PENNSTATE

Regression

Overview

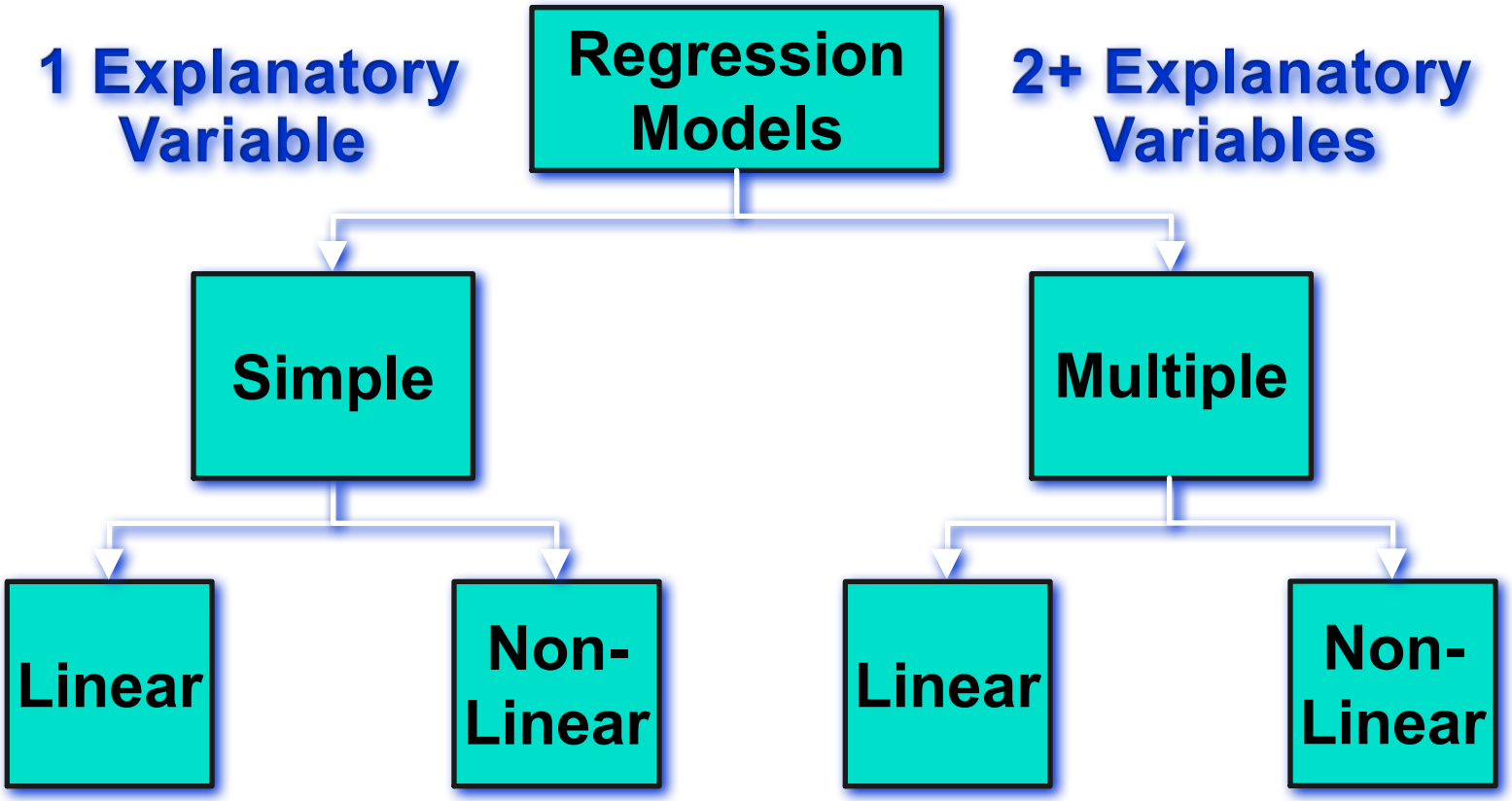
- Linear Regression Model
- Univariate Regression
- Gradient Descent & Normal Equation
- Multivariate Regression

Materials prepared based on PSU STAT414 (https://onlinecourses.science.psu.edu/stat414/print/book/export/html/262/) and machine learning by Andrew Ng.

Regression Model

- Models are usually used for representation of some phenomenon
- Regression is a predictive modeling technique where the target to be estimated has a continuous variable type.
- Examples:
  - Predicting S&P 500 index using economic indicators
  - Forecast the amount if precipitation in a region based on characteristics of jet stream
  - Projecting the total sales of a company based on the amount of spending for advertisement

Types of Regression Models



4

Preliminaries

- Let D denote a dataset of N observations

  $$D = \{(\boldsymbol{x}_i, y_i) | i=1..N\}$$

- Each explanatory variable $\boldsymbol{x}_i$ corresponds to the set of attributes of the i-th observation ($\boldsymbol{x}_i$ is also called predictor variable or independent variable)
- The corresponding target variable y_i is also called response variable, outcome variable or dependent variable.
- The explanatory variables of a regression can be either discrete or continuous.

Preliminaries

- Predict housing price based on size, number of bed rooms, number of floors, and age of the house.

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

x

Explanatory variable
Predictor variable
Independent variable

y

Target variable
Response variable
Dependent variable

Regression

- Regression is the data mining task of learning a target function *f* that maps each attribute set ***x*** into a continuous-valued output *y*.
- The goal of regression is to find a target function that fits the input data with minimum error.
- Possible error functions:
  - Absolute Error = $\sum_i | y_i - f(x_i)|$
  - Squared Error = $\sum_i \ (y_i - f(x_i))^2$

Simple Linear Regression

- Simple linear regression is a way of evaluating the relationship between two variables.
- For example, one might be interested in the relationship between:
  - heights and weights
  - high school grade point averages and college entrance exam scores
  - Housing price and size
  - speed and gas mileage
- Type of relationships
  - deterministic relationships
  - statistical relationships

Deterministic Relationships

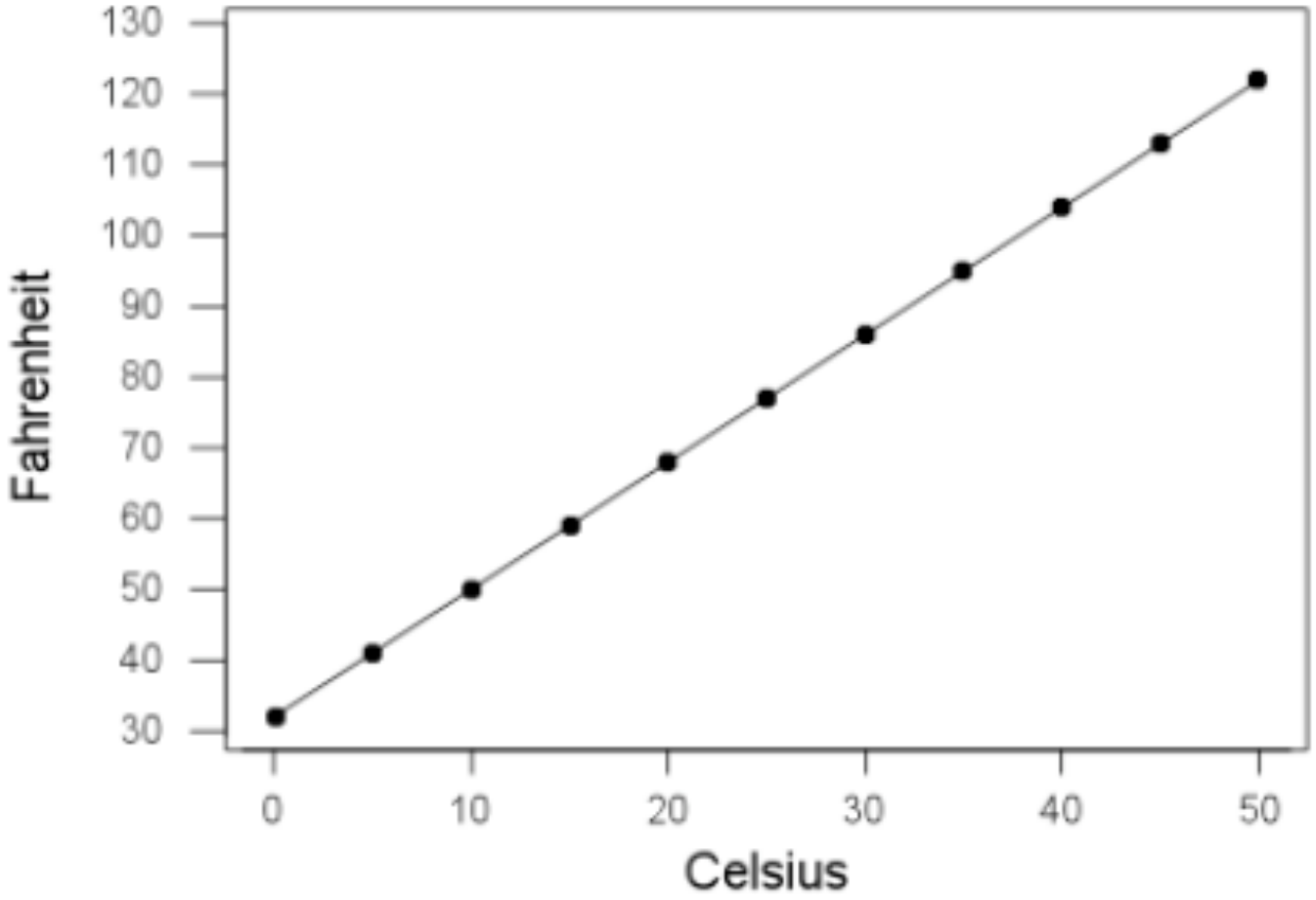
- A deterministic (or functional) relationship is a hypothesized exact relationship between the predictor *x* and the response *y*.
- Example: the conversion between temperature in degrees Celsius (C) and degrees Fahrenheit (F)

$$F = \frac{9}{5}C + 32$$

- C=10

$$F = \frac{9}{5}(10) + 32 = 50$$

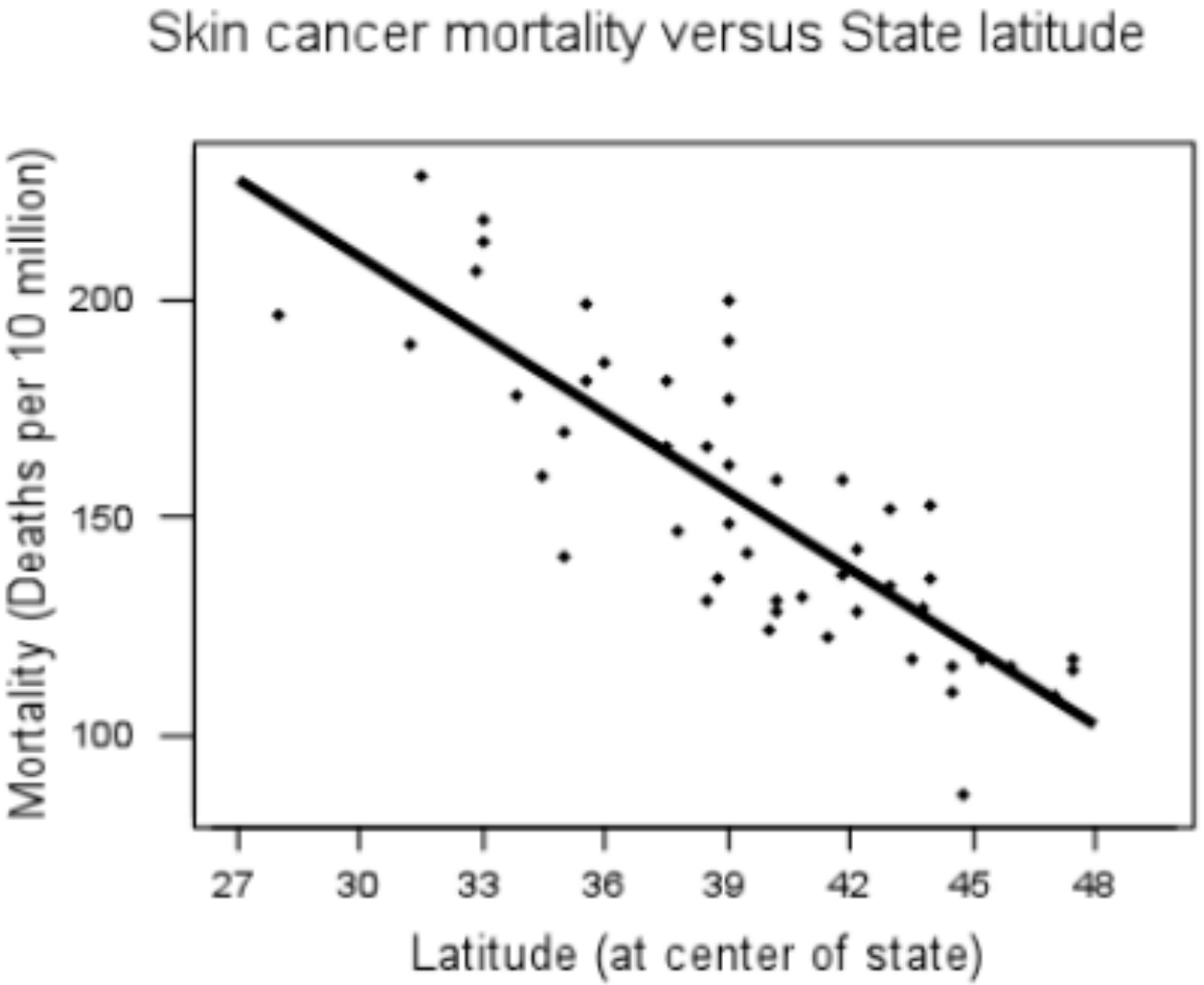
Celsius vs Fahrenheit



Statistical Relationship

- A statistical relationship is not an exact relationship.
- It is a scattered trend (less-than-perfect statistical relationships) existing between the predictor *x* and the response *y*.
- Example: relationship between the latitude (in degrees) at the center of each of the 50 U.S. states and the mortality (in deaths per 10 million) due to skin cancer in each of the 50 U.S. states
- Additional example: positive relationship between height and weight

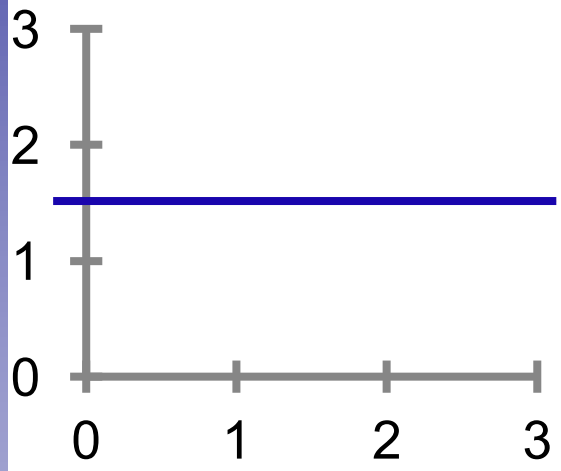
Skin Cancer Mortality vs. State Latitude



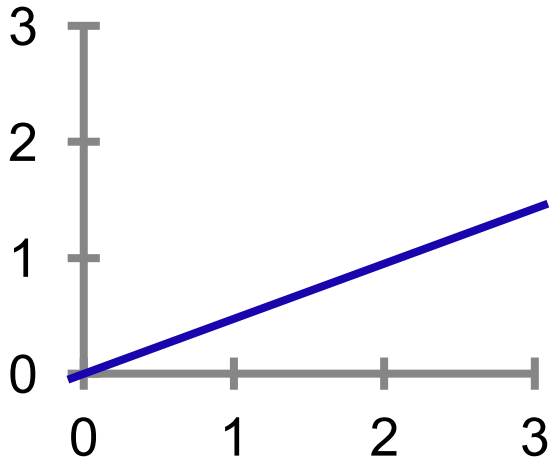
Linear Relationships

- Linear regression models model relationships by linear functions!
  - Use a linear function, i.e., line, to fit the observations.

