Machine Learning ZG565

BITS Pilani
Pilani Campus

Dr. Sugata Ghosal, PhD
Sugata.ghosal@pilani.bits-pilani.ac.in

innovate achieve lead

BITS Pilani

Pilani Campus

Lecture No. – 3 | Linear Regression

Date – 26/11/2022

Time – 4:15 PM – 6:15 PM

Session Content

- What is Linear regression
- Direct Method for linear regression
- Iterative Method based on gradient descent
- Avoiding overfitting
- Linear Basis Function Models
- Accuracy Analysis

Regression

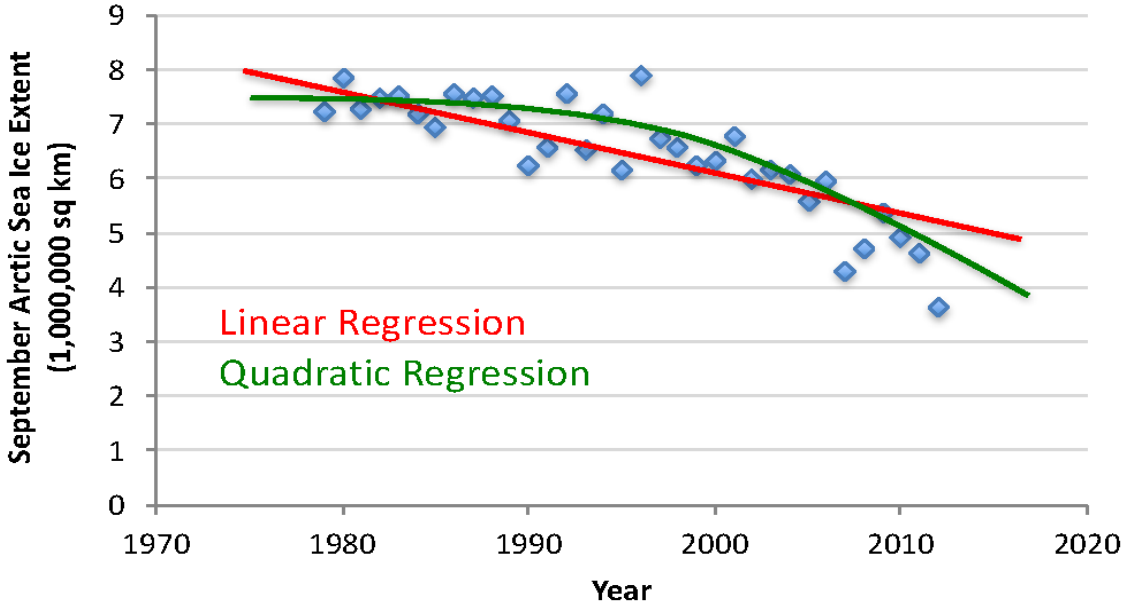
- Wish to learn a function f :: X → Y, where predicted output Y is real, given the n real training instances $\{<x^1,y^1>\ldots<x^n,y^n>\}$.

- Examples include
  - predict weight from gender, height, age, …
  - Predict house price from locality, area, income, …
  - predict Google stock price today from Google, Yahoo, MSFT prices yesterday
  - predict each pixel intensity in robot's current camera image, from previous image and previous action

Least Squares Approach

Given:

- Data $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\right\}$ where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$
- Corresponding labels $\boldsymbol{y} = \left\{y^{(1)}, \ldots, y^{(n)}\right\}$ where $y^{(i)} \in \mathbb{R}$



Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

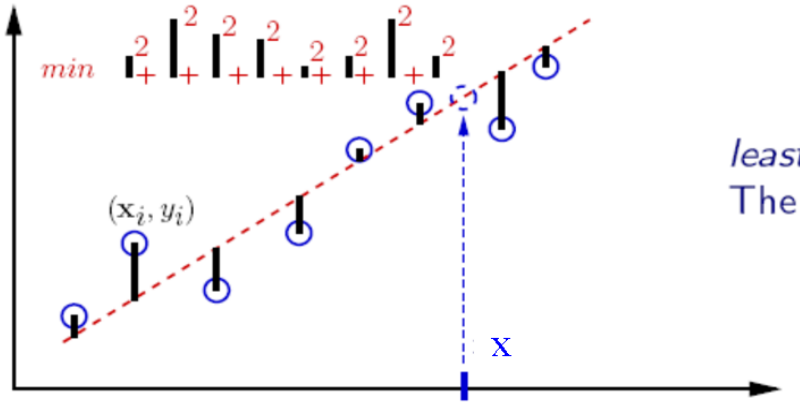
Linear Regression

- Hypothesis:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d = \sum_{j=0}^{d} \theta_j x_j = h_{\boldsymbol{\theta}}(x)$$

Assume $x_0 = 1$

- Fit model by minimizing sum of squared errors



least squares (LSQ)
The fitted line is used as a predictor

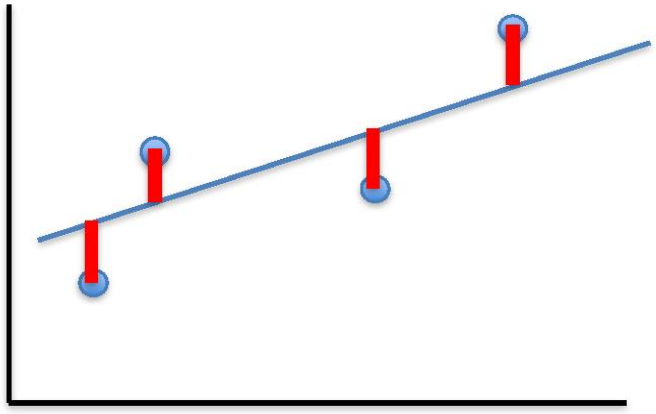
Figures are courtesy of Greg Shakhnarovich

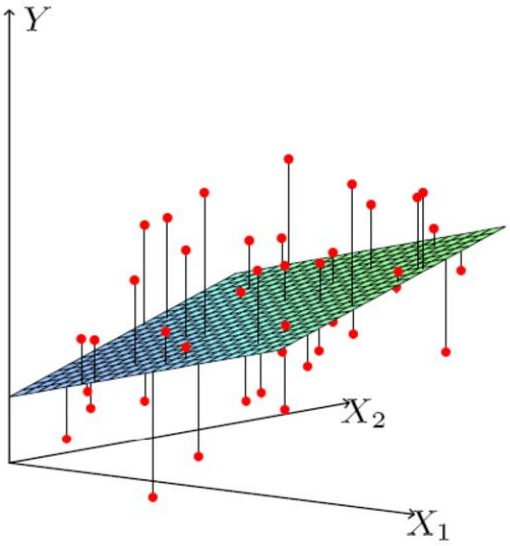
Least Squares Linear Regression

- Cost Function

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

- Fit by solving $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$





Least Squares Based Solution

- Benefits of vectorization
  - More compact equations
  - Faster code (using optimized matrix libraries)
- Consider our model:

$$h(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j x_j$$

- Let

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \qquad \boldsymbol{x}^\mathsf{T} = \begin{bmatrix} 1 & x_1 & \dots & x_d \end{bmatrix}$$

- Can write the model in vectorized form as $h(\boldsymbol{x}) = \boldsymbol{\theta} \cdot \boldsymbol{x}$

Least Squares Based Solution

- Consider our model for n instances:

$$h\left(\boldsymbol{x}^{(i)}\right) = \sum_{j=0}^{d} \theta_j x_j^{(i)}$$

- Let

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \quad \boldsymbol{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(i)} & \dots & x_d^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \dots & x_d^{(n)} \end{bmatrix}$$

$\mathbb{R}^{(d+1)\times 1}$ $\mathbb{R}^{n\times(d+1)}$

- Can write the model in vectorized form as $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{X}\boldsymbol{\theta}$

Least Squares Based Solution

- For the linear regression cost function:

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

$$= \frac{1}{2n}\sum_{i=1}^{n}\left(\boldsymbol{\theta}^{\intercal}\boldsymbol{x}^{(i)} - y^{(i)}\right)^2$$

$$= \frac{1}{2n}\underbrace{(\boldsymbol{X\theta} - \boldsymbol{y})^{\intercal}}_{\mathbb{R}^{1\times n}}\underbrace{(\boldsymbol{X\theta} - \boldsymbol{y})}_{\mathbb{R}^{n\times 1}}$$

$\boldsymbol{X} \in \mathbb{R}^{n\times(d+1)}$, $\boldsymbol{\theta} \in \mathbb{R}^{(d+1)\times 1}$

Let:

$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

Least Squares Based Solution

- Solve for optimal θ analytically
  - Notice that the solution is when $\frac{\partial}{\partial \boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0$
- Derivation:

$$\mathcal{J}(\boldsymbol{\theta}) = \frac{1}{2n} (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})^{\mathsf{T}} (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})$$

1 x 1

$$\propto \boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}$$

$$\propto \boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{\theta} - 2\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}$$

Take derivative and set equal to 0, then solve for $\boldsymbol{\theta}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}} (\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{\theta} - 2\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y}) = 0$$

$$(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X})\boldsymbol{\theta} - \boldsymbol{X}^{\mathsf{T}}\boldsymbol{y} = 0$$

$$(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X})\boldsymbol{\theta} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{y}$$

Closed Form Solution: $\boldsymbol{\theta} = (\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{y}$

Least Squares Based Solution

- Can obtain $\boldsymbol{\theta}$ by simply plugging $\boldsymbol{X}$ and $\boldsymbol{y}$ into

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

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(i)} & \dots & x_d^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \dots & x_d^{(n)} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

- If $\boldsymbol{X}^\intercal \boldsymbol{X}$ is not invertible (i.e., singular), may need to:
  - Use pseudo-inverse instead of the inverse
    - In python, `numpy.linalg.pinv(a)`
  - Remove redundant (not linearly independent) features
  - Remove extra features to ensure that $d \leq n$

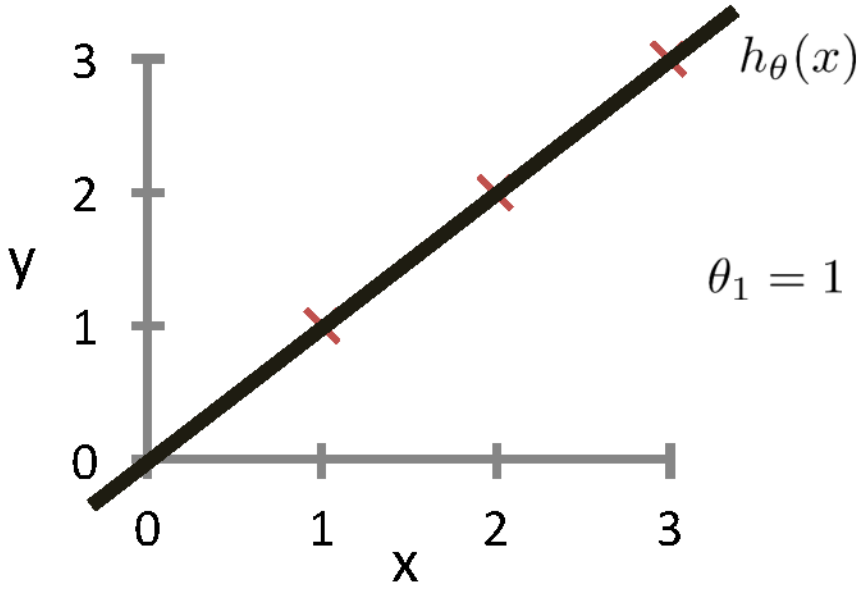
Intuition Behind Cost Function

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

For insight on J(), let's assume $x \in \mathbb{R}$ so $\boldsymbol{\theta} = [\theta_0, \theta_1]$

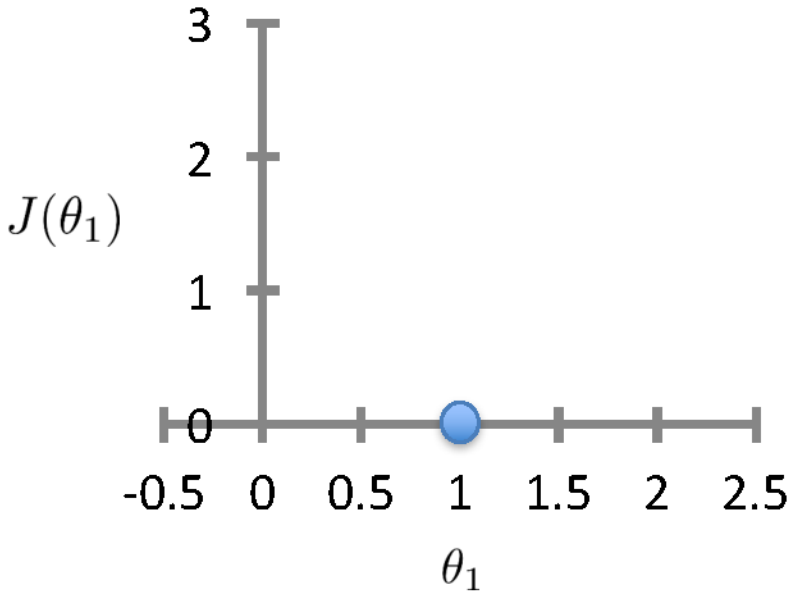
$h_\theta(x)$

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



$J(\theta_1)$

(function of the parameter θ_1)



Based on example by Andrew Ng

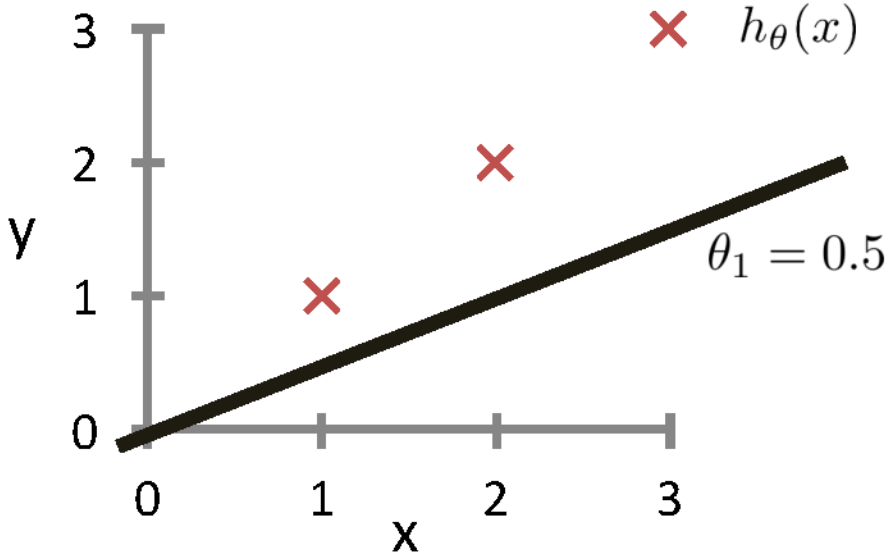
Intuition Behind Cost Function

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

For insight on J(), let's assume $x \in \mathbb{R}$ so $\boldsymbol{\theta} = [\theta_0, \theta_1]$

