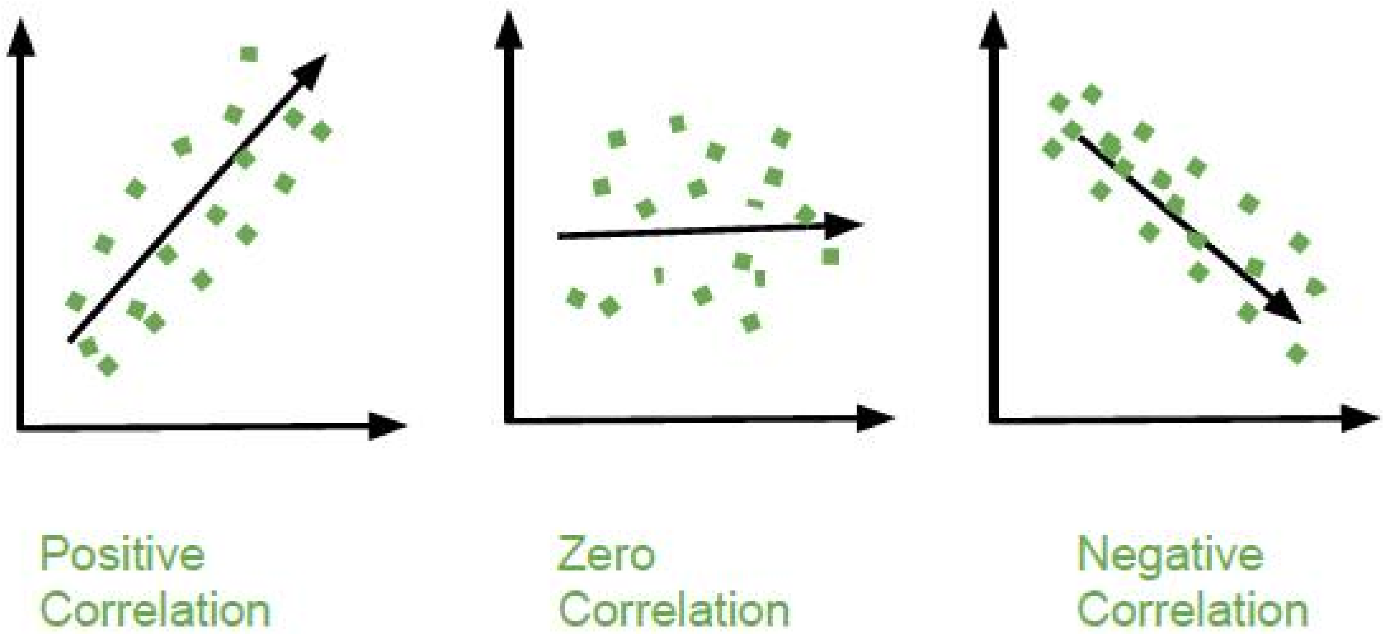
(Linear) Regression

Nazerfard, Ehsan
nazerfard@aut.ac.ir

Basics

- Supervised learning
  - Regression (target output is a numeric variable/quantity)
  - Classification (target is a discrete variable/label)
- Discrete vs. numeric variable
- Discretizing a numeric variable
- Francis Galton first discovered the concepts of regression and correlation both.

Correlation



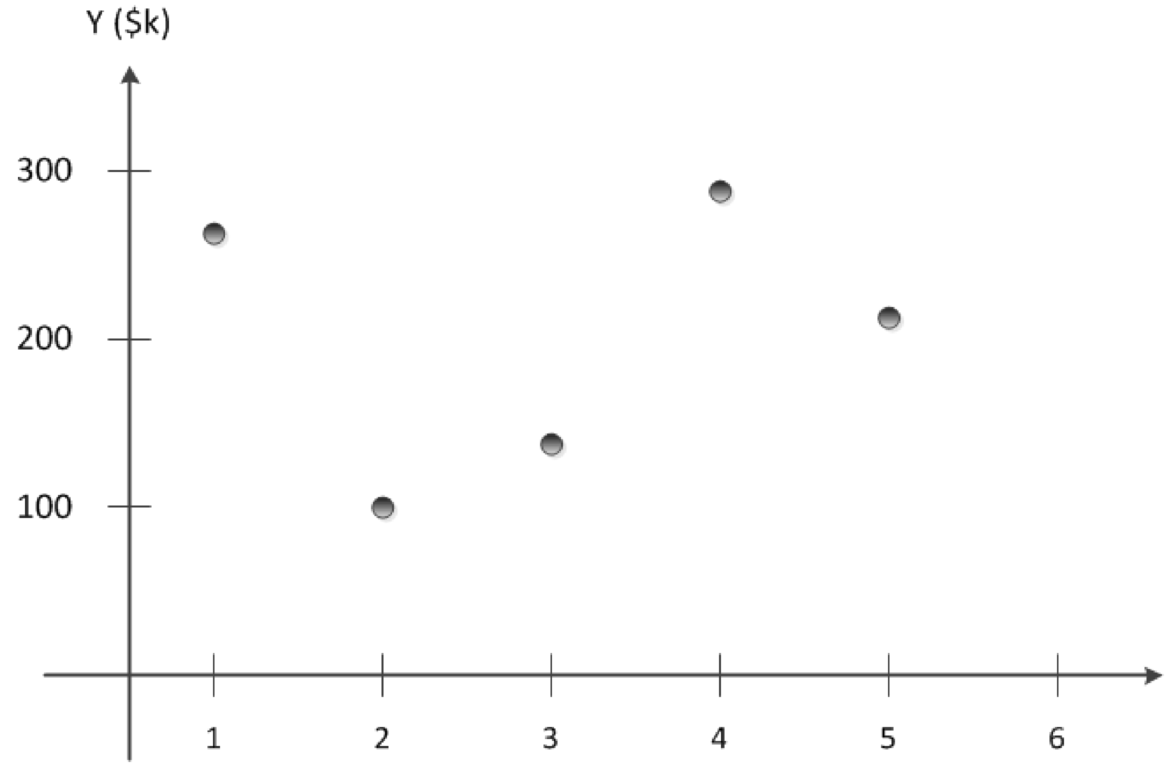
- $\rho_{XY} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathbb{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}, \ -1 \leq \rho_{XY} \leq 1$