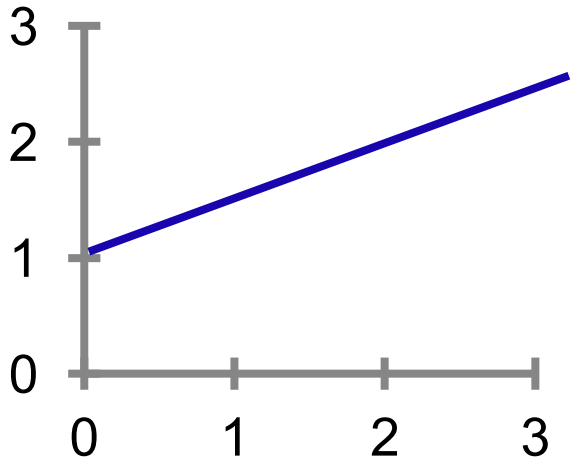
$$h_\theta(x) = \theta_0 + \theta_1 x$$



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



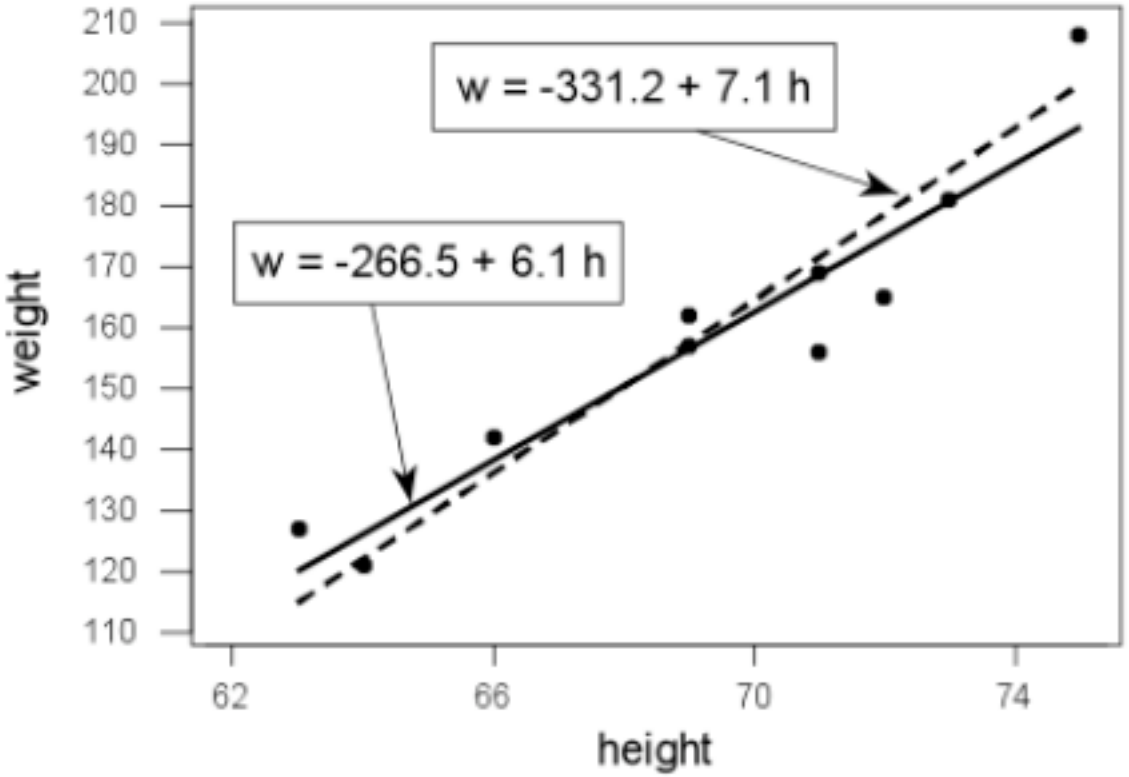
$\theta_0 = 0$
$\theta_1 = 0.5$



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

Least Squares: The Idea

- For example, to quantify an *assumed* linear relationship between height and weight
- Given ten observations, there are many fitting line. Summarize the relationship with the best fitting line, in terms of squared errors



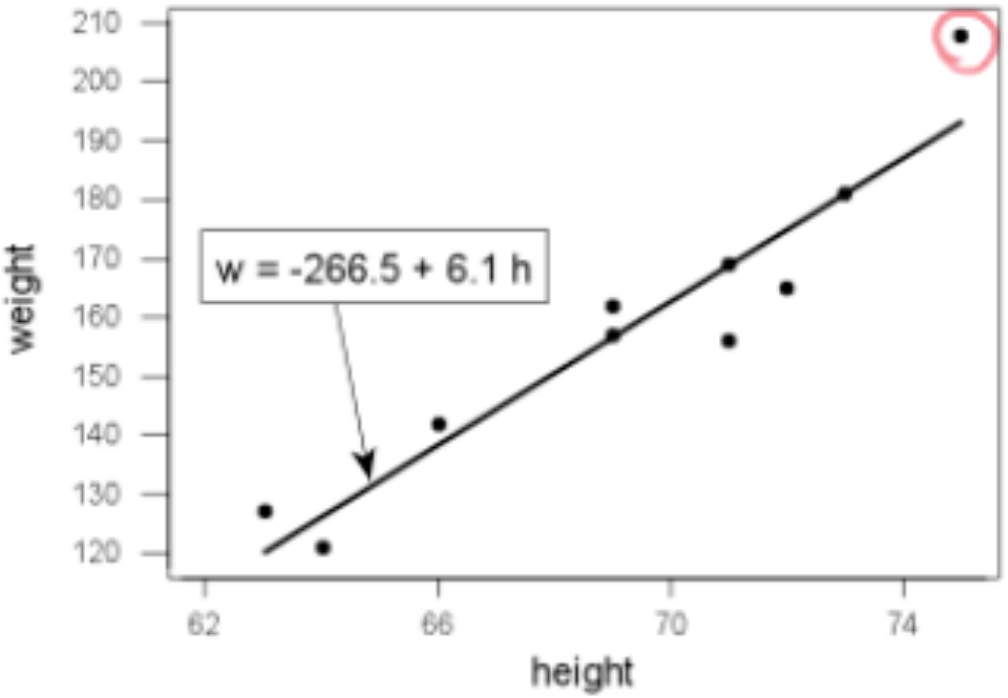
Finding Best Fitting Line

- Let y_i, x_i and $f(x_i)$ denote the observed response, the predictor value, and the predicted response (or fitted value) for the i-th observation.
- For the circled point
  - $x_i = 75$ $y_i = 208$
  - $f(75) = -266.534 + 6.13758(75) = 193.8$
  - Prediction error:

    $e_i = y_i - f(x_i)$

    $= 208 - 193.8$

    $= 14.2$



Least Squares Criterion

- A line that fits the data well will be one for which the *n* prediction errors are as small as possible.
- In order to find the best fitting line:

$$f(x_i) = a_1 + bx_i$$

We find a_1 and b that minimize the Sum of Squared prediction Errors:

$$SSE = \Sigma_{i=1..n} (y_i - f(x_i))^2 = \Sigma_{i=1..n} (y_i - (a_1 + bx_i))^2$$

Weight = −331.2 + 7.1 × Height

i	x_i	y_i	$f(x_i)$	$y_i - f(x_i)$	$(y_i - f(x_i))^2$
1	64	121	123.2	-2.2	4.84
2	73	181	187.1	-6.1	37.21
3	71	156	172.9	-16.9	285.61
4	69	162	158.7	3.3	10.89
5	66	142	137.4	4.6	21.16
6	69	157	158.7	-1.7	2.89
7	75	208	201.3	6.7	44.89
8	71	169	172.9	-3.9	15.21
9	63	127	116.1	10.9	118.81
10	72	165	180.0	-15.0	225.00

					766.51

Weight = −266.5 + 6.1 × Height

i	x_i	y_i	$f(x_i)$	$y_i - f(x_i)$	$(y_i - f(x_i))^2$
1	64	121	126.271	-5.3	28.09
2	73	181	181.509	-0.5	0.25
3	71	156	169.234	-13.2	174.24
4	69	162	156.959	5.0	25.00
5	66	142	138.546	3.5	12.25
6	69	157	156.959	0.0	0.00
7	75	208	193.784	14.2	201.64
8	71	169	169.234	-0.2	0.04
9	63	127	120.133	6.9	47.61
10	72	165	175.371	-10.4	108.16

					597.28

Least Squares Method

- Given $f(x) = a_1 + bx$, then

  $SSE = \Sigma_{i=1..n} (y_i - f(x_i))^2 = \Sigma_{i=1..n} (y_i - (a_1 + bx_i))^2$

- We minimize *SSE* to find a_1 and b
  - Statistical software, e.g., Minitab, can be used to find a_1 and b
  - However, the intercept *a1* (when *x=0*) is not very meaningful, e.g., let *x* and *y* be the height and weight. Then a_1 is the weight of a person with *height = 0.*
- Let $f(x) = a + b(x-\bar{x})$, where $\bar{x}$ is the average of *x* values
  - The intercept *a* is the predicted weight of the person with average height.

Least Squares Estimate

- The least squares regression line is:

$$f(x_i) = a + b(x_i - \bar{x})$$

with least squares estimates:

$$a = \bar{y} \quad \text{and} \quad b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- *a* is an estimate of *intercept*, and *b* is an estimate of the *slope*

Simple Linear Regression Model

- Consider the linear relationship between high school GPA x and scores on a college entrance exam Y, then $x-\bar{x}$ is the centered GPA
- Assumption 1: Within the entire population of college students, there is a linear relationship between the average entrance score E(Y) and the centered GPA $x-\bar{x}$.

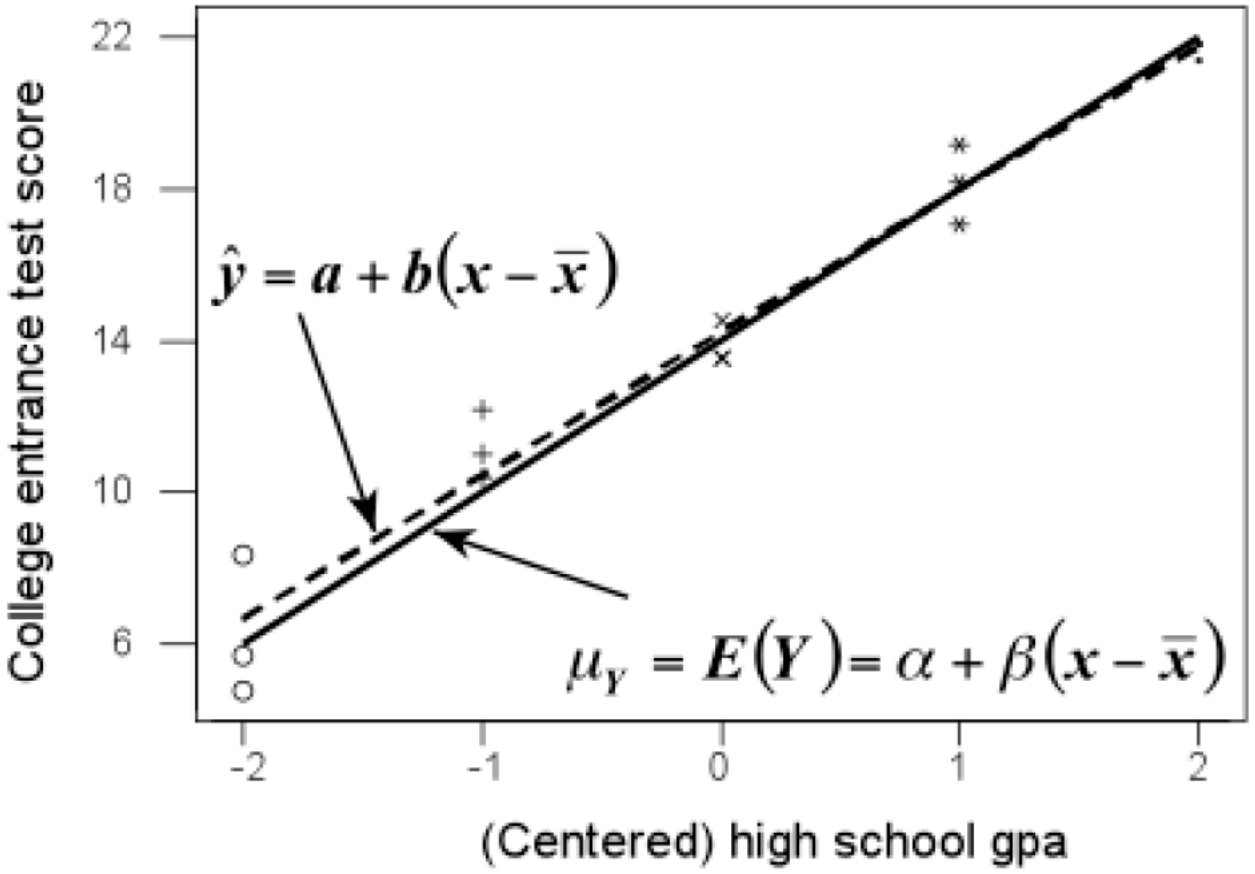
$$E(Y) = \alpha + \beta(x-\bar{x})$$

- Assumption 2: individual students deviate from the mean entrance score of students with same centered GPA by some unknown amount ϵ_i.