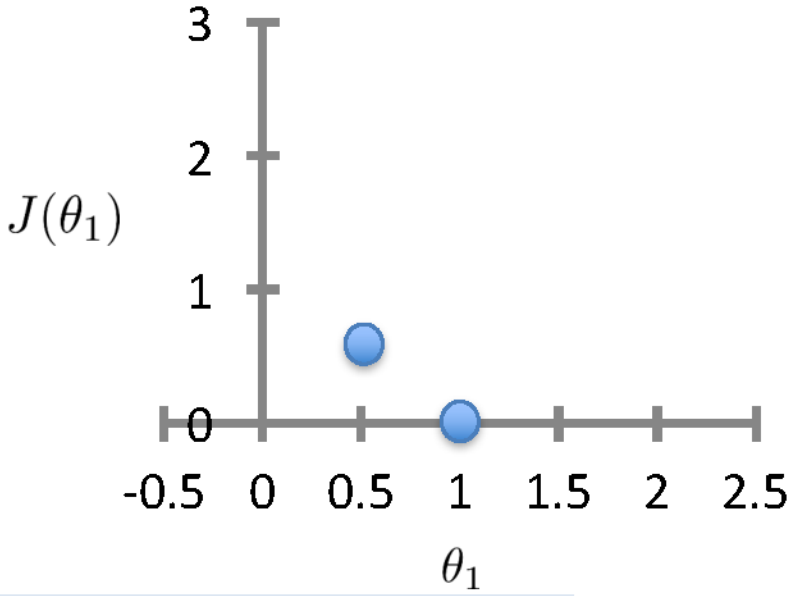
$h_\theta(x)$

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



$J(\theta_1)$

(function of the parameter θ_1)



$$J([0, 0.5]) = \frac{1}{2 \times 3}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right] \approx 0.58$$

Based on example by Andrew Ng

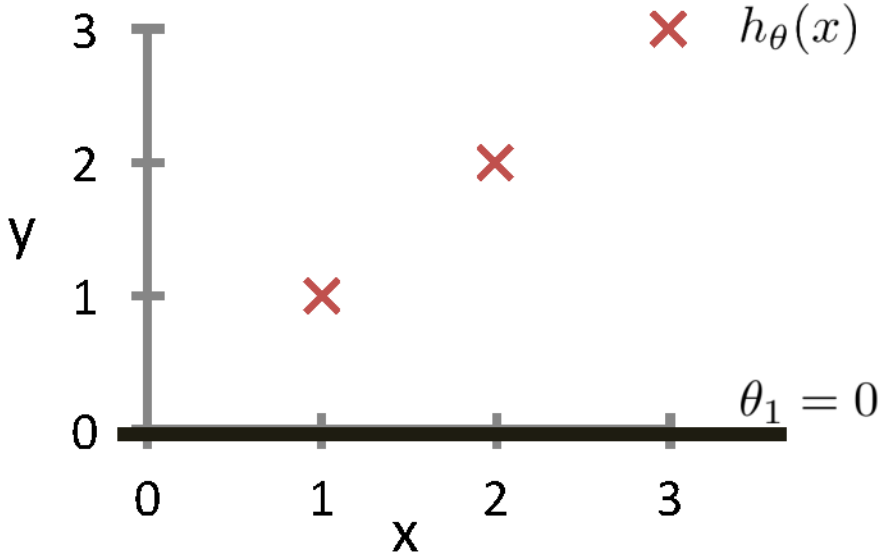
Intuition Behind Cost Function

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

For insight on J(), let's assume $x \in \mathbb{R}$ so $\boldsymbol{\theta} = [\theta_0, \theta_1]$

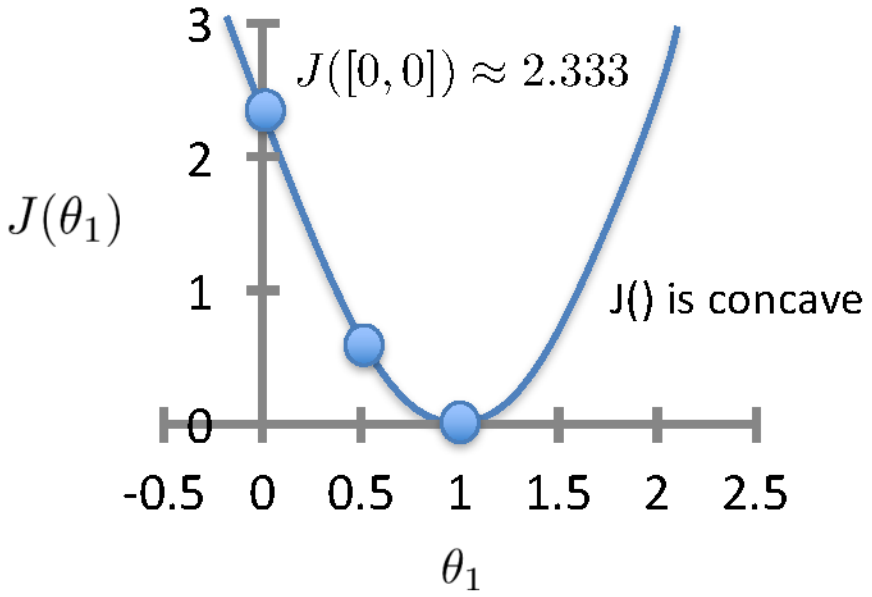
$h_\theta(x)$

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



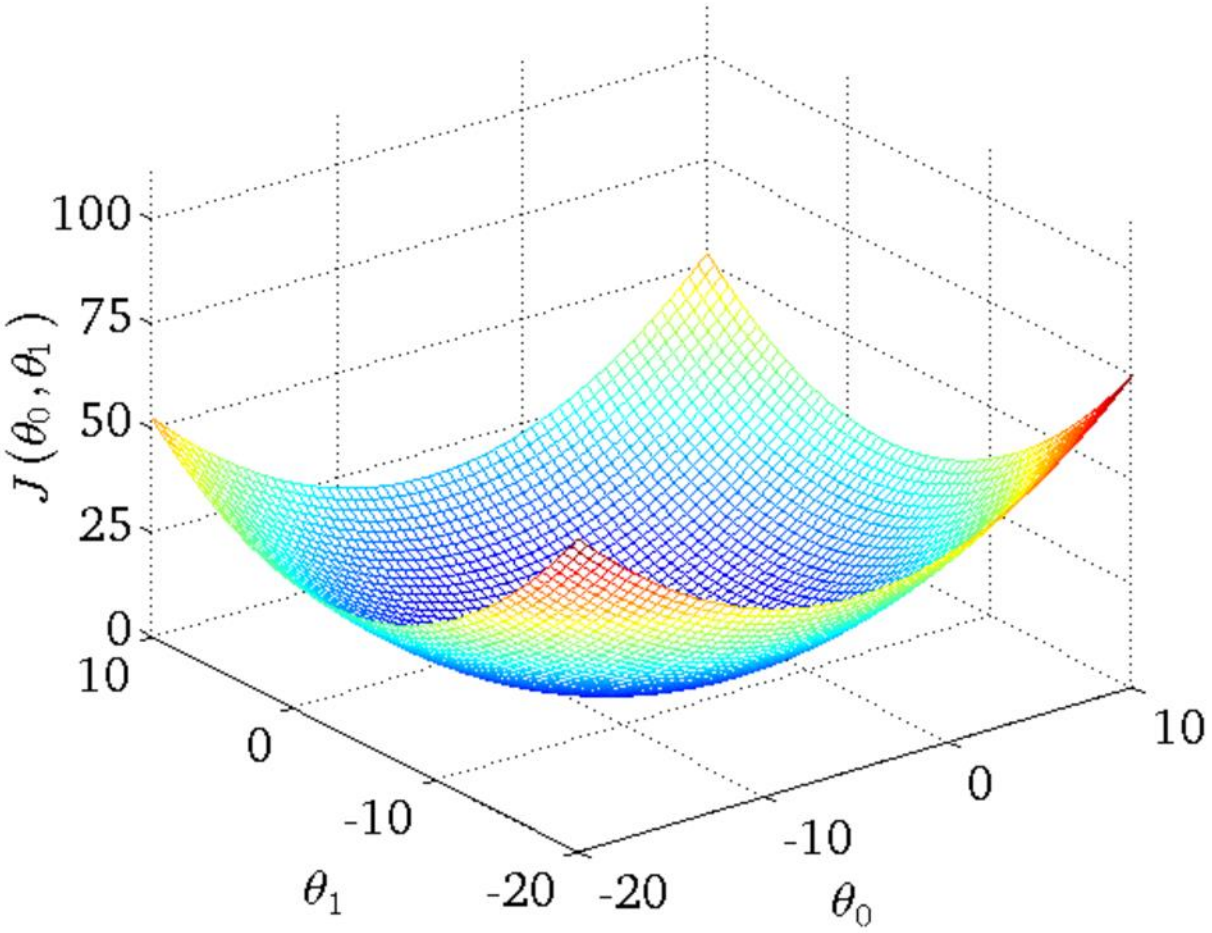
$J(\theta_1)$

(function of the parameter θ_1)



Based on example by Andrew Ng

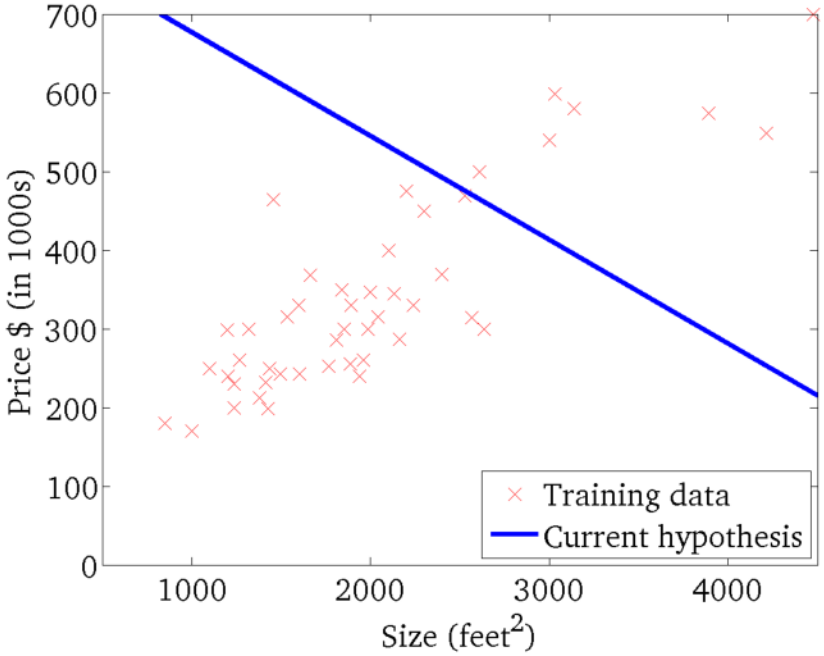
Intuition Behind Cost Function



Intuition Behind Cost Function

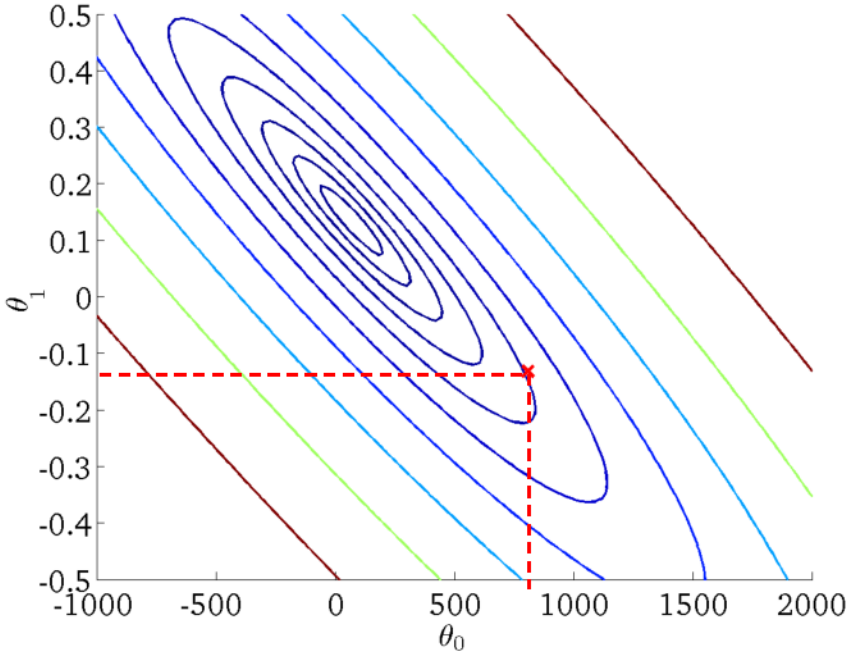
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

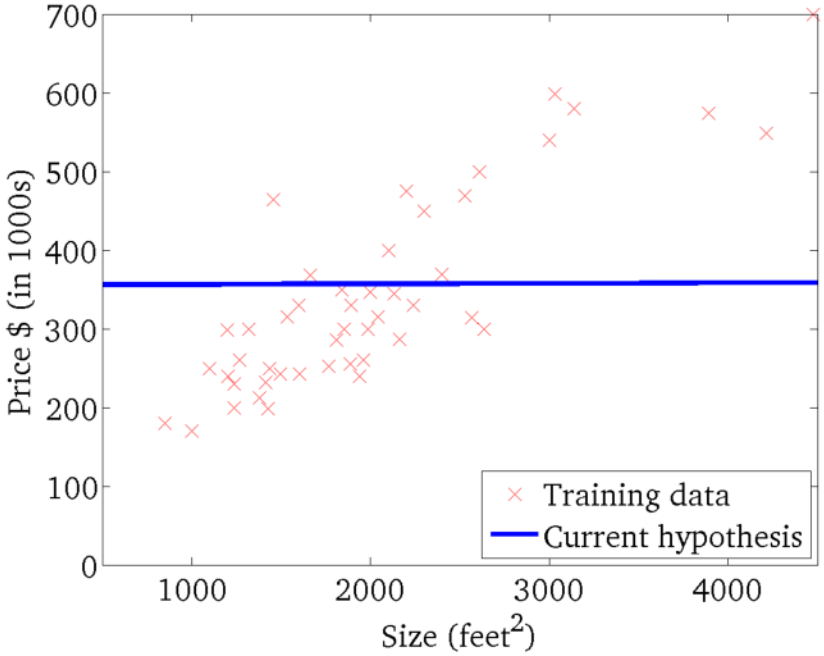
(function of the parameters θ_0, θ_1)



Intuition Behind Cost Function

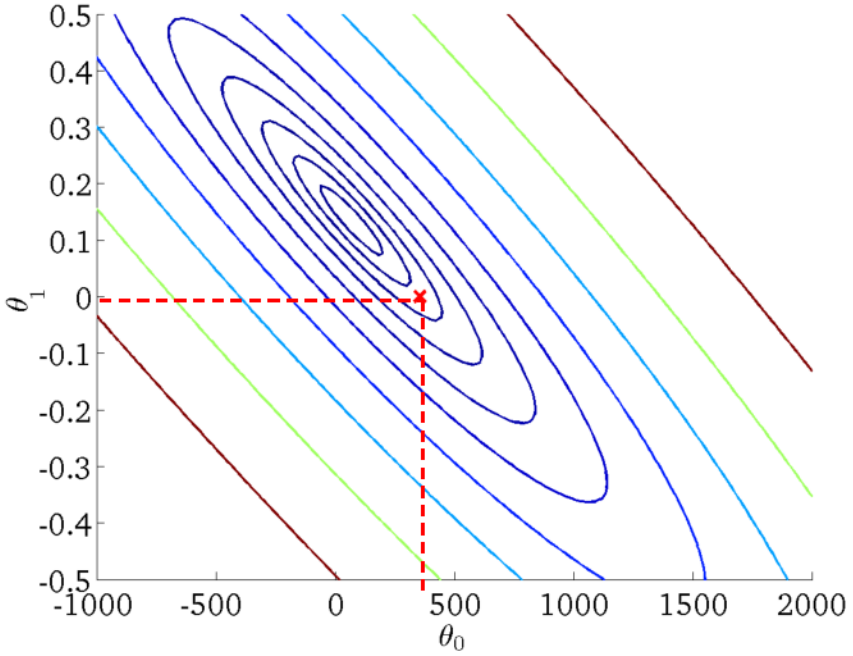
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