Figures © https://www.geeksforgeeks.org

House Price Prediction

- Training data

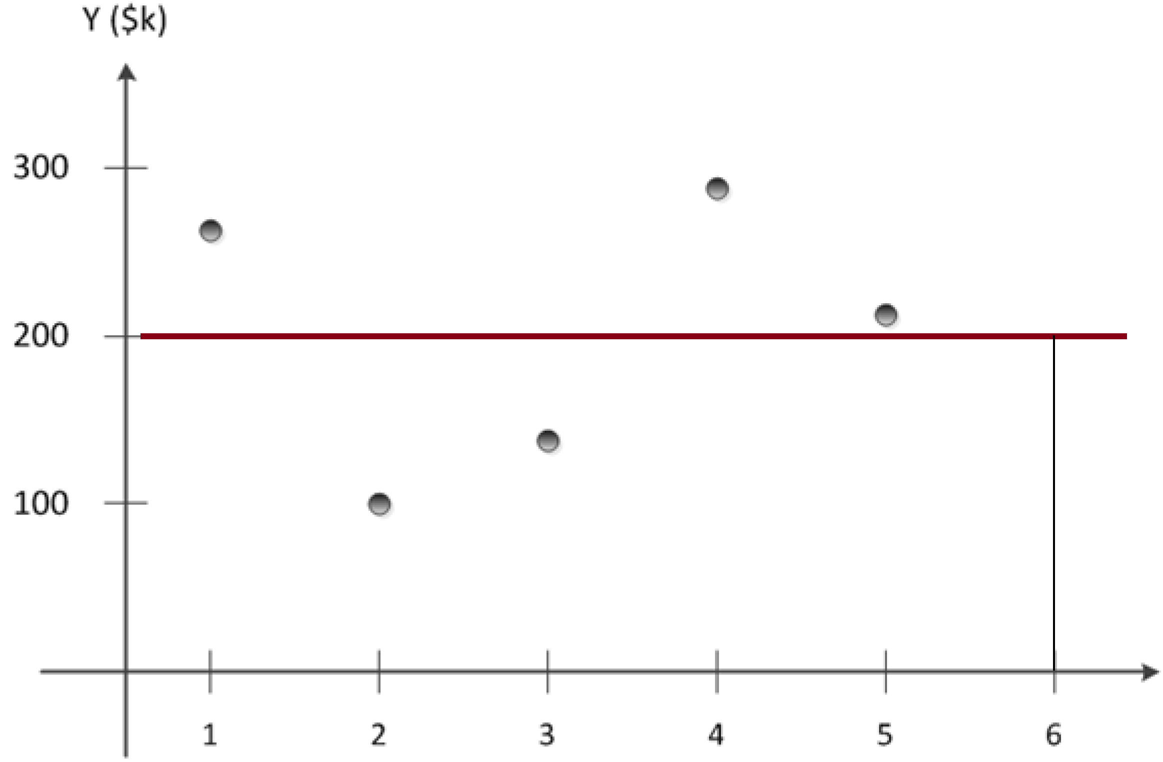
	Price ($k)
1	260
2	100
3	140
4	290
5	210



House Price Prediction

- Training data

	Price ($k)
1	260
2	100
3	140
4	290
5	210



$$\bar{y} = \$200k \text{ (a high bias answer)}$$

$$residual\ error_i = \bar{y} - y^i$$

Training Data

- A sample training data
  - m=#of training data (here m=10)
  - n= #of features (here n=1) → Univariate

m	Size in $feet^2$ (X)	Price in $k (Y)
1	700	158
2	1060	230
3	582	120
4	830	175
..	.	.
..	.	.
10	940	194

Regression

- Goal is to learn a hypothesis function *h*, which maps input variable(s) to a numeric output variable:

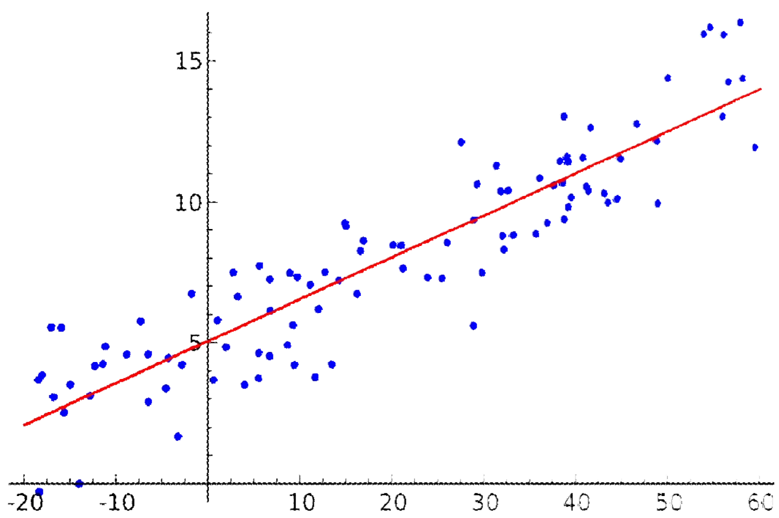
$$h: X \to \mathbb{R}$$

Regression (cont.)