$$Y_i = \alpha + \beta(x-\bar{x}) + \epsilon_i$$

Simple Linear Regression Model

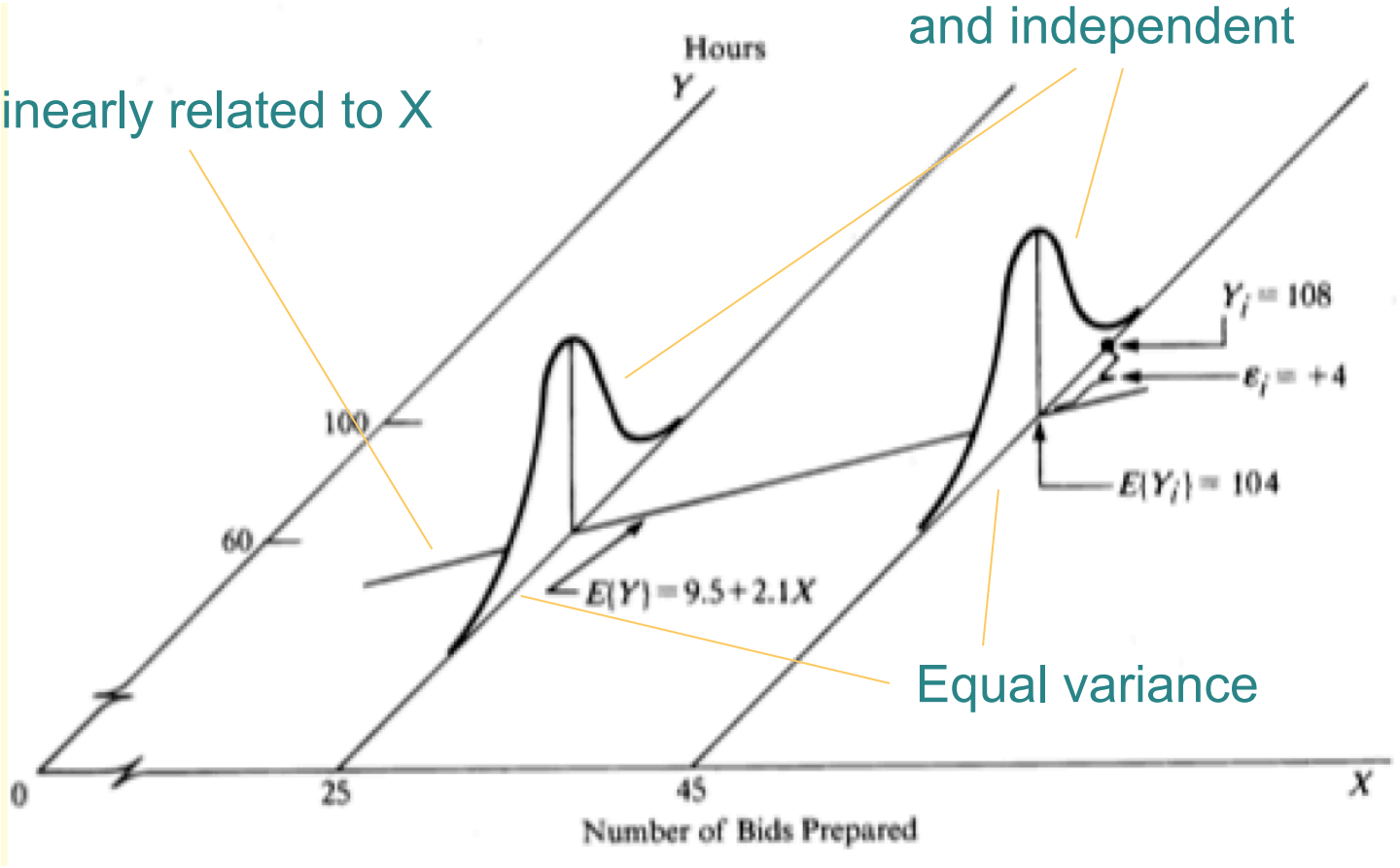
- As we do not have the data for the whole population, we cannot derive α and β.
- Best way is to estimae α, β using *random samples*.



Conditions for Simple Linear Regression Model

Error is normally distributed and independent

Y linearly related to X



Equal variance

Simple Linear Regression Model

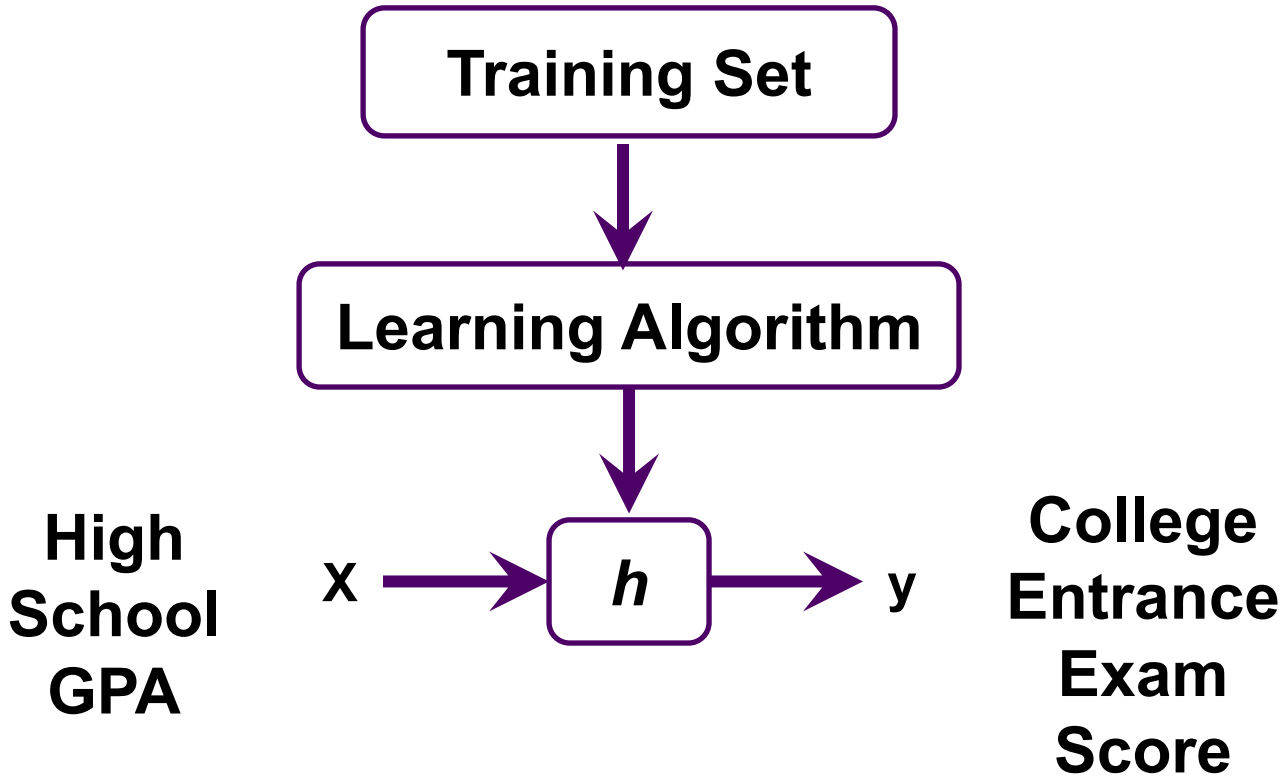
- If the four conditions hold true, then:

$$a = \bar{y} \quad \text{and} \quad b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- *a* and *b* are the maximum likelihood estimators of *α* and *β*

Model Representation

- Computer Science Point of View



- How to learn/represent the model h as a good predictor of y?

Univariate Linear Regression

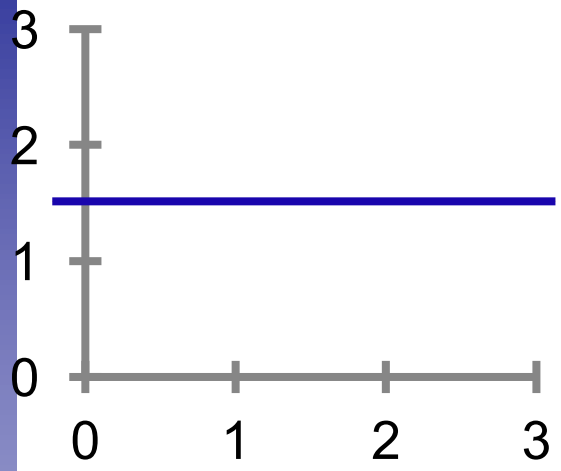
- Training Set:

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

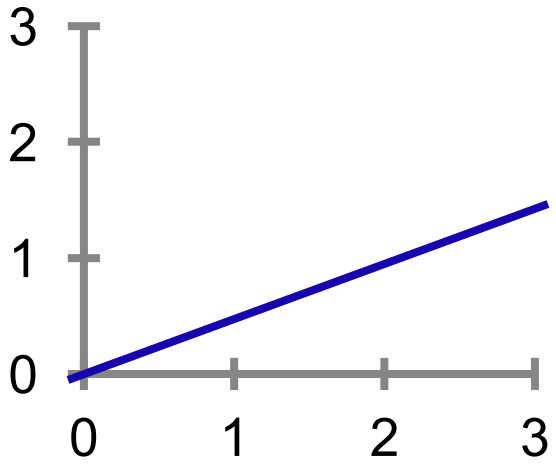
- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
- Parameters: θ_i
- Regression Problem: How to choose θ_i such that $h_\theta(x)$ best fits the training set.

Linear Functions

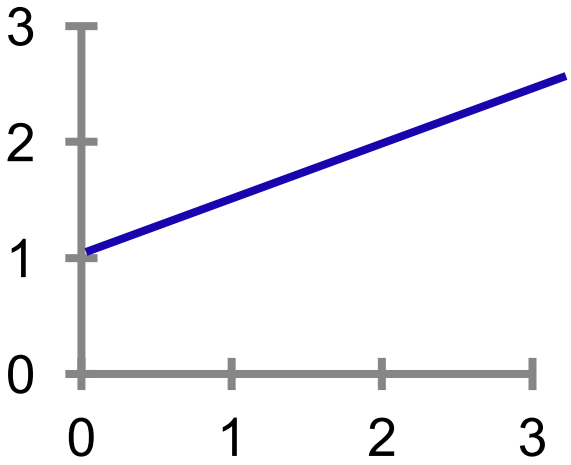
$$h_\theta(x) = \theta_0 + \theta_1 x$$



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



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



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

Linear Regression Learning

- Optimization problem: Choose θ (i.e., θ_0, θ_1) so that h_θ is close to y in our training examples (x, y)

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

(each represents one hypothesis function, i.e., one line.)

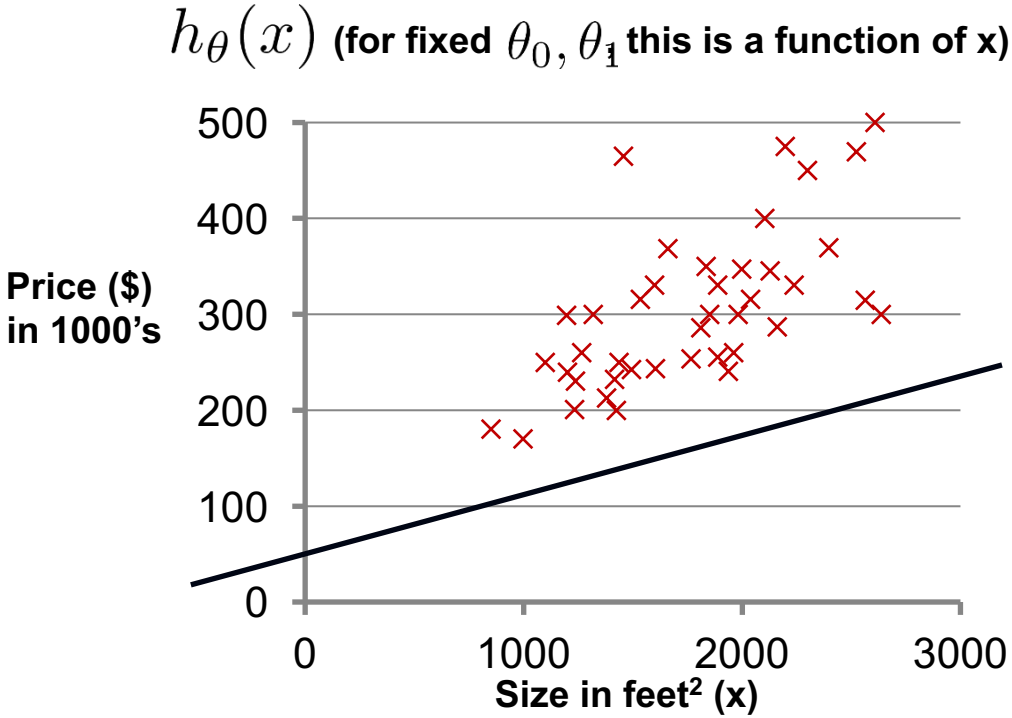
Objective (Cost) Function: Average Squared Errors

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$ (function of θ_0, θ_1)

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

Hypothesis vs. Objective (Cost)

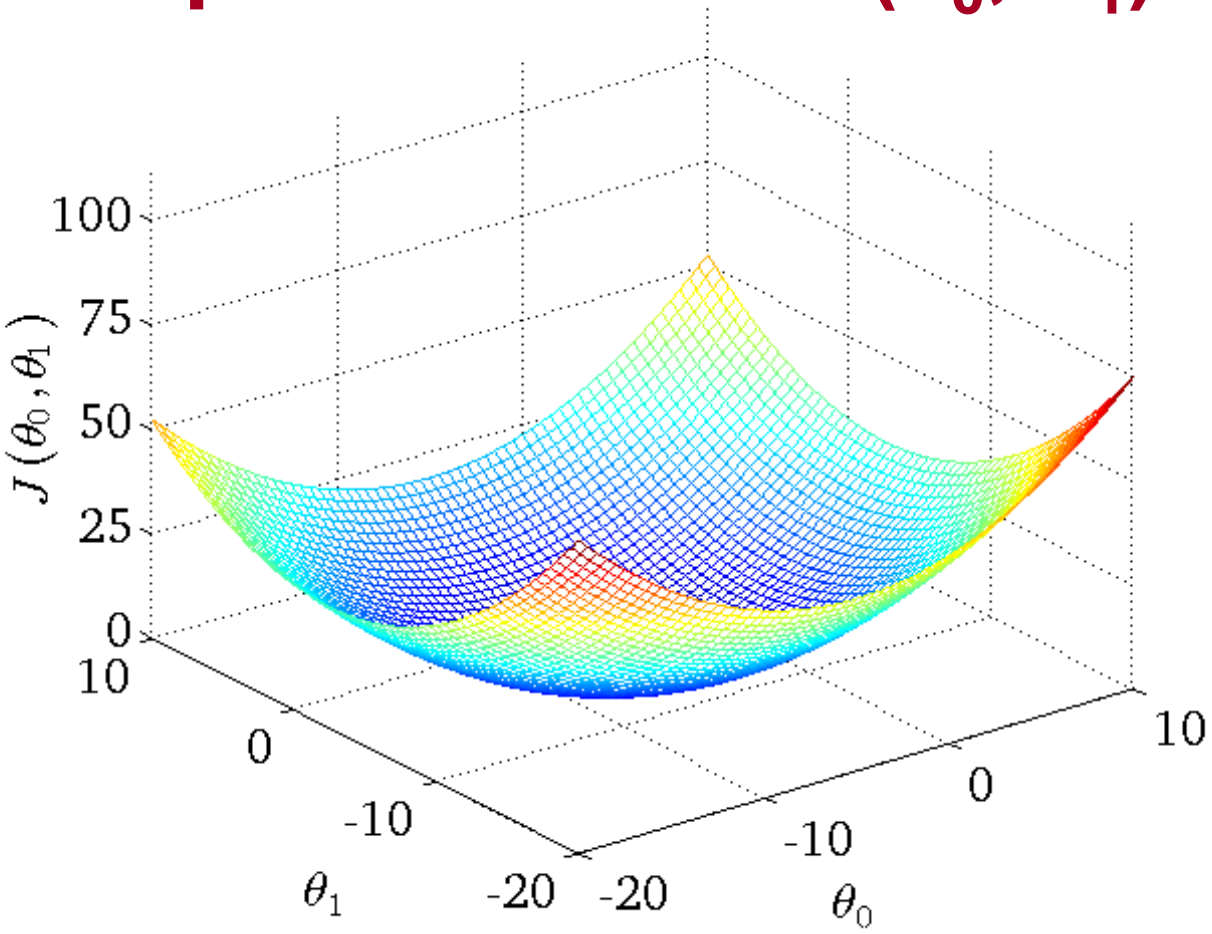
- Plot of Hypothesis function $h_\theta(x)$



$$h_\theta(x) = 50 + 0.06x$$

- How does the objective function $J(\theta_0, \theta_1)$ look like graphically?

