SE125 Machine Learning

Linear Regression

Yue Ding
School of Software, Shanghai Jiao Tong University
dingyue@sjtu.edu.cn

References and Acknowledgement

- Prof. Weinan Zhang's machine learning course for ACM class (CS420)
    - http://wnzhang.net/teaching/cs420/slides/2-linear-model.pdf
    - http://wnzhang.net

- Getting Started with Machine Learning, Jim Liang

Linear Regression

课程难度:

掌握程度:

Introduction to Linear Regression

- Regression is a statistical procedure that determines the equation for the straight line that best fits a specific set of data.

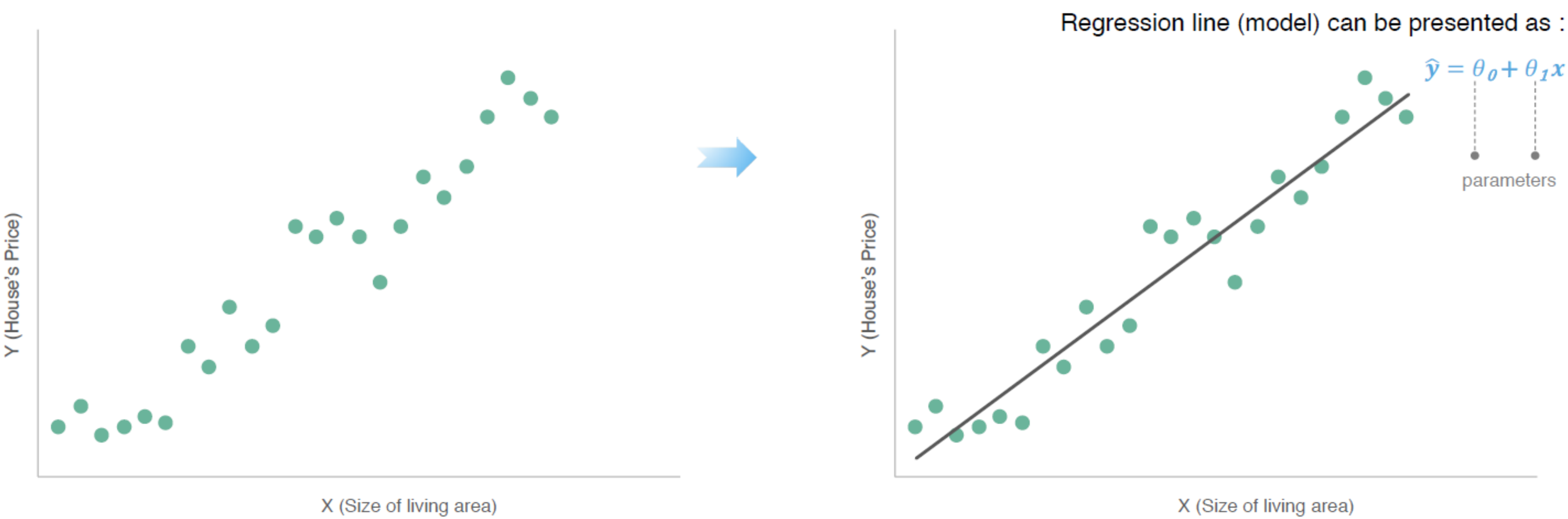
Introduction to Linear Regression

Living Area (Feet^2)	Price ($)
1180	221,900
2570	538,000
770	180,000
1960	604,000
1680	510,000
5420	1,225,000
1715	257,500
1060	291,850
1780	229,500
1890	323,000
3560	662,500
1160	468,000
1430	310,000
1370	400,000
1810	530,000
...	...
x	y

Data set

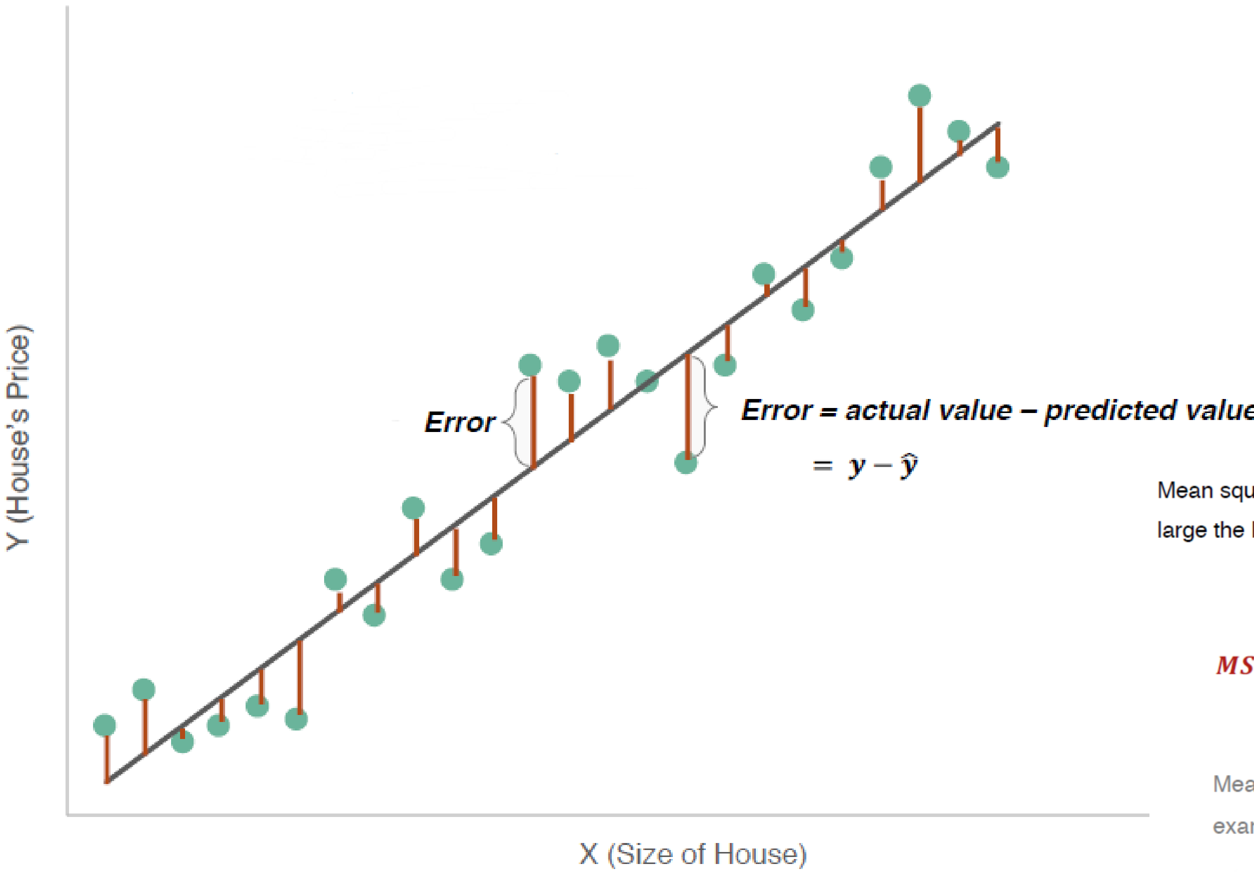
Introduction to Linear Regression

- The straight line can be seen in the plot, showing how linear regression attempts to find the best-fit line to represent the relationship between the input feature *x* and the target *y*.



