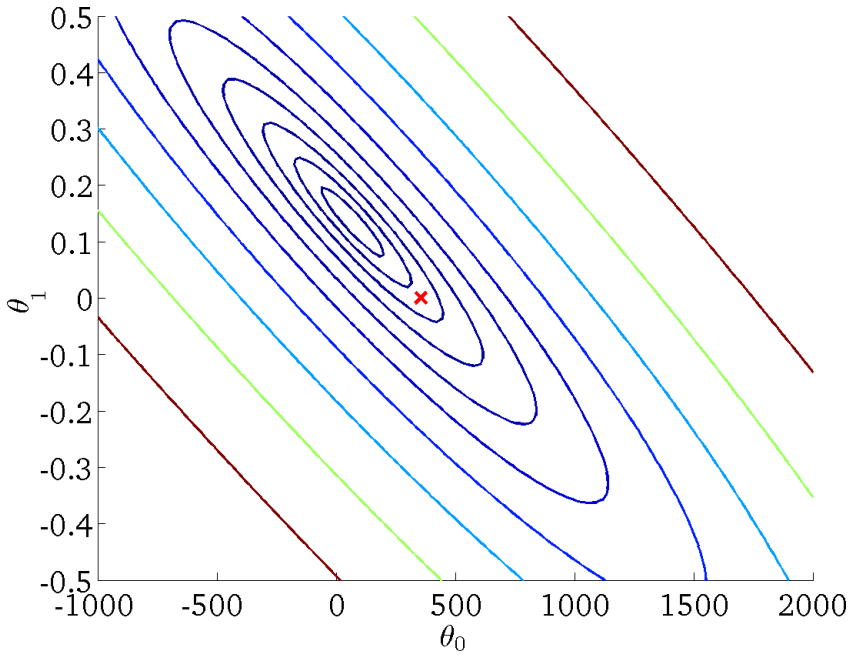


Cost function: contour curve

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

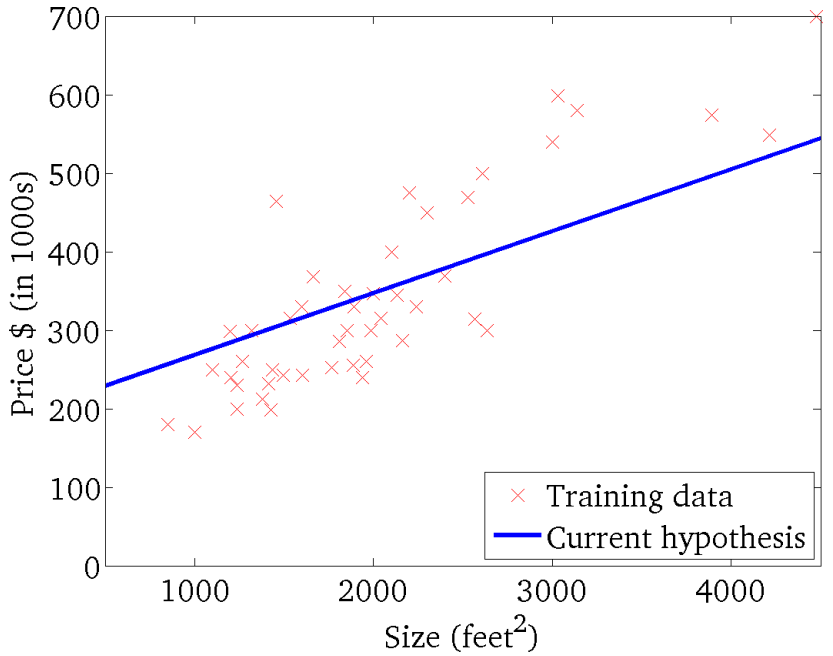


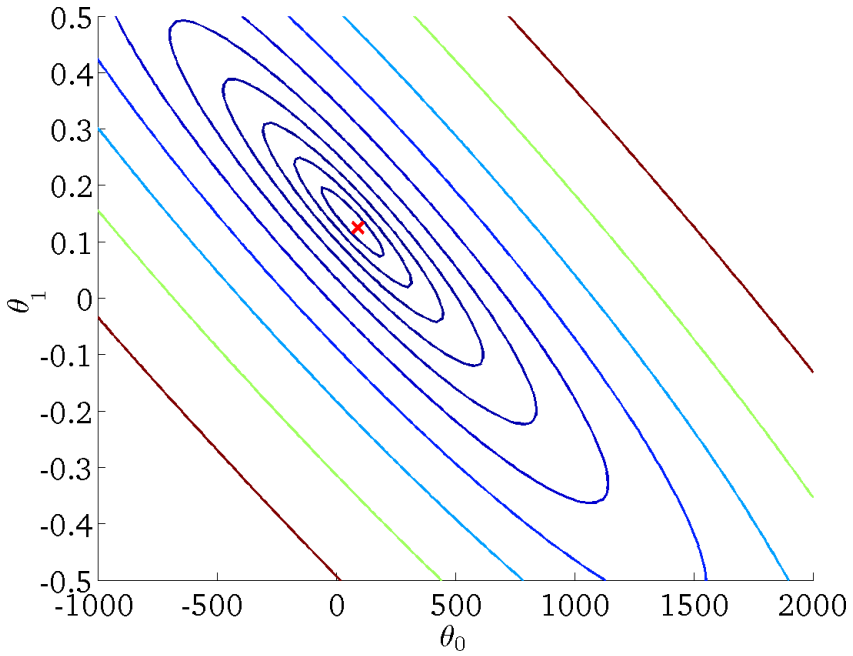


Cost function: contour curve

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$





Gradient Descent

Question

☐hypothesis

$$J(\theta_0, \theta_1)$$

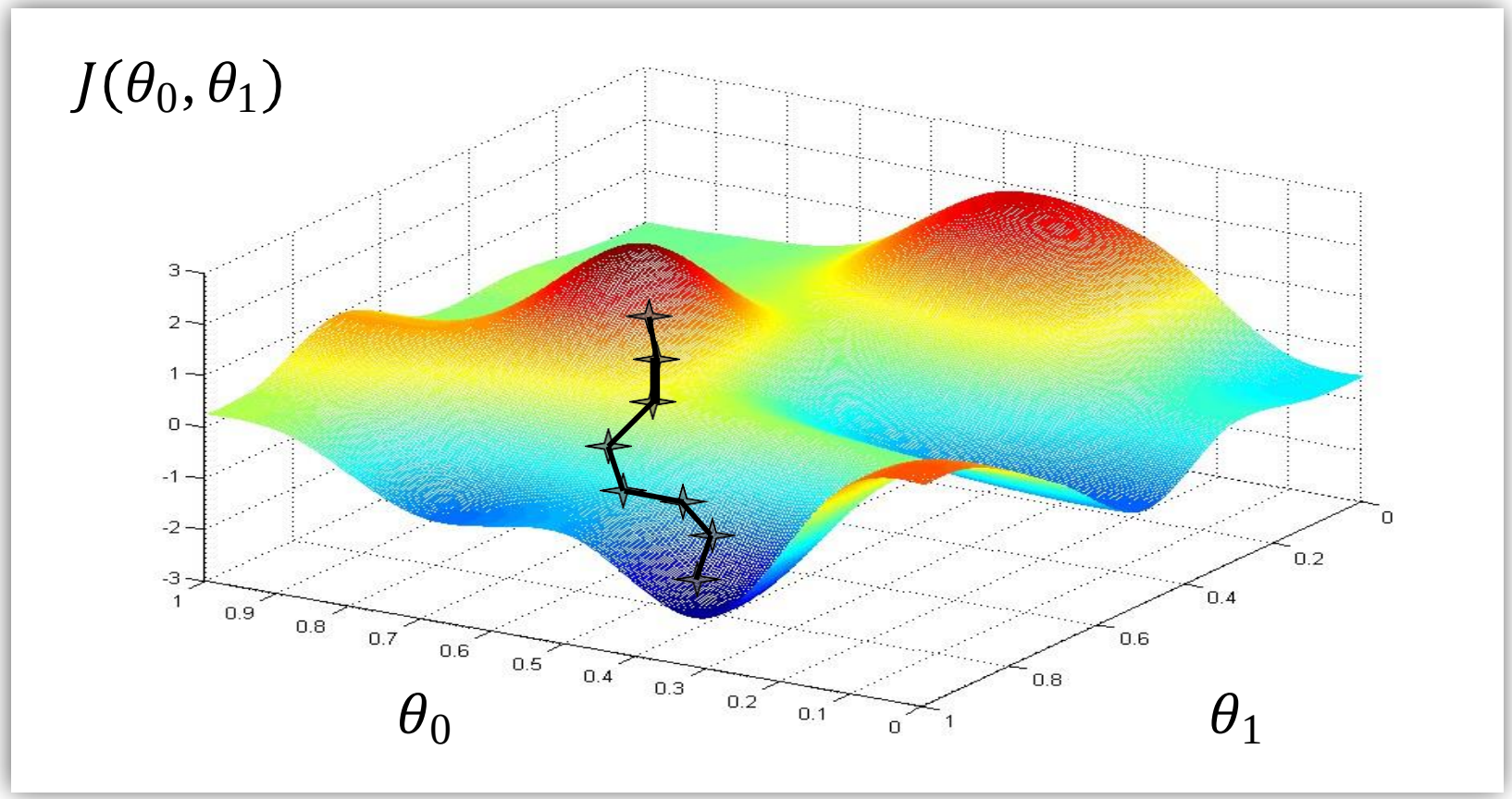
☐purpose

$$\underset{\theta_0,\theta_1}{\text{minimize}} \quad J(\theta_0, \theta_1)$$

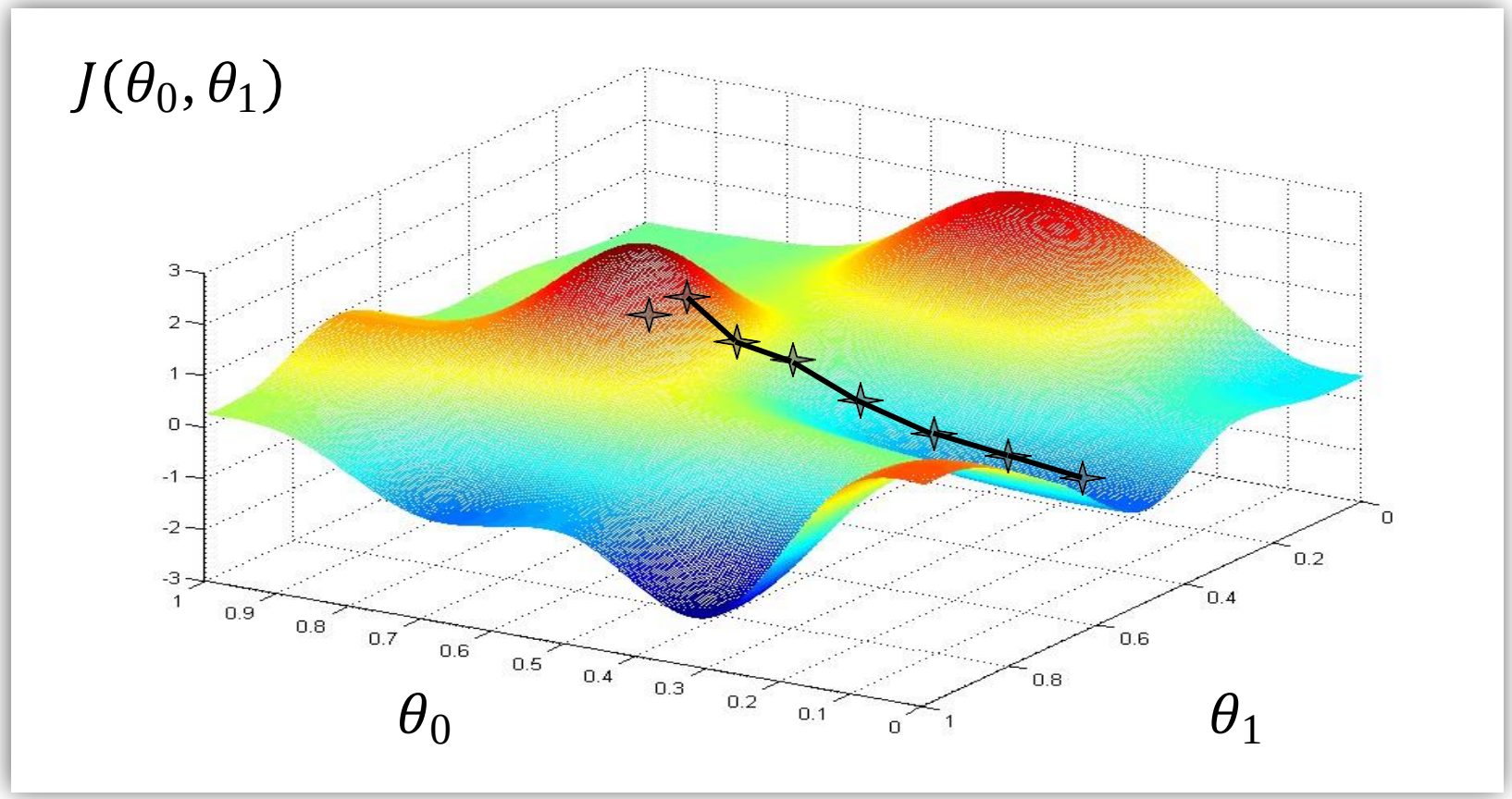
☐General method

- ❖ Start with a random initial value for the parameters $\theta_0$and θ_1. [eg zero value]
- ❖ Change the value of the parameters in such a way that the value of the cost function $J(\theta_0, \theta_1)$ decreases.
- ❖ Repeat the above operation until we reach a minimum value for the cost function. [convergence]

Gradient Descent: Global Optimum



Gradient Descent: Local Optimum



Gradient descent algorithm

repeat until convergence {

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

Learning rate

}

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Correct implementation. Update parameters value simultaneously