Graphical Plot of $J(\theta_0, \theta_1)$

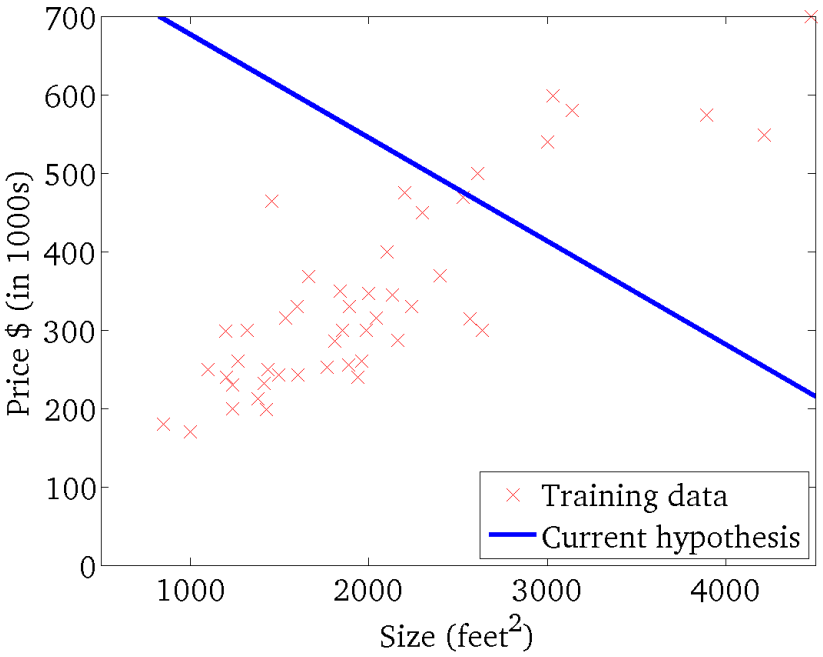


- Each θ (corresponding to a hypothesis function $h_\theta(x)$) is a point in the above figure of objective function $J(\theta_0, \theta_1)$

Contour Plot of $J(\theta_0, \theta_1)$

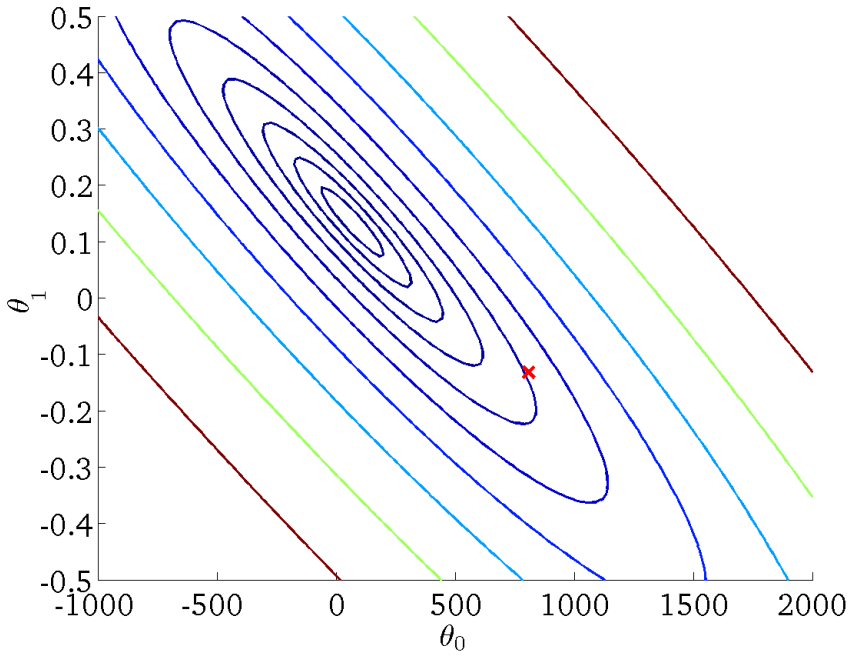
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



$J(\theta_0, \theta_1)$

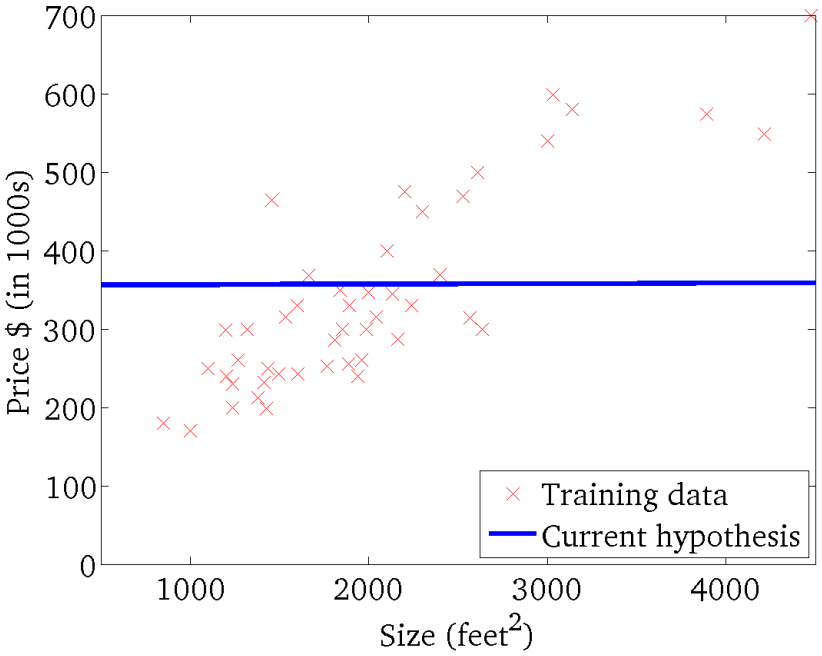
(function of the parameters θ_0, θ_1)



Contour Plot of $J(\theta_0, \theta_1)$

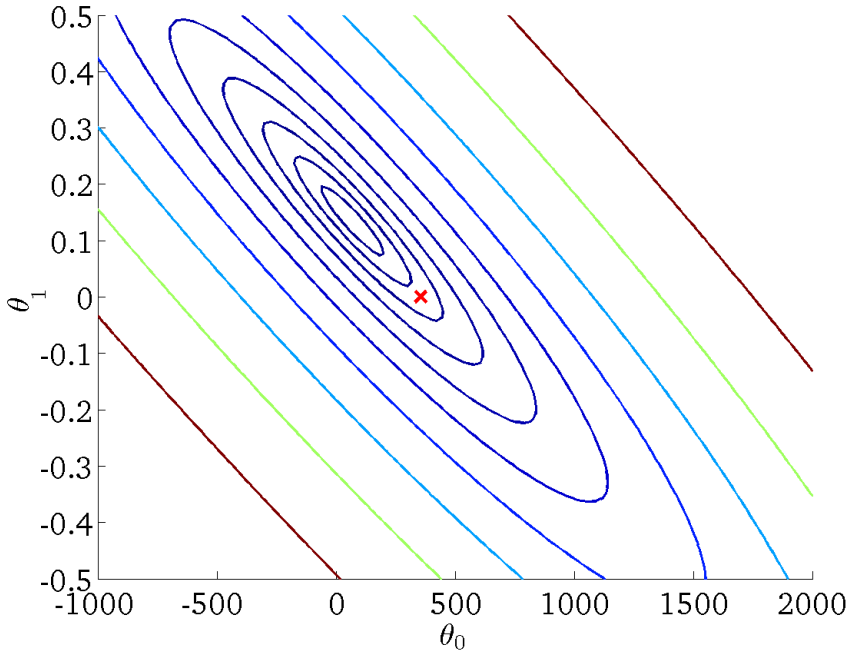
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

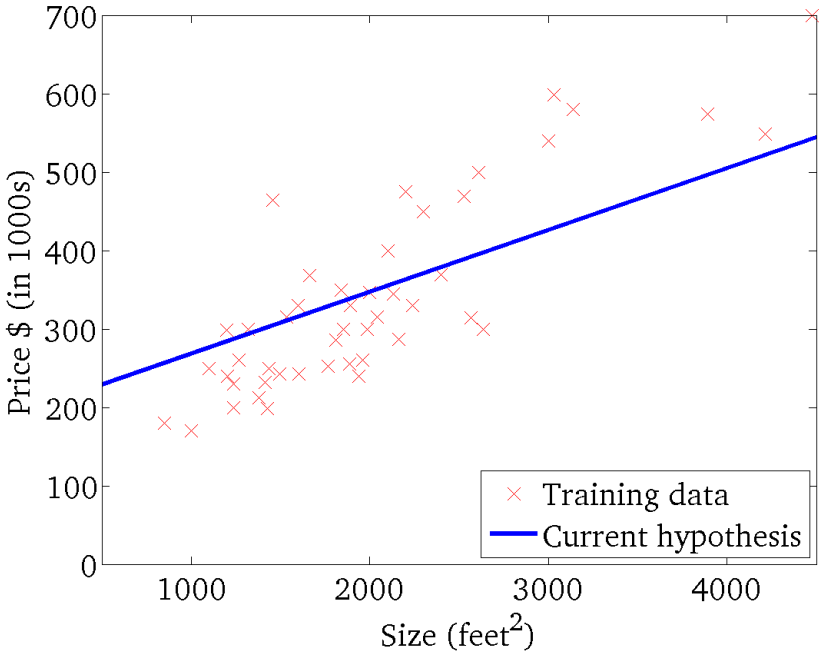
(function of the parameters θ_0, θ_1)



Contour Plot of $J(\theta_0, \theta_1)$

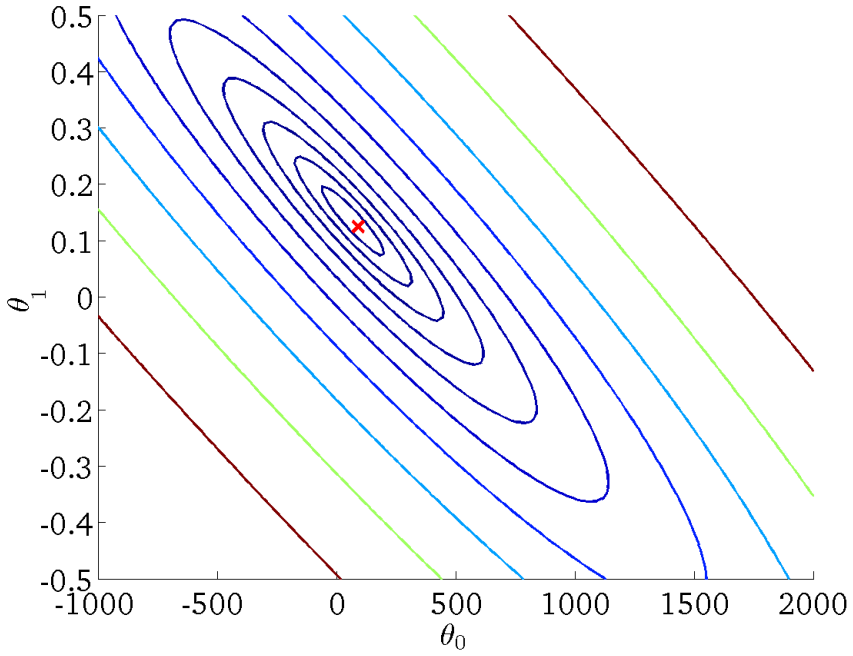
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