Introduction to Linear Regression

- Loss (i.e. error) is a number indicating how bad the model's prediction is on a single example. The smaller the error, the better the fit of the line to the data.



Mean square error (MSE) is a commonly-used function to measure how large the loss is. It's called as **Loss function** or **Cost function**.

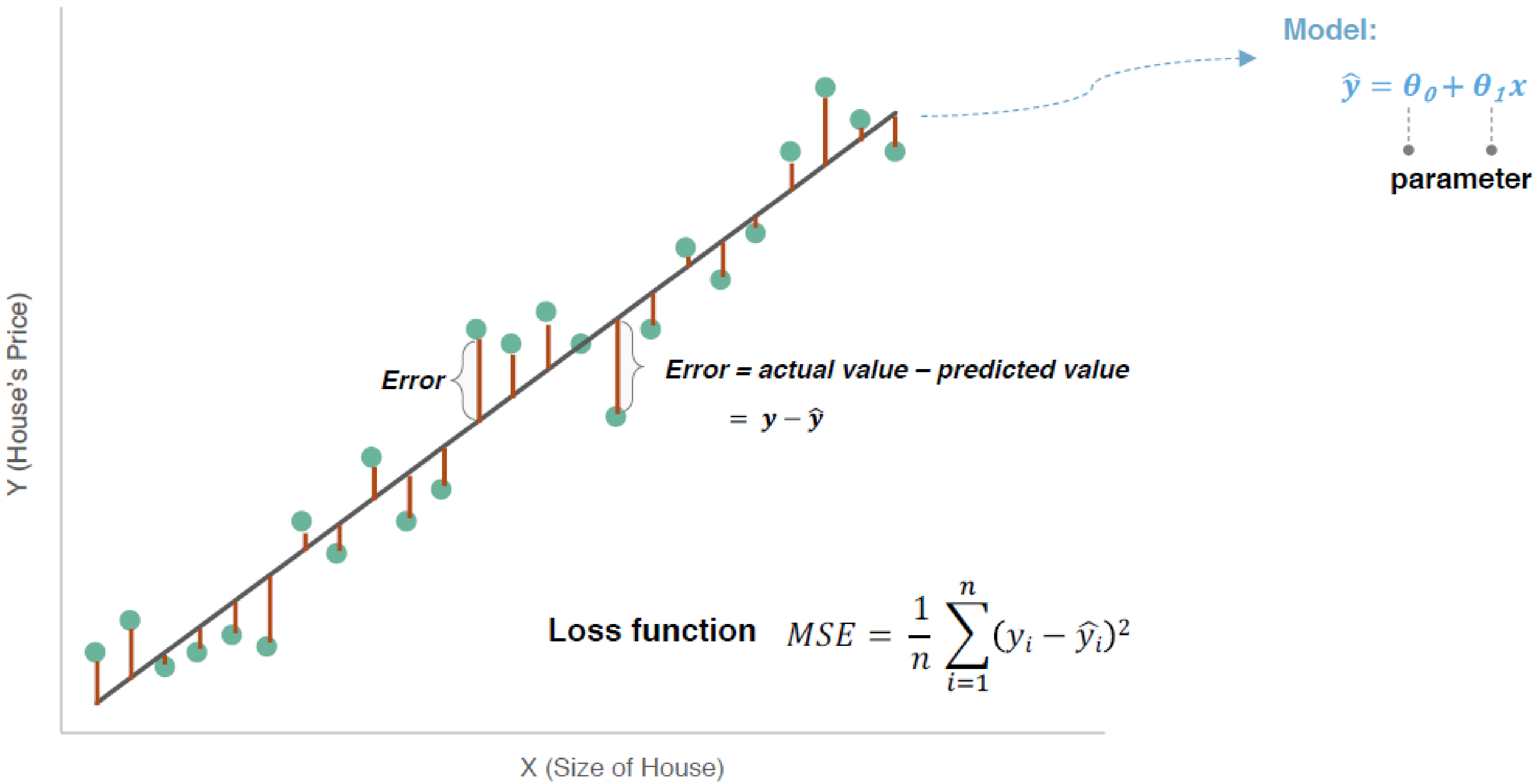
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$\hat{y}_i$ *is the prediction*
y_i *is the actual value*

Mean square error (MSE) is the average squared loss per example over the whole dataset.