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

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

$$\theta_0 := \theta_0 + \Delta\theta_0$$

$$\theta_1 := \theta_1 + \Delta\theta_1$$

Gradient descent algorithm

repeat until convergence {

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

Learning rate

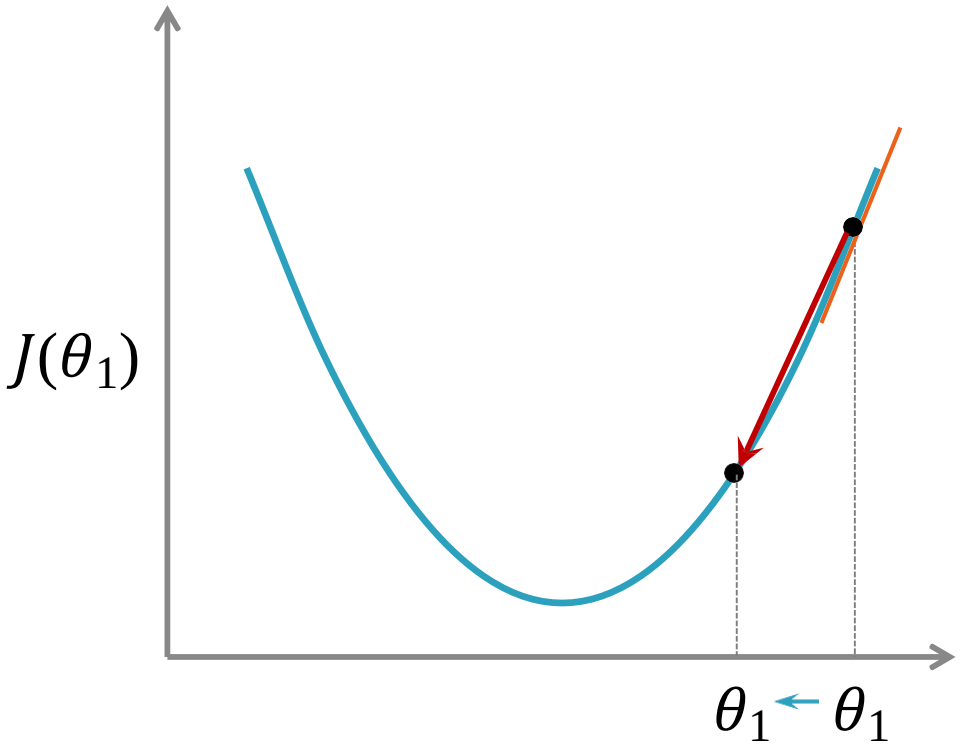
$$h_\theta(x) = \theta_0 + \theta_1 x$$

Incorrect implementation. Update parameters value sequentially

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

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

Gradient descent algorithm



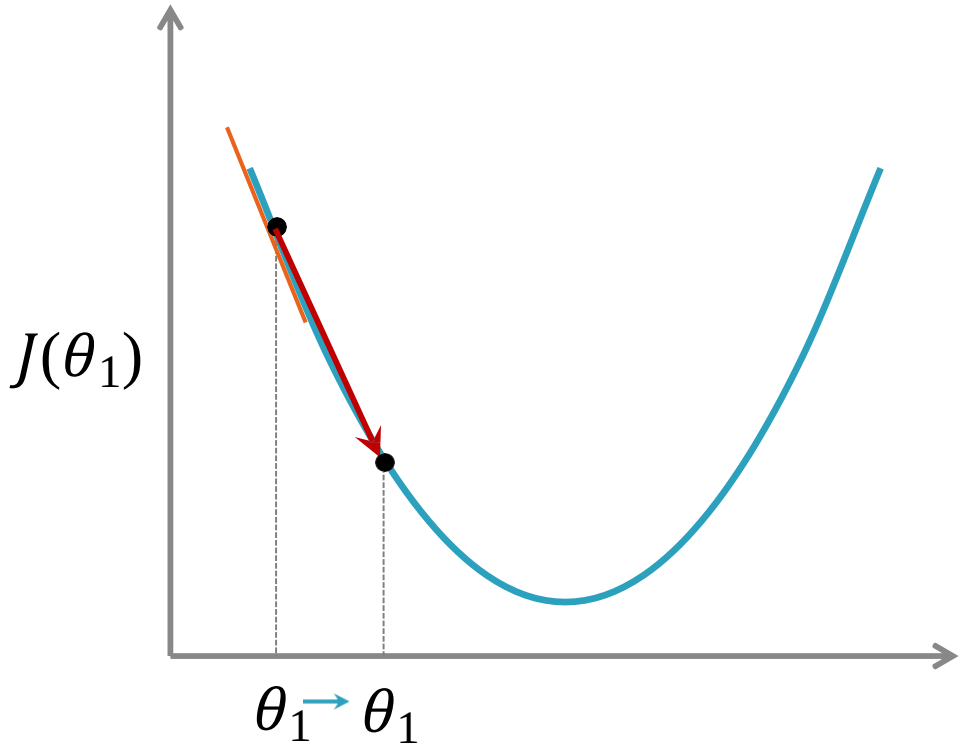
Positive slope

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta)$$

1

$$\theta_1 := \theta_1 - \underbrace{\alpha (\geq 0)}_{\geq 0}$$

Gradient descent algorithm



negetive slope

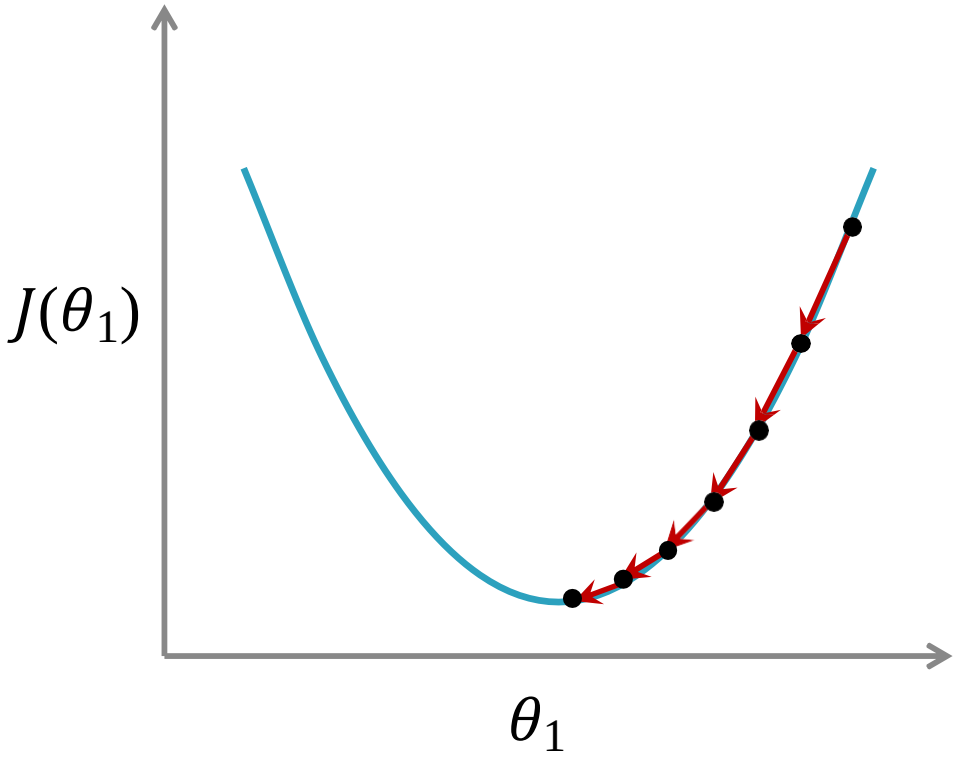
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta)$$

1

$$\theta_1 := \theta_1 - \underbrace{\alpha (\leq 0)}_{\leq 0}$$

Gradient descent algorithm: learning rate

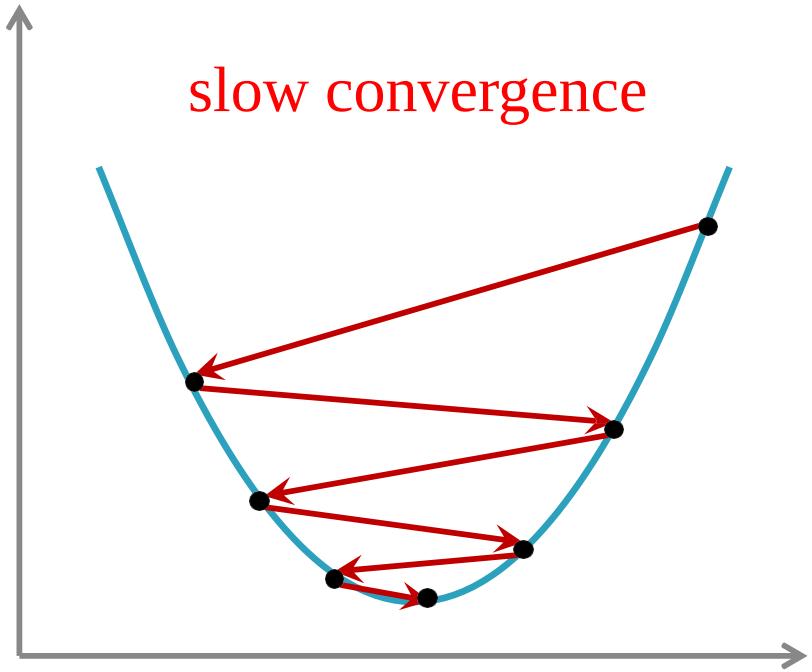
If the learning rate is too small, the gradient descent will converge slowly.