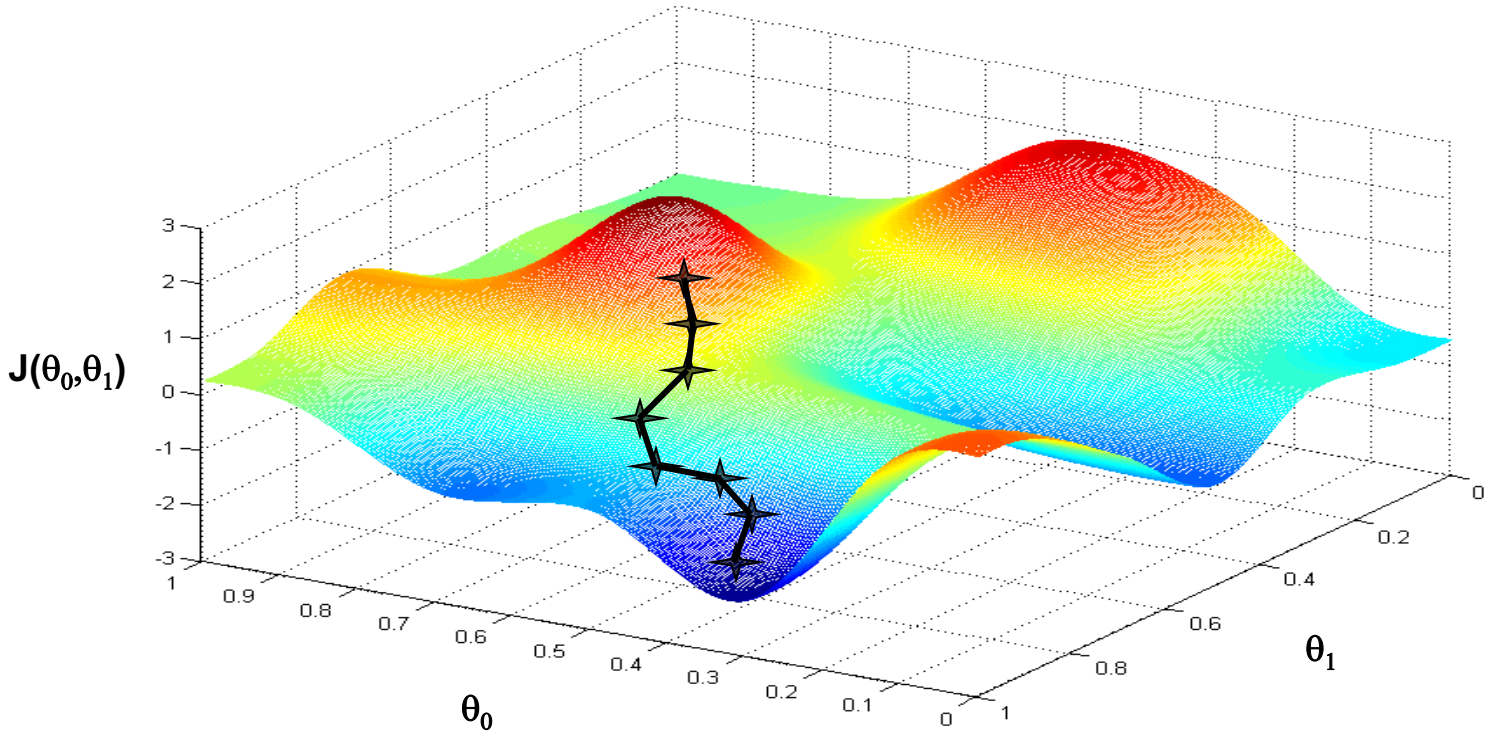
$J(\theta_0, \theta_1)$

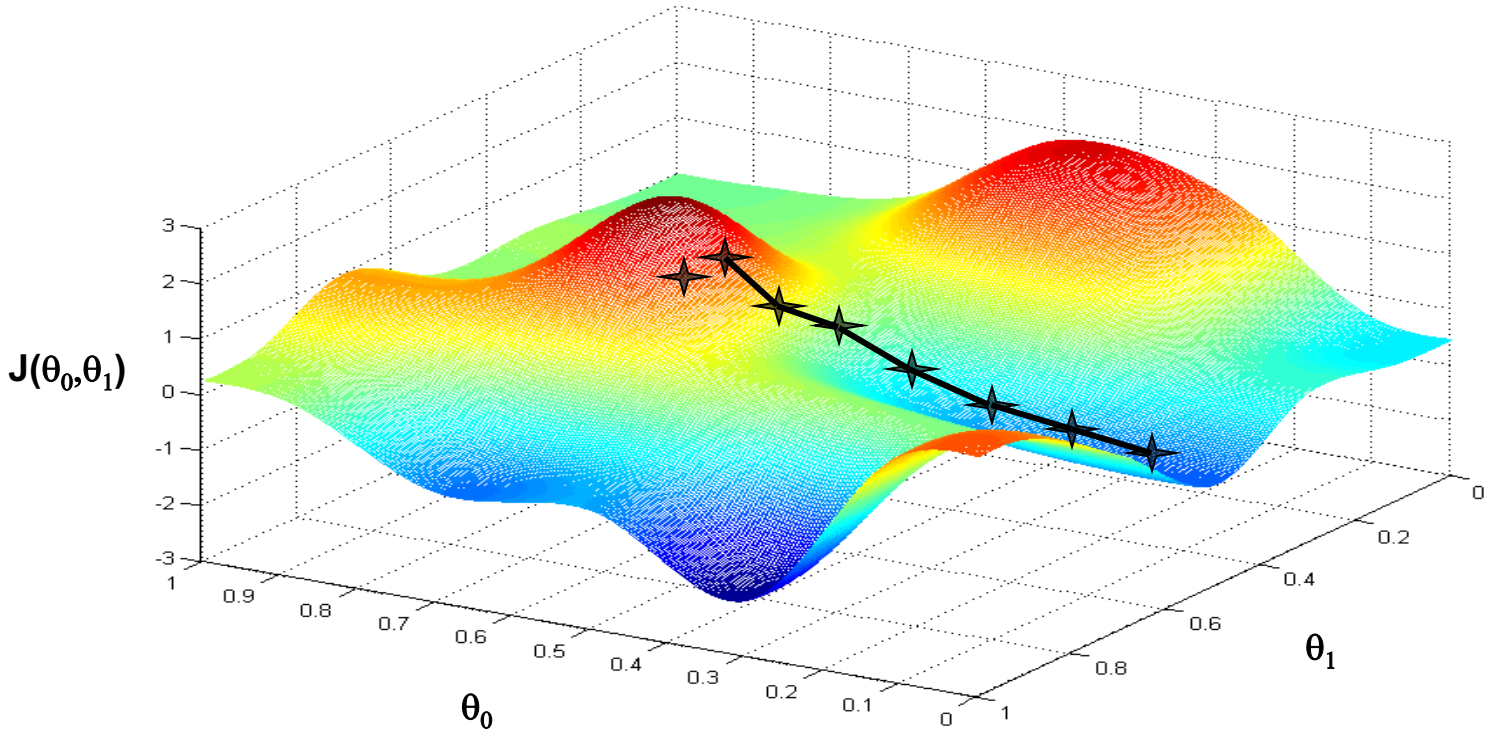
(function of the parameters θ_0, θ_1)



Gradient Descent

- Univariate Regression problem
  - Given: an objective function $J(\theta_0, \theta_1)$
  - Goal: $\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$
- Idea:
  1. Start with some point (θ_0, θ_1)
  2. Iteratively updating (θ_0, θ_1) towards reducing $J(\theta_0, \theta_1)$ until reaching a minimum point
- Applicable for more general minimization problems, not only for regression problems.


J(θ0,θ1)
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
θ0
0
0.2
0.4
0.6
0.8
1
θ1


J(θ0,θ1)
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
θ0
0
0.2
0.4
0.6
0.8
1
θ1

Gradient Descent Algorithm

- Repeat

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad \begin{array}{l}(\text{simultaneously update} \\ j = 0 \text{ and } j = 1)\end{array}$$

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

- α is the learning rate, which determines the size of each step
- Update of parameters

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

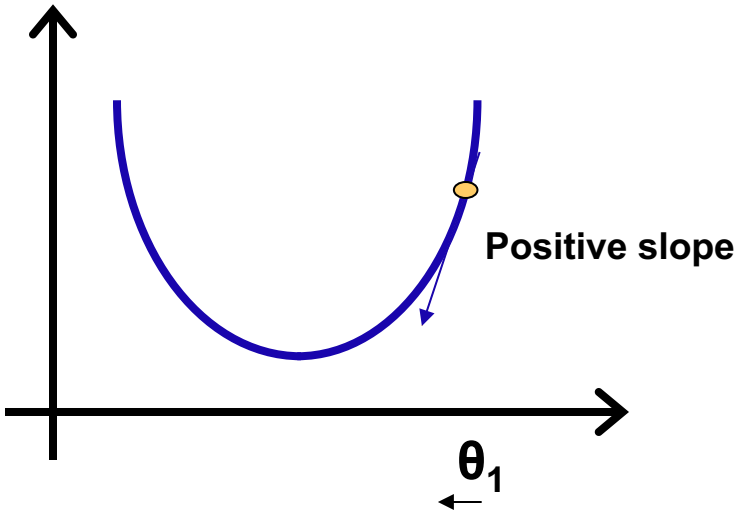
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

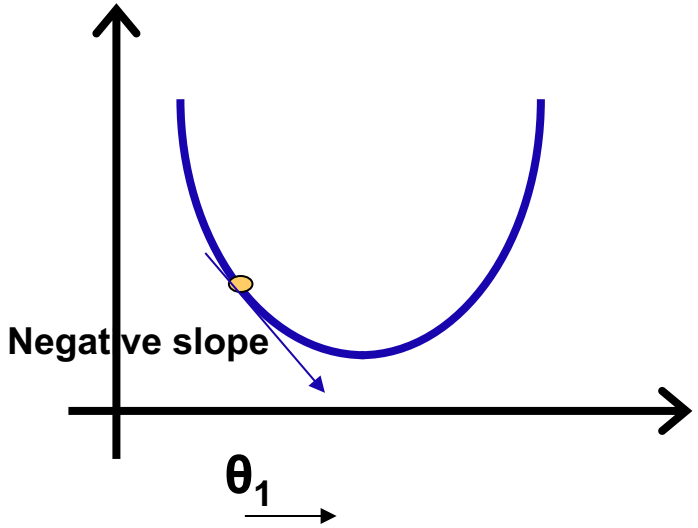
- Incorrect (why?)

Gradient Descent: Intuition

- Consider a simplified version of $J(\theta_1)$, i.e., $\theta_0,=0$

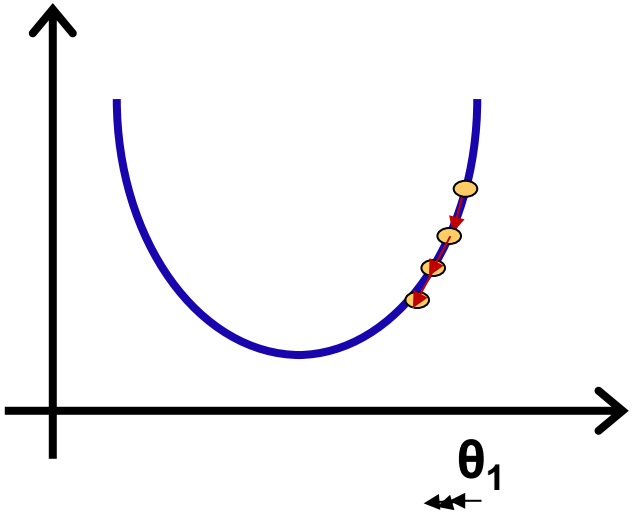
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

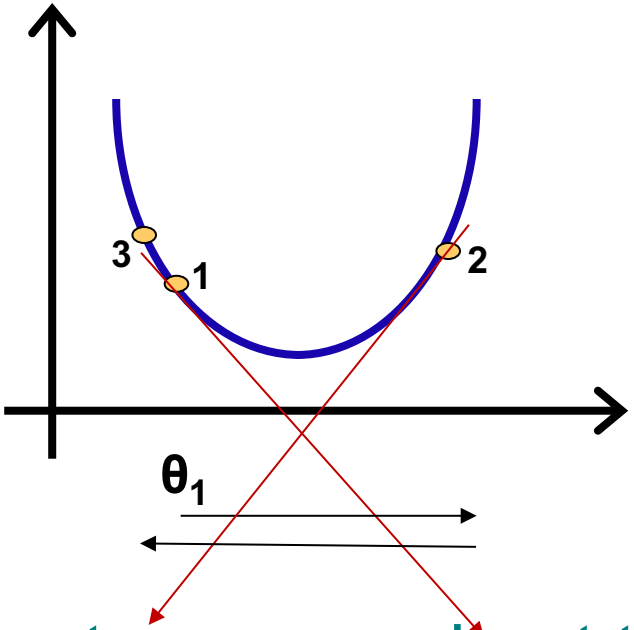




Effect of Learning Rate α

- Gradient descent can be very slow with a small *α*

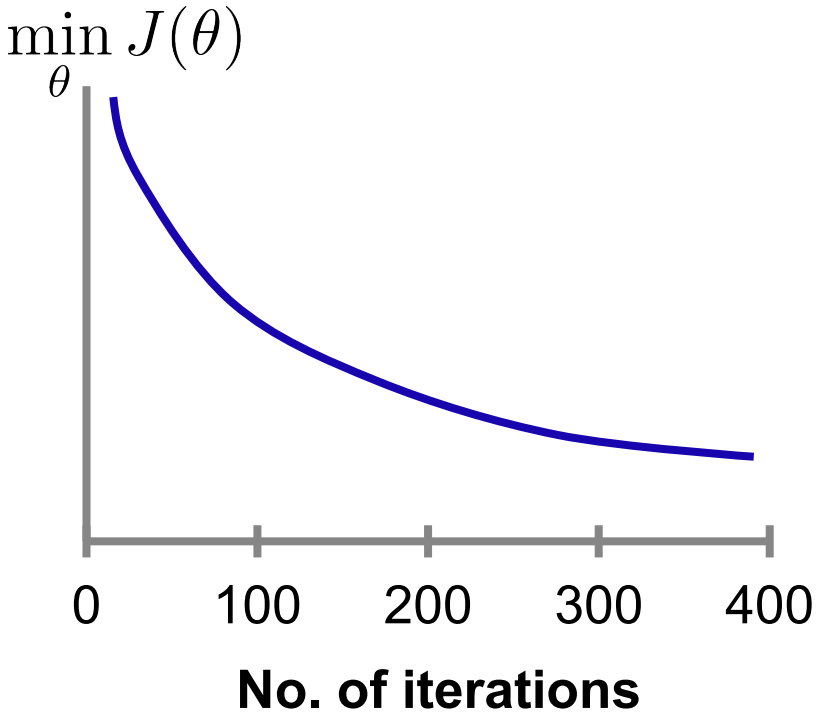


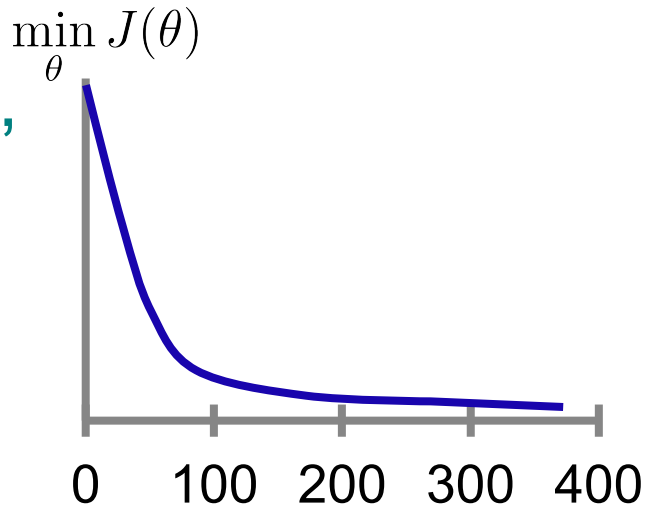


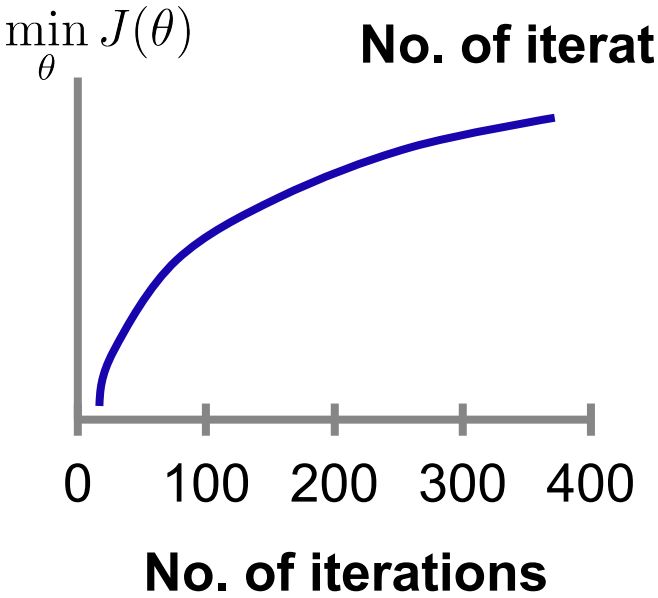
- With a large *α*, gradient descent can overshoot the minimum, and fail to converge, or even diverge.
- It's essential to find a proper learning date!

Setting of Learning Rate α

- To make sure gradient descent working correctly, *α* needs to be set properly.
- As the no. of iterations increases, $J(\theta_1)$ is expected to decreased.

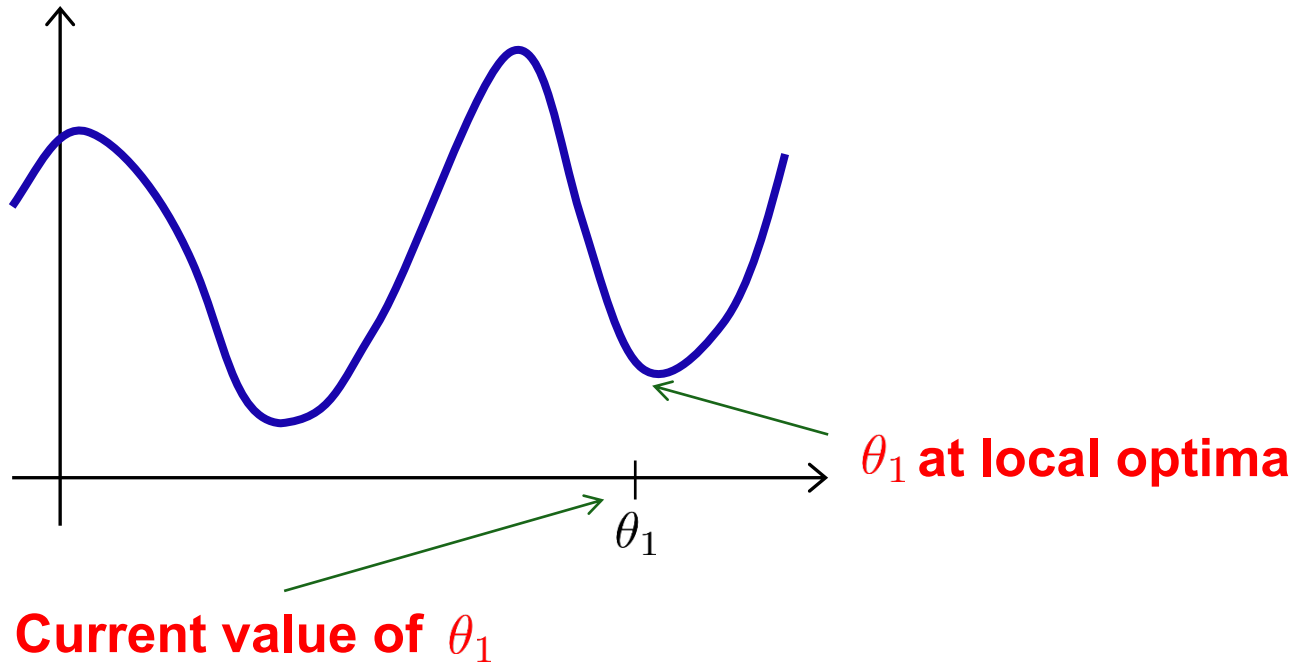






Local Minimum

- Gradient descent can converge to a local minimum, even with the learning rate α fixed.