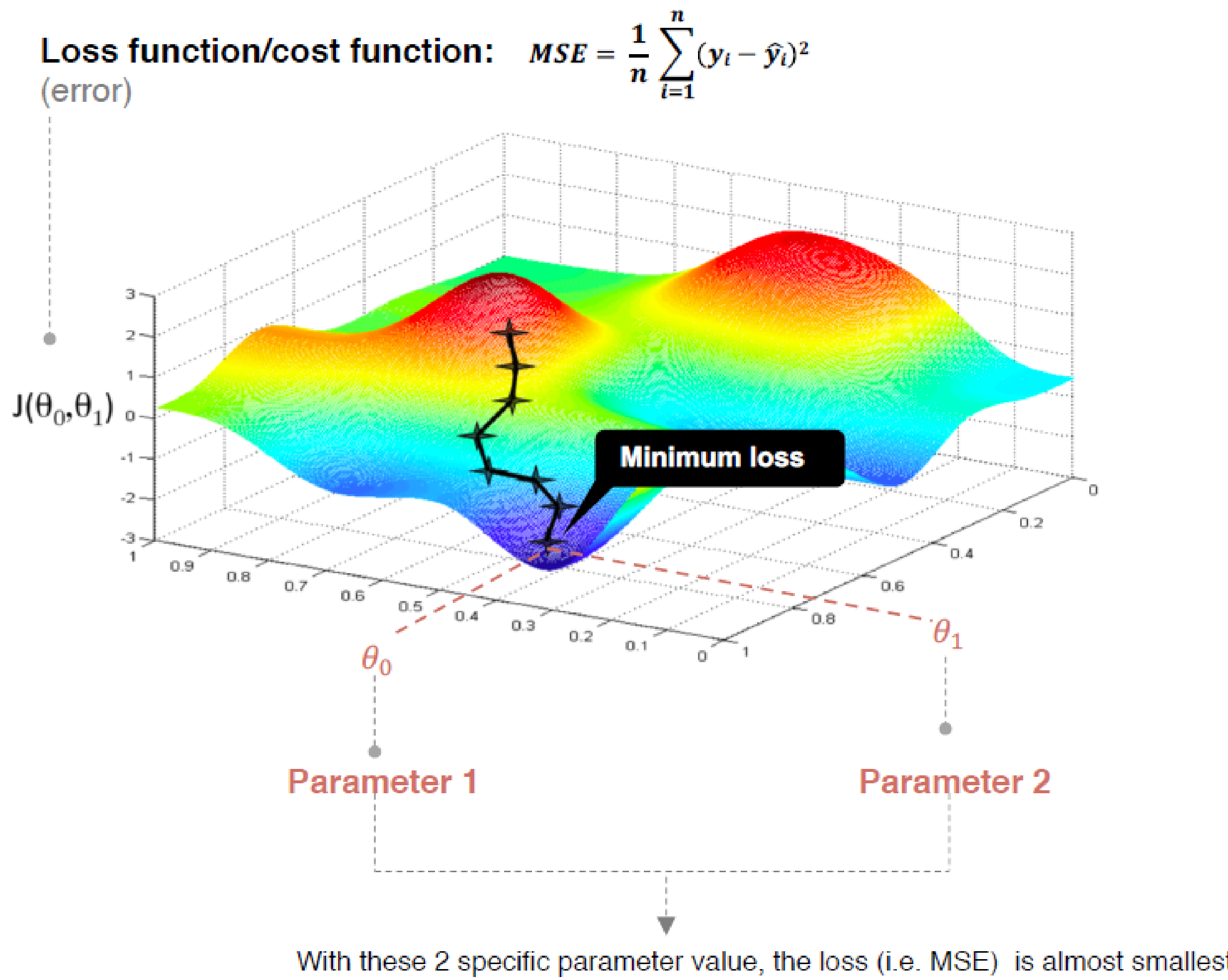
Introduction to Linear Regression

- The goal of training a model is to find a set of parameters that have low loss, on average, across all examples.



Introduction to Linear Regression

- Gradient Descent is commonly used to find the good parameters.



Recall: Key Components in Machine Learning

- 1# Data (Experience): What kind of data do we have?
- 2# Model (Hypothesis): What hypothesis do we make about this data?
- 3# Loss Function (Objective): How to evaluate a model?
- 4# Optimization Algorithm (Improvement): How to find the optimal model?