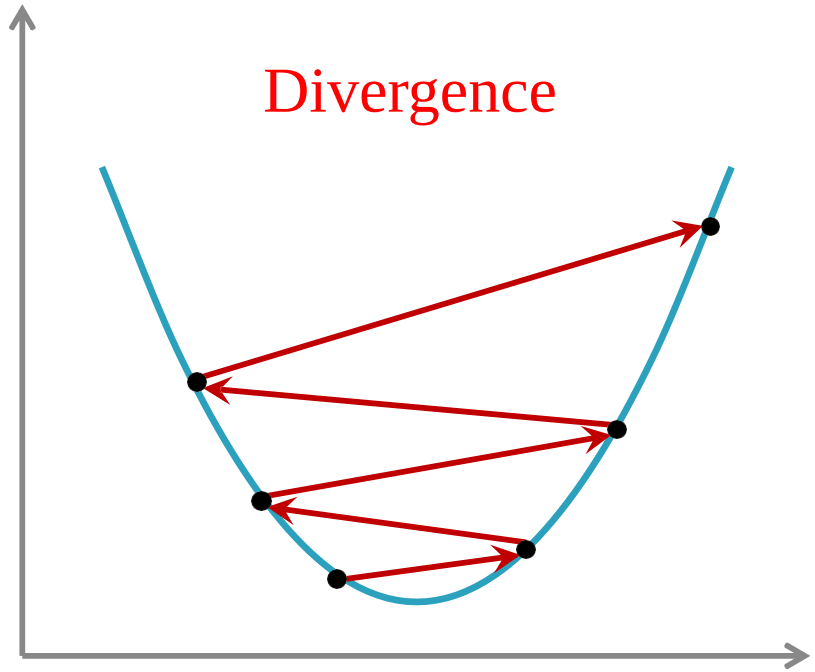


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Gradient descent algorithm: learning rate

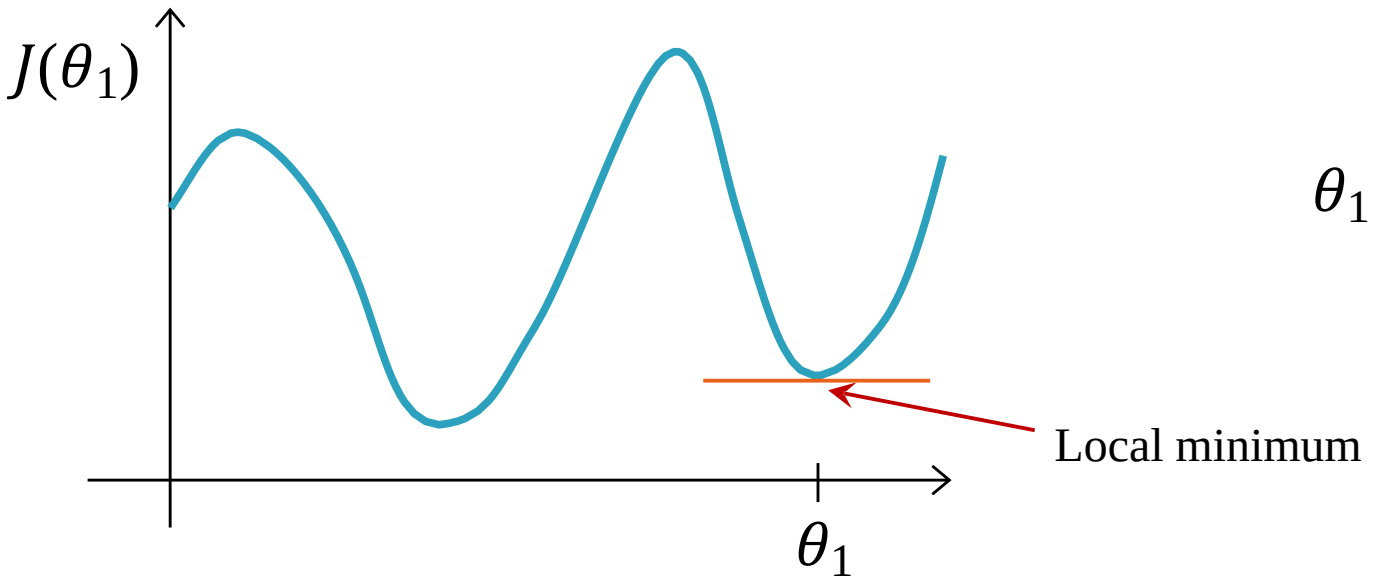
If the learning rate is too large, the gradient descent may converge slowly or even diverge.





Gradient Descent Algorithm: Convergence

Convergence When the value of the parameter θ_1 is in a local minimum.



$$\theta_1 := \theta_1 - \alpha \underbrace{\frac{\partial}{\partial \theta_1} J(\theta_1)}_{0}$$

Application of Gradient Descent in linear regression

Gradient Descent and linear regression

repeat until convergence {

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

}

linear regression

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

$$h_\theta(x) = \theta_0 + \theta_1 x$$

The gradient of the cost function

$$\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1)=\frac{\partial}{\partial\theta_j}\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2=\frac{\partial}{\partial\theta_j}\frac{1}{2}\sum_{i=1}^{m}\left(\left(\theta_0+\theta_1x^{(i)}\right)-y^{(i)}\right)^2$$

$$j=0\Rightarrow\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1)=\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)$$

$$j=1\Rightarrow\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1)=\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)\cdot x^{(i)}$$

Decreasing gradient and linear regression

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)$$

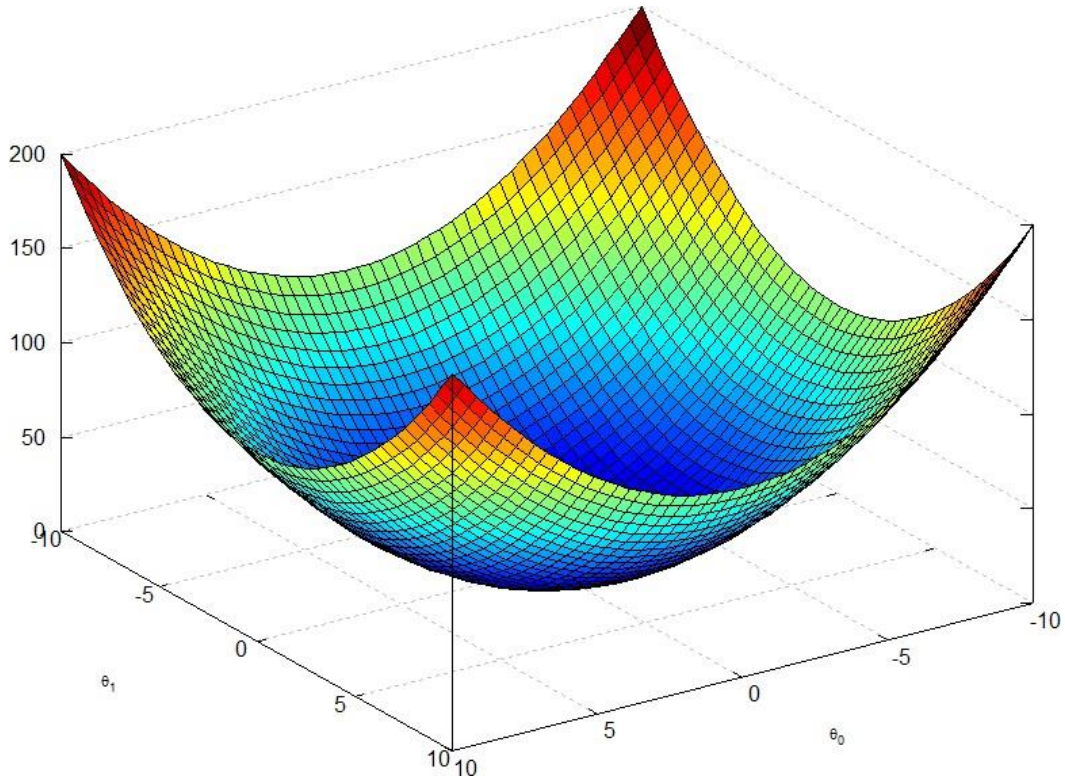
$$\theta_1 := \theta_1 - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x^{(i)}$$

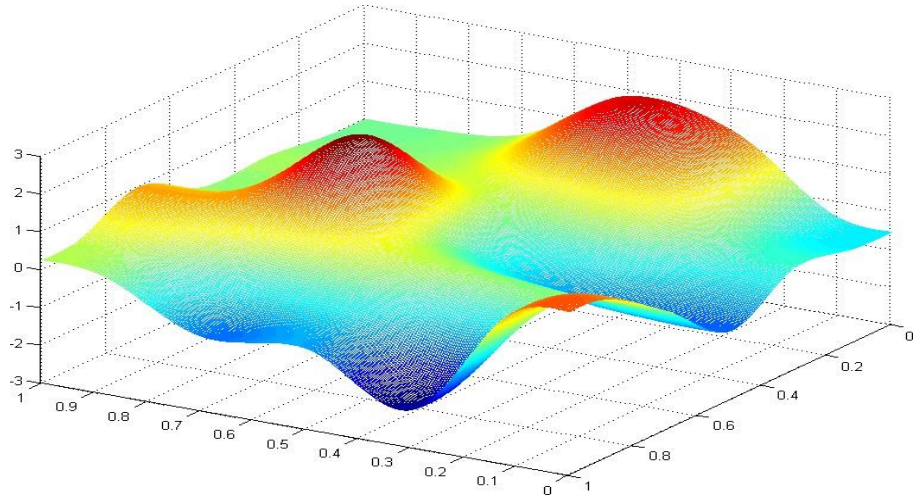
}

Simultaneous update

Gradient Descent and linear regression

☐Attention. In linear regression, the cost function is a cognate function, and as a result, the decreasing gradient necessarily converges to the global optimum in case of convergence.