- Gradient descent automatically takes *smaller* steps, when it approaches a local minimum,

Gradient Descent Algorithm for Linear Regression Model

- Gradient Descent Algorithm

  Repeat

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad \begin{array}{l}(\text{simultaneously update} \\ j = 0 \text{ and } j = 1)\end{array}$$

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

- Linear Regression Model

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

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

Gradient Descent Algorithm for Linear Regression Model

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

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

- Gradient Descent Algorithm for Linear Regression

  Repeat

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \qquad \text{(simultaneously update } j = 0 \text{ and } j = 1\text{)}$$

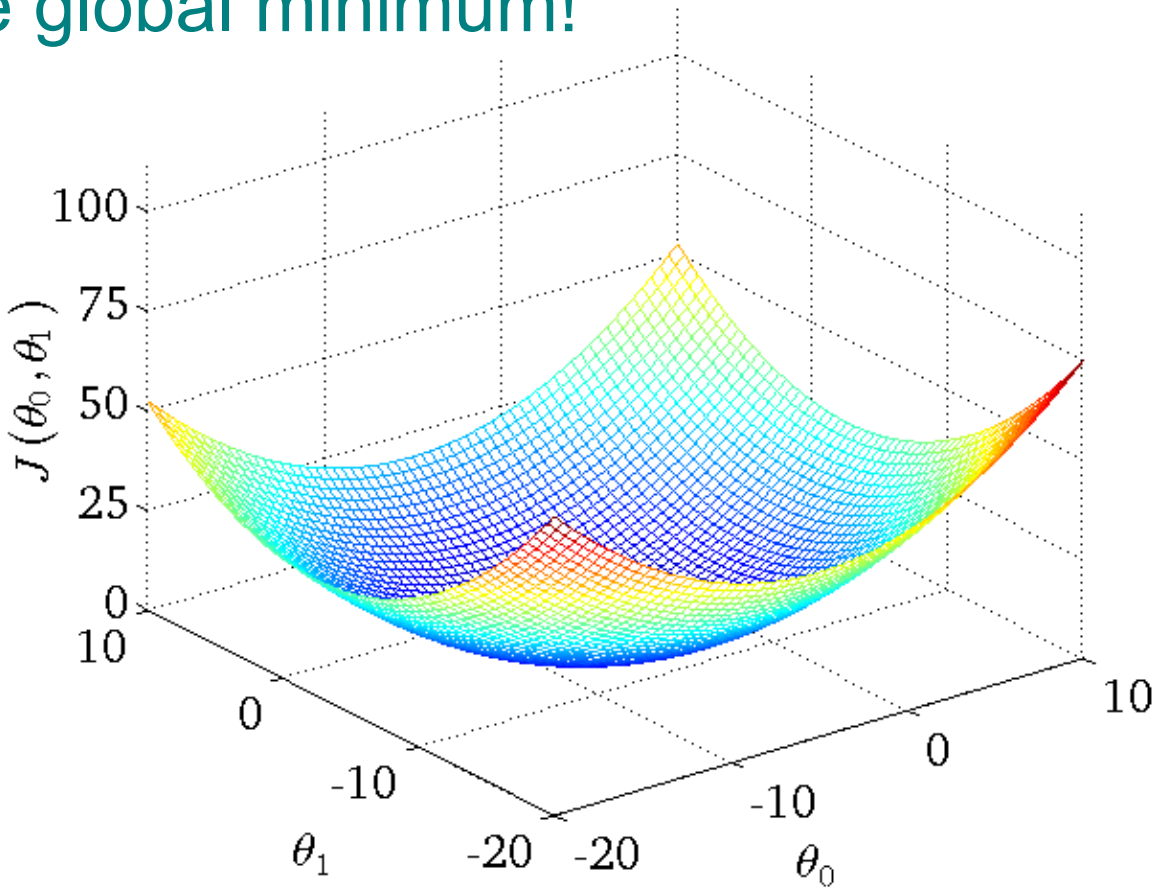
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}$$

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

- Each iteration involves all training samples

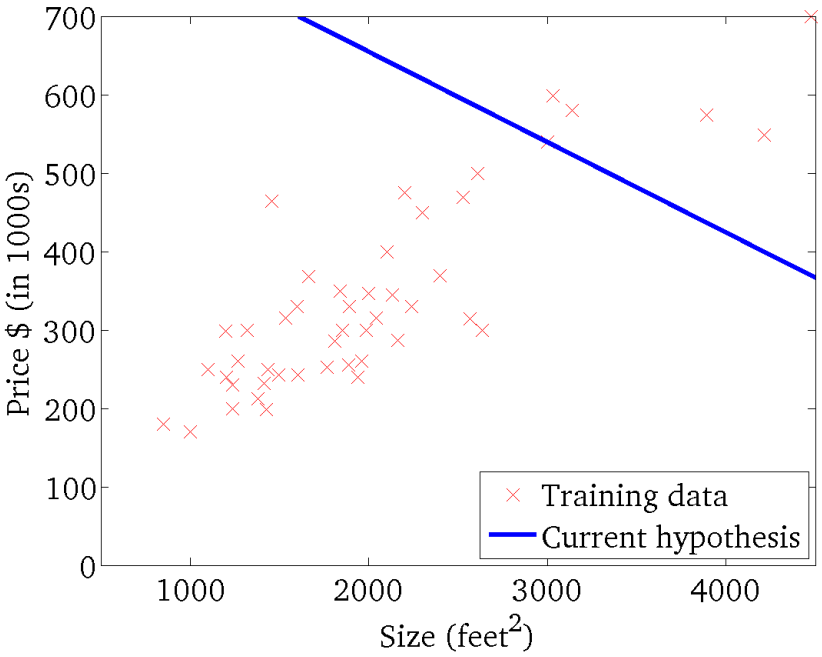
Convex Function

- Objective function for Linear Regression is a bowl-shaped convex function, i.e., there is only one global minimum!



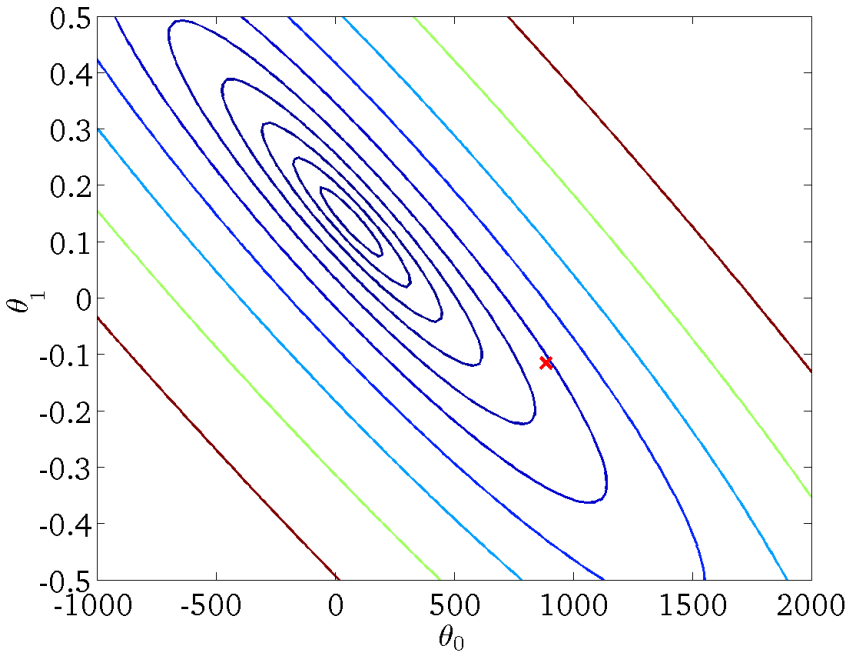
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



$J(\theta_0, \theta_1)$

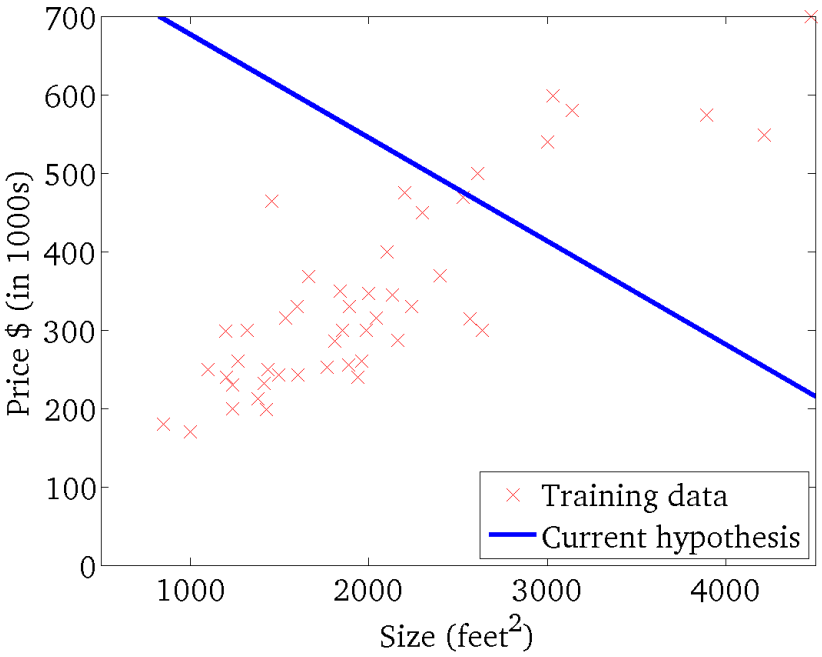
(function of the parameters θ_0, θ_1)



Initial point usually set at (0, 0) or randomly selected

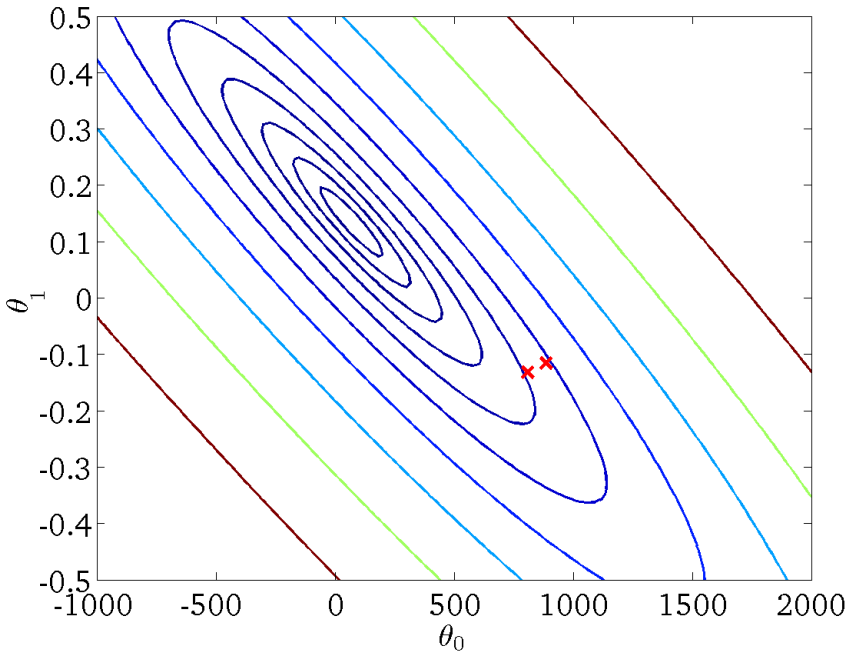
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



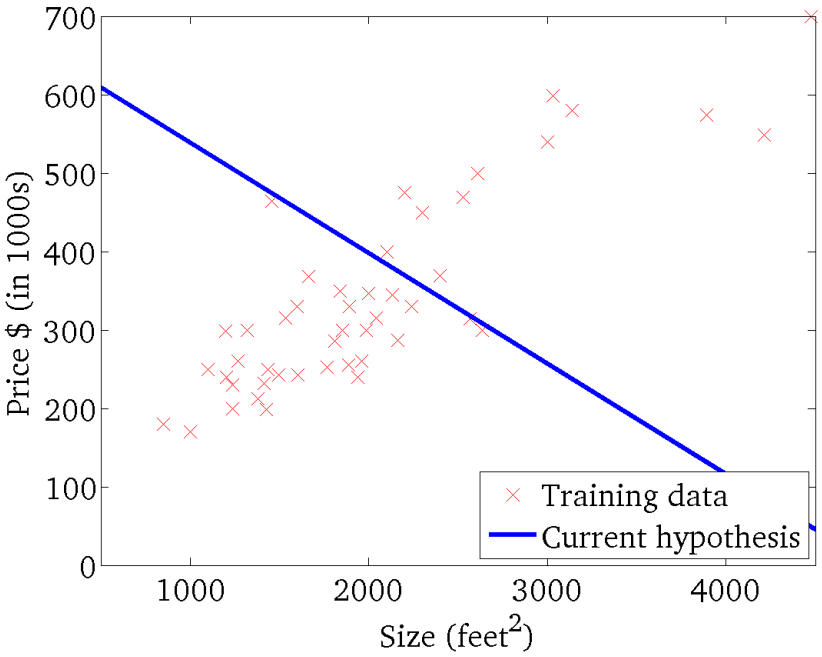
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



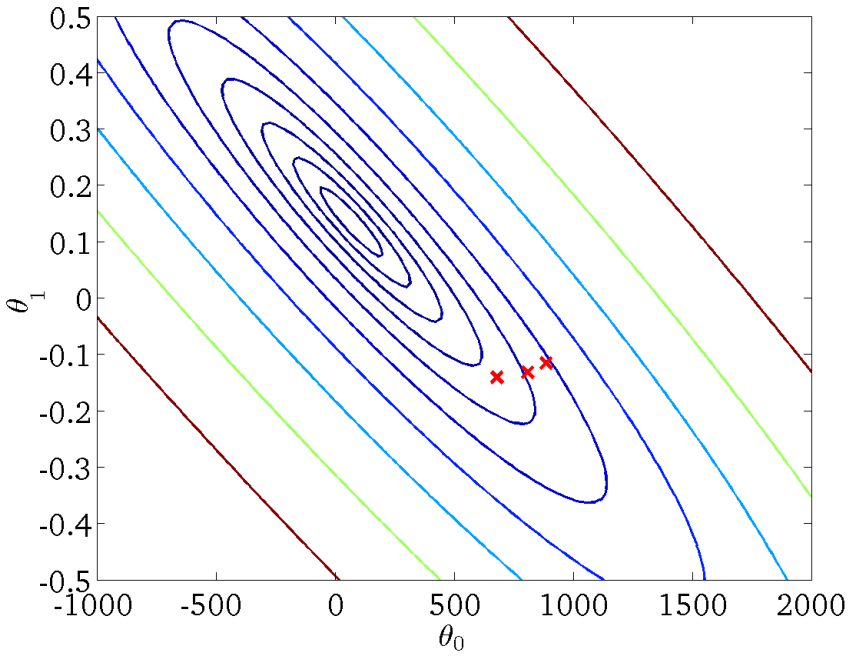
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