- Goal is to learn a hypothesis function *h*, which maps input variable(s) to a numeric output:

$$h: X \rightarrow \mathbb{R}$$

- An example for Linear regression (LR):



LR Hypothesis Space

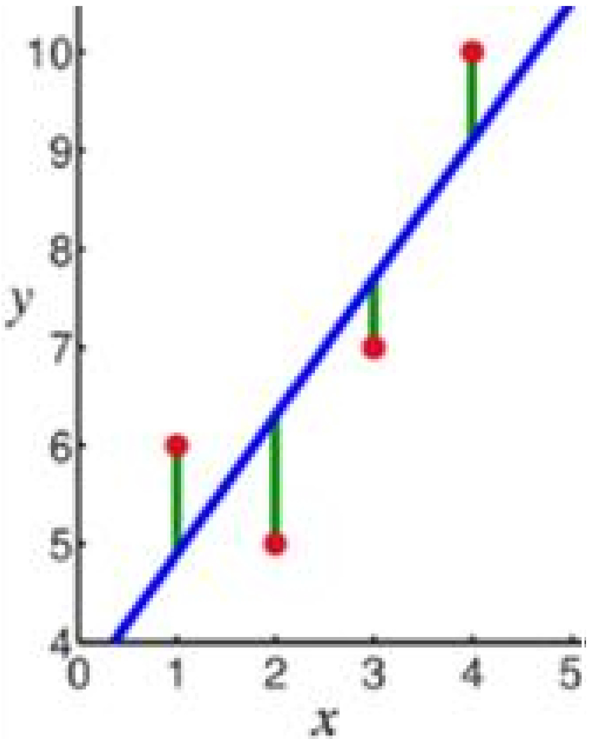
- Linear Regression Hypothesis space:

$$h(x) = \theta_0 + \theta_1 x$$

  - Goal is to learn parameters θ_0, θ_1
  - A parametric model
  - Occam's razor principle

Best Fit Regression Line

- How to find the best fit regression line?



How to Learn Parameters

- Parameters θ_0 and θ_1 are estimated by minimizing a cost/loss function:

$$\underset{\theta_0, \theta_1}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^{m} \left(h\left(x^i\right) - y^i \right)^2$$

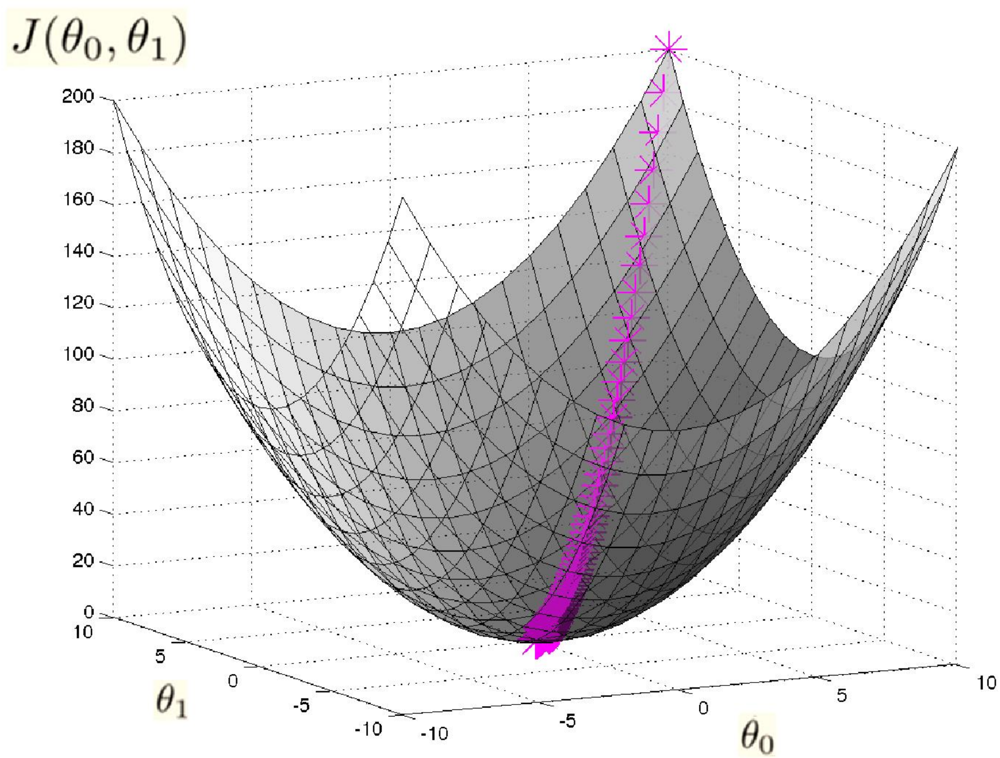
- The above loss function is referred to as mean square error (MSE) function → L2 loss

- Other loss functions: MAE, Hubber, Log-Cosh, Quantile, …

Mean Square Error Cost Function

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

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



Minimizing the Cost Function

- Minimizing the cost function in order to learn parameters $\theta's$:

- Two approaches:
  - Iterative Gradient Descent
  - Non-iterative Normal Equation (later on in this lecture)

Simplified Gradient Descent Algorithm

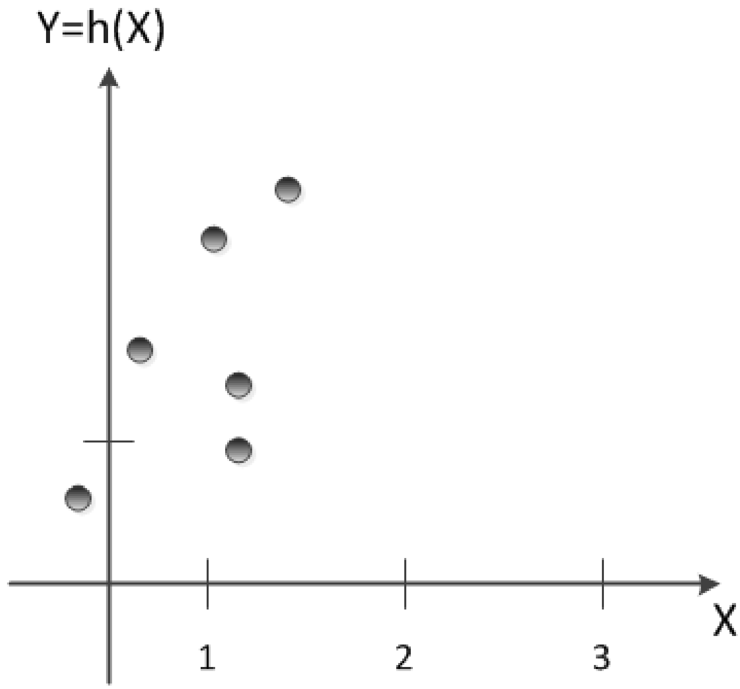
- For simplicity, assume that θ_0 is zero.