Linear Regression

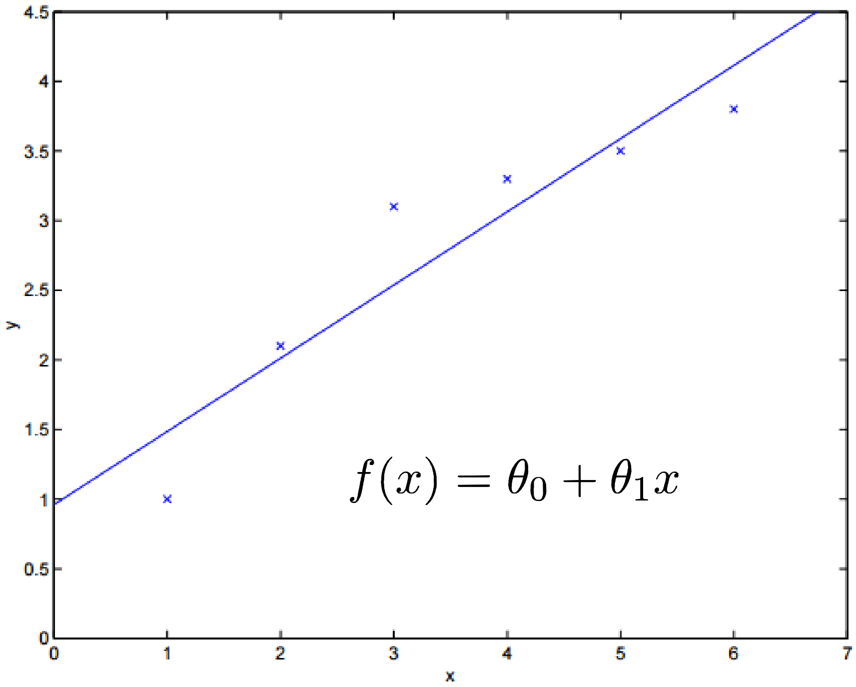
- Linear regression model

$$y = f_\theta(x) = \theta_0 + \sum_{j=1}^{d} \theta_j x_j = \theta^\top x$$

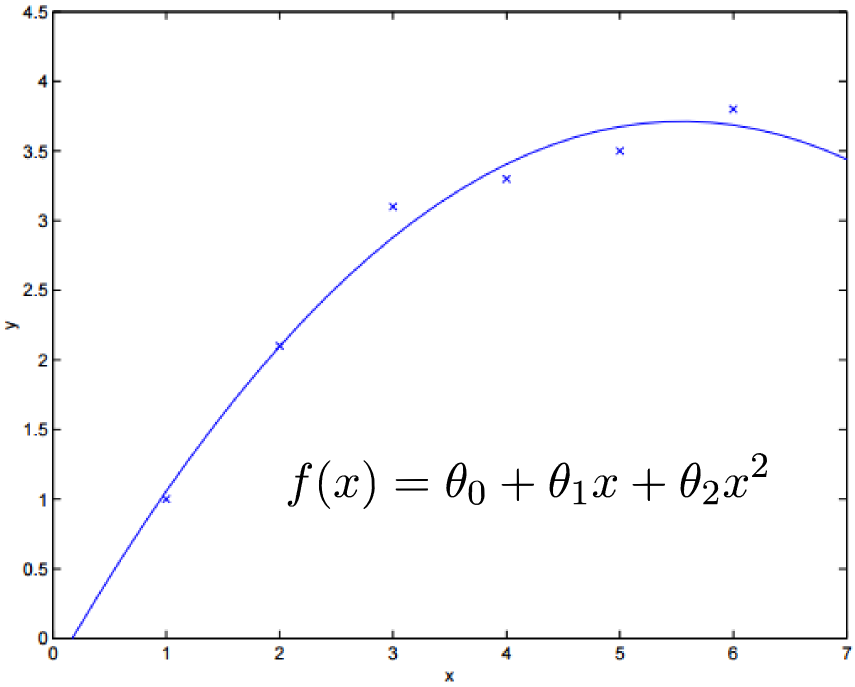
$$x = (1, x_1, x_2, \ldots, x_d)$$

Linear Regression

- One-dimensional linear & quadratic regression



Linear Regression

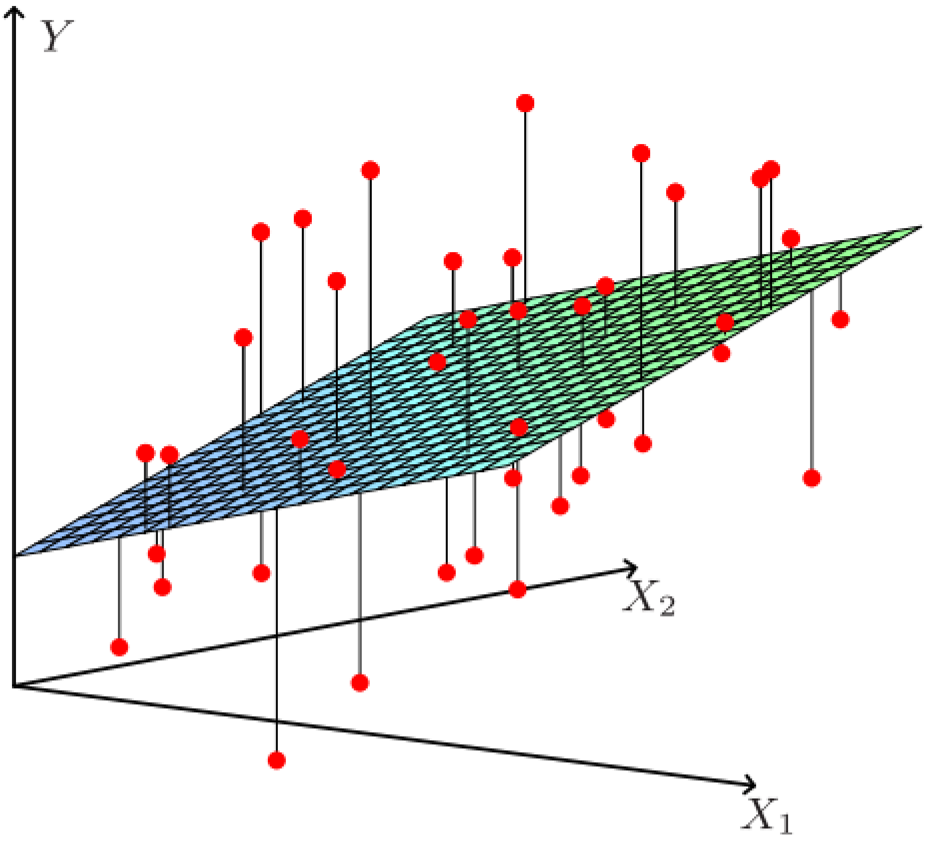


Quadratic Regression
(A kind of generalized linear model)

Linear Regression

- Two-dimensional linear regression

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$



Learning Objective

- Make the prediction close to the corresponding label

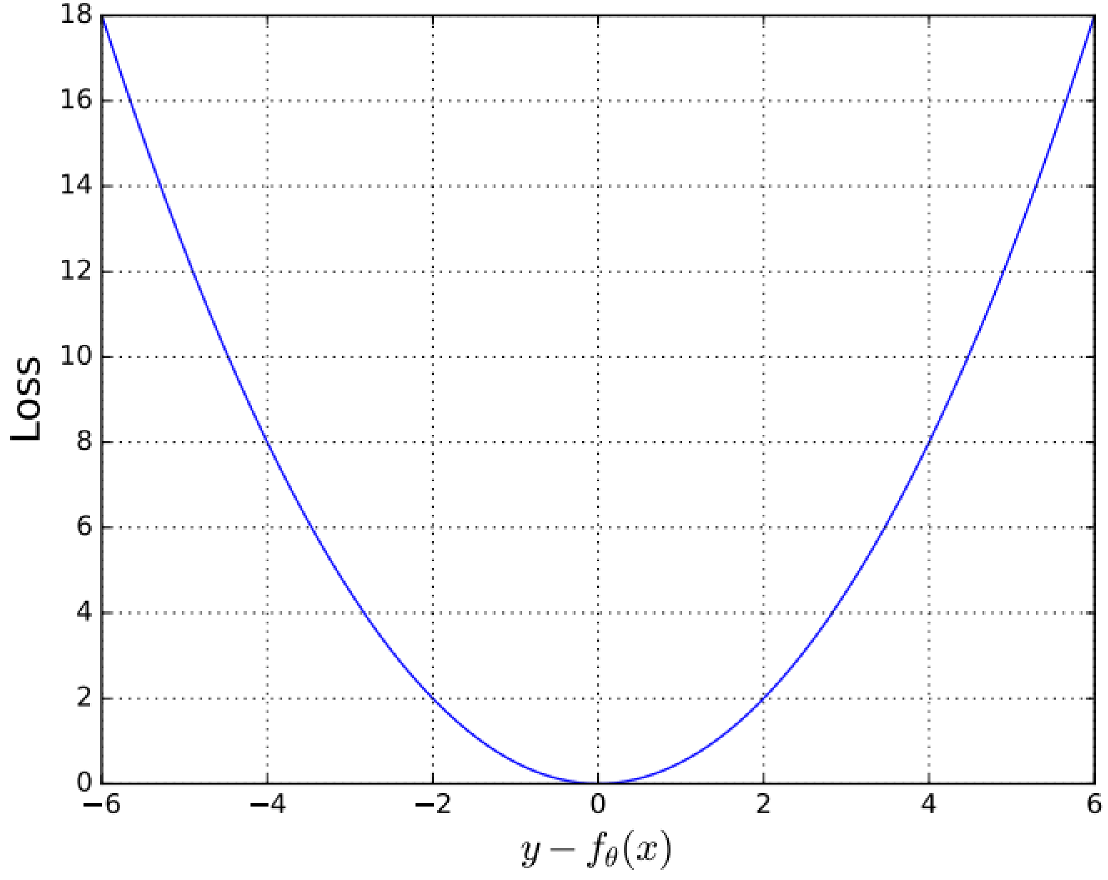
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f_\theta(x_i))$$

- The definition of loss function depends on the data and task
- Most popular loss function: squared loss

$$J_\theta = \frac{1}{2N} \sum_{i=1}^{N} (y_i - f_\theta(x_i))^2 \qquad \min_{\theta} J_\theta$$