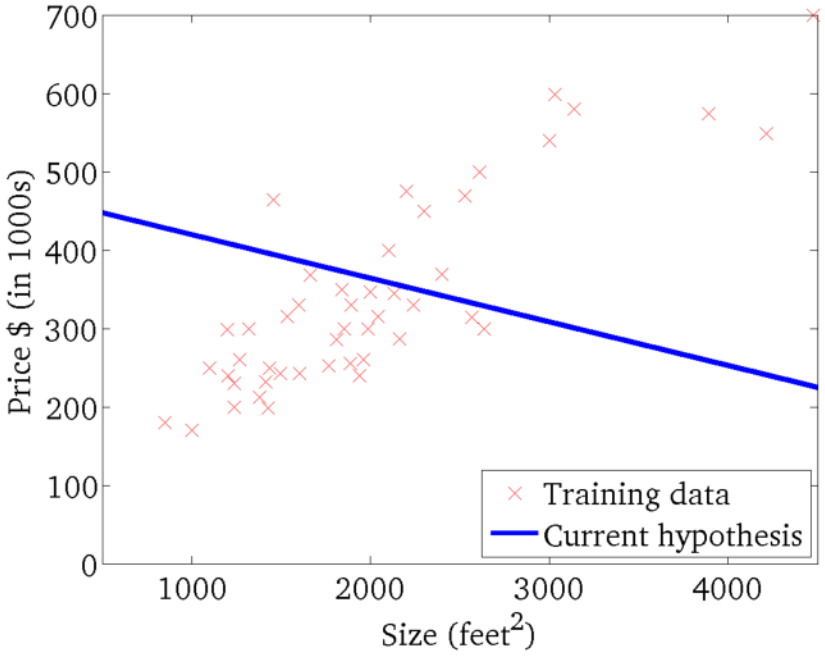
(function of the parameters θ_0, θ_1)



Intuition Behind Cost Function

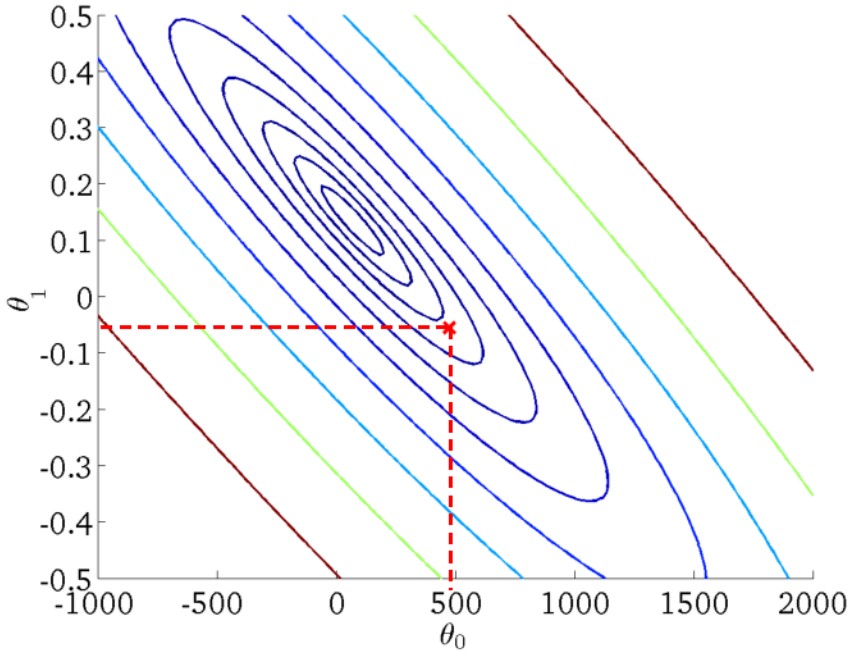
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

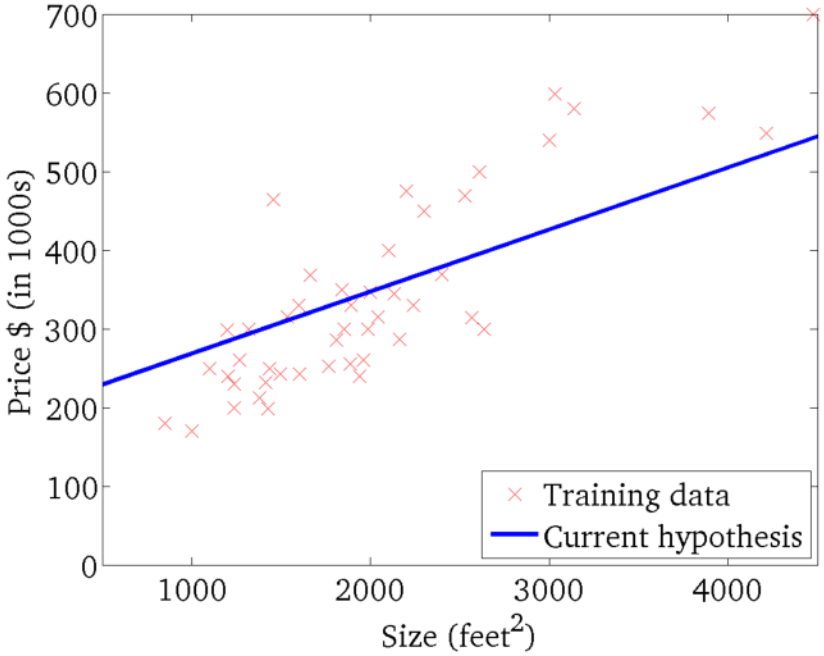
(function of the parameters θ_0, θ_1)



Intuition Behind Cost Function

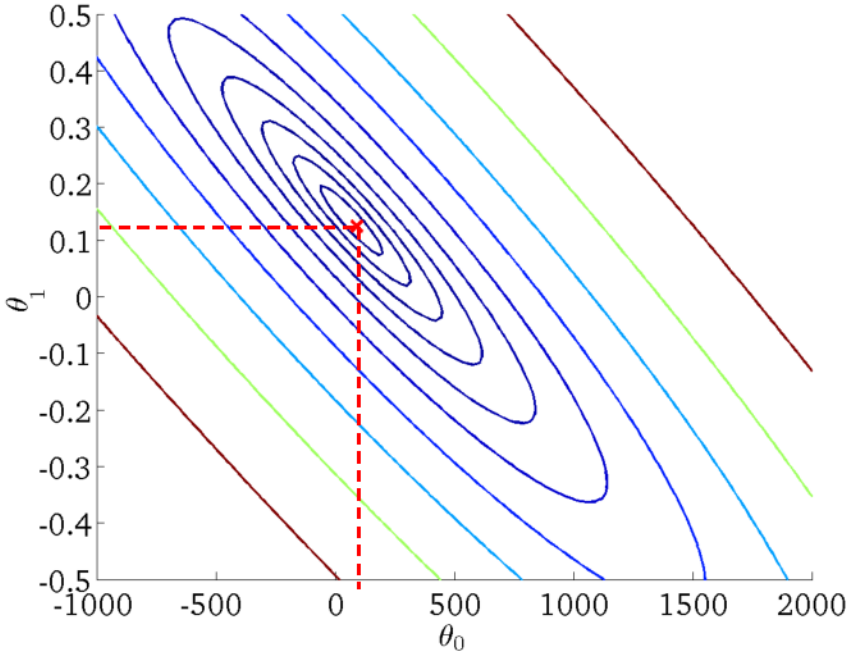
$h_\theta(x)$

(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)



Basic Search Procedure

- Choose initial value for θ
- Until we reach a minimum:
  - Choose a new value for θ to reduce $J(\theta)$

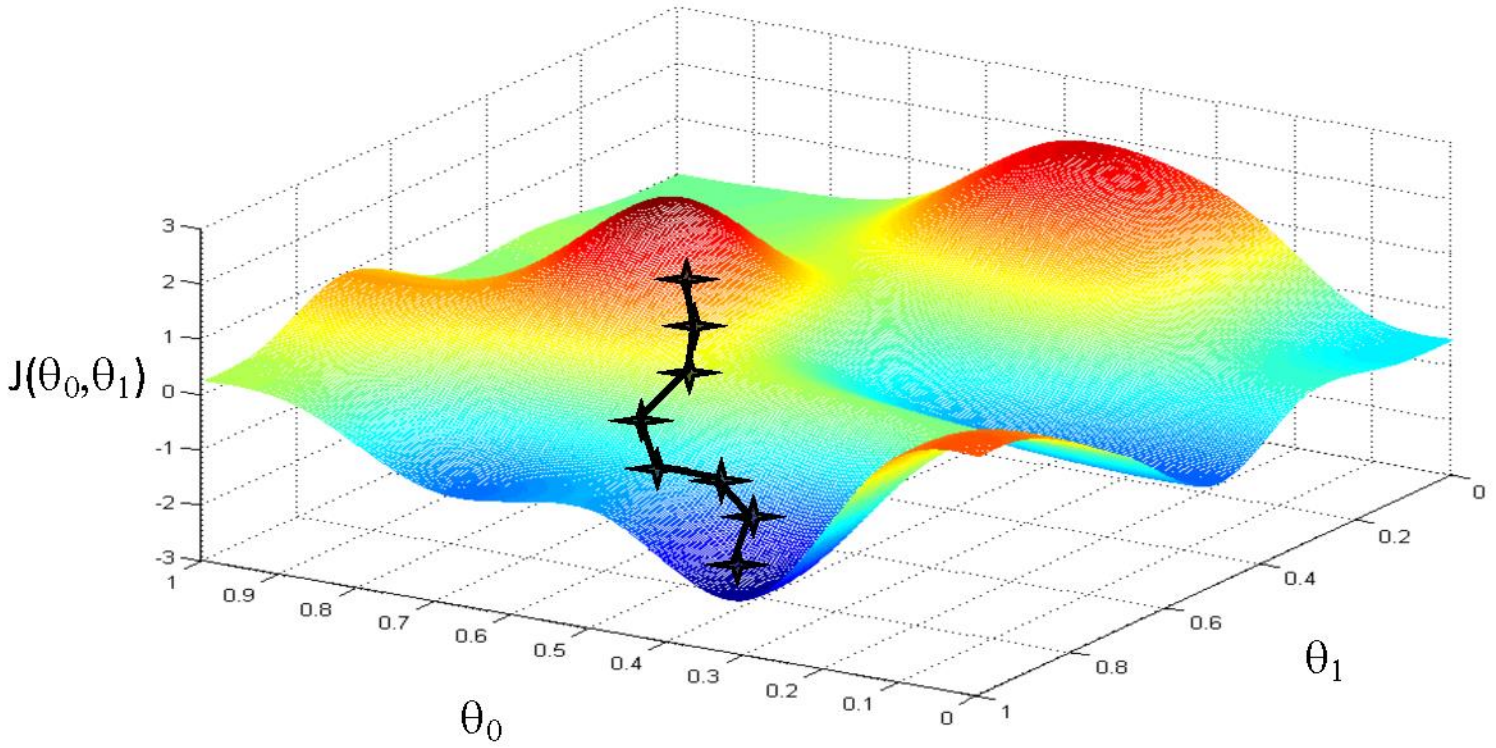


Figure by Andrew Ng

Basic Search Procedure

- Choose initial value for θ
- Until we reach a minimum:
  - Choose a new value for θ to reduce $J(\theta)$