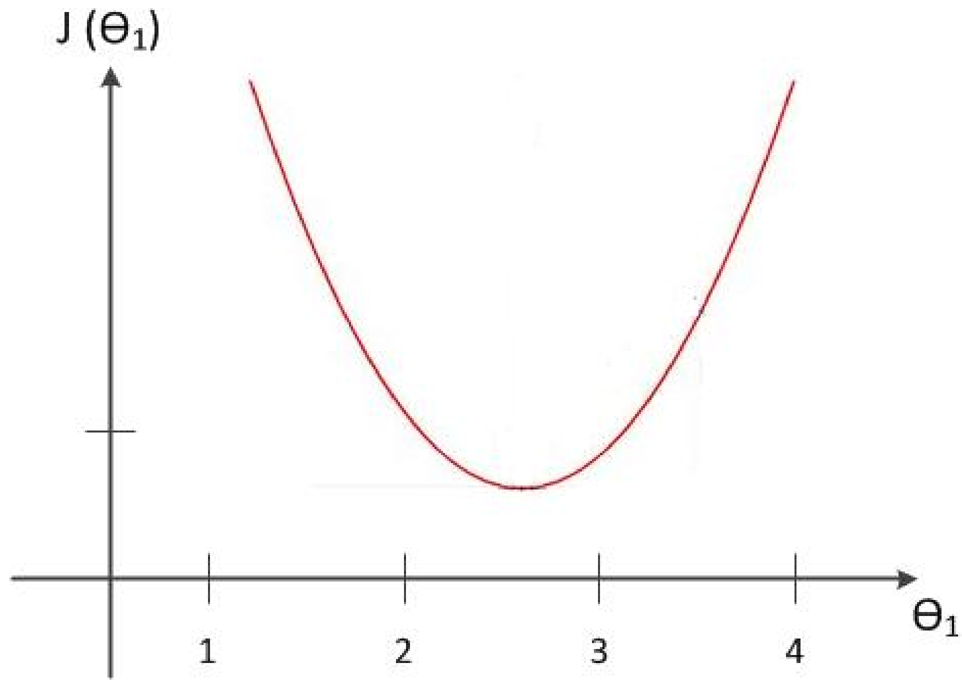
- Initialize θ_1

- Repeat until convergence:

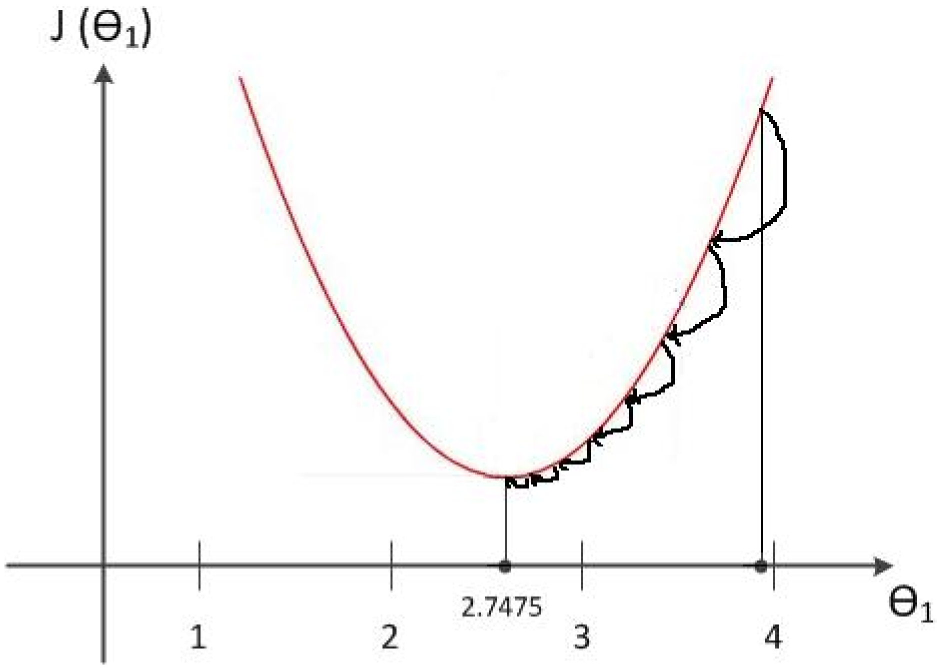
$$\theta_1 \leftarrow \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1) \quad (\text{subject to: } \alpha > 0)$$

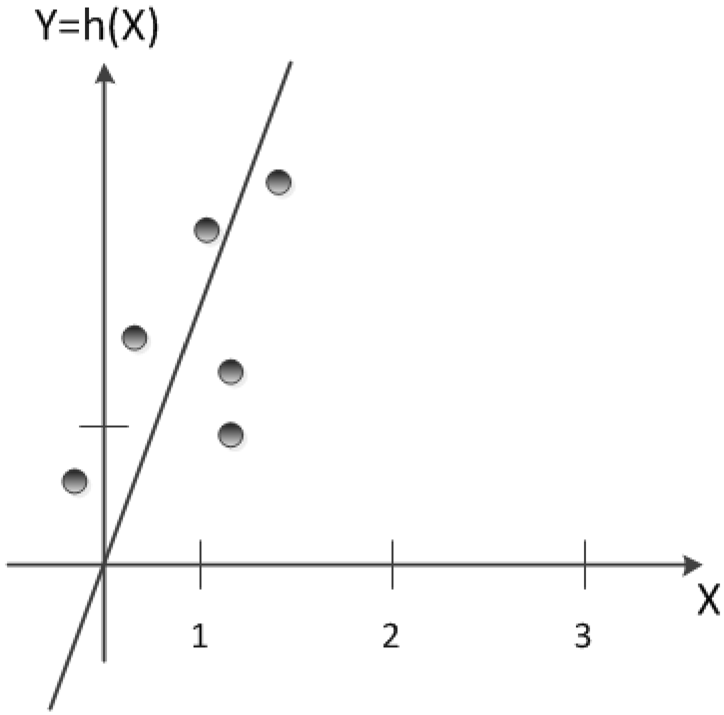
Gradient Descent Example





Gradient Descent Example (cont.)