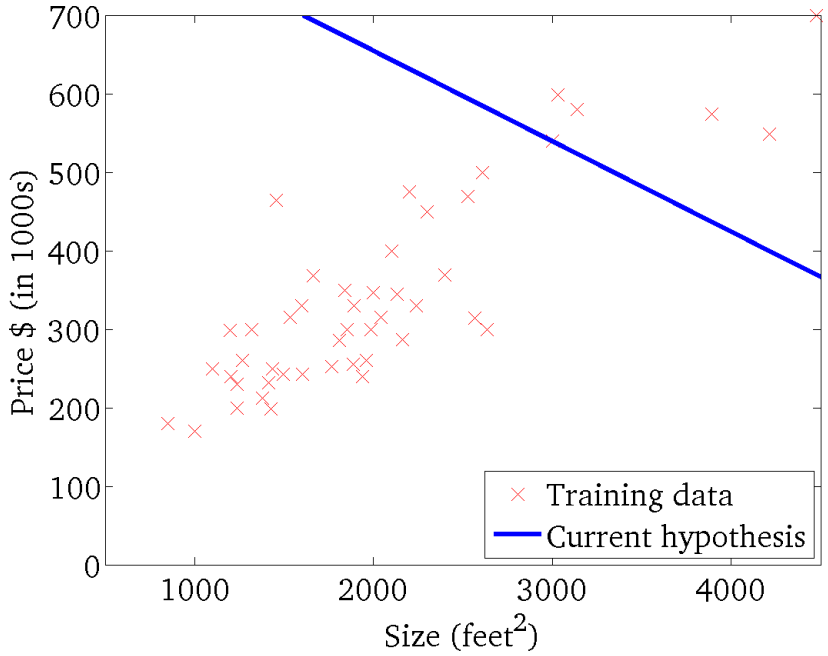


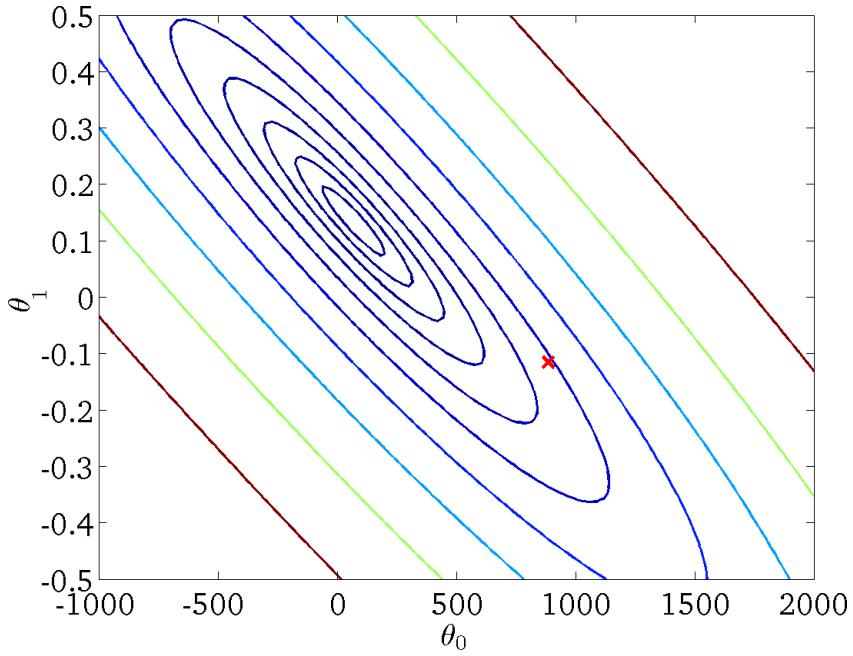


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

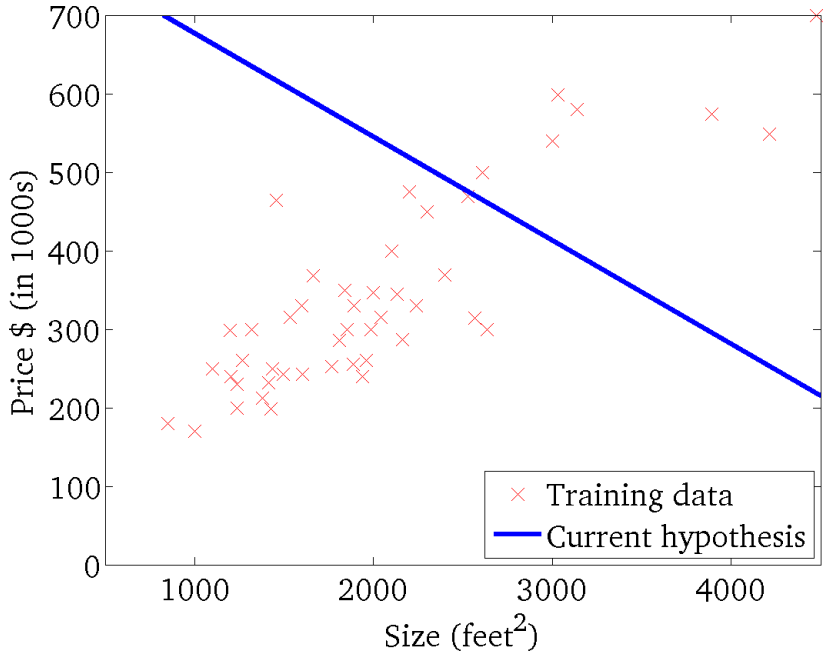


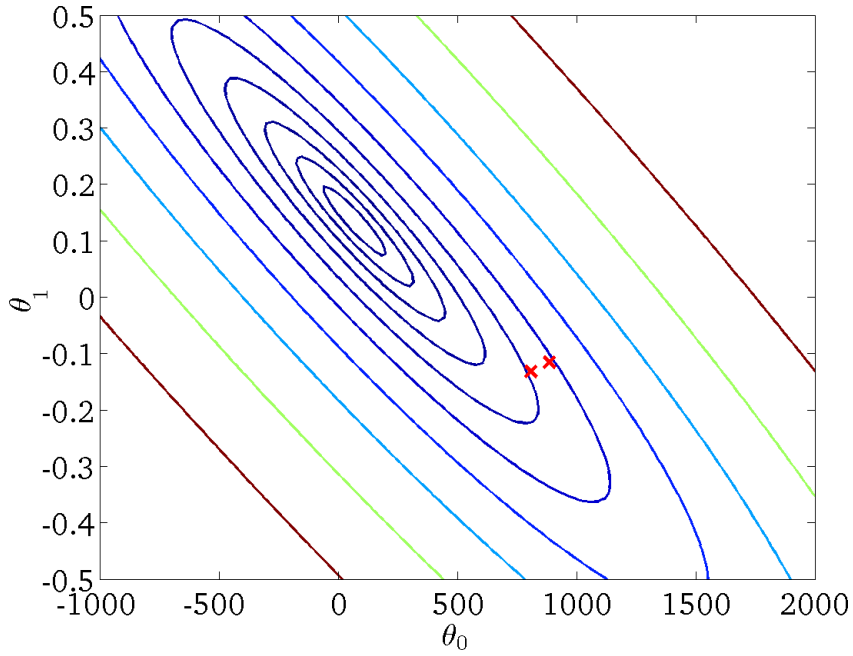


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

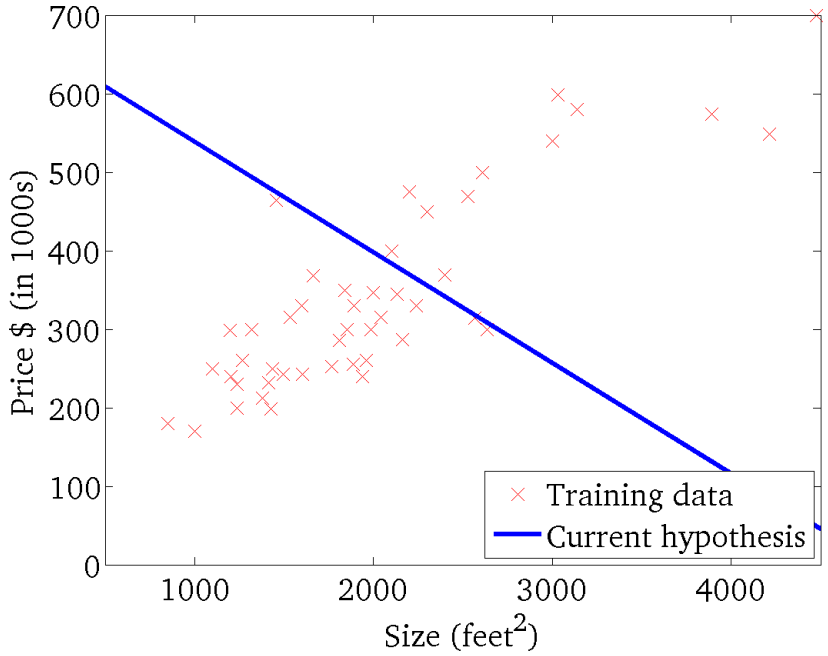


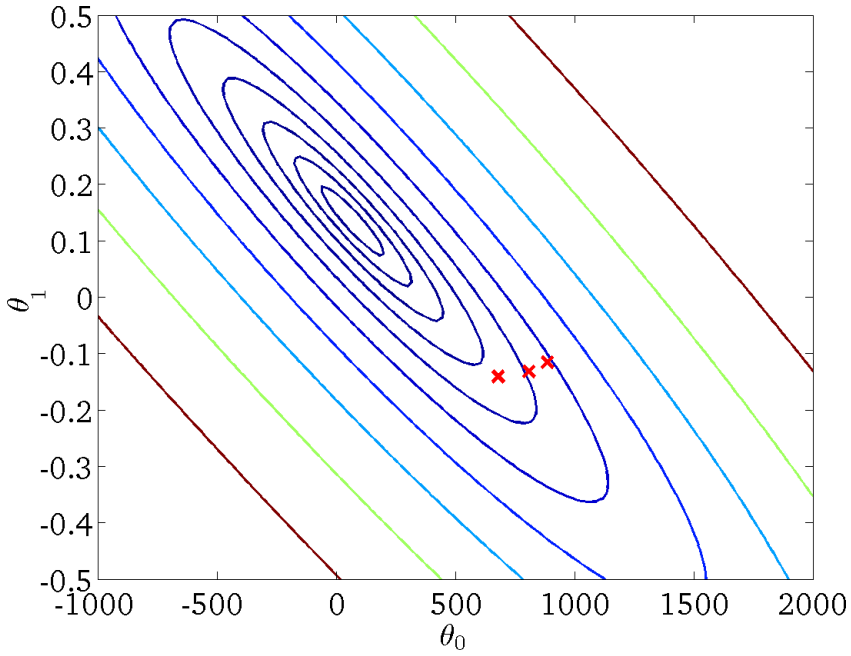


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

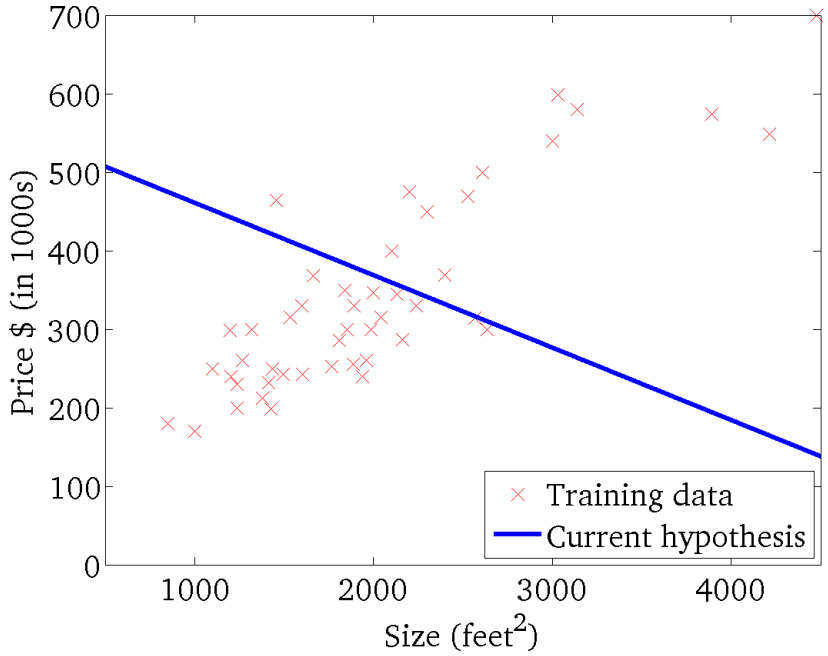


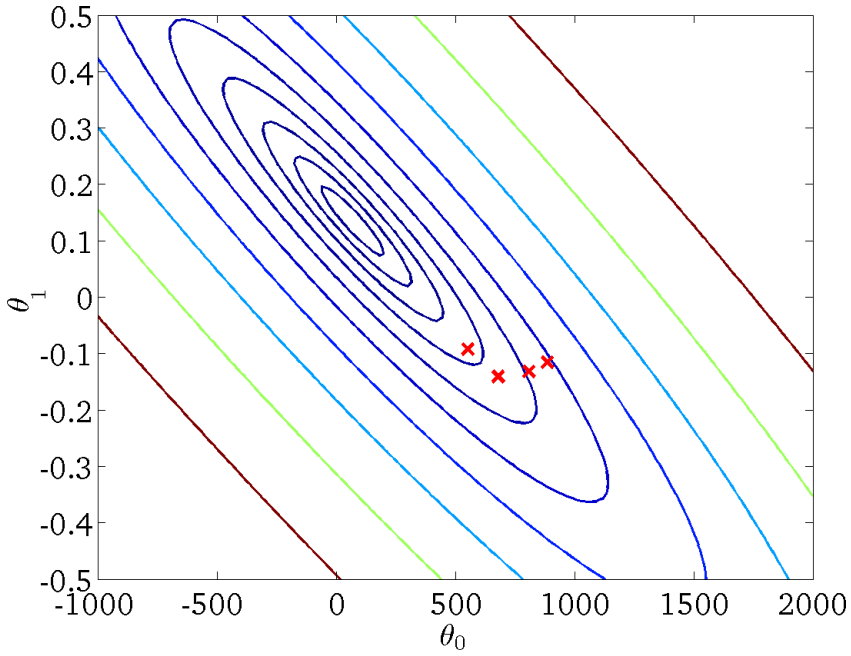


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

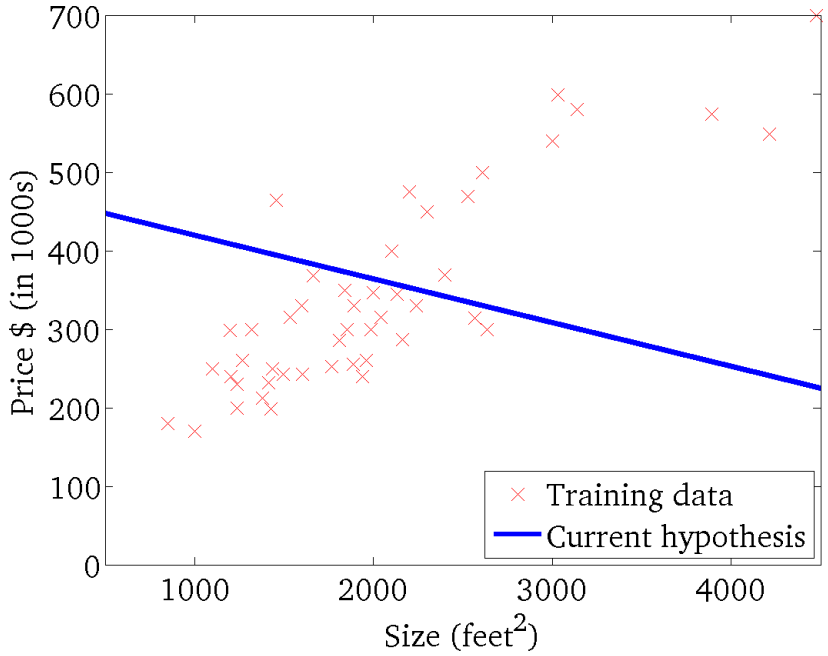


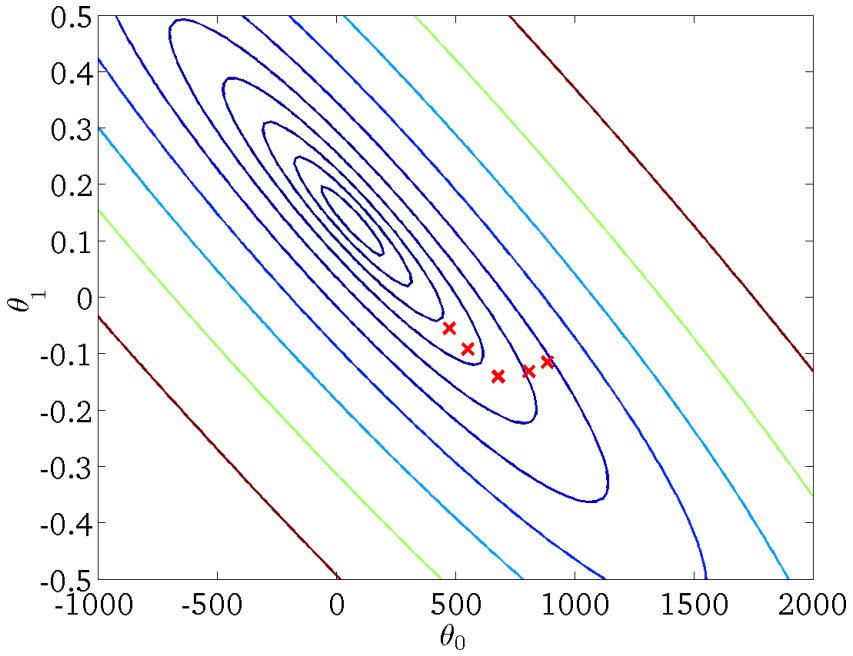


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

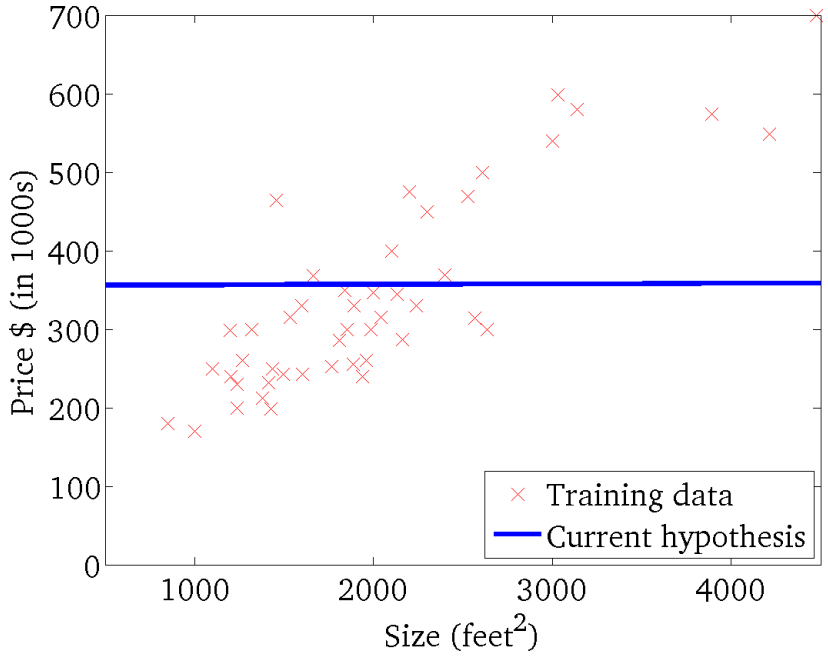


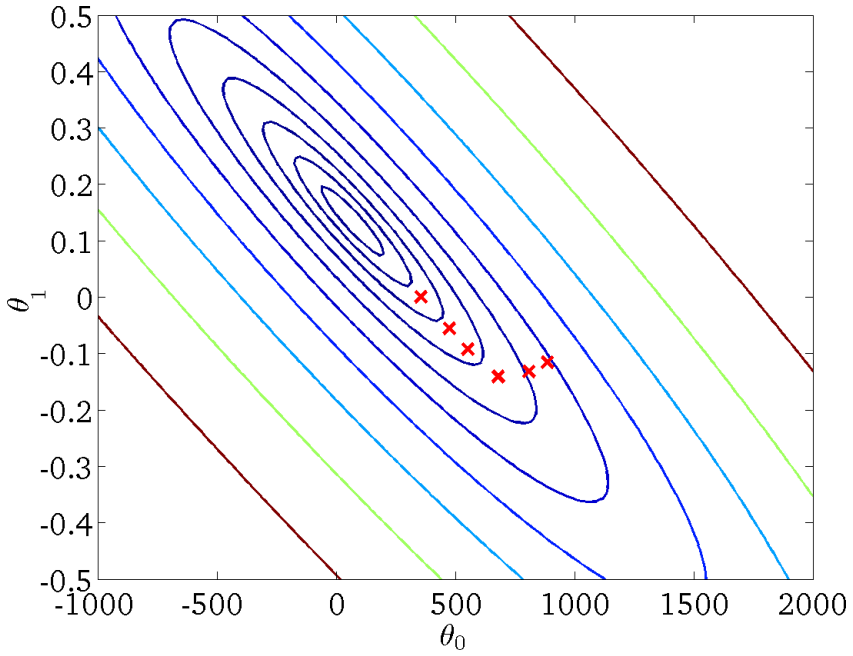


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

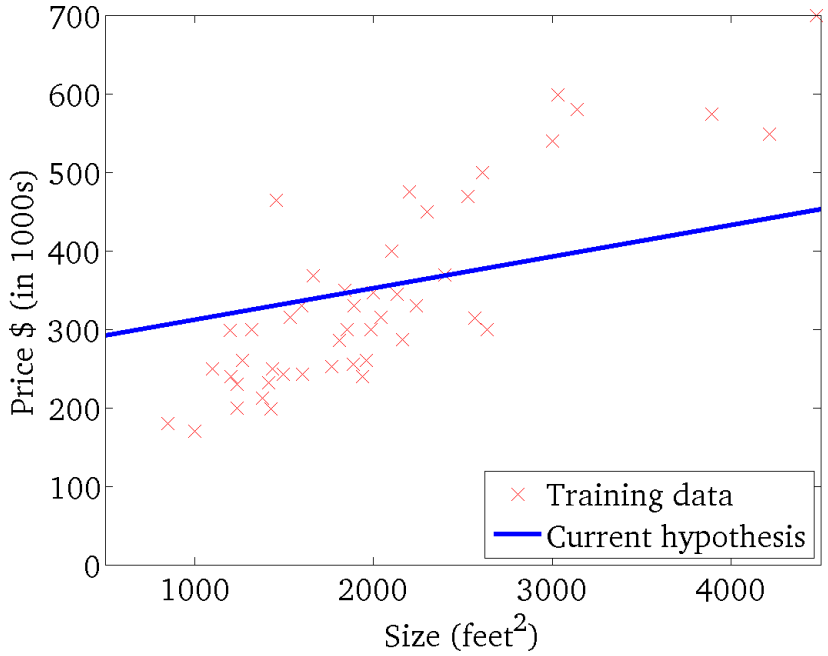


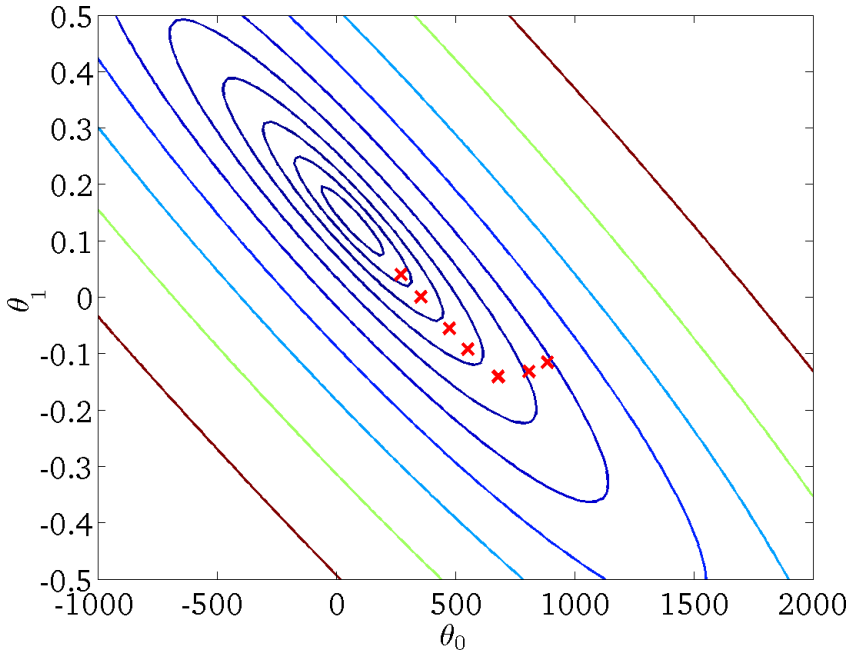