Squared Loss

$$\mathcal{L}(y_i, f_\theta(x_i)) = \frac{1}{2}(y_i - f_\theta(x_i))^2$$

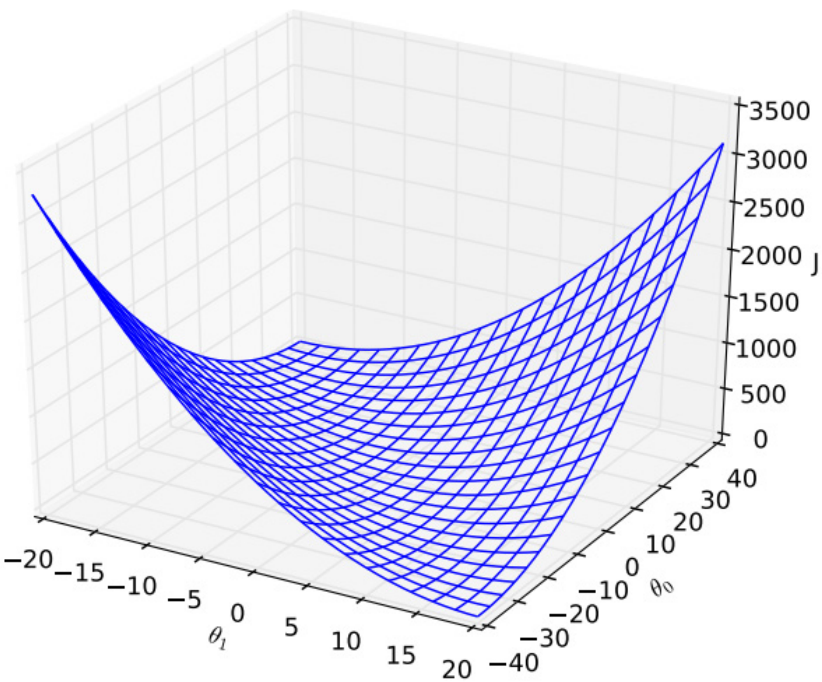


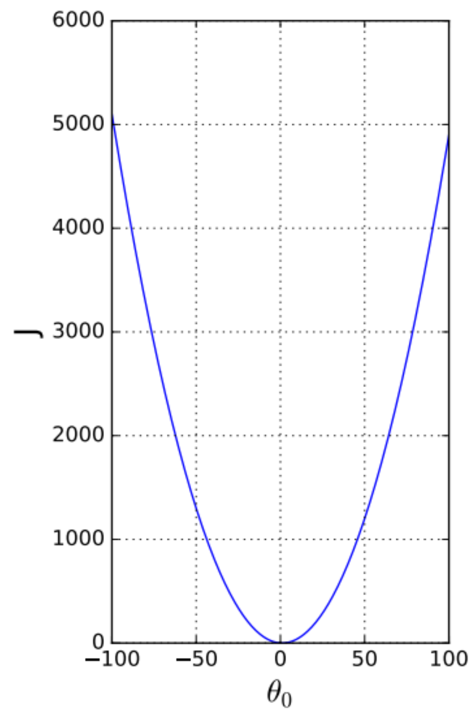
- Penalty much more on larger distances
- Accept small distance (error)
  - Observation noise etc.
  - Generalization

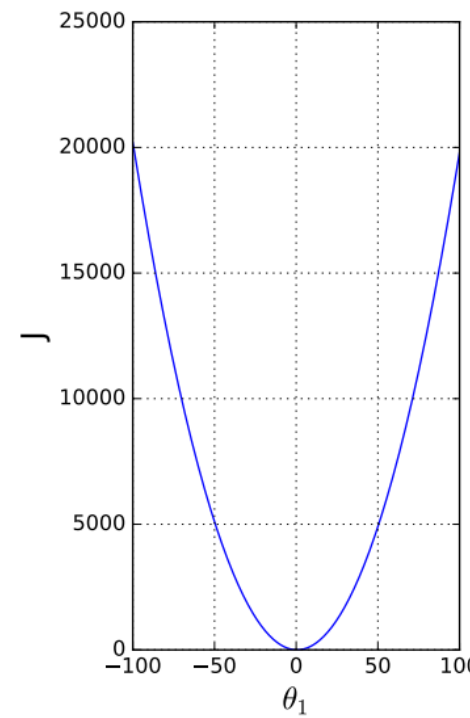
Minimize the Objective Function

- Let *N*=1 for a simple case, for (*x*,*y*)=(2,1)

$$J(\theta) = \frac{1}{2}(y - \theta_0 - \theta_1 x)^2 = \frac{1}{2}(1 - \theta_0 - 2\theta_1)^2$$







Gradient Descent

- https://en.wikipedia.org/wiki/Gradient

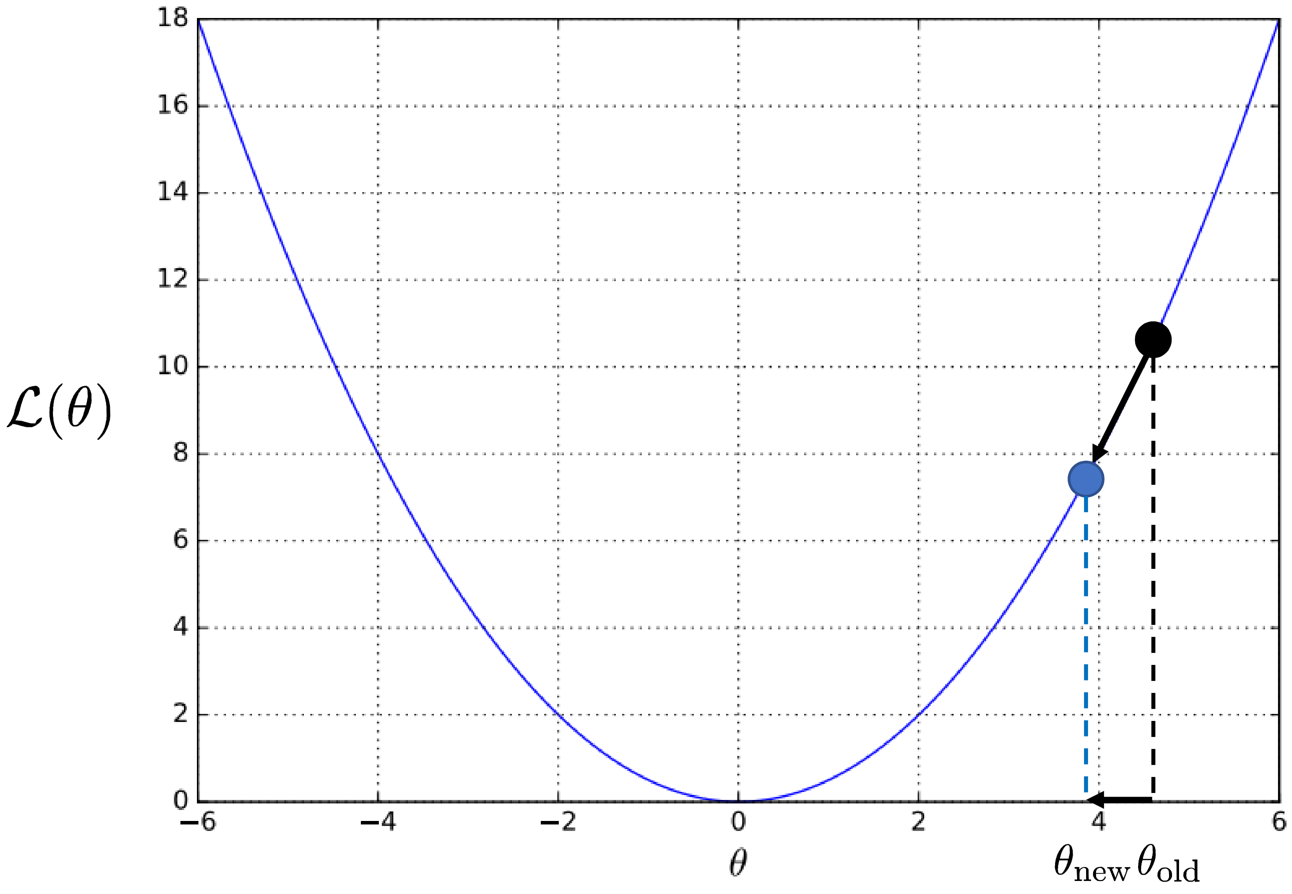
In vector calculus, the **gradient** of a scalar-valued differentiable function f of several variables is the vector field (or vector-valued function) ∇f whose value at a point p is the vector[a] whose components are the partial derivatives of f at p.[1][2][3][4][5][6][7][8][9][*excessive citations*] That is, for $f\colon \mathbb{R}^n \to \mathbb{R}$, its gradient $\nabla f\colon \mathbb{R}^n \to \mathbb{R}^n$ is defined at the point $p = (x_1, \ldots, x_n)$ in n-dimensional space as the vector:[b]

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_n}(p) \end{bmatrix}.$$

The gradient vector can be interpreted as the "direction and rate of fastest increase". If the gradient of a function is non-zero at a point p, the direction of the gradient is the direction in which the function increases most quickly from p, and the magnitude of the gradient is the rate of increase in that direction, the greatest absolute directional derivative.