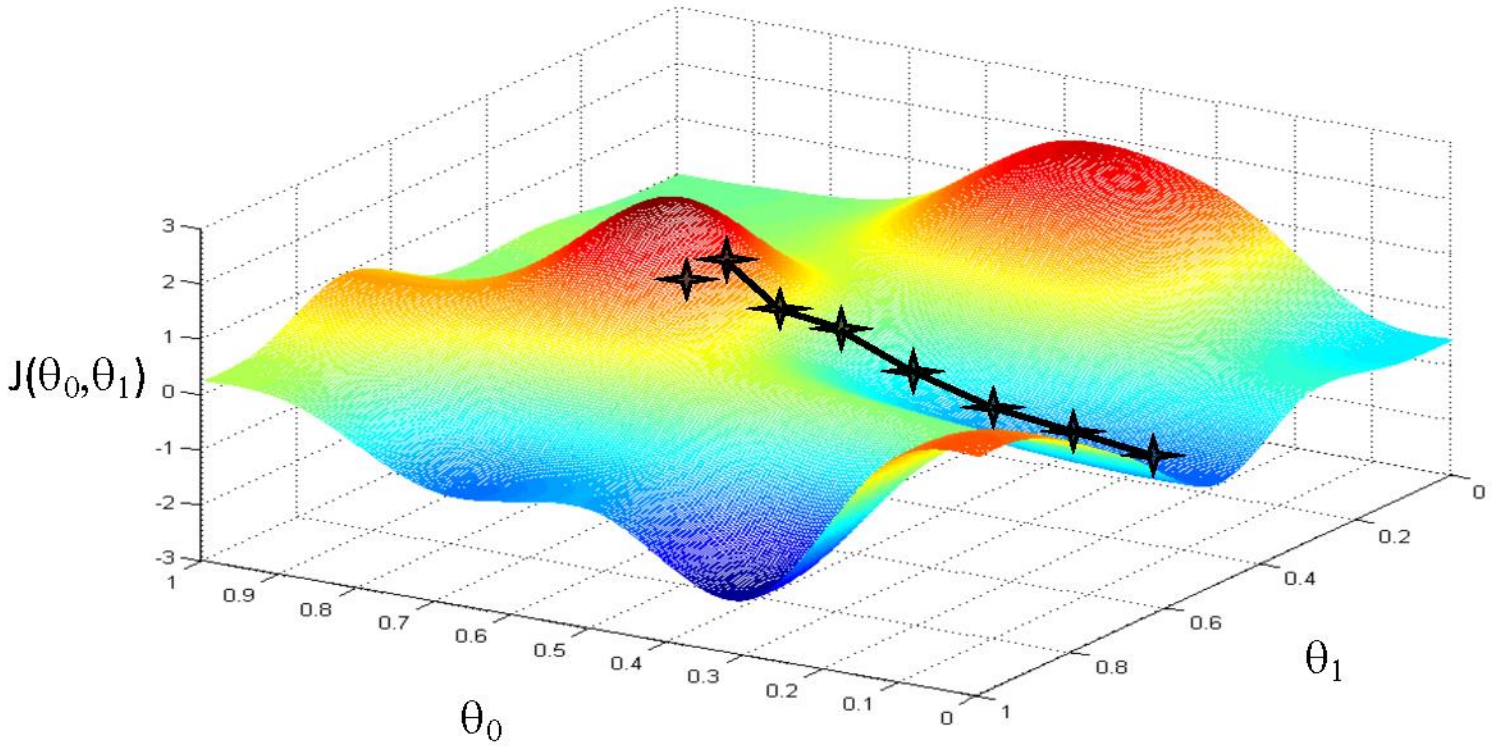
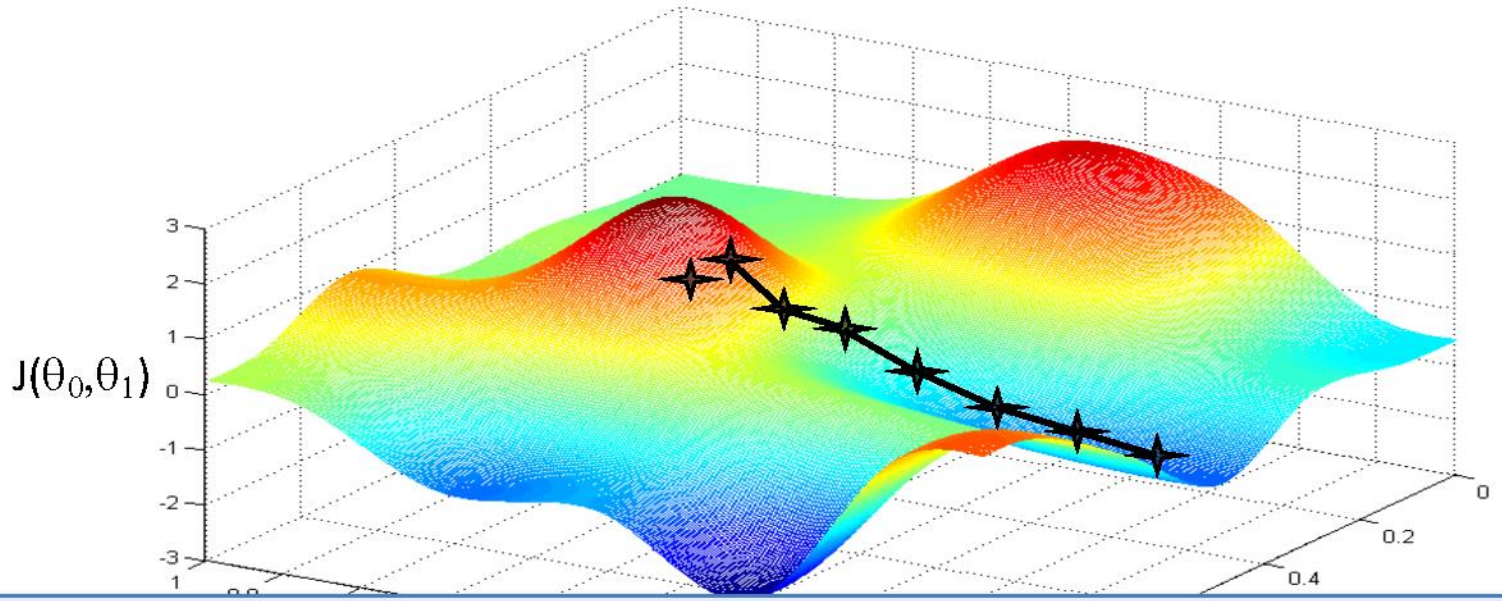


Figure by Andrew Ng

Basic Search Procedure

- Choose initial value for θ
- Until we reach a minimum:
  - Choose a new value for θ to reduce $J(\theta)$



Since the least squares objective function is convex (concave), we don't need to worry about local minima

Figure by Andrew Ng

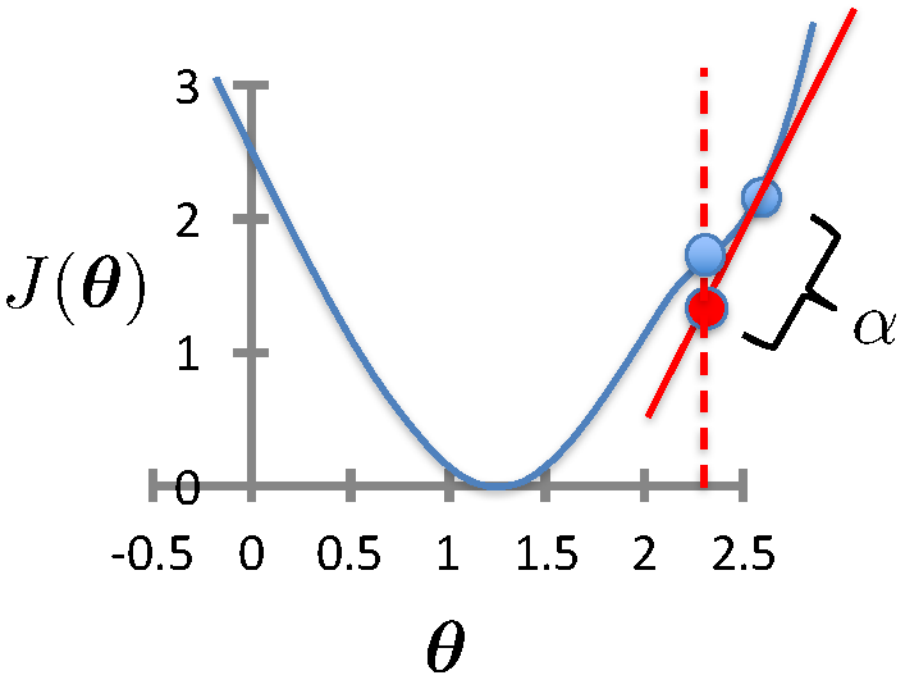
Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

learning rate (small)
e.g., α = 0.05



Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

For Linear Regression: $\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2$

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

For Linear Regression:

$$\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2$$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2$$

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

For Linear Regression:

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2 \\
&= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2 \\
&= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \times \frac{\partial}{\partial \theta_j} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)
\end{aligned}$$

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

For Linear Regression:

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2 \\
&= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2 \\
&= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \times \frac{\partial}{\partial \theta_j} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \\
&= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) x_j^{(i)}
\end{aligned}$$

Gradient Descent for Linear Regression

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right) x_j^{(i)}$$

simultaneous update for j = 0 ... d

- To achieve simultaneous update
  - At the start of each GD iteration, compute $h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right)$
  - Use this stored value in the update step loop

- Assume convergence when $\left\| \boldsymbol{\theta}_{new} - \boldsymbol{\theta}_{old} \right\|_2 < \epsilon$

L_2 norm: $\|\boldsymbol{v}\|_2 = \sqrt{\sum_i v_i^2} = \sqrt{v_1^2 + v_2^2 + \ldots + v_{|v|}^2}$

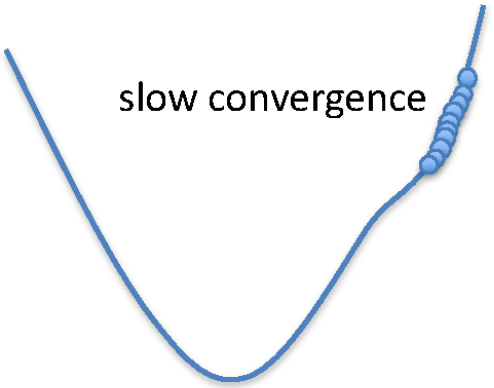
Closed Form Solution Vs. Gradient Descent

innovate achieve lead

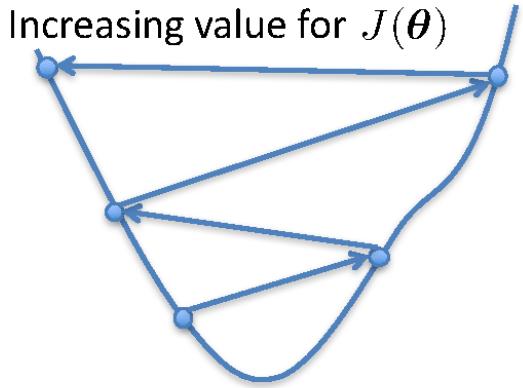
Gradient Descent	Closed Form Solution
• Requires multiple iterations • Need to choose α • Works well when n is large • Can support incremental learning	• Non-iterative • No need for α • Slow if n is large – Computing $(X^T X)^{-1}$ is roughly $O(n^3)$

Choosing Step Size (*learning rate*)

α too small



α too large



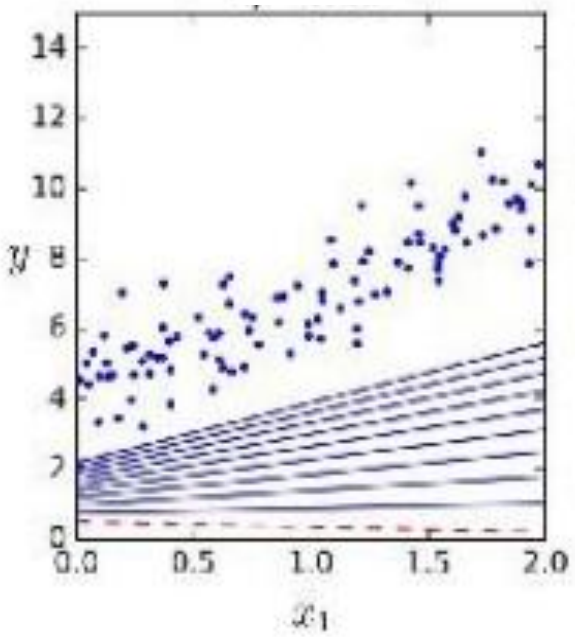
- May overshoot the minimum
- May fail to converge
- May even diverge

To see if gradient descent is working, print out $J(\theta)$ each iteration

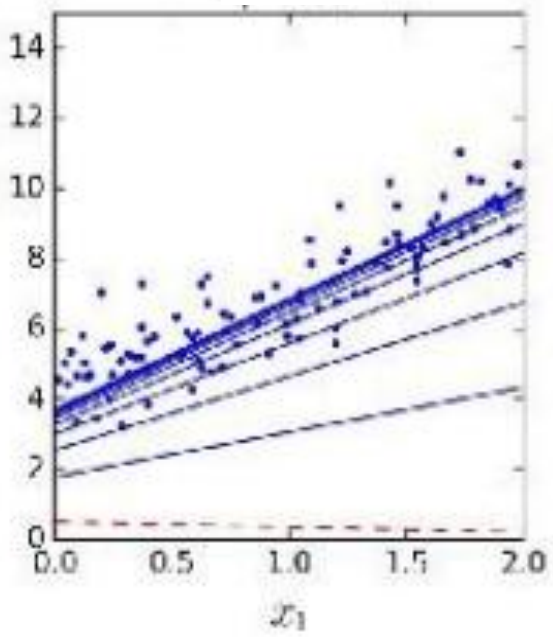
- The value should decrease at each iteration
- If it doesn't, adjust α

Impact of Learning Rate

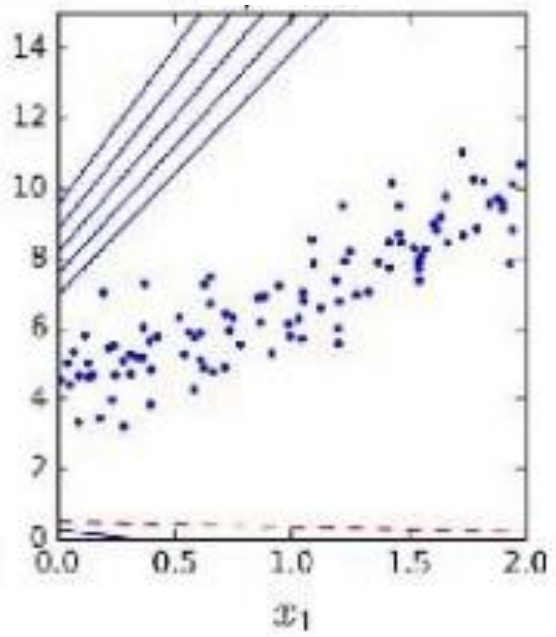
α=0.02



α=0.1

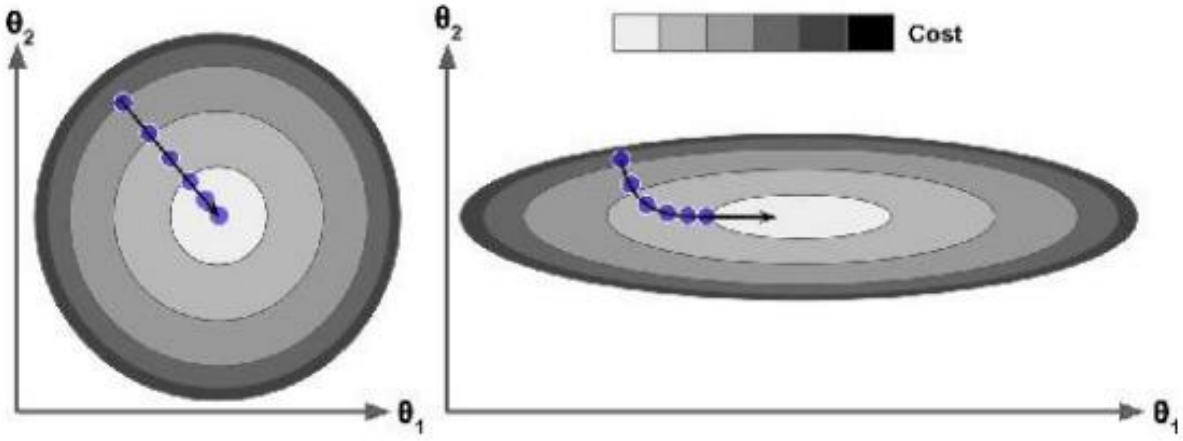


α=0.5



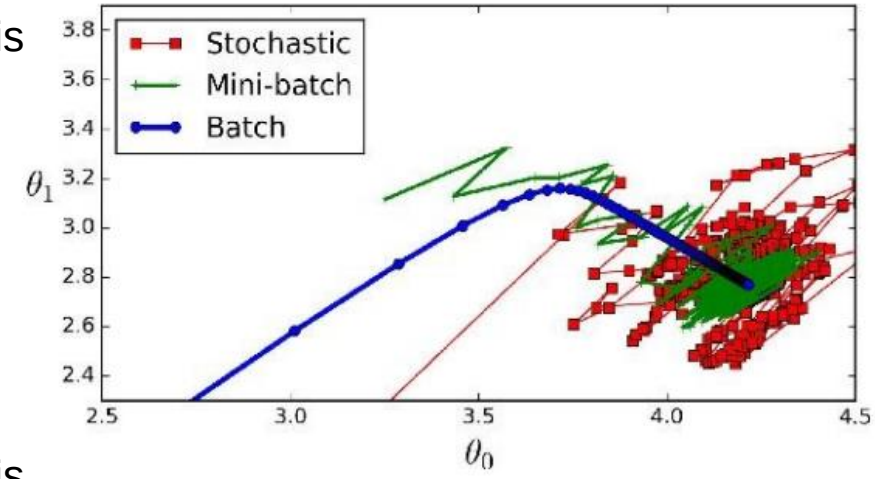
Feature Normalization

- Gradient Descent can be slow when feature scales are widely different
- Normalization of feature values for same variance needed for faster convergence