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

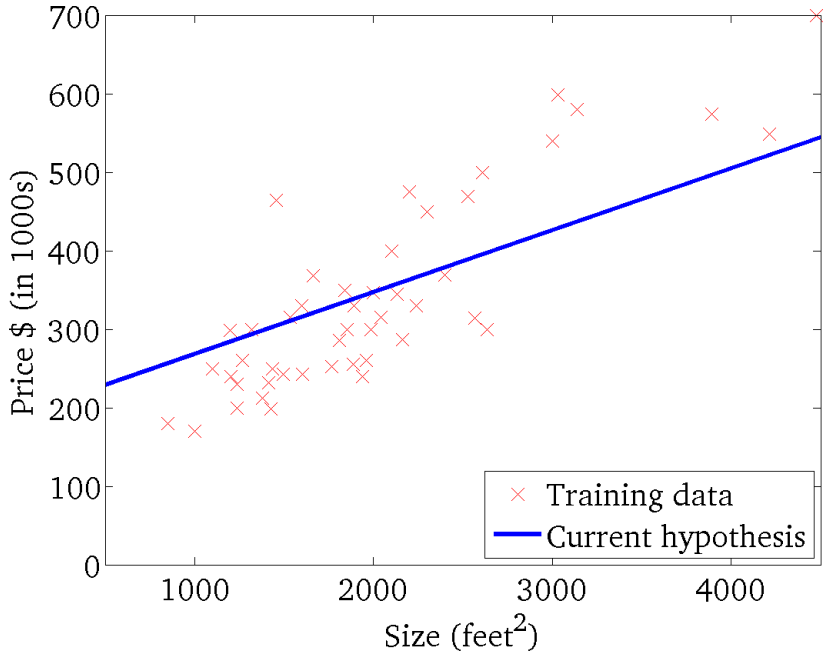


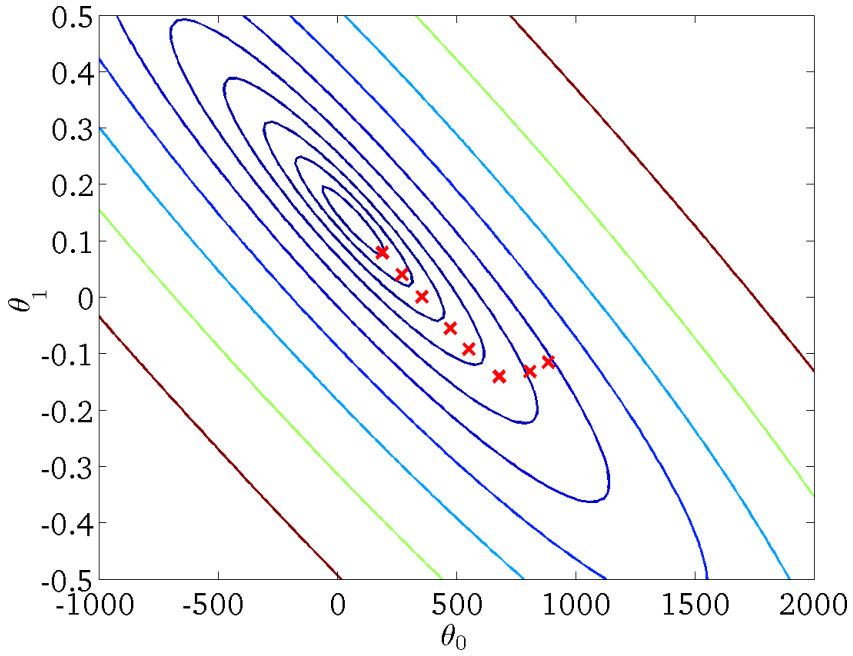


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$

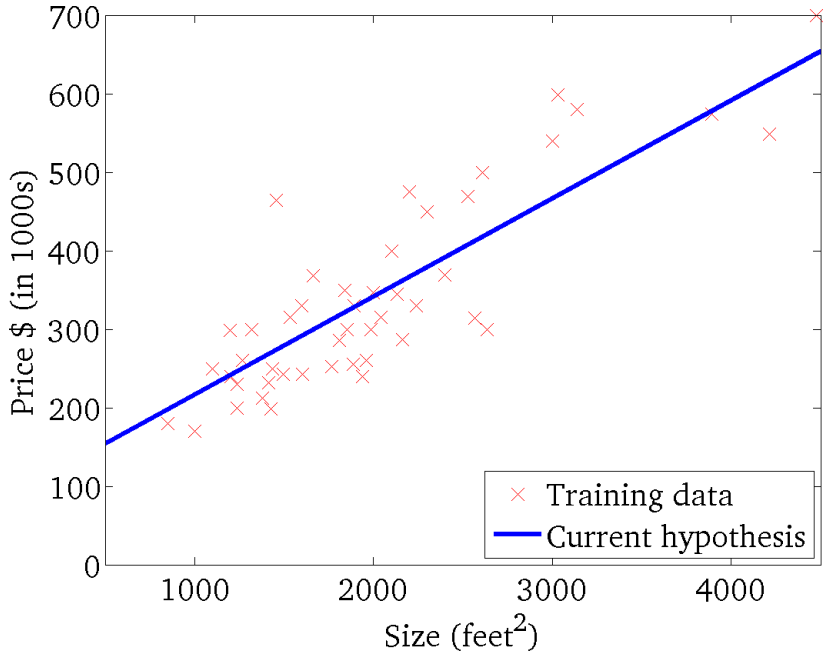


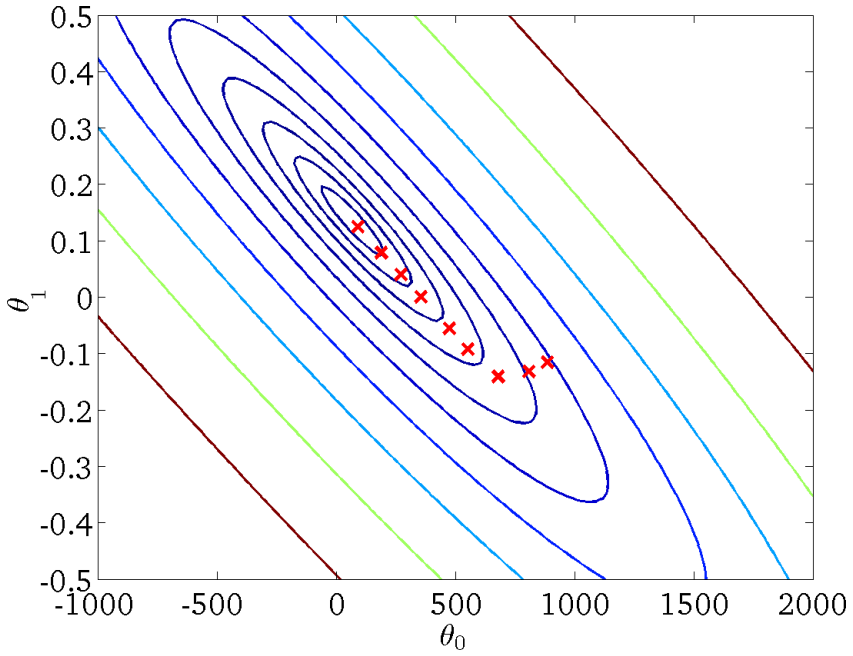


Gradient Descent

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1)$





Batch Gradient Descent

☐ Batch Gradient Descent. In each iteration of the algorithm, all training samples are used to update the parameter values.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad \text{(for } j = 0 \text{ and } j = 1\text{)}$$

$$j = 0 \Rightarrow \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)$$

$$j = 1 \Rightarrow \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x^{(i)}$$

Multivariate linear regression

Univariate regression (one feature)

- Linear regression with one feature.

Price(1000 dollars) y	Meterage (Square foot) x
460	2104
232	1416
315	1534
178	852
…	…

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Multivariate regression (multiple features)

- Linear regression with multiple features.

Price(1000 dollars) y	age of the house(Year) x_4	Number of floors x_3	Number of bedrooms x_2	Meterage (Square foot) x_1
460	45	1	5	2104
232	40	2	3	1416
315	30	2	3	1534
178	36	2	2	852
...	...	...	...	...

Multivariate regression (multiple features)

Price(1000 dollars) y	age of the house(Year) x_4	Number of floors x_3	Number of bedrooms x_2	Meterage (Square foot) x_1
460	45	1	5	2104
232	**40**	**2**	**3**	**1416**
315	30	2	3	1534
178	36	1	2	852
...	...	...	...	...

m

Number of Train set

n

- symbols.

n	Number of features
$x^{(i)}$	Inputs in the i-th training sample
$x_j^{(i)}$	The j th feature value in the i-th training sample

$$x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix}$$

hypothesis

- Univariate linear regression.