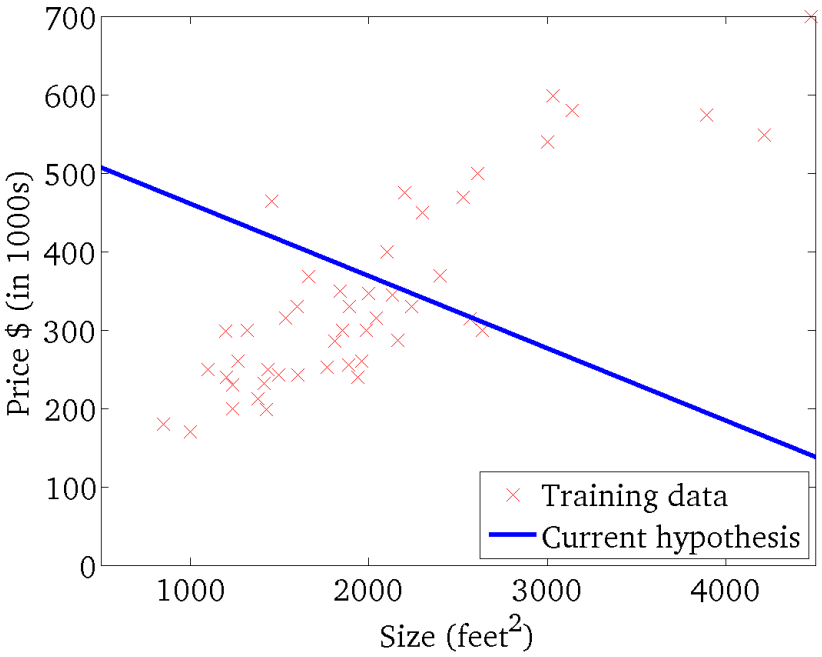
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



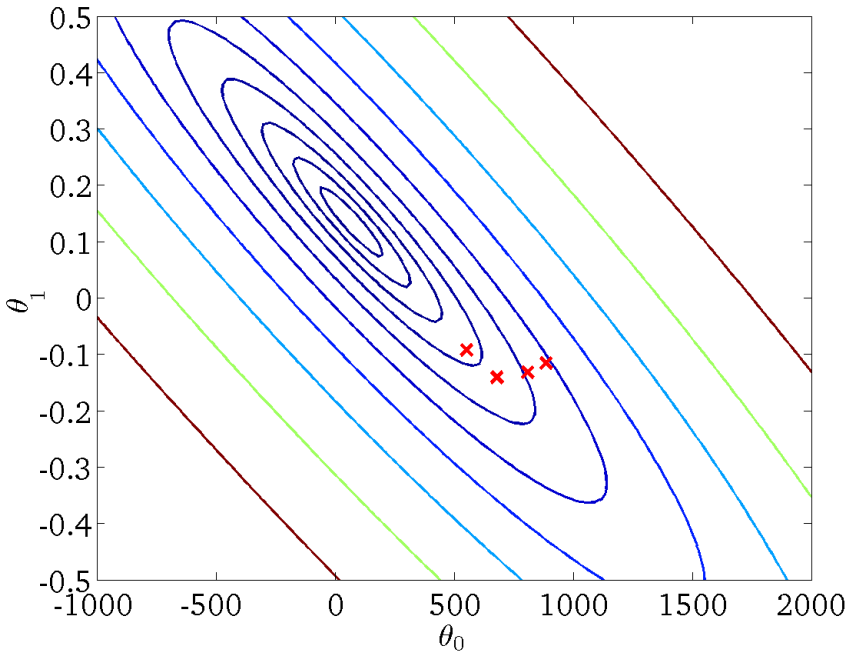
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



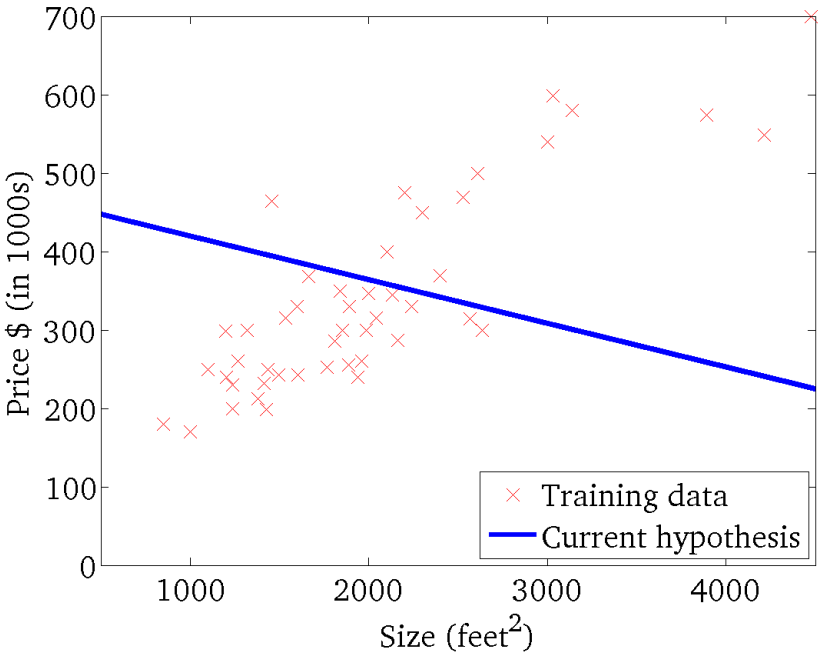
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



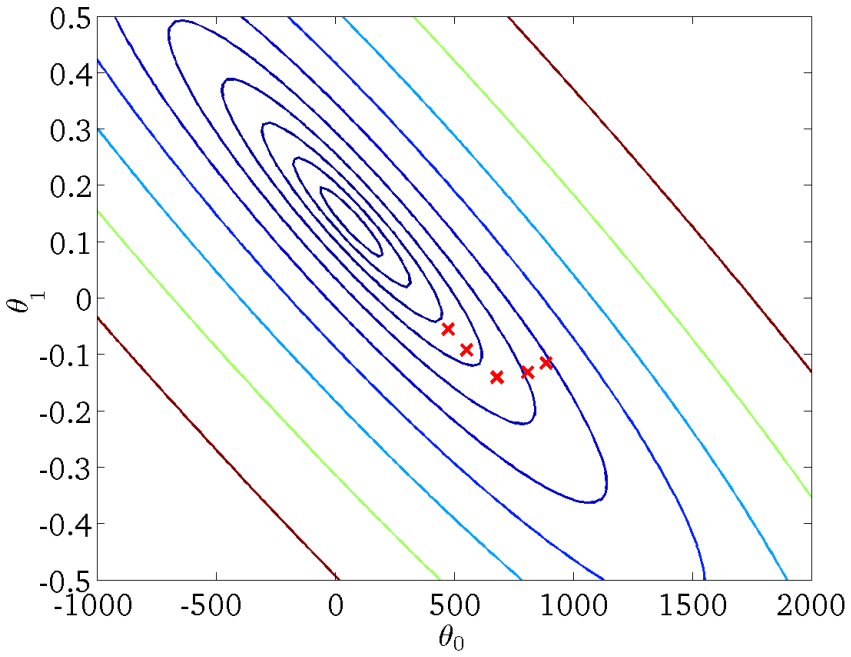
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



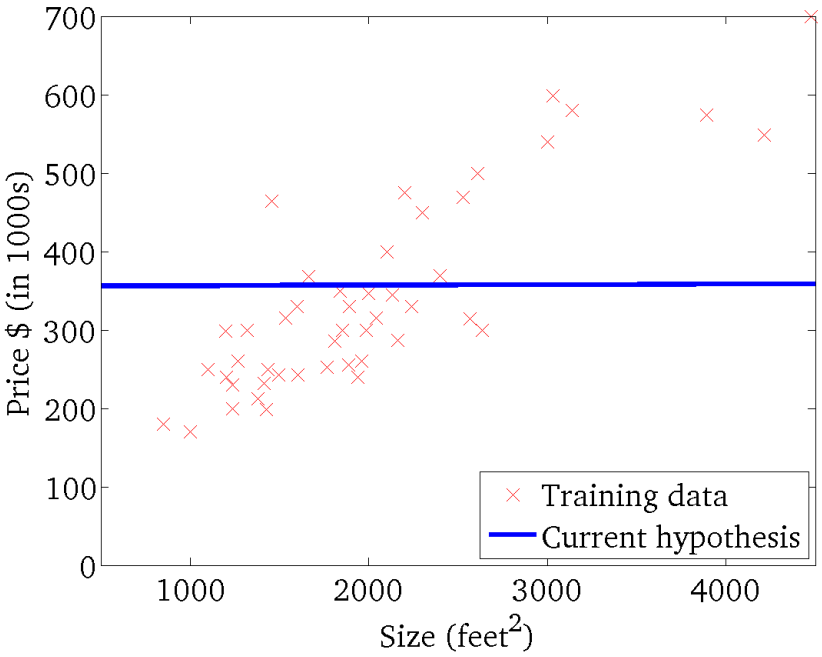
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



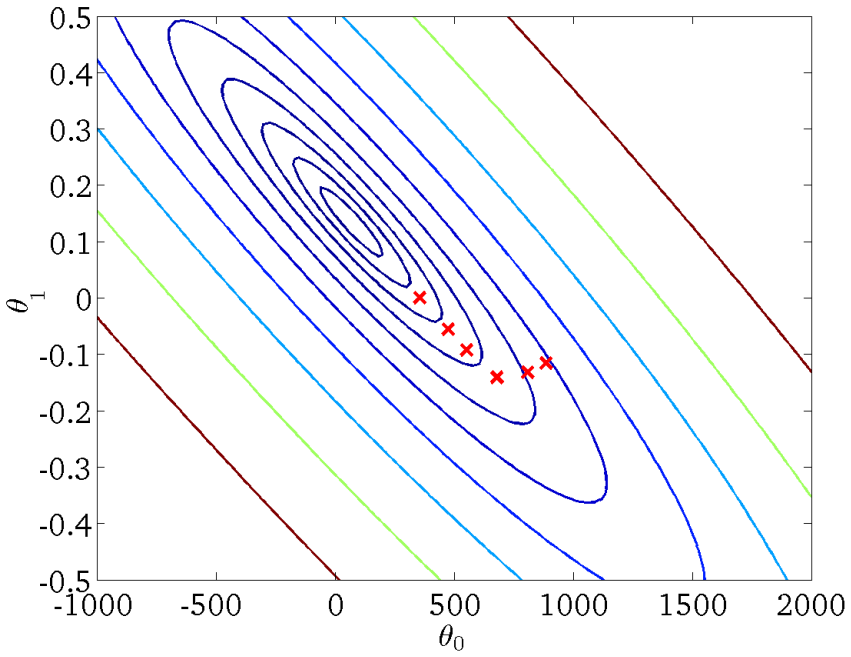
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



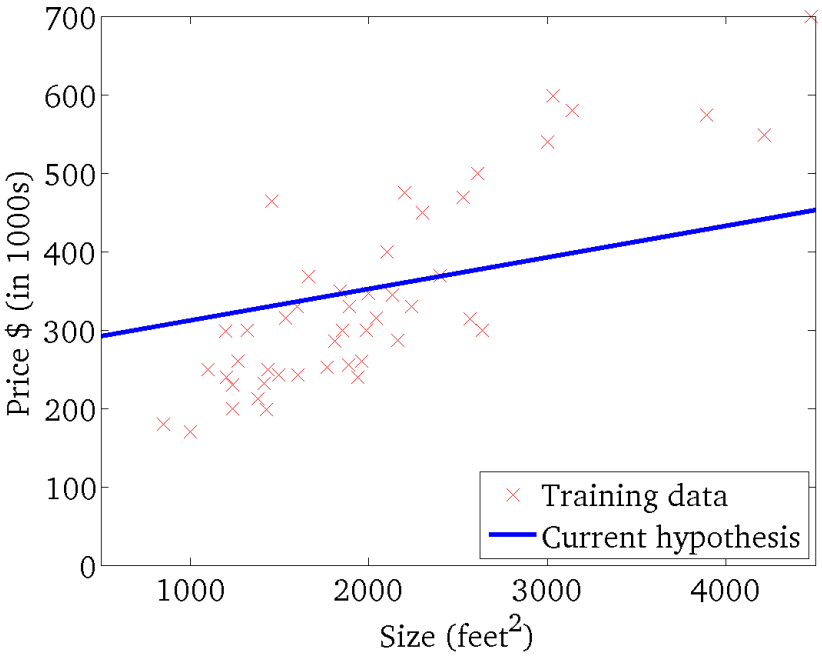
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



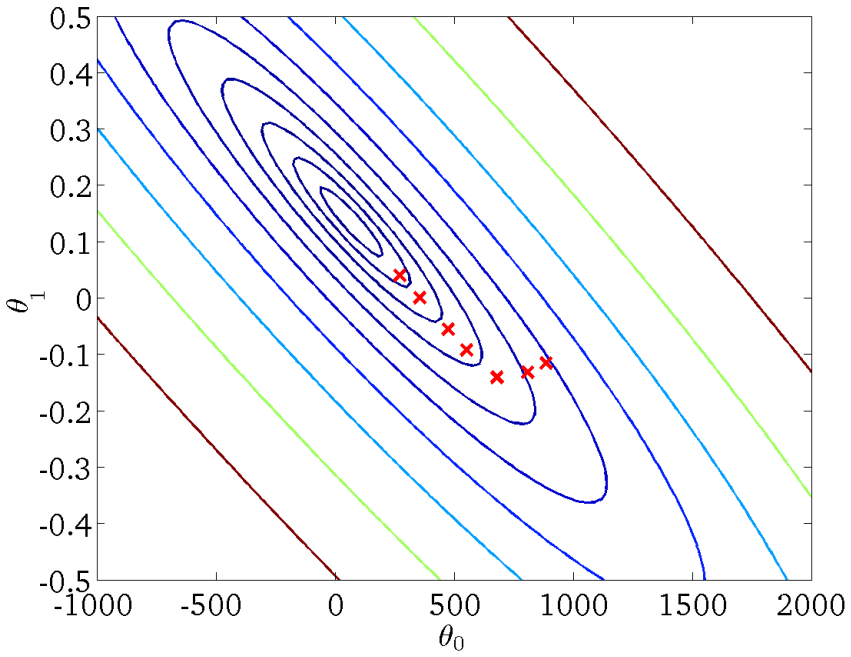
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