Multivariate Linear Regression

- Multivariate linear regression:

$$h(\boldsymbol{x}) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \ ,$$

where $\boldsymbol{x} = [x_1, x_2, \dots, x_n]^T$

- Feature engineering

- Feature Selection
  - Domain expert knowledge
  - Feature scaling
- Feature Extraction
- Adding Polynomial Features
- Dimensionality Reduction
  - Principal component analysis (PCA)
  - Auto Encoders
- …
- Computers Teach Themselves to Recognize Cats, Faces: by Peter Norvig
- Representation Learning, Yoshua Bengio et al.

Self Learned Cat!

Gradient Descent Algorithm

- Initialization

- Repeat until convergence:

for j=1 to n do

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_{0..n})$$

(subject to: $0 \leq j \leq n,\ \alpha > 0$)

Gradient Descent Algorithm (cont.)

- Initialization
- Repeat until convergence:

$$\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^{k} - \alpha \boldsymbol{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Where gradient vector is defined as follows:

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} J(\boldsymbol{\theta}) \\ \frac{\partial}{\partial \theta_1} J(\boldsymbol{\theta}) \\ \vdots \\ \frac{\partial}{\partial \theta_n} J(\boldsymbol{\theta}) \end{bmatrix}$$

How to Use Training Data

- Holdout method: split up your data into a train and a test set.

Size	Price
1020	247
806	133
1584	390
550	117
1682	403
…	…
1054	275
988	168

Training set
60% -- 70%

Test set
30% -- 40%

Polynomial Regression

- A type of univariate non-linear regression:

$$h(x) = \theta_0 + \cdots + \theta_i \boxed{x^i} + \cdots + \theta_d x^d$$

$$\Downarrow$$

$$h(\boldsymbol{x}) = \theta_0 + \cdots + \theta_i \boxed{x_i} + \cdots + \theta_d x_d$$

- Parameter Learning:
  - Moving to a higher dimension allows us to treat the polynomial regression as a linear regression problem.

Example

- Moving to a higher dimension, an example:

  a) $h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$

  b) $h(\boldsymbol{x}) = \theta_0 + \theta_1 x + \theta_2 x_2$

- Suppose that parameters in (b), namely $\theta_0, \theta_1, \theta_2$, are estimated as $(1, 0, 1\ )$, using the gradient descent algorithm:

$$y = x_2 + 1 \xrightarrow{\text{going back to the original space}} y = x^2 + 1$$

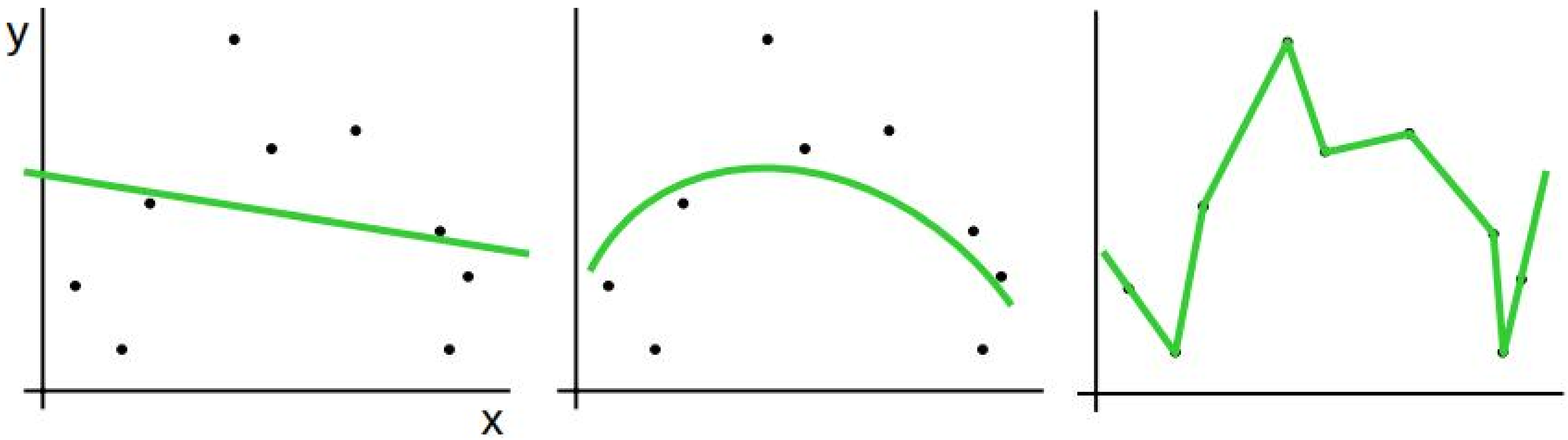
Polynomial Regression (cont.)

- Model Selection:
  - How to select the hyper-parameter d, for a given set of data points?

$$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d$$

Over/Under Fitting

- Considering the given data points, which model is best and why?



Figures © https://stackoverflow.com

Validation Set

- Parameter Learning vs. Model Selection
- Why do we need a validation set?

Size	Price
1020	247
806	133
1584	390
550	117
1682	403
…	…
1054	275
988	168

Training data 60%
m_{tr}

Validation data 20%
m_v

Test data 20%
m_{ts}

Model Selection

- Model selection boils down to choosing the hyper parameter d (polynomial degree).

d	Hypothesis	Type	Parameter Learning $\boldsymbol{min\, J_{tr}(\theta)}$	Error on Validation data
1	$h(x) = \theta_0 + \theta_1 x$	linear	$\boldsymbol{\theta_1} = [\theta_0, \theta_1]^T$	$J_v(\boldsymbol{\theta_1})$
2	$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$	quadratic	$\boldsymbol{\theta_2} = [\theta_0, \theta_1, \theta_2]^T$	$J_v(\boldsymbol{\theta_2})$
3	$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	cubic	$\boldsymbol{\theta_3} = [\theta_0, \theta_1, \theta_2, \theta_3]^T$	$J_v(\boldsymbol{\theta_3})$
4	…	quartic	…	
5	…	quintic	…	

Model Selection (cont.)