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- Multivariate linear regression.

$$(h_\theta)\ x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- For simplicity, we define $x_0 = 1$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

$$h_\theta(x) = \theta^T x$$

Gradient Descent in multivariate linear regression

Gradient Descent

- hypothesis

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- parameters

$$\theta = (\theta_0, \theta_1, \ldots, \theta_n)$$

- Cost Function

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

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

$$\frac{\partial}{\partial \theta_j} J(\theta) = \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)}$$

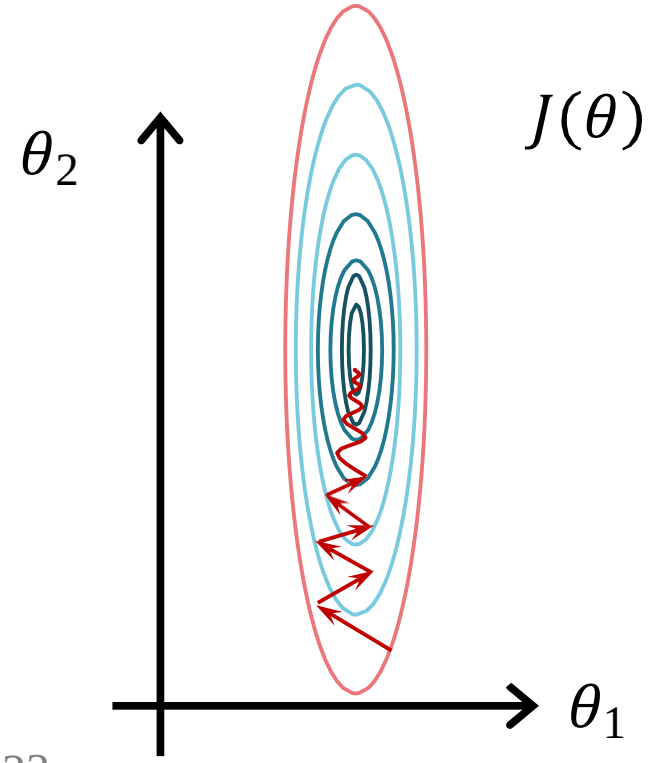
Practical tricks in multivariate regression

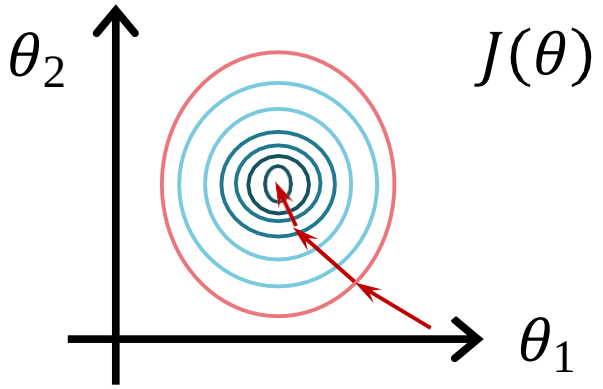
Scaling features
Determine the learning rate

Scaling of features (normalization)

- **Ideas.** Ensuring that feature values are on the same scale.
- **Target.** Increasing the speed of convergence in decreasing gradient.
- **Example.** The value of each feature is in a small coefficient from -1 to +1

$x1$: house size (0 to 2000)
$x2$: Number of bedrooms (1 to 5).





Scaling features

- Scaling. The value of each feature is in a small coefficient from -1 to +1

- Example

$$x_1 = \frac{size - 1000}{2000} \qquad -0.5 \le x_1 \le 0.5$$

$$x_2 = \frac{\#\ bedrooms - 2}{5} \qquad -0.5 \le x_2 \le 0.5$$

- Average normalization

$$x_i = \frac{x_i - \mu_i}{s_i}$$

μ_i → Average

s_i → Deviation from the norm

Gradient descent

- gradient descent

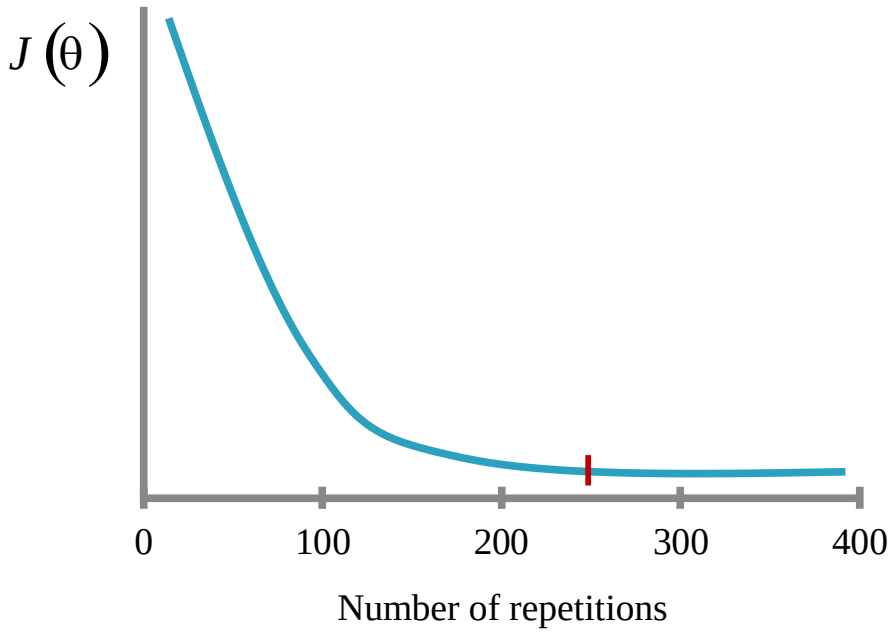
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- Question. How can you make sure that gradient descent works correctly?
- Question. What is the right value for the learning rate?

Correct function for gradient descent

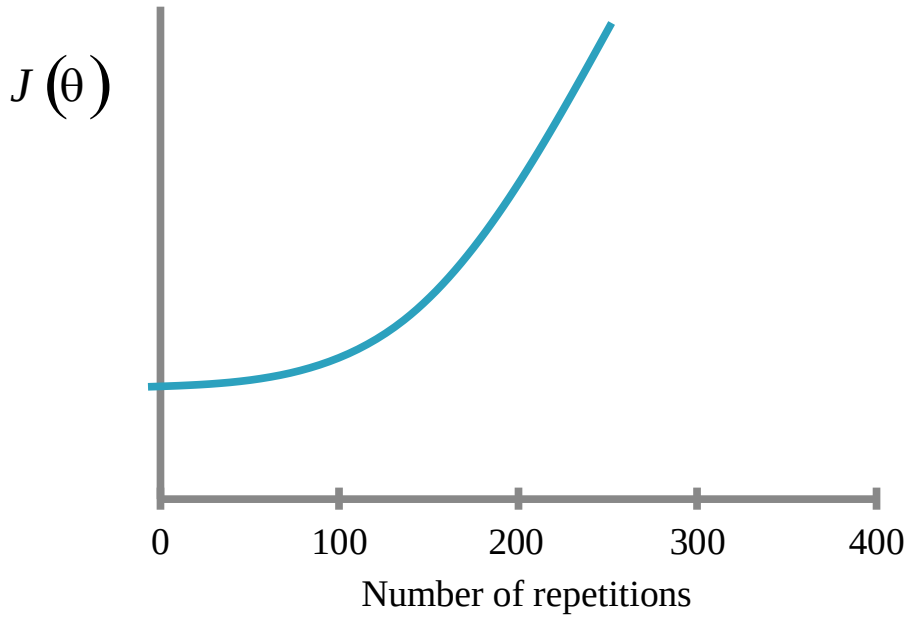
☐ Convergence test

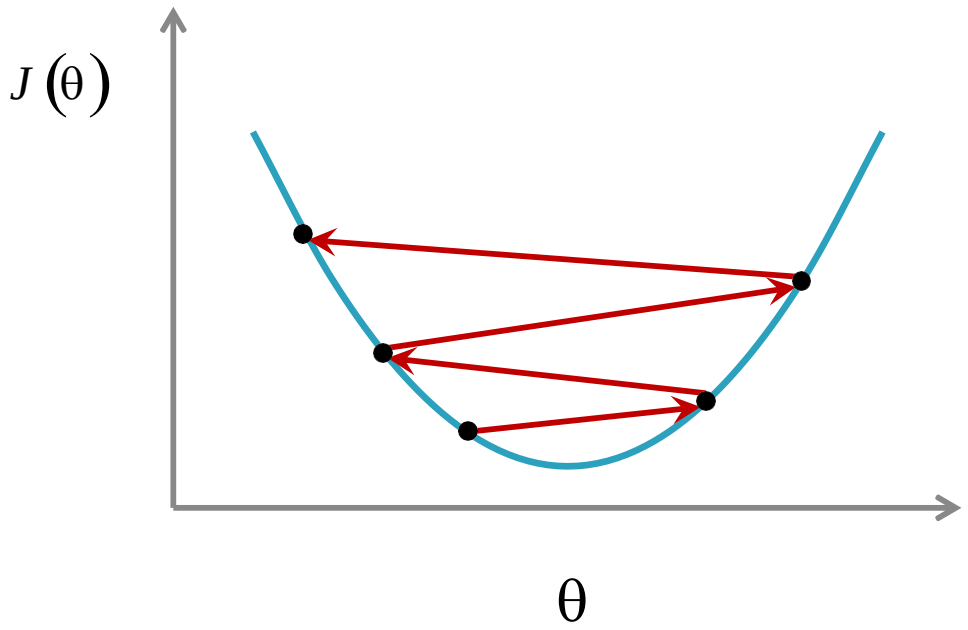
If the value of $J(0)$ changes by an amount less than 10^{-3} in one iteration, convergence has occurred.



Indications of gradient descent malfunction

- lack of convergence
- solution. Use smaller values for the learning rate, but if the learning rate is too small the convergence will be very slow.





Summary

- Learning rate.
  - Very small: Very slow convergence
  - Very large: slow convergence or no convergence

- Choice of learning rate.
  - In order to choose a suitable value for the learning rate, try the following values:

$$\ldots,\quad 0.001,\quad 0.003,\quad 0.01,\quad 0.03,\quad .01,\quad 0.3,\quad 1.0,\quad \ldots$$

Multinomial regression

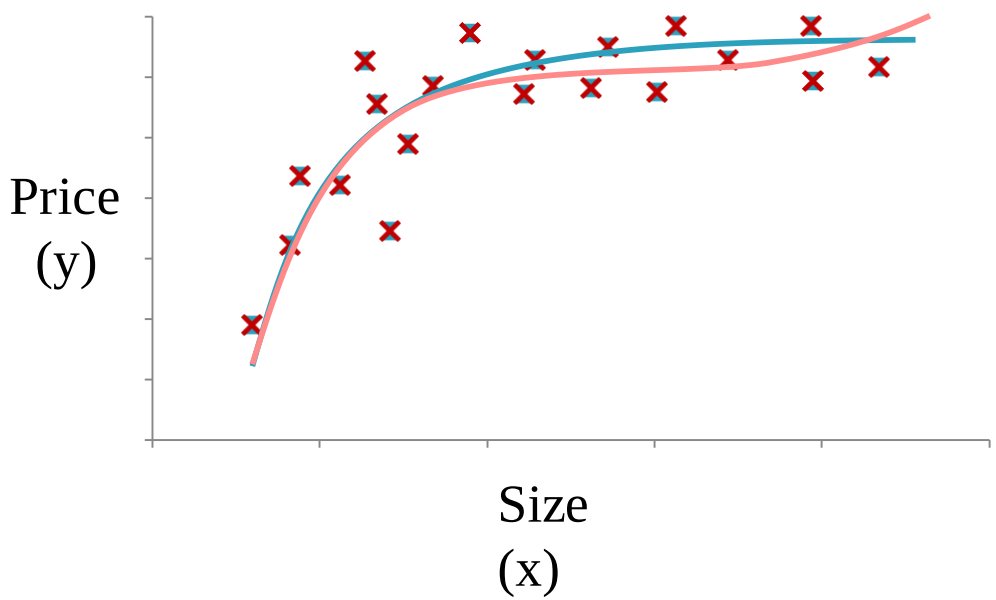
Pricing a Home: Selecting Features

$$h_\theta(x) = \theta_0 + \theta_1 \times \underbrace{(\text{frontage})}_{x_1} \times {}_2\theta + \underbrace{(\text{depth})}_{x_2}$$