- 简而言之，关于梯度的两个重要结论
  - 梯度是向量，由目标函数对变量的每一维求偏导后组成
  - 最小化目标函数更新参数时沿着梯度的反方向，反之，最大化目标函数更新参数时沿着梯度的正方向 (why? 不计分课后作业)

Gradient Descent



$$\theta_{\text{new}} \leftarrow \theta_{\text{old}} - \eta \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

Batch Gradient Descent

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - f_\theta(x_i))^2 \qquad \min_\theta J(\theta)$$

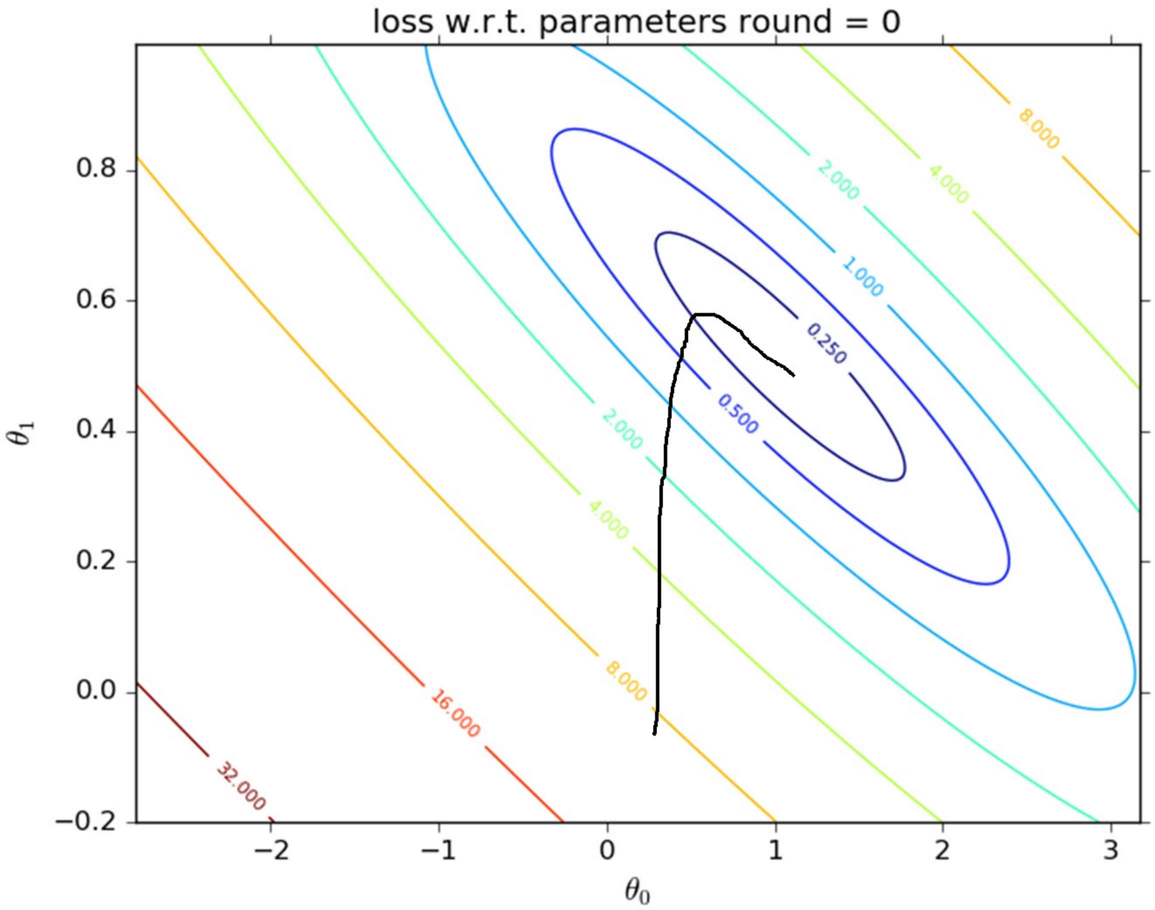
- Update $\theta_{\text{new}} \leftarrow \theta_{\text{old}} - \eta \frac{\partial J(\theta)}{\partial \theta}$ for the whole batch

$$\frac{\partial J(\theta)}{\partial \theta} = -\frac{1}{N}\sum_{i=1}^{N}(y_i - f_\theta(x_i))\frac{\partial f_\theta(x_i)}{\partial \theta}$$

$$= -\frac{1}{N}\sum_{i=1}^{N}(y_i - f_\theta(x_i))x_i$$

$$\theta_{\text{new}} = \theta_{\text{old}} + \eta\frac{1}{N}\sum_{i=1}^{N}(y_i - f_\theta(x_i))x_i$$

Learning Linear Model - BGD



Stochastic Gradient Descent

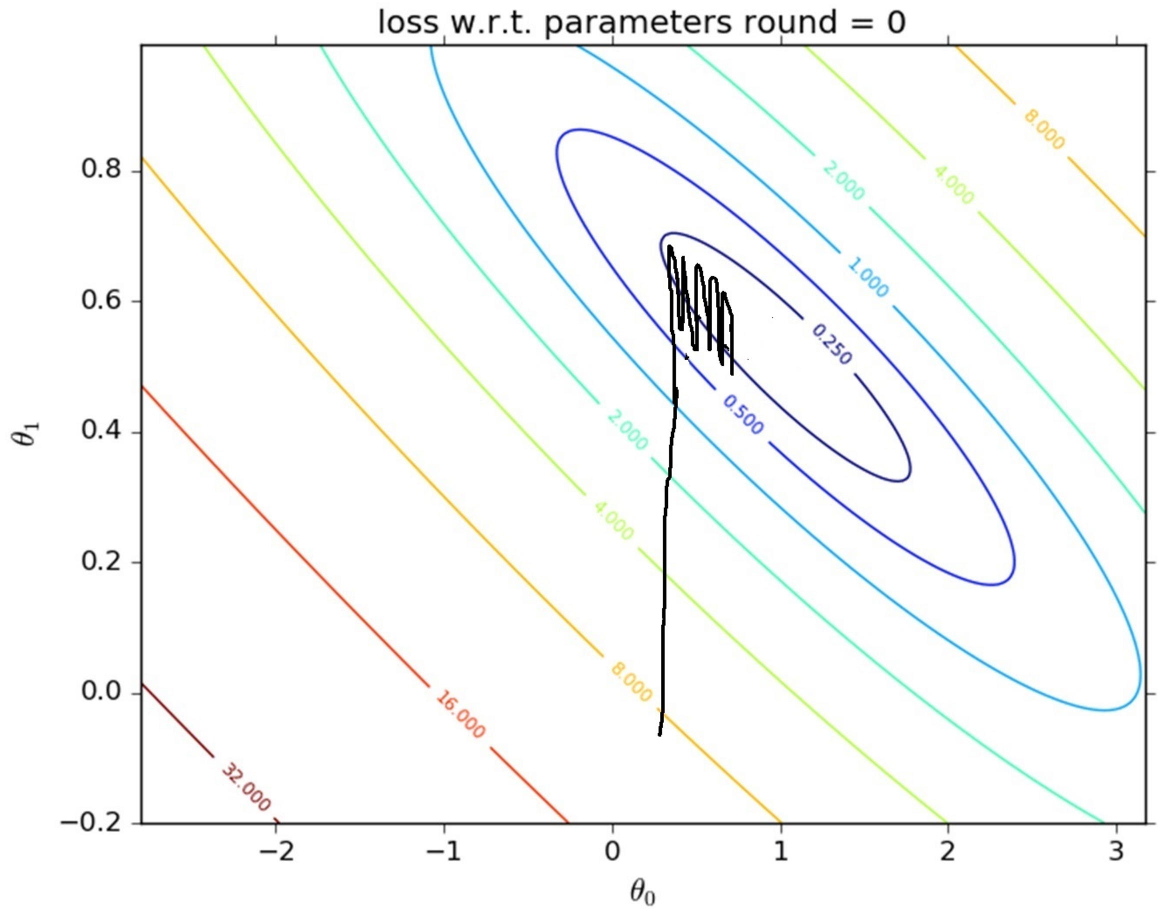
$$J^{(i)}(\theta) = \frac{1}{2}(y_i - f_\theta(x_i))^2 \qquad \min_\theta \frac{1}{N}\sum_i J^{(i)}(\theta)$$