- Model selection boils down to choosing the hyper parameter d (polynomial degree).

d	Hypothesis	Type	Parameter Learning $\min J_{tr}(\boldsymbol{\theta})$	Error on Validation data
1	$h(x) = \theta_0 + \theta_1 x$	linear	$\boldsymbol{\theta_1} = [\theta_0, \theta_1]^T$	$J_v(\boldsymbol{\theta_1})$
2	$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$	quadratic	$\boldsymbol{\theta_2} = [\theta_0, \theta_1, \theta_2]^T$	$J_v(\boldsymbol{\theta_2})$
3	$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$	cubic	$\boldsymbol{\theta_3} = [\theta_0, \theta_1, \theta_2, \theta_3]^T$	$J_v(\boldsymbol{\theta_3})$
4	…	quartic	…	…
5	…	quintic	…	…

$$\boxed{d = \underset{k}{\operatorname{argmin}}\, J_v(\boldsymbol{\theta_k})}$$

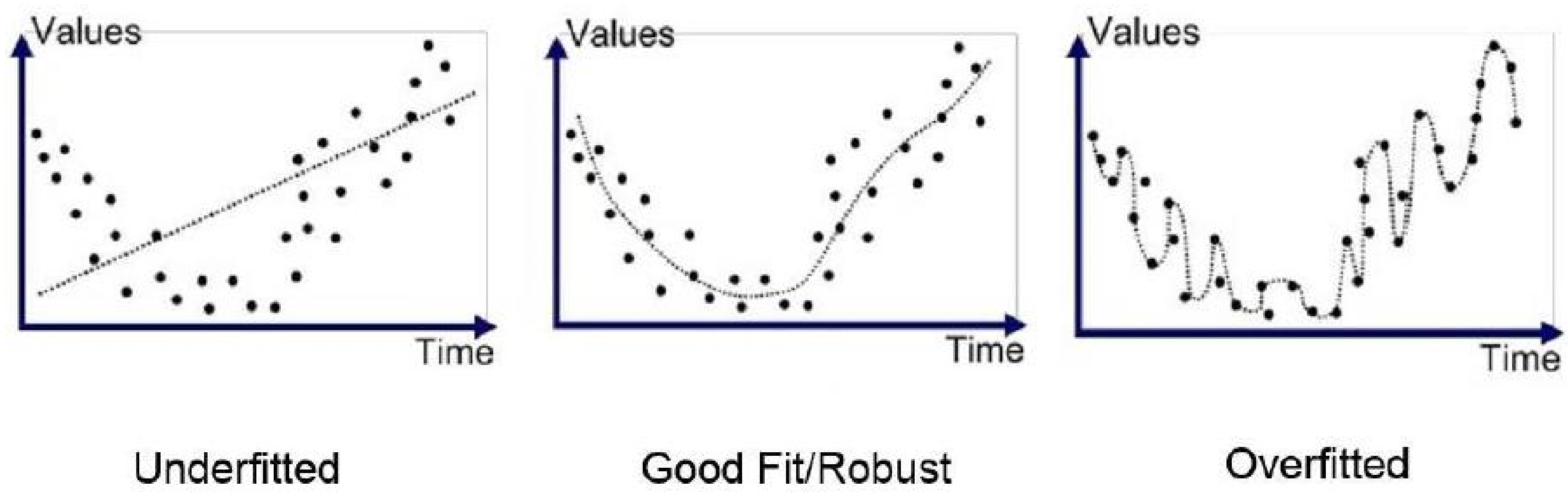
Why Validation Data: Intuition

- Borrowed from a Data Mining Contest

You are limited to one submission per day and three total. On the final day, you are no longer restricted to one submission per day.