Impact Due To Batch Size

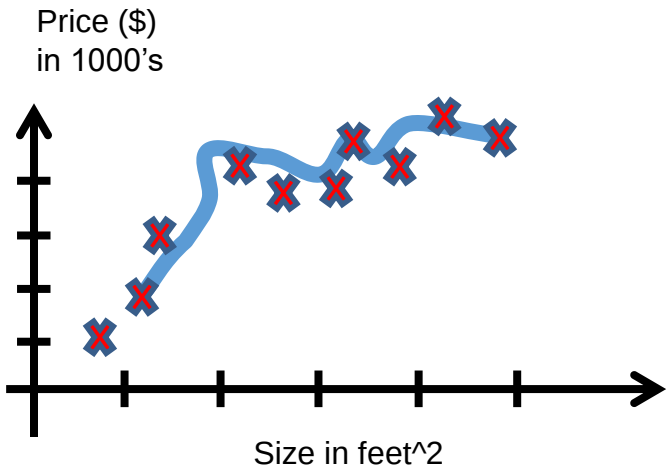
- Choice of batch size impacts the rate of convergence of gradient descent (GD)
- In Batch GD, entire training set is used to calculate the training error in each iteration/epoch and gradient is calculated and used for weight updates
  - Convergences in least number of iterations, i.e., rate of convergence is highest [$O(1/\text{iterations})$]
  - Computation requirement per iteration is highest
  - Memory requirement is also highest
- In Stochastic gradient descent, a **randomly** selected training instance is used
  - Convergences in highest number of iterations, i.e., rate of convergence is slowest [~$O(1/\sqrt{iterations})$]
  - Computation requirement per iteration is lowest
  - Memory requirement is also lowest



- In mini batch GD, a subset of training data of size, say 64,128, 256 is used
- very efficient implementation possible leveraging vector processing using GPUs

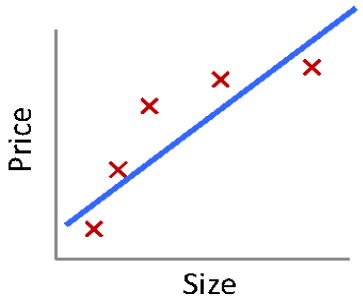
Addressing overfitting

- $x_1 =$ size of house
- $x_2 =$ no. of bedrooms
- $x_3 =$ no. of floors
- $x_4 =$ age of house
- $x_5 =$ average income in neighborhood
- $x_6 =$ kitchen size
- $\vdots$
- x_{100}



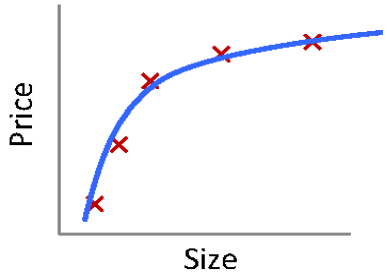
Slide credit: Andrew Ng

Quality of Fit



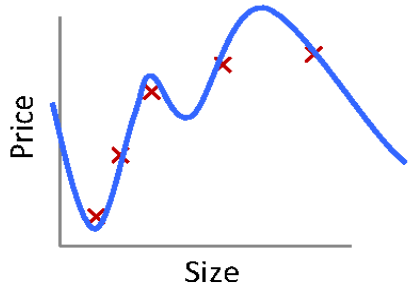
$\theta_0 + \theta_1 x$

Underfitting
(high bias)



$\theta_0 + \theta_1 x + \theta_2 x^2$

Correct fit



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfitting
(high variance)

Overfitting:

- The learned hypothesis may fit the training set very well ($J(\boldsymbol{\theta}) \approx 0$)
- ...but fails to generalize to new examples

Based on example by Andrew Ng

Addressing overfitting

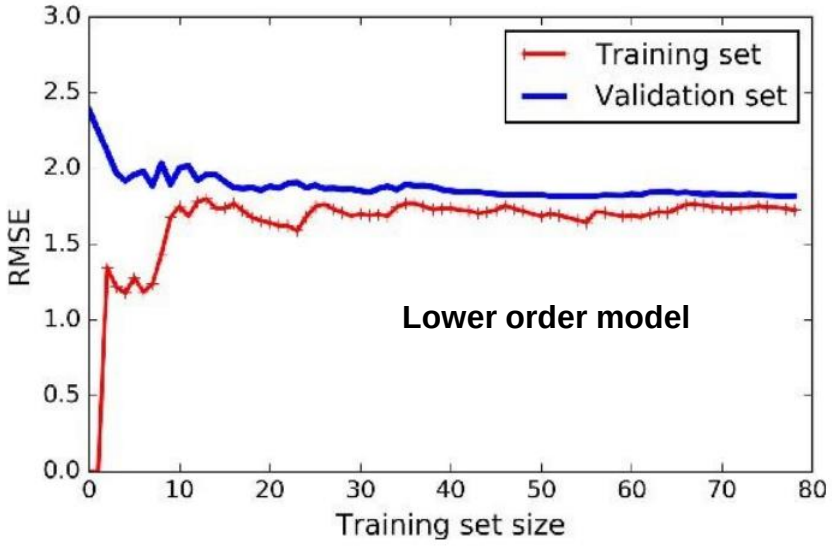
- Reduce number of features.
  - Manually select which features to keep.
  - Model selection algorithm

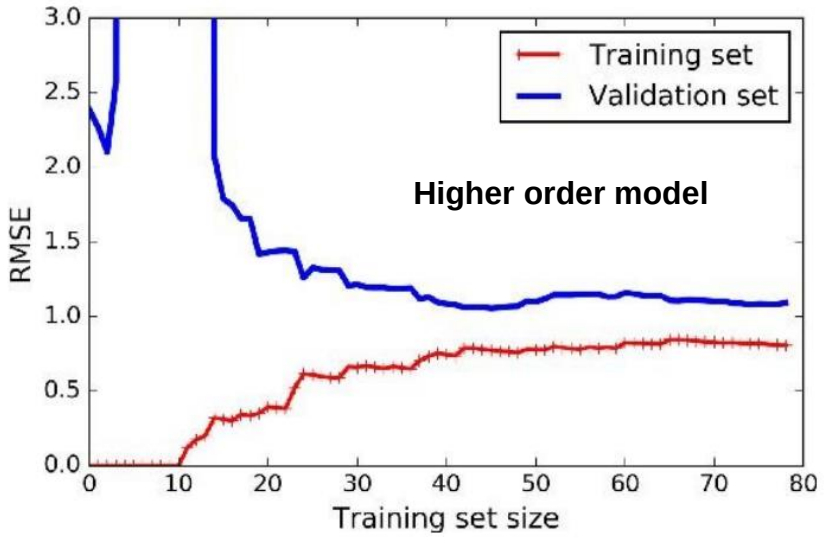
- Regularization.
  - Keep all the features, but reduce magnitude/values of parameters θ_j.
  - Works well when we have a lot of features, each/many of which contributes a bit to predicting y.

Slide credit: Andrew Ng

Effect of Training Size on Overfitting

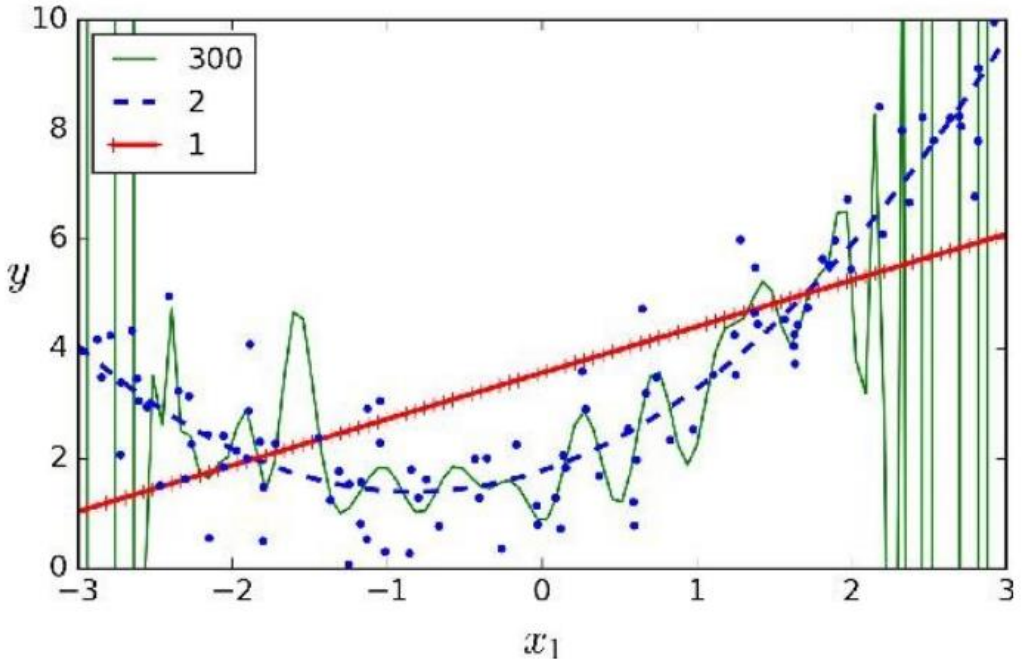
- Size of training dataset needs to be large to prevent when higher order model is used.





Polynomial Fitting can lead to Overfitting

- Underlying target function is quadratic
- Linear model results in underfitting with large bias
- Polynomial of order 300 results in a large variance



Regularization

- A method for automatically controlling the complexity of the learned hypothesis
- **Idea**: penalize for large values of θ_j
  - Can incorporate into the cost function
  - Works well when we have a lot of features, each that contributes a bit to predicting the label
- Can also address overfitting by eliminating features (either manually or via model selection)

Ridge Regularization

- Linear regression objective function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

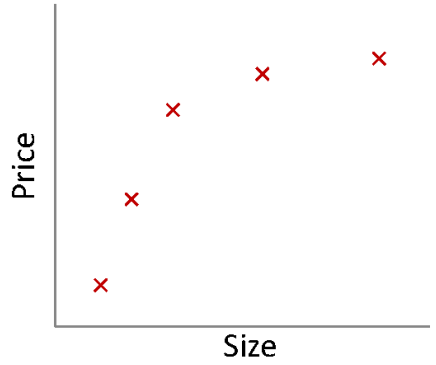
model fit to data regularization

  - λ is the regularization parameter ($\lambda \geq 0$)
  - No regularization on θ_0!

Understanding Regularization

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

- What happens if we set λ to be huge (e.g., 10^{10})?



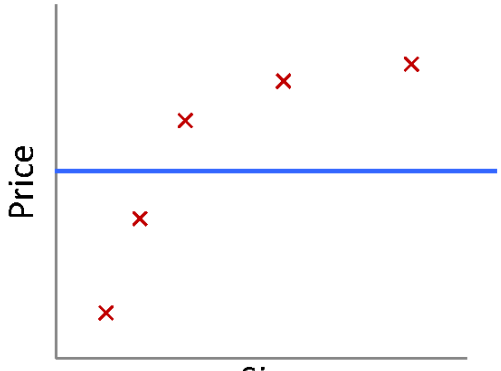
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Based on example by Andrew Ng

Understanding Regularization

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

- What happens if we set λ to be huge (e.g., 10^{10})?



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

0 0 0 0

Based on example by Andrew Ng

Ridge regression

- Cost Function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

- Fit by solving $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Gradient update:

$\frac{\partial}{\partial\theta_0}J(\theta)$

$$\theta_0 \leftarrow \theta_0 - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)$$

$\frac{\partial}{\partial\theta_j}J(\theta)$

$$\theta_j \leftarrow \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} - \lambda\theta_j$$

regularization

Ridge Regression

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

$$\theta_0 \leftarrow \theta_0 - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)$$

$$\theta_j \leftarrow \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} - \lambda\theta_j$$

- We can rewrite the gradient step as:

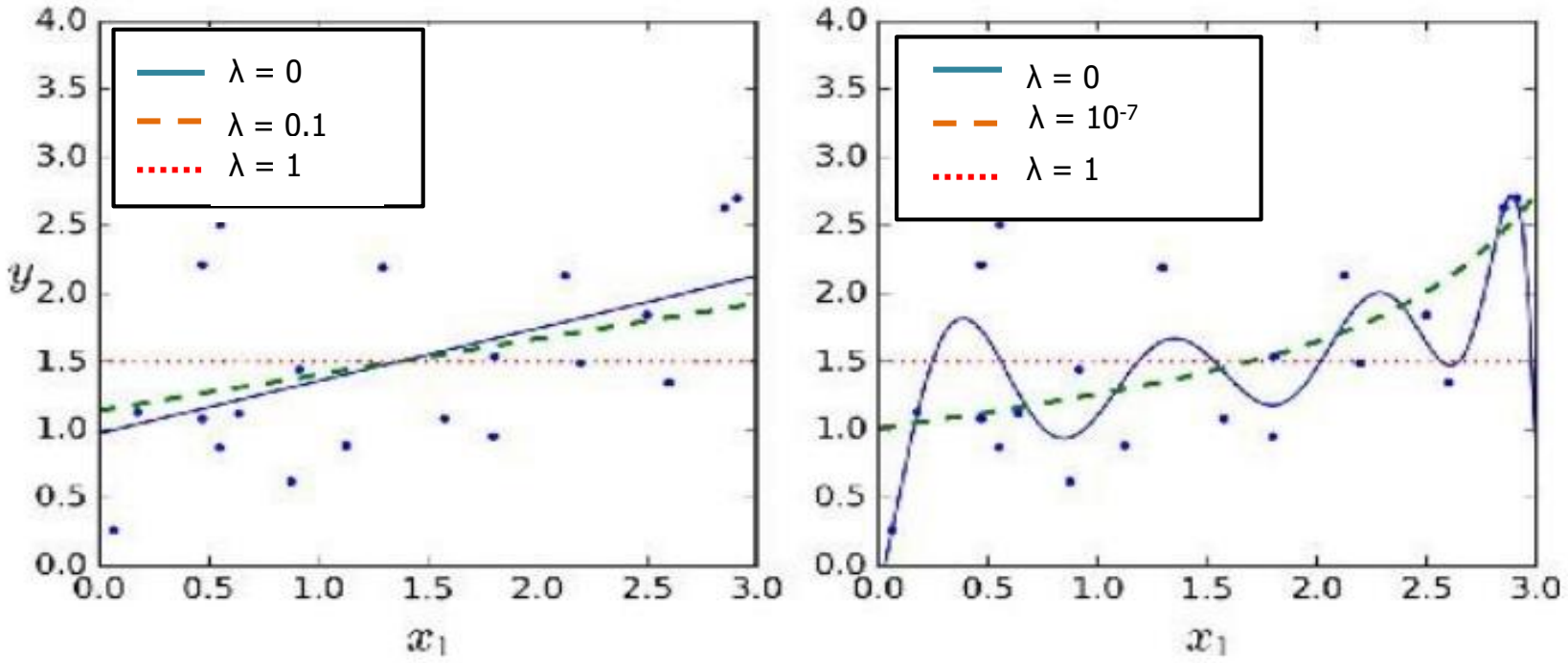
$$\theta_j \leftarrow \theta_j\left(1 - \alpha\lambda\right) - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