- Update $\theta_{\text{new}} = \theta_{\text{old}} - \eta \frac{\partial J^{(i)}(\theta)}{\partial \theta}$ for every single instance

$$\begin{aligned}\frac{\partial J^{(i)}(\theta)}{\partial \theta} &= -(y_i - f_\theta(x_i))\frac{\partial f_\theta(x_i)}{\partial \theta} \\ &= -(y_i - f_\theta(x_i))x_i \\ \theta_{\text{new}} &= \theta_{\text{old}} + \eta(y_i - f_\theta(x_i))x_i\end{aligned}$$

- Compare with BGD
  - Faster learning
  - Uncertainty or fluctuation in learning

Learning Linear Model - SGD



Mini-Batch Gradient Descent

- A combination of batch GD and stochastic GD

- Split the whole dataset into *K* mini-batches

$$\{1, 2, 3, \ldots, K\}$$

- For each mini-batch *k*, perform one-step BGD towards minimizing

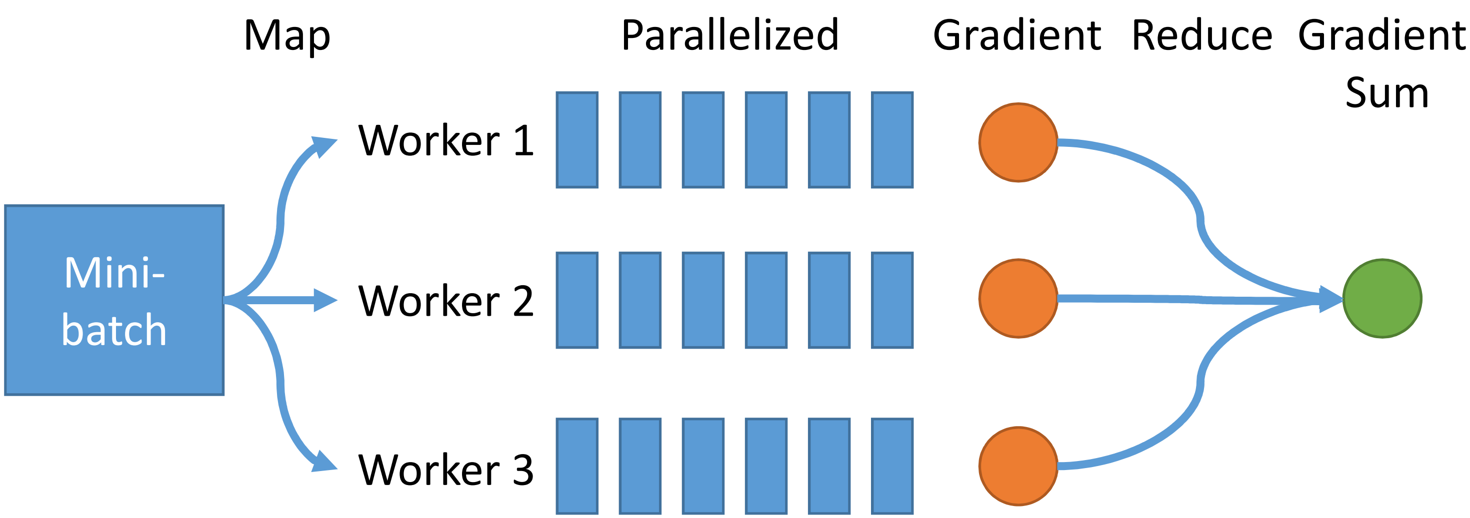
$$J^{(k)}(\theta) = \frac{1}{2N_k} \sum_{i=1}^{N_k} (y_i - f_\theta(x_i))^2$$

- Update $\theta_{\text{new}} = \theta_{\text{old}} - \eta \frac{\partial J^{(k)}(\theta)}{\partial \theta}$ for each mini-batch

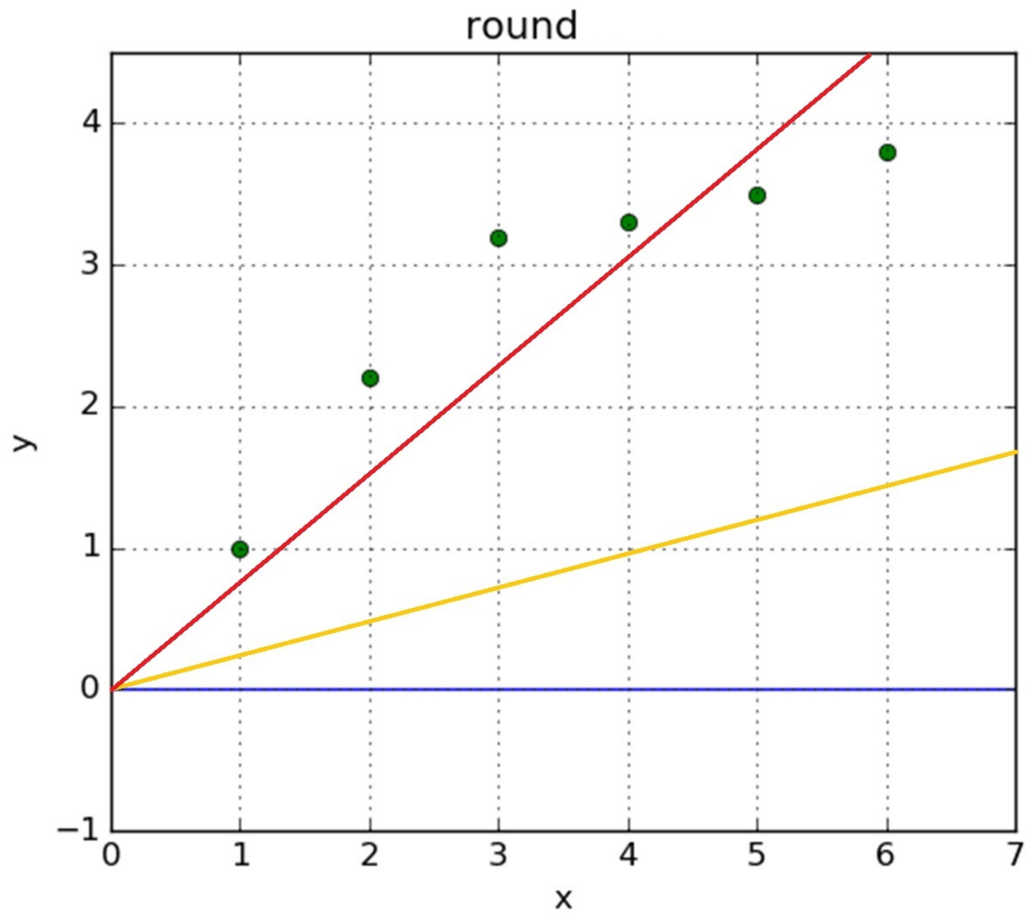
Mini-Batch Gradient Descent

- Good learning stability (BGD)
- Good convergence rate (SGD)