Under/Over Fitting

- High bias vs. high variance model



Bias/Variance Tradeoff

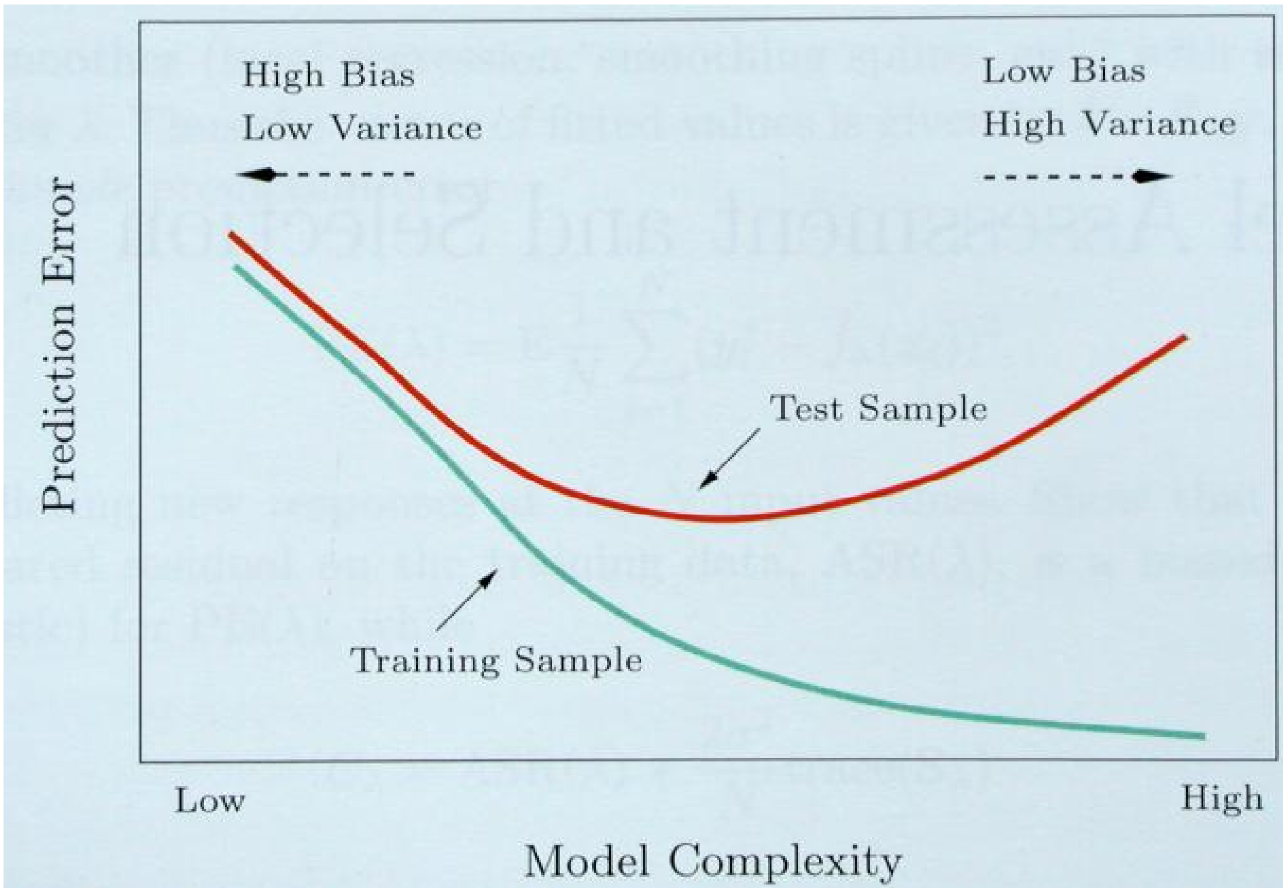


Figure © Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001

Bias

1. Likelihood that a learner doesn’t change its mind.

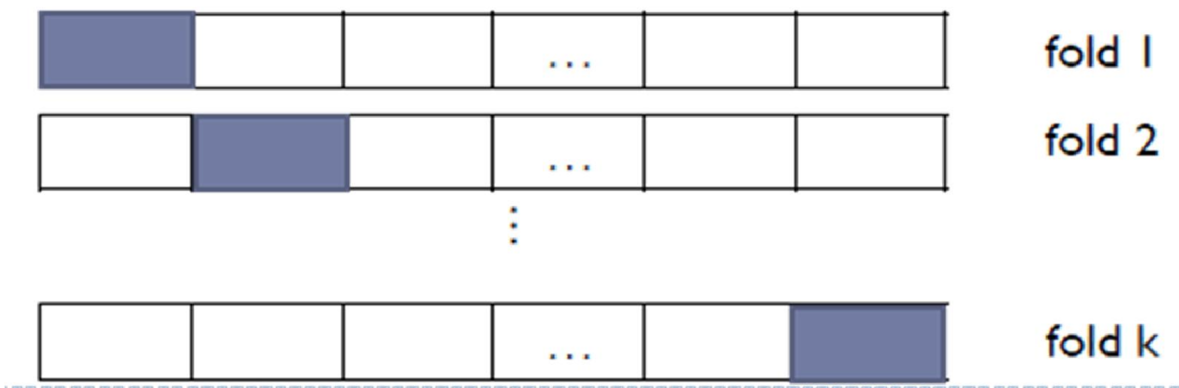
2. How to push an ML algorithm in one direction, which is imposed from outside. (eg. Regularization, maximizing margin)

3. Each learning algorithm has an inductive bias that imposes a preference on the space of all possible hypotheses.

Cross Validation

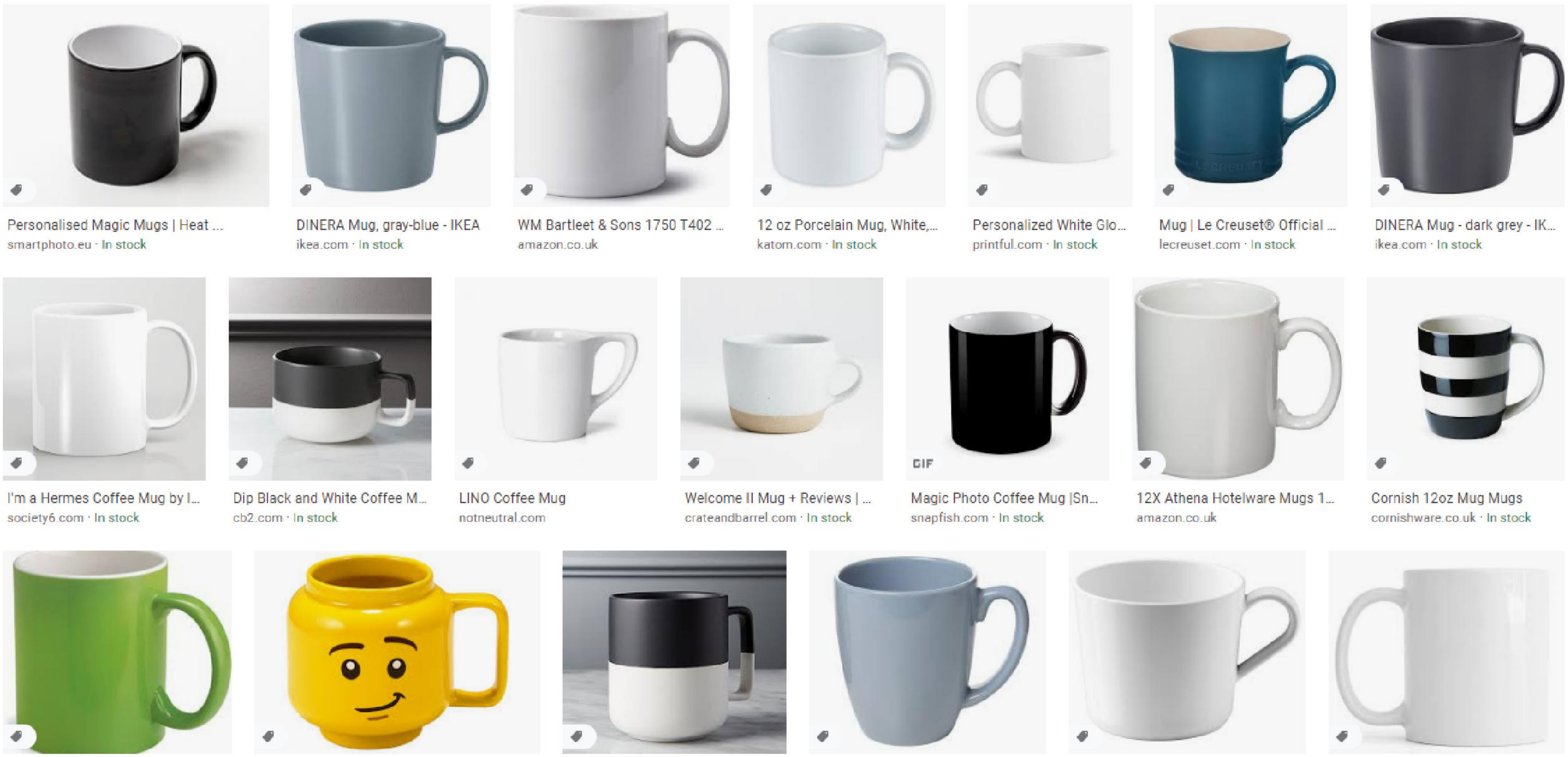
- Holdout method may be misleading if we happen to get an unfortunate split.