Lasso Regularization

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{d}|\theta_j|$$

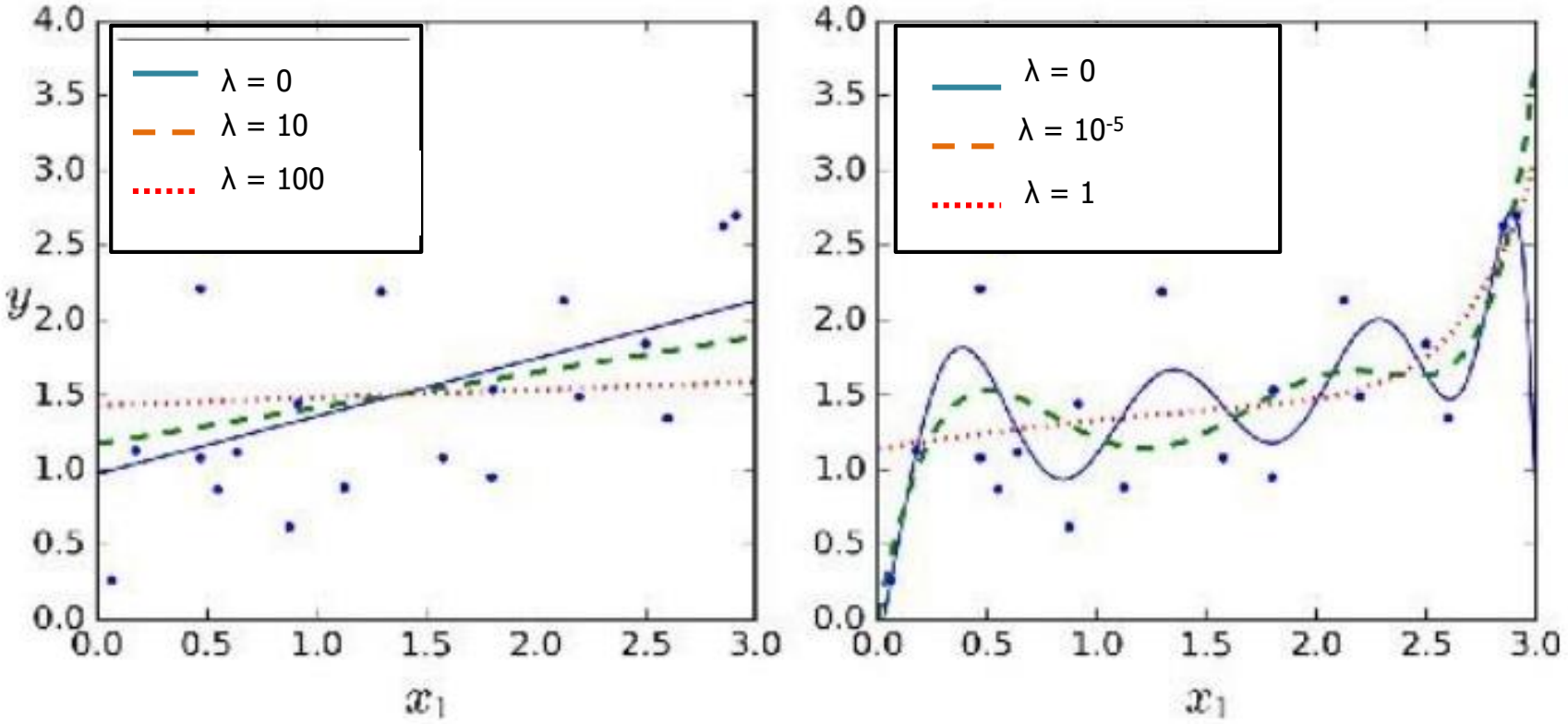
$$\theta_j = \theta_j - \frac{\alpha}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)\boldsymbol{x}_j^{(i)} - \alpha\lambda\,\text{sign}(\theta_j)$$

Ridge Vs Lasso Regularization



Lasso

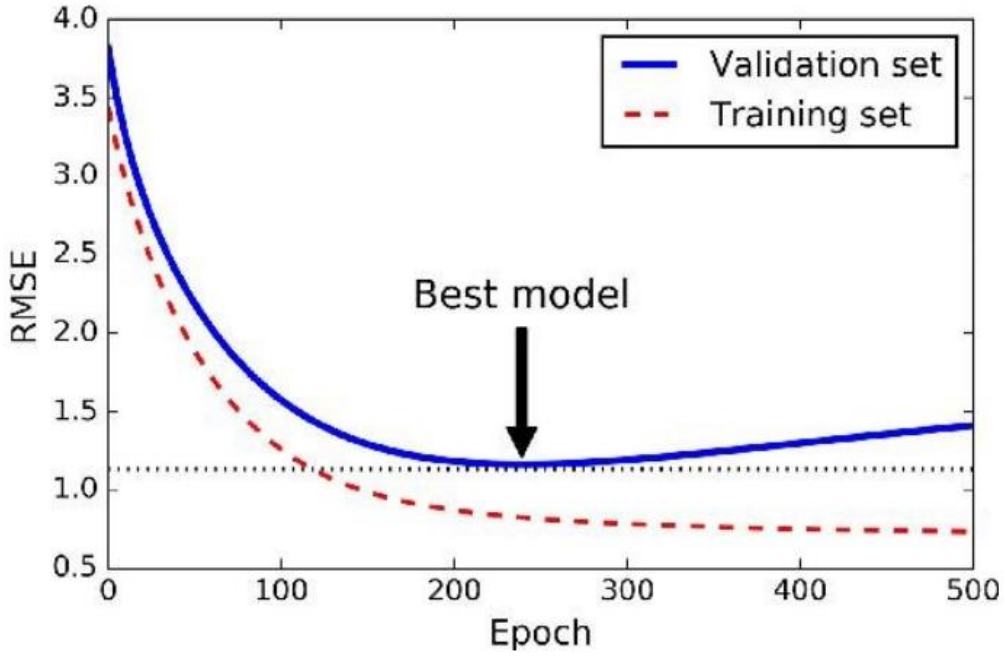
Ridge Vs Lasso Regularization



Ridge

Early Stopping

- Do Not Over train to prevent overfitting
- Stop training once error on the validation set starts showing an upward trend, even if the error on the training set keeps decreasing



Extending Linear Regression to More Complex Models

- The inputs **X** for linear regression can be:
  - Original quantitative inputs
  - Transformation of quantitative inputs
    - e.g. log, exp, square root, square, etc.
  - Polynomial transformation
    - example: $y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$
  - Basis expansions
  - Dummy coding of categorical inputs
  - Interactions between variables
    - example: $x_3 = x_1 \cdot x_2$

This allows use of linear regression techniques to fit non-linear datasets.

Linear Basis Function

- Simplest linear model for regression is one that involves a linear combination of the input variables

$$y(x,w)=w_0 +w_1x_1 +...+w_Dx_D$$

- Extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form

$$y(x,w)=w_0 + \sum_{j=1}^{M-1} w_j\varphi_j(x)$$

  - where $\varphi_j(x)$ are known as basis functions.
  - By denoting the maximum value of the index j by M −1, the total number of parameters in this model will be M.

Linear Basis Function

- Convenient to define an additional dummy 'basis function' $\phi_0(x)=1$. So,

$$y(x,w) = \sum_{j=1}^{M-1} w_j \varphi_j(x) = \mathbf{w}^T \boldsymbol{\phi}_j(\mathbf{x})$$

where $\mathbf{w} = (w_0,\ldots,w_{M-1})^T$ and $\boldsymbol{\phi} = (\phi_0, \phi_1, \ldots\ldots \phi_n)$

- If the original variables comprise the vector x, then the features can be expressed in terms of the basis functions $\{\phi j(x)\}$

Linear Basis Function Models

- Generally,

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j \phi_j(\boldsymbol{x})$$

basis function

- Typically, $\phi_0(\boldsymbol{x}) = 1$ so that θ_0 acts as a bias
- In the simplest case, we use linear basis functions :

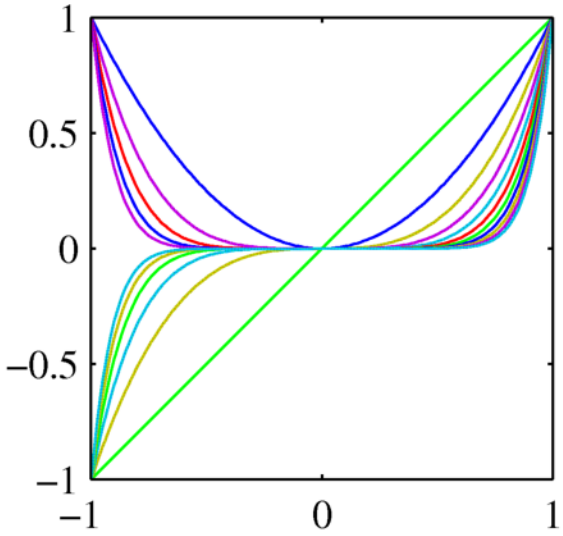
$$\phi_j(\boldsymbol{x}) = x_j$$

Linear Basis Function Models

- Polynomial basis functions:

$$\phi_j(x) = x^j$$

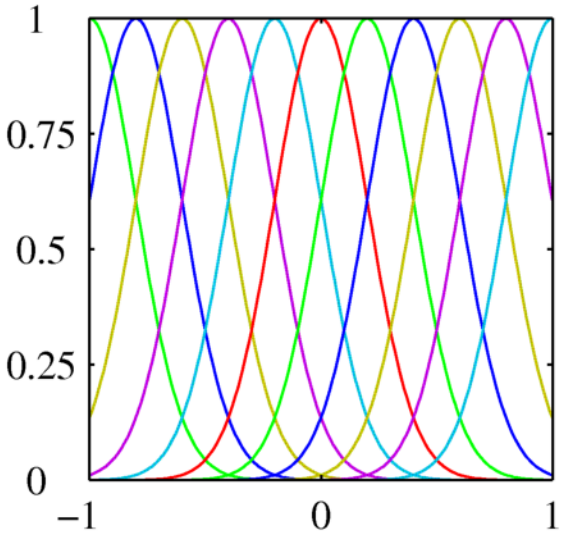
  - These are global; a small change in x affects all basis functions



- Gaussian basis functions:

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

  - These are local; a small change in x only affect nearby basis functions. μ_j and s control location and scale (width).



Linear Basis Function Models

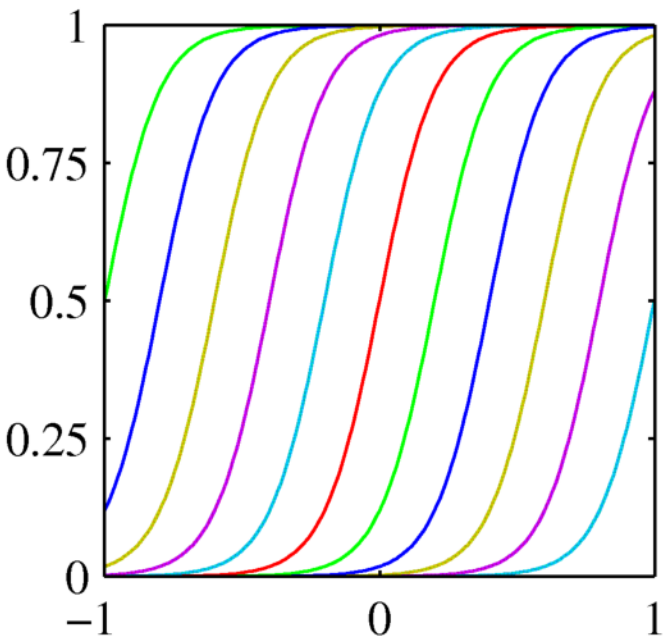
- Sigmoidal basis functions:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

  - These are also local; a small change in x only affects nearby basis functions. μ_j and s control location and scale (slope).

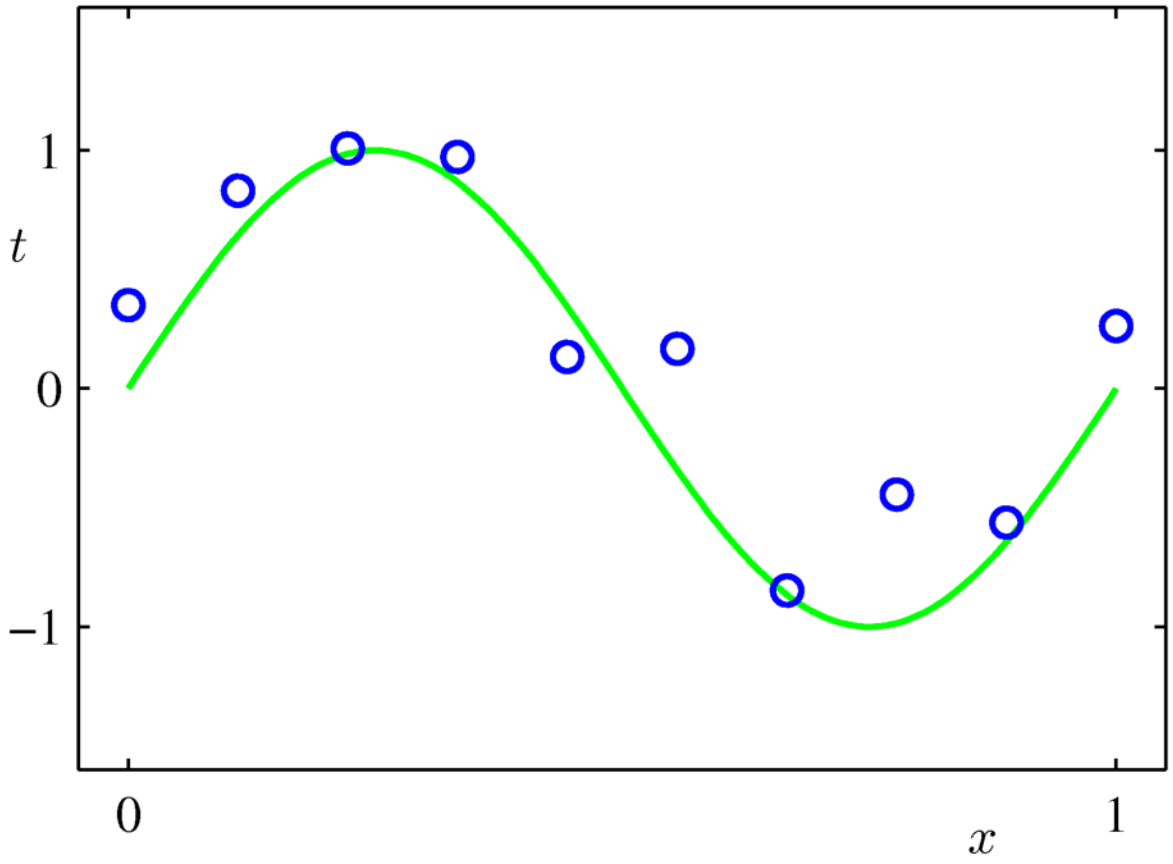


Linear Basis Function Models

- Basic Linear Model: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j x_j$
- Generalized Linear Model: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j \phi_j(\boldsymbol{x})$
- Once we have replaced the data by the outputs of the basis functions, fitting the generalized model is exactly the same problem as fitting the basic model
  - Unless we use the kernel trick – more on that when we cover support vector machines
  - Therefore, there is no point in cluttering the math with basis functions