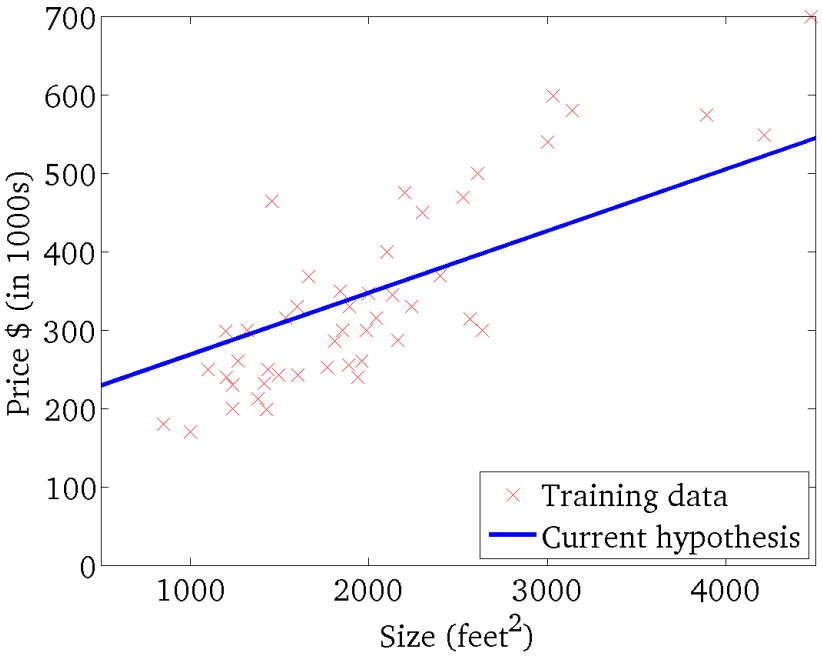
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



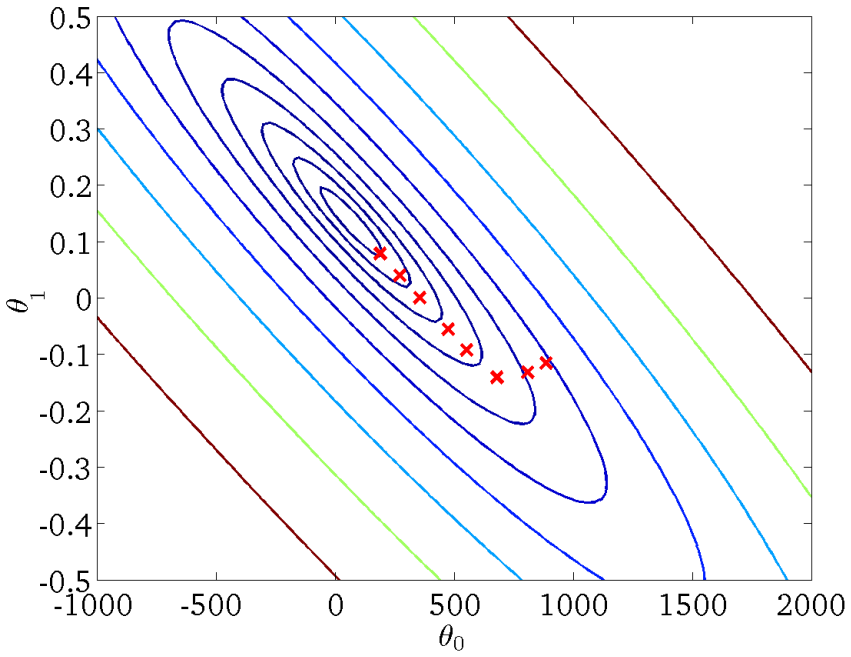
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



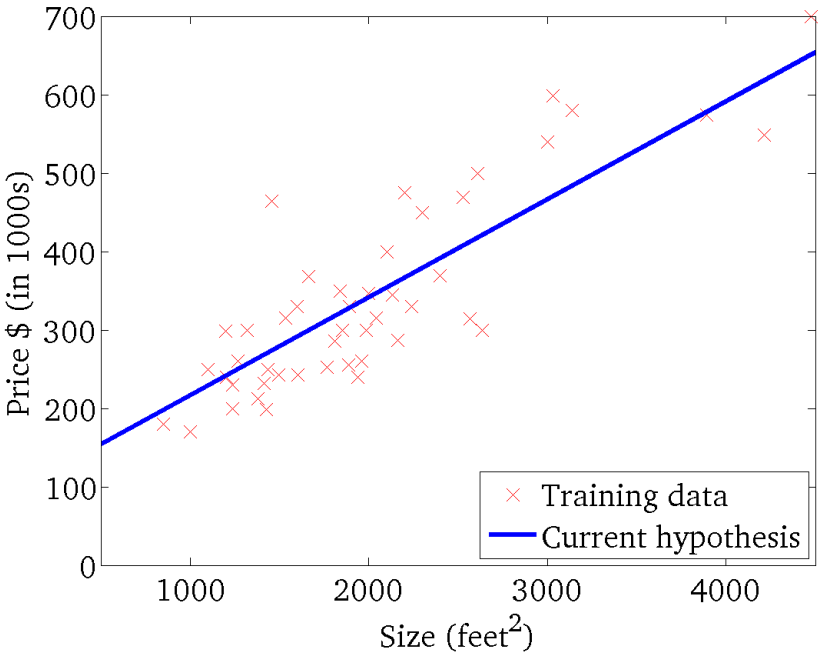
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



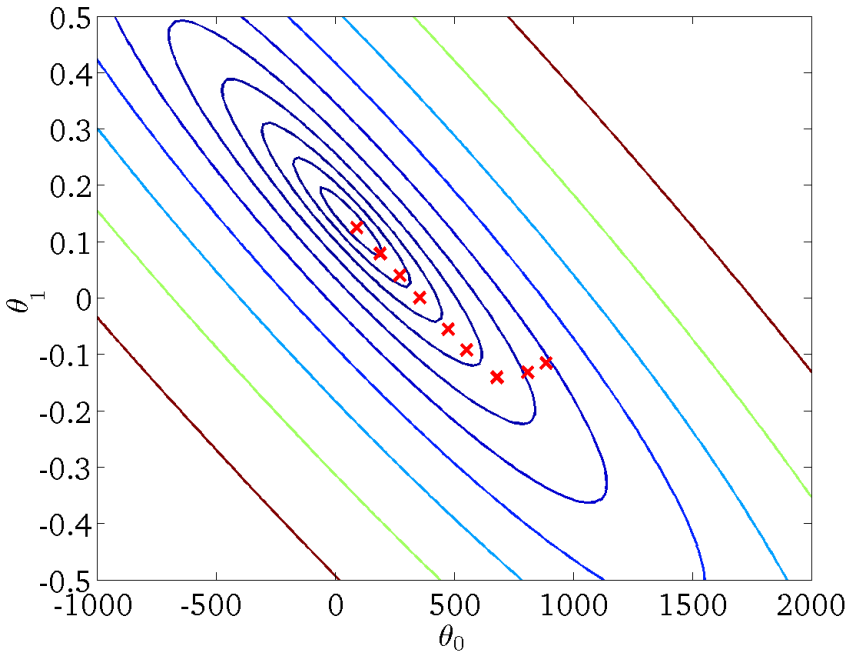
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



Alternative Approach

- Normal Equation: a numerical solution for solving linear regression equations without relying on the iterative approach of gradient descent.
- Gradient descent scales better than normal equation approach on large datasets

Normal Equation

- Consider the objective function *J(θ)* for an arbitrary *θ*.
  - *J(θ)* is quadratic

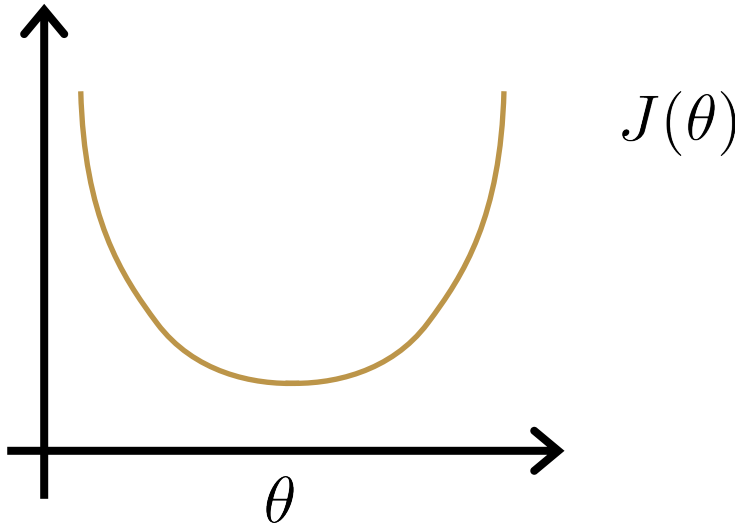
$$J(\theta) = a\theta^2 + b\theta + c$$

  - To solve *θ*, we take

$$\frac{\partial}{\partial\theta} J(\theta) = 0$$

  - Solution:

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



Design Matrix: Univariate Regression

- Predict housing price by size of the house

Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

- Design Matrix: X

$$X = \begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

Gradient Descent vs Normal Eq.

- Gradient descent (GD) is a iterative approach, while normal equation (NE) is numerical.
- GD needs to set a proper learning rate α but GD does not.
- GD needs perform normalization; NE does not
- GD works well even when the dataset is very large, but NE does not work that well.
  - NE needs to compute $(X^TX)^{-1}$ which is expensive when data size is large
  - $O(kn^2)$ vs $O(n^3)$ where n is the size of dataset

Multivariate Linear Regression

- Multiple features

Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
$x^{(3)}$ 1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…
				y

x

- Notation:
  - n = number of features
  - $x^{(i)}$ = input feature of *i*-th training example
  - $x_j^{(i)}$ = value of *j*-th feature in *i*-th training example

Hypothesis function

- $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n = \theta^T x$
  - $x_0 = 1$ for convenience of notation
  - $x = [x_0 \; x_1 \; x_2 \ldots x_n]^T = [1 \; x_1 \; x_2 \ldots x_n]^T$
  - $\theta = [\theta_0 \; \theta_1 \; \theta_2 \ldots \theta_n]^T$

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

Multivariate Linear Regression Learning

- Optimization problem: Choose θ ($\theta_0, \theta_1, \ldots \theta_n$) so that h_θ is close to y in our training examples ($\boldsymbol{x}$, y)

Hypothesis:

$$h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Parameters:

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

Objective (Cost) Function:

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

Goal: $\underset{\theta}{\text{minimize}}\ J(\theta)$

Gradient Descent for Multivariate Linear Regression

- Repeat

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$x_0 = 1$

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

(simultaneously update θ_j for $j = 0, \ldots, n$)

Data Preprocessing

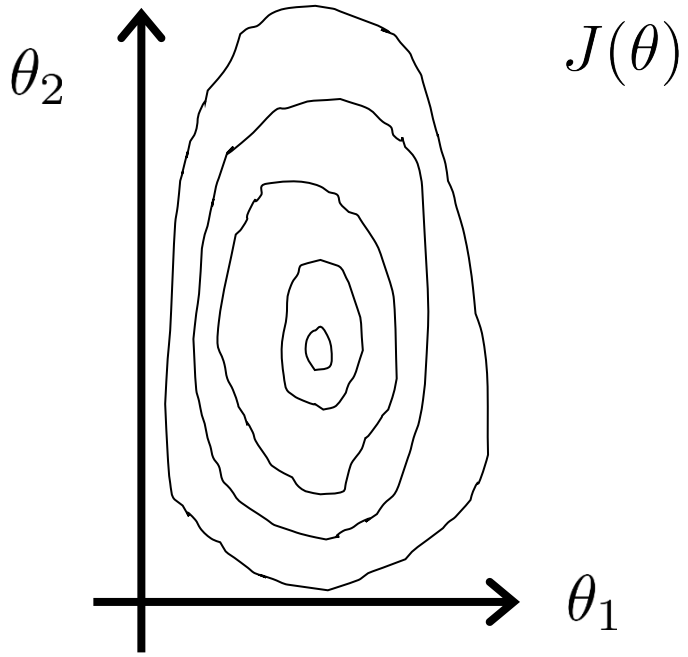
- Different scales of features/attributes Consider the housing dataset:

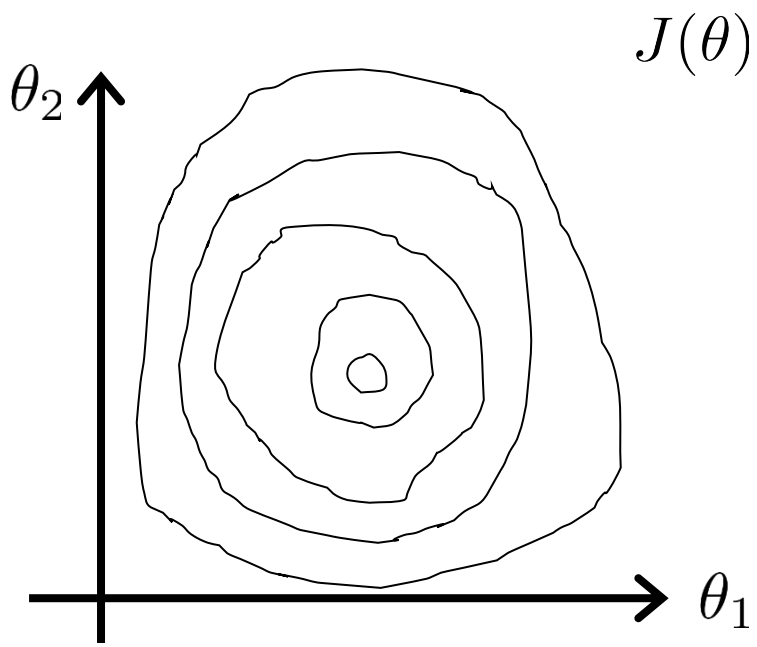
Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

- The regression coefficients θ_1, θ_2… *and so on* corresponding to features of different scales are expected to have different value ranges

Feature Normalization

- Gradient descent may be slowed down significantly.
- Descent slower in dimension of larger scale range





- Normalization may bring scales into similar scale.

Feature Scaling

- Get every feature into approximately a [-1,1] range.
- Feature scaling divides the feature values by the range (i.e. the maximum value minus the minimum value) of the input variable.
- For example,
  - x_1 = *size (0-2000 feet^2)*

    → x_1 = *size /2000*
  - x_2 = *number of bedrooms (1-5)*

    → x_2 = *number of bedroom/5*

Mean Normalization

- Get every feature into approximately a [-0.5,0.5] range.
- Mean Normalization replaces the feature value x with (x- μ) divided the range (i.e. the maximum value minus the minimum value) of the input variable, where μ is the average feature value.
- Resulted features have approximately zero mean.
- For example,
  - x_1 = *size (0-2000 feet^2)* → x_1 = *(size-1000) /2000*
  - x_2 = *number of bedrooms (1-5)*

    → x_2 = *(number of bedroom-2)/5*

Polynomial Regression

- Housing prices prediction
  - $h_\theta(x) = \theta_0 + \theta_1 Width + \theta_2 Depth$
- Feature creation:
  - $Area = Width * Depth$
  - $h_\theta(x) = \theta_0 + \theta_1 Area$
- Depends on domain knowledge of the application
  - $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ (Quadratic)
  - $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$ (Cubic)
  - $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ (Linear)

  $where\ x_1 = size\ \ x_2 = size^2\ \ x_3 = size^3$
- The tricks of linear regression can be applied to handle polynomial regression

Normal Equation (General Case)

- Objective function J(θ) $\theta \in \mathbb{R}^{n+1}$

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

  - To solve θ, for every j we take

$$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots = 0$$

  - Solution:

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

Design Matrix: Multivariate Regression

	Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
x_0	x_1	x_2	x_3	x_4	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

- Design Matrix X

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \\ 1 & 3000 & 4 & 1 & 38 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \\ 540 \end{bmatrix}$$