- k–fold cross validation (or out-of-sample testing):



  - Leave-one-out cross validation

Dataset Bias – Example



Regularization

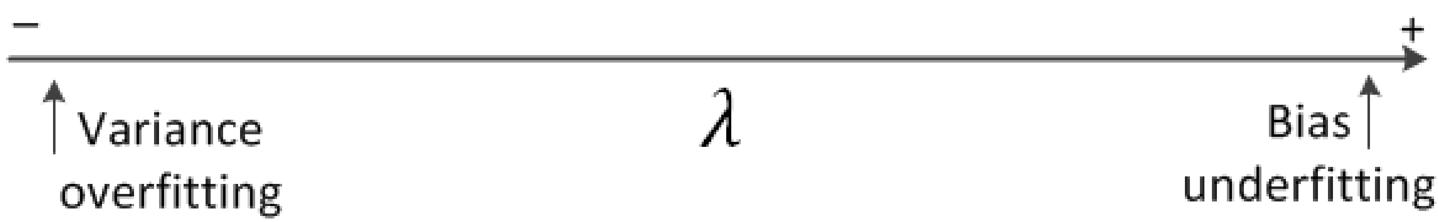
- Modify cost function to add preference or **bias** for certain parameters near zero to prevent overfitting.

$$J(\theta) = \frac{1}{2m}[\sum_{i=1}^{m}(h\left(x^{i}\right) - y^{i})^2 + \lambda\sum_{j=1}^{n}\theta_j^2]$$

Regularization (cont.)

- Modify cost function to add preference or **bias** for certain parameters near zero to prevent overfitting.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h\left(x^{i}\right) - y^{i}\right)^{2} + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$



Minimizing the Cost Function

- Minimizing the cost function in order to learn parameters $\theta's$:

- Two approaches:
  - Iterative Gradient Descent (already discussed)
  - Non-iterative Normal Equation

- Note: Either approach can be used for both Univariate and Multivariate models.

Notations

- Capital letters → random variables
- Lower case letters → values of random variables also for scalars
- Bold lower case letters → real-valued vectors (even if random variables)
- Bold capital letters → matrices

Normal Equation

- Here we discuss the normal equation for multivariate models:

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

goal is to minimize $J(\theta_{0,..n}) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x^i) - y^i\right)^2$

- Let's assume $\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x_0 = 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$: $h(x) = \boldsymbol{\theta}^{\boldsymbol{T}} \boldsymbol{x}$

Design Matrix

- We define the design matrix as the following:

$$\mathbf{X}_{\mathrm{m}\times(\mathrm{n}+1)} = \begin{bmatrix} (\boldsymbol{x}^1)^T \\ (\boldsymbol{x}^2)^T \\ \vdots \\ (\boldsymbol{x}^m)^T \end{bmatrix}$$

- The matrix form of the cost function (why?):

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

Justification

- $\boldsymbol{y} = \boldsymbol{X\theta} + \boldsymbol{\epsilon}$ (goal is to minimize error $\boldsymbol{\epsilon}$)

$$\boldsymbol{\epsilon} = \boldsymbol{y} - \boldsymbol{X\theta} \rightarrow \|\boldsymbol{\epsilon}\| = \|\boldsymbol{y} - \boldsymbol{X\theta}\|$$

$$\|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{y} - \boldsymbol{X\theta}\|^2 = (\boldsymbol{y} - \boldsymbol{X\theta}).(\boldsymbol{y} - \boldsymbol{X\theta})$$

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

Normal Equation

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

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

$$= (\boldsymbol{X\theta})^T\boldsymbol{X\theta} - 2(\boldsymbol{X\theta})^T\boldsymbol{y} + \mathbf{y}^{\mathrm{T}}\mathbf{y}$$

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = 2\boldsymbol{X}^T\boldsymbol{X\theta} - 2\boldsymbol{X}^T\mathbf{y} = 0 \rightarrow \boxed{\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}$$

Normal Equation (cont.)

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

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

$$= (\boldsymbol{X\theta})^T\boldsymbol{X\theta} - 2(\boldsymbol{X\theta})^T\boldsymbol{y} + \mathbf{y}^{\mathbf{T}}\mathbf{y}$$

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = 2\boldsymbol{X}^T\boldsymbol{X\theta} - 2\boldsymbol{X}^T\mathbf{y} = 0 \rightarrow \boxed{\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}$$

- When $\boldsymbol{X}^T\boldsymbol{X}$ is not invertible?

Evaluation

- L2 error: LSE → MSE → RMSE

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

- L1 error: LAD → MAD

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left|h\left(x^i\right) - y^i\right|$$

- Which one is less sensitive to outliers? MSE or MAE?

Further Reading

- Regression toward the mean
- Stochastic Gradient Descent
  - SAG: Stochastic Average Gradient
  - SAGA: A Fast Incremental Gradient Method
- K-nearest Neighbors Regressionm Support Vector Regression (SVR), Random Forest Regression, Gradient Boosting Machine (GBM), Recurrent Neural Networks (RNNs), LASSO Regression, Ridge Regression
- Sparsity and the Lasso
- Logistic Regression (!)
- …
- Representation Learning
  - Yoshua Bengio, Aaron Courville, Pascal Vincent, "Representation Learning: A Review and New Perspectives", 2014.