$$Area = frontage \times depth$$

$$h_\theta(x) = \theta_0 + \theta_1 \times (\text{Area})$$

Polynomial regression



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

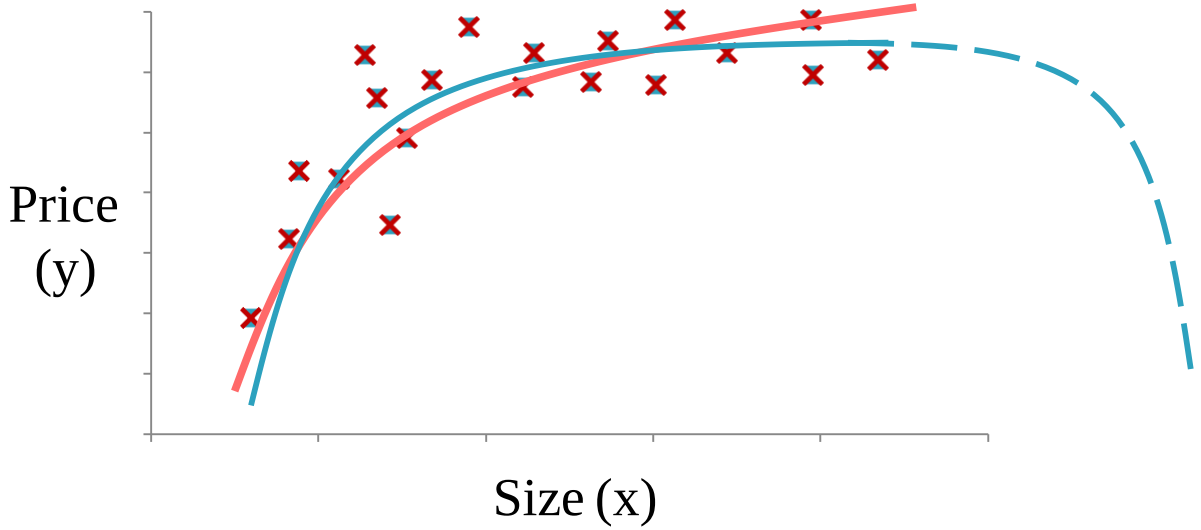
Polynomial regression

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

$$\begin{aligned} x_1 &= (Size)^1 & 1 \le x_1 \le 10^3 \\ x_2 &= (Size)^2 & 1 \le x_2 \le 10^6 \\ x_3 &= (Size)^3 & 1 \le x_3 \le 10^9 \end{aligned}$$

features Selection

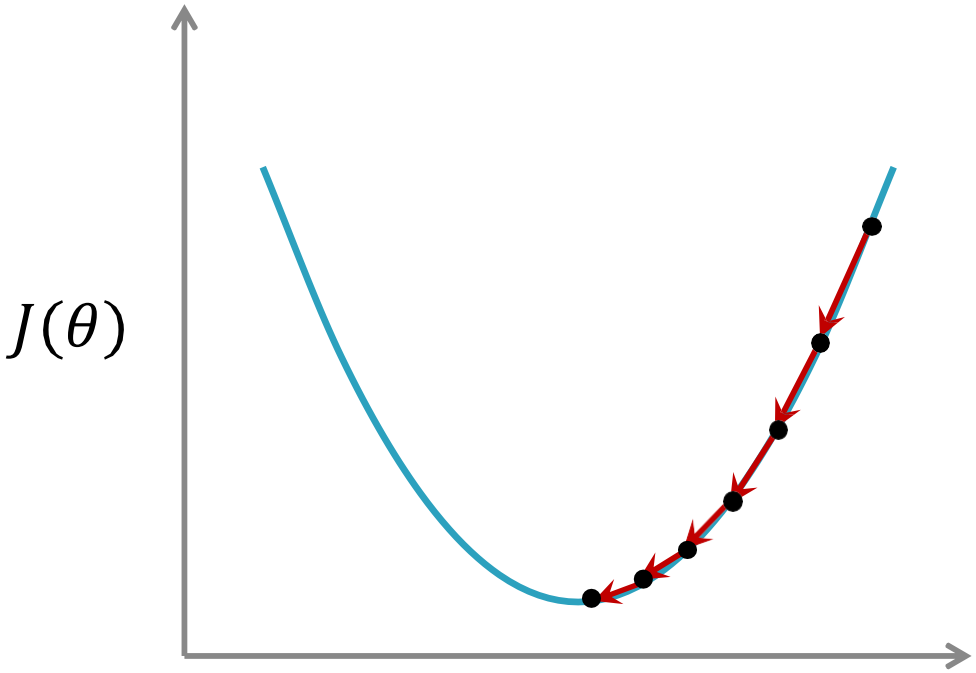


$$h_\theta(x) = \theta_0 + \theta_1\ (size) + \theta_2\,(size)^2$$

$$h_\theta(x) = \theta_0 + \theta_1\ (size) + \theta_2\left(\sqrt{size}\right)$$

Linear regression: normal equation

Decreasing gradient and normal equation



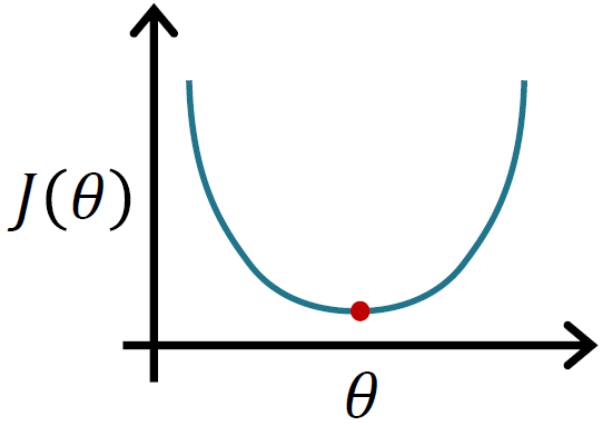
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- Normal equation. An analytical method to determine the value of parameters

Normal equation

$\theta \in \mathbb{R}$: $J(\theta) = a\theta^2 + b\theta + c$

$$\frac{d}{d\theta} J(\theta) \stackrel{\text{def}}{=} 0$$



$\theta \in \mathbb{R}^{n+1}$: $J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2$

$$\frac{\partial}{\partial \theta_j} J(\theta) \stackrel{\text{def}}{=} 0 \qquad (j = 0,1,2, \dots, n)$$

Normal equation: Example (m = 4)

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$$X\theta = y$$

Normal equation: Example (m = 5)

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178
1	3000	4	1	38	540

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \\ 1 & 3000 & 4 & 1 & 38 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \quad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \\ 540 \end{bmatrix}$$

$$X\theta = y$$

Normal equation: general state

- m educational sample; n feature

$$X\theta = y$$

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1} \qquad X = \begin{bmatrix} \text{———} \left(x^{(1)}\right)^T \text{———} \\ \text{———} \left(x^{(2)}\right)^T \text{———} \\ \text{———} \left(x^{(3)}\right)^T \text{———} \\ \vdots \\ \text{———} \left(x^{(m)}\right)^T \text{———} \end{bmatrix} \in \mathbb{R}^{m\times(n+1)} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^{m}$$

Design matrix

Normal equation

Solving linear equations

$$X\theta = y$$

$$X^T X\theta = X^T y$$ ← Normal equation

$$\theta = \underbrace{(X^T X)^{-1} X^T}_{X^+} y$$ ← Pseudo-inverse matrix

Octave:

```
theta = pinv(X'*X)*X'*y
```

Gradient Descent and normal equation

normal equation	Gradient Descent
☐No need to choose α	☐need to choose α
☐No need to repeat	☐need to repeat
☐It is slow for large n due to the need to calculate the inverse of the $X^T X$ matrix.	☐It works well even for very large values of n.
$n < 10000$	$n \geq 10000$

Normal equation and irreversibility of $X^T X$

Normal equation

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

Question. But what if $X^T X$ is not invertible?

Octave: **theta = pinv(X'*X)*X'*y**

pseudo reverse

Causes of $X^T X$ irreversibility

☐Redundancy of features (linear dependence).

$$x_1 = size\ (feet^2)$$

$$x_2 = size\ (m^2)$$

$$x_1 = (3.28)^2\, x_2$$

☐ A very large number of features (eg $n \geq m$).

❖ solution. Remove some features by setting [Next]

Regression with local weighting

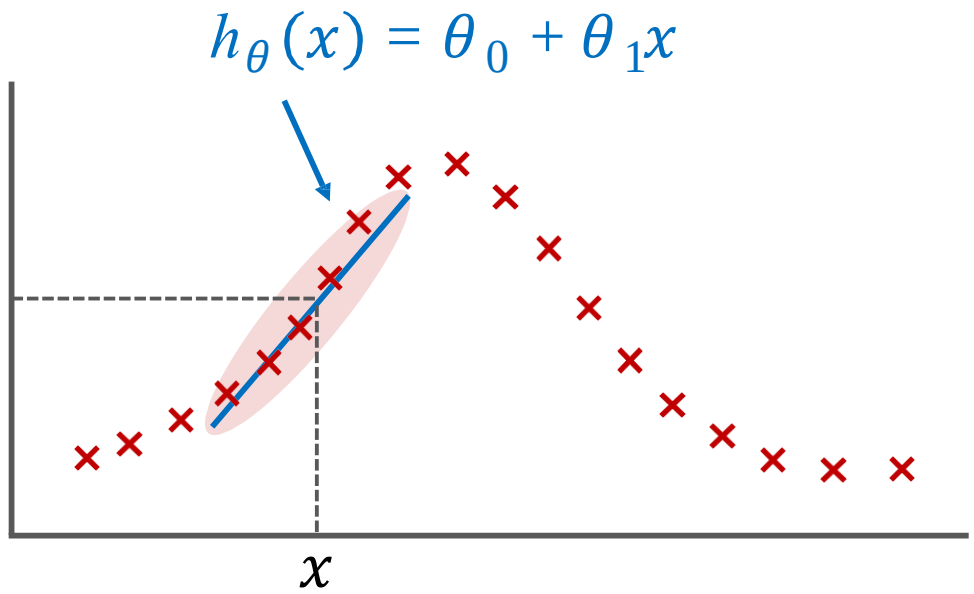
Parametric and non-parametric learning methods

- Parametric learning methods.
  - There is a fixed set of parameters.
  - We do not need a training set to predict new data.
  - Example: regression andlogistic regression [continue].

- Non-parametric learning methods.
  - The number of parameters increases (linearly) with the increase in the size of the training set.
  - In order to predict new data, we need the entire training set.
  - Example: regression andlogistic regression [continue].

Regression with local volatility

☐Ideas. Giving more importance to closer data.



$$w^{(i)} = \exp\left(-\frac{\left(x^{(i)} - x\right)^2}{2r^2}\right)$$

Bandwidth

$$J(\theta) = \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right)^2$$

Regression: probabilistic interpretation

Regression: probabilistic interpretation

☐model

error

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

☐Error

- ❖ Considering the influence of other factors
  - ❖ such as unintended features
- ❖ Considering the effect of noise.

$$\epsilon^{(i)} \sim N(0, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$$

☐The errors are independent and follow a uniform Gaussian distribution. [iid]

$$y^{(i)} \sim N\left(\theta^T x^{(i)}, \sigma^2\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

Maximum likelihood estimation

- Exponentiation function. Probability of seeing training data as a function of parameters θ

$$L(\theta) = p(Y|X;\theta) = \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

- The logarithm of the exponential function.

$$\begin{aligned} l(\theta) = \ln L(\theta) &= \ln \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{m} \ln\left[\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)\right] \\ &= m \ln \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2 \end{aligned}$$

Maximum likelihood estimation

Maximum likelihood estimation.

Choosing a value for the parameter θ so that $l(\theta)$ is maximized.

$$\theta_{MLE} = \arg\max_{\theta} l(\theta)$$

$$= \arg\max_{\theta} \; m \ln \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

$$= \arg\max_{\theta} \; -\frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

$$= \arg\min_{\theta} \; \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2 \longleftarrow \text{Cost function}$$