Based on slide by Geoff Hinton

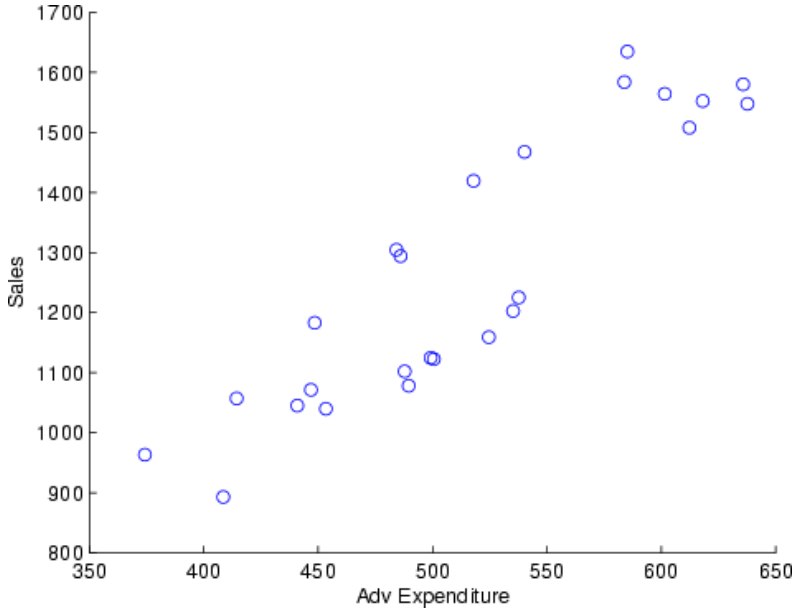
Example of Fitting a Polynomial Curve with a Linear Model



$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p = \sum_{j=0}^{p} \theta_j x^j$$

Regression analysis

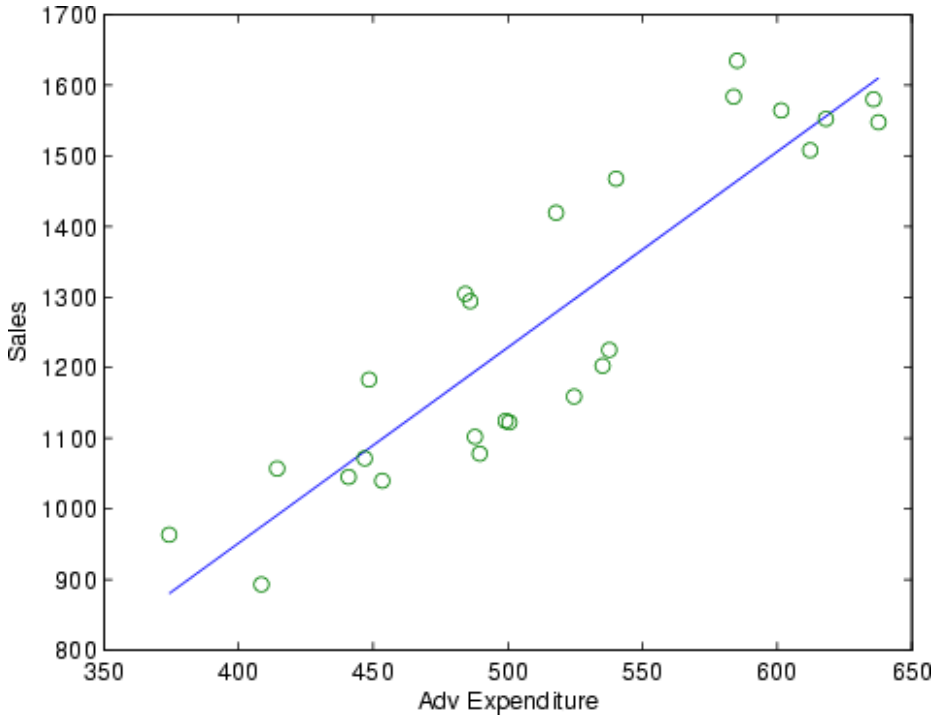
- Step 1: graphical display of data — scatter plot: sales vs. advertisement cost



- calculate correlation

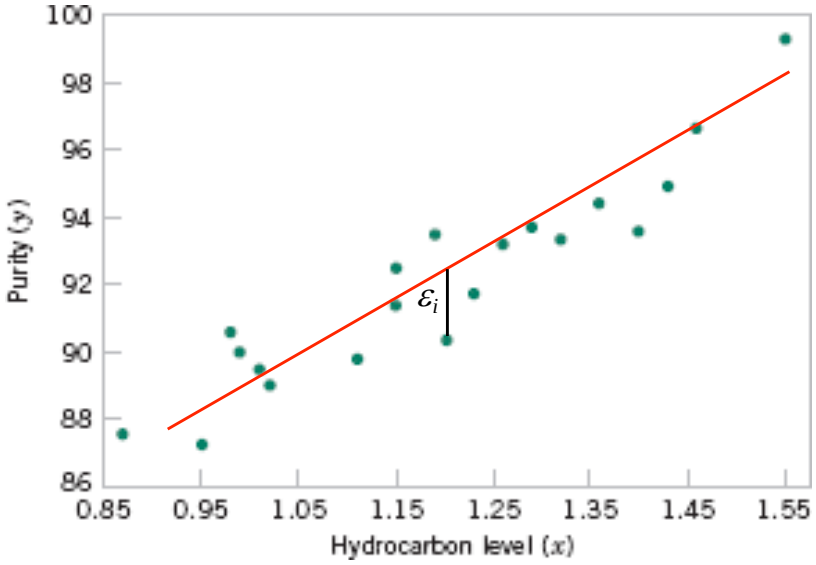
$$\hat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \times \sum_{i=1}^{n}(y_i - \bar{y})^2}} \qquad -1 \le \hat{\rho} \le 1$$

- Step 2: find the relationship or association between Sales and Advertisement Cost — Regression



Simple linear regression

Based on the scatter diagram, it is probably reasonable to assume that the mean of the random variable Y is related to X by the following simple linear regression model:



Response

Regressor or Predictor

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \qquad i = 1, 2, !, n$$

Intercept

Slope

Random error

$$\varepsilon_i \sim N(0, \sigma^2)$$

where the slope and intercept of the line are called regression coefficients.

•The case of simple linear regression considers a single regressor or predictor x and a dependent or response variable Y.

Regression coefficients

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}$$

$$S_{xy} = \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) = \sum_{i=1}^{n} x_i y_i - \frac{\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

Fitted (estimated) regression model

Caveat: regression relationship are valid only for values of the regressor variable within the range the original data. Be careful with extrapolation.

Estimation of variance

- Using the fitted model, we can estimate value of the response variable for given predictor

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

- Residuals: $r_i = y_i - \hat{y}_i$
- Our model: $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$, $i = 1,\ldots,n$, $Var(\varepsilon_i) = \sigma^2$
- Unbiased estimator (MSE: Mean Square Error)

$$\hat{\sigma}^2 = MSE = \frac{\sum_{i=1}^{n} r_i^2}{n-2}$$

Punchline

- the coefficients

$$\hat{\beta}_1 \text{ and } \hat{\beta}_0$$

and both calculated from data, and they are subject to error.

- if the true model is $y = \beta_1 x + \beta_0$, $\hat{\beta}_1$ and $\hat{\beta}_0$ are point estimators for the true coefficients
- we can talk about the ``accuracy'' of $\hat{\beta}_1$ and $\hat{\beta}_0$

Assessing linear regression model

- Test hypothesis about true slope and intercept

$$\beta_1 = ?, \qquad \beta_0 = ?$$

- Construct confidence intervals

$$\beta_1 \in \left[\hat{\beta}_1 - a, \hat{\beta}_1 + a\right] \quad \beta_0 \in \left[\hat{\beta}_0 - b, \hat{\beta}_0 + b\right] \quad \text{with probability } 1-\alpha$$

- Assume the errors are normally distributed

$$\varepsilon_i \sim \mathrm{N}\left(\ 0,\ \ \sigma^2\ \right)$$

Properties of Regression Estimators

slope parameter β_1	intercept parameter β_0
$E(\hat{\beta}_1) = \beta_1$	$E(\hat{\beta}_0) = \beta_0$
$V(\hat{\beta}_1) = \dfrac{\sigma^2}{S_{xx}}$	$V(\hat{\beta}_0) = \sigma^2 \left[\dfrac{1}{n} + \dfrac{\bar{x}^2}{S_{xx}} \right]$
unbiased estimator	unbiased estimator

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i \right)^2}{n}$$

Standard errors of coefficients

- We can replace σ^2 with its estimator $\hat{\sigma}^2$ ...

$$\hat{\sigma}^2 = MSE = \frac{\sum_{i=1}^{n} r_i^2}{n-2}$$

$$r_i = y_i - \hat{y}_i \qquad \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

- Using results from previous page, estimate the

$$se(\hat{\beta}_1) = \sqrt{\frac{\hat{\sigma}^2}{S_{xx}}} \qquad \text{and} \qquad se(\hat{\beta}_0) = \sqrt{\hat{\sigma}^2\left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]}$$

Hypothesis test in simple linear regression

- we wish to test the hypothesis whether the slope equals a constant $\beta_{1,0}$

$$H_0: \beta_1 = \beta_{1,0}$$
$$H_1: \beta \; != \; \beta_{1,0}$$

- e.g. relate **ads** to **sales**, we are interested in study whether or not increase a $ on ads will increase $ $\beta_{1,0}$ in sales?
- sale = $\beta_{1,0}$ ads + constant?

A related and important question…

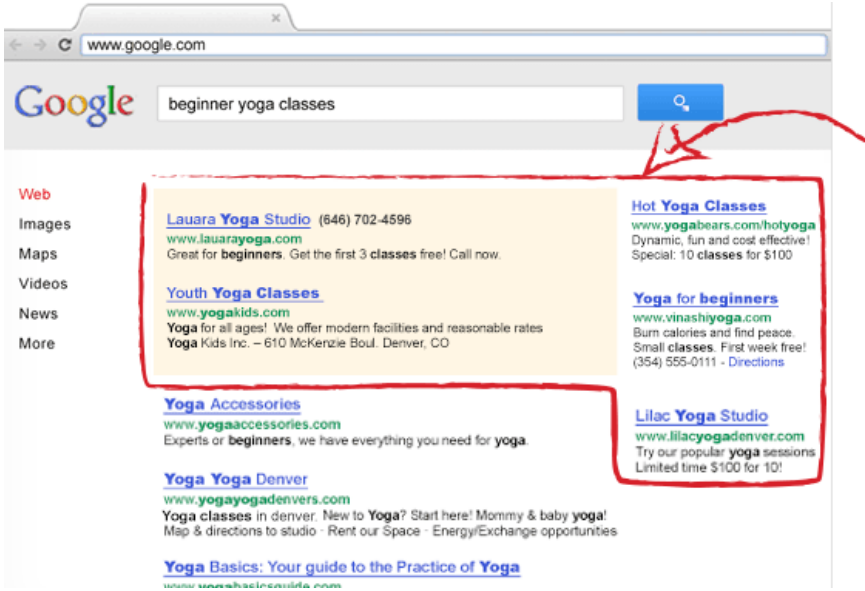
- whether or not the slope is zero ?

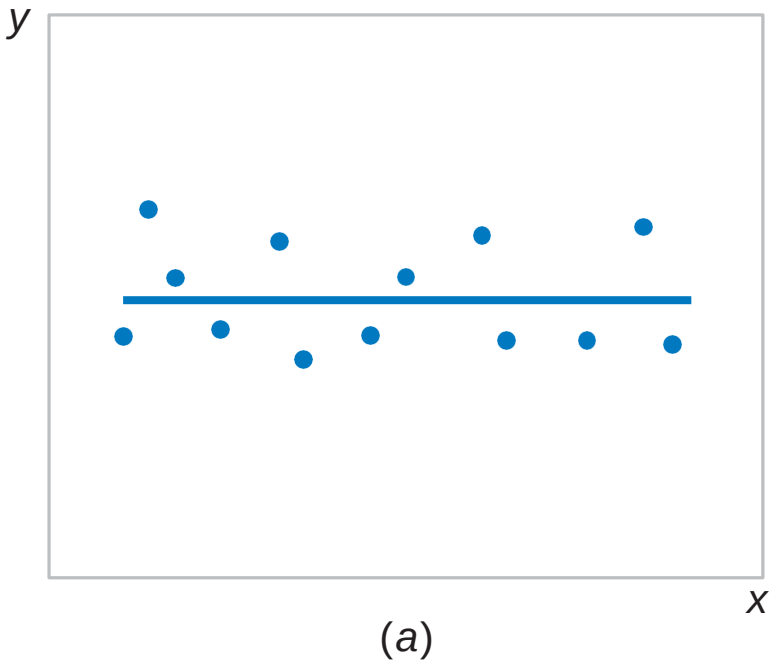
$$H_0: \beta_1 = 0$$

$$H_1: \beta_1 \; != 0$$

Significance of regression

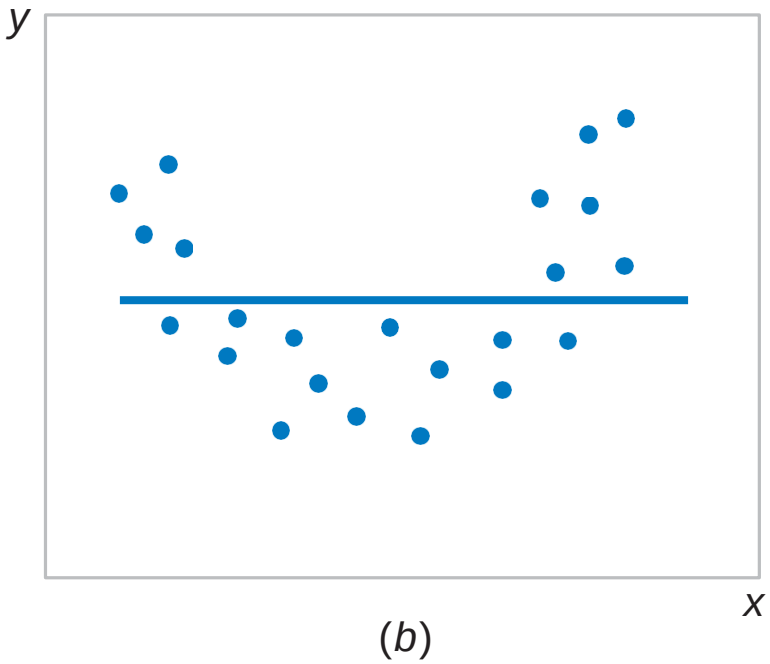
- if $\beta_1 = 0$, that means Y does not depend on X, i.e.,
- Y and X are **independent**
- In the advertisement example, does **ads** increase **sales**? or no effect?





(*a*)

- H_0 not rejected



(*b*)

- H_0 rejected

Use t-test for slope

Under H_0

slope parameter β_1

$$E(\hat{\beta}_1) = \beta_{1,0}$$

$$V(\hat{\beta}_1) = \frac{\sigma^2}{S_{xx}}$$

$$\hat{\beta}_1 \sim N\left(\beta_{1,0}, \sigma^2 / S_{xx}\right)$$

- Under H_0, test statistic

$$T_0 = \frac{\hat{\beta}_1 - \beta_{1,0}}{\sqrt{\hat{\sigma}^2 / S_{xx}}}$$

~ t distribution with n-2 degree of freedom

- Reject H_0 if

$$|t_0| > t_{\alpha/2, n-2}$$

(two-sided test)

Example: oxygen purity tests of coefficients

- Consider the test

$$H_0: \beta_1 = 0$$
$$H_1: \beta_1 \neq 0$$

$\hat{\beta}_1 = 14.947 \quad n = 20,$

$S_{xx} = 0.68088, \quad \hat{\sigma}^2 = 1.18$

- Calculate the test statistic

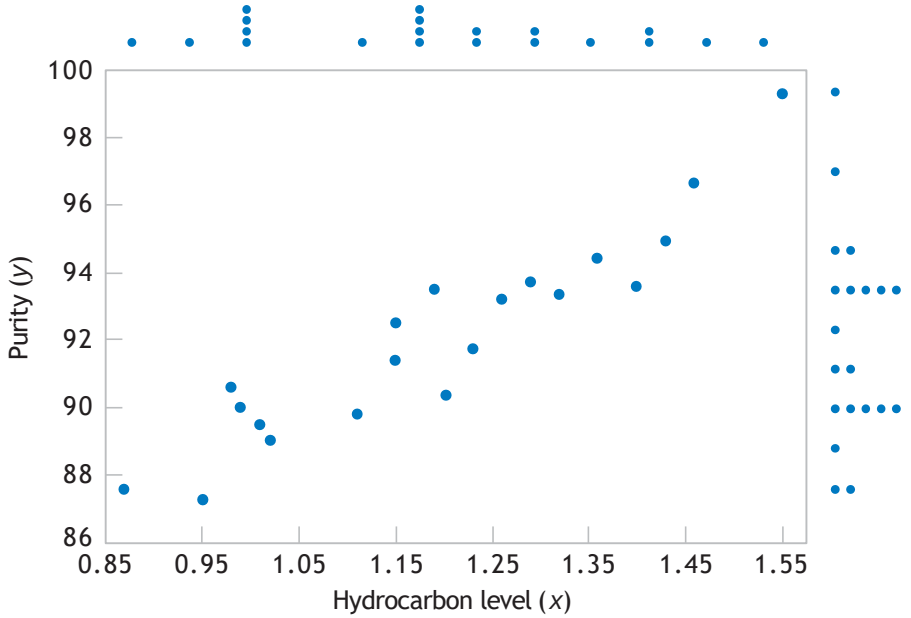


Figure 11-1 Scatter diagram of oxygen purity versus hydrocarbon level from Table 11-1.

$$t_0 = \frac{\hat{\beta}_1}{\sqrt{\hat{\sigma}^2/S_{xx}}} = \frac{\hat{\beta}_1}{se(\hat{\beta}_1)} = \frac{14.947}{\sqrt{1.18/0.68088}} = 11.35$$

- Threshold $t_{\alpha/2,n-2} = t_{0.005,18} = 2.88$
- Reject H_0 since $|t_0| > t_{\alpha/2,n-2}$

Use t-test for intercept

- Use a similar form of test

H_0: $\beta_0 = \beta_{0,0}$

H_1: $\beta_0 \; != \beta_{0,0}$

- Test statistic $$T_0 = \frac{\hat{\beta}_0 - \beta_{0,0}}{\sqrt{\hat{\sigma}^2\left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]}} = \frac{\hat{\beta}_0 - \beta_{0,0}}{se(\hat{\beta}_0)}$$

Under H_0, T_0 ~ t distribution with n-2 degree of freedom

- Reject H_0 if $|t_0| > t_{\alpha/2,n-2}$

Confidence interval

- we can obtain confidence interval estimates of slope and intercept
- width of confidence interval is a measure of the overall quality of the regression

true parameter

slope	intercept
$T_0 = \frac{\hat{\beta}_1 - \beta_{1,0}}{\sqrt{\hat{\sigma}^2/S_{xx}}}$	$T_0 = \frac{\hat{\beta}_0 - \beta_{0,0}}{\sqrt{\hat{\sigma}^2\left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]}}$
~ t distribution with n-2 degree of freedom	~ t distribution with n-2 degree of freedom

Confidence intervals

a $100(1-\alpha)\%$ **confidence interval on the slope** β_1

$$\hat{\beta}_1 - t_{\alpha/2,n-2}\sqrt{\frac{\hat{\sigma}^2}{S_{xx}}} \le \beta_1 \le \hat{\beta}_1 + t_{\alpha/2,n-2}\sqrt{\frac{\hat{\sigma}^2}{S_{xx}}}$$

a $100(1-\alpha)\%$ **confidence interval on the intercept** β_0

$$\hat{\beta}_0 - t_{\alpha/2,n-2}\sqrt{\hat{\sigma}^2\left[\frac{1}{n}+\frac{\bar{x}^2}{S_{xx}}\right]} \le \beta_0 \le \hat{\beta}_0 + t_{\alpha/2,n-2}\sqrt{\hat{\sigma}^2\left[\frac{1}{n}+\frac{\bar{x}^2}{S_{xx}}\right]}$$

Example: oxygen purity tests of coefficients

find a 95% confidence interval on the slope $(\alpha = 0.05)$

$$\hat{\beta}_1 = 14.947, S_{xx} = 0.68088, \text{and } \hat{\sigma}^2 = 1.18$$

$$\hat{\beta}_1 - t_{0.025,18}\sqrt{\frac{\hat{\sigma}^2}{S_{xx}}} \le \beta_1 \le \hat{\beta}_1 + t_{0.025,18}\sqrt{\frac{\hat{\sigma}^2}{S_{xx}}}$$

$$14.947 - 2.101\sqrt{\frac{1.18}{0.68088}} \le \beta_1 \le 14.947 + 2.101\sqrt{\frac{1.18}{0.68088}}$$

$$12.181 \le \beta_1 \le 17.713$$

The confidence interval does not include 0, so enough evidence saying there is enough correlation between X and Y.

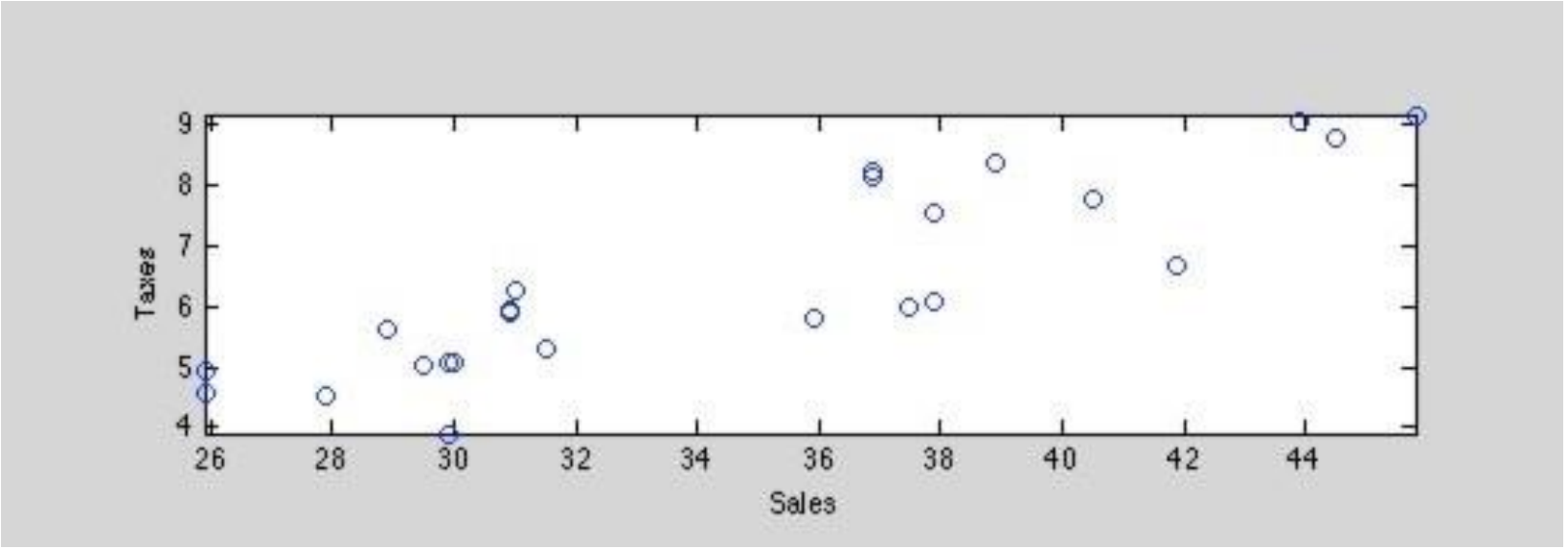
Example: house selling price and annual taxes

Sale Price/1000	Taxes (Local, School), County)/1000	Sale Price/1000	Taxes (Local, School), County)/1000
25.9	4.9176	30.0	5.0500
29.5	5.0208	36.9	8.2464
27.9	4.5429	41.9	6.6969
25.9	4.5573	40.5	7.7841
29.9	5.0597	43.9	9.0384
29.9	3.8910	37.5	5.9894
30.9	5.8980	37.9	7.5422
28.9	5.6039	44.5	8.7951
35.9	5.8282	37.9	6.0831
31.5	5.3003	38.9	8.3607
31.0	6.2712	36.9	8.1400
30.9	5.9592	45.8	9.1416

Independent variable X: SalePrice

Dependent variable Y: Taxes

- qualitative analysis



Calculate correlation

$$\hat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \times \sum_{i=1}^{n}(y_i - \bar{y})^2}} = 0.8760$$

Independent variable Y: SalePrice

Dependent variable X: Taxes

$$n = 24 \qquad \bar{x} = 34.6125 \qquad \bar{y} = 6.4049$$

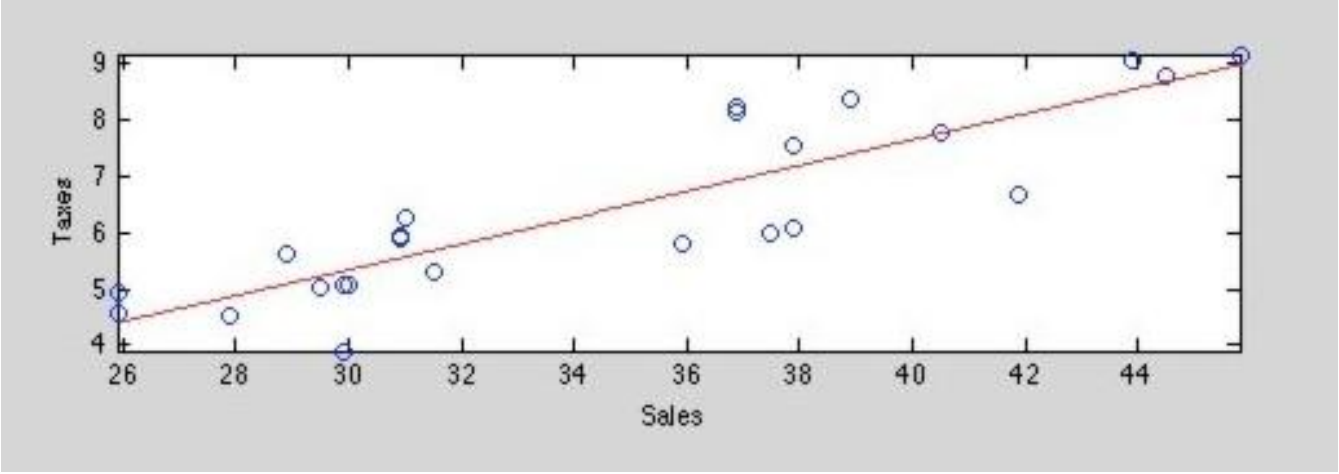
$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2 = 829.0462$$

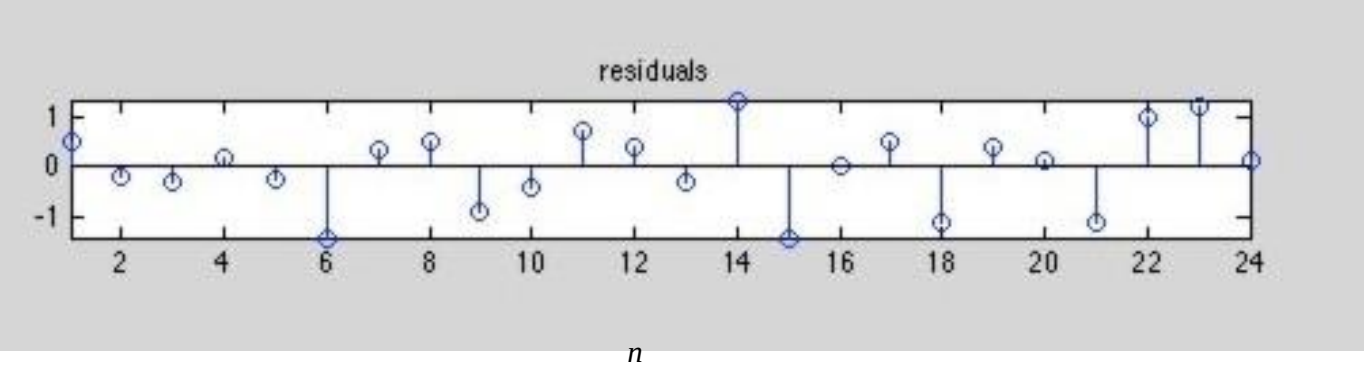
$$S_{xy} = \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) = 191.3612$$

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{191.3612}{829.0462} = 0.2308$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 6.4049 - 0.2308 \times 34.6125 = -1.5837$$

Fitted simple linear regression model $\hat{y} = -1.5837 + 0.2308x$





- residuals: $\hat{\sigma}^2 = MSE = \frac{\sum_{i=1}^{n} r_i^2}{n-2} = 0.6088$

- standard error of regression coefficients

$$se(\hat{\beta}_1) = \sqrt{\frac{\hat{\sigma}^2}{S_{xx}}} = \sqrt{\frac{0.6088}{829.0462}} = 0.0271$$

$$se(\hat{\beta}_0) = \sqrt{\hat{\sigma}^2\left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]} = \sqrt{0.6088\left[\frac{1}{24} + \frac{34.6125^2}{829.0462}\right]} = 0.9514$$

- test

Test H_0: $\beta_1 = 0$ using the t-test; use $\alpha = 0.05$

- calculate test statistics

$$t_0 = \frac{\hat{\beta}_1}{\sqrt{\hat{\sigma}^2/S_{xx}}} = \frac{\hat{\beta}_1}{se(\hat{\beta}_1)} = \frac{0.2308}{0.0271} = 8.5166$$

- threshold

$$t_{\alpha/2,n-2} = t_{0.0025,22} = 3.119$$

- value of test statistic is greater than threshold
-  reject H_0

- construct confidence interval for slope parameter

$$\hat{\beta}_1 - t_{\alpha/2,n-2}\sqrt{\frac{\hat{\sigma}^2}{S_{xx}}} \le \beta_1 \le \hat{\beta}_1 + t_{\alpha/2,n-2}\sqrt{\frac{\hat{\sigma}^2}{S_{xx}}}$$

$$t_{\alpha/2,n-2} = t_{0.0025,22} = 3.119$$

$$0.2308 - 3.119 \times 0.0271 \le \beta_1 \le 0.2308 + 3.119 \times 0.0271$$

$$0.14631 \le \beta_1 \le 0.3153$$