- Easy to be parallelized
  - Parallelization within a mini-batch



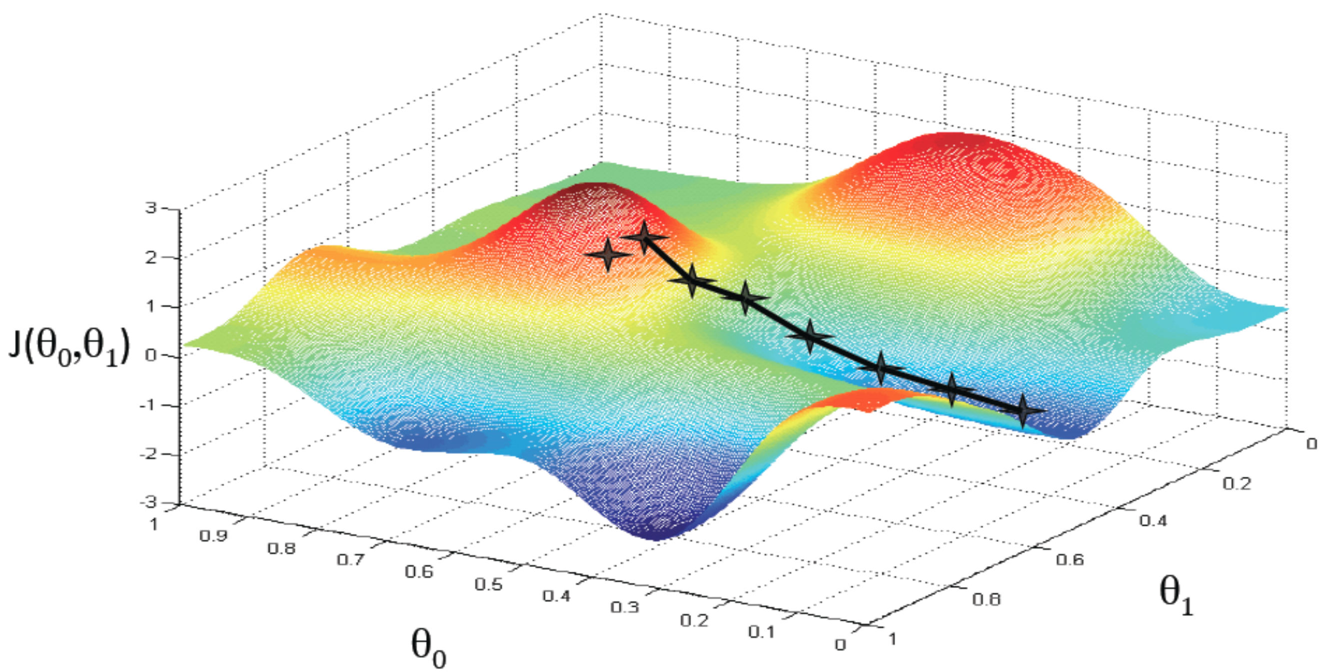
Learning Linear Model - Curve



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

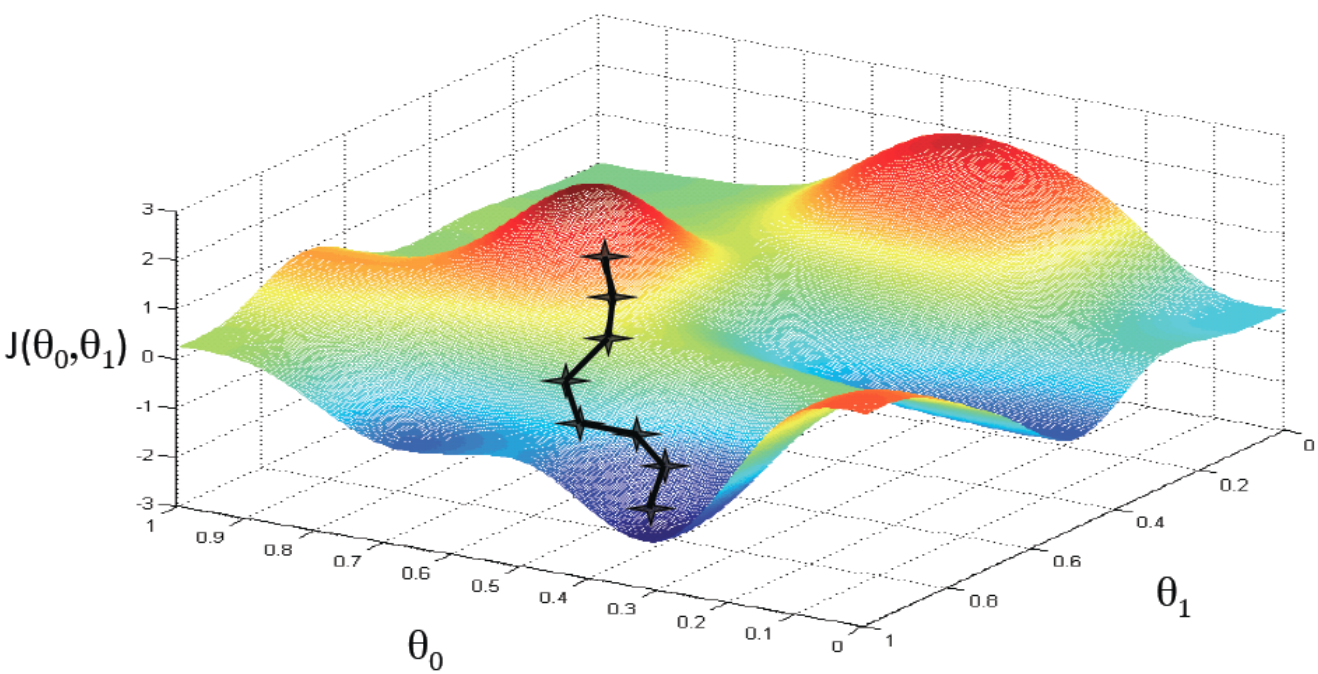
Basic Search Procedure

- Choose an initial value for θ
- Update θ iteratively with the data
- Until we research a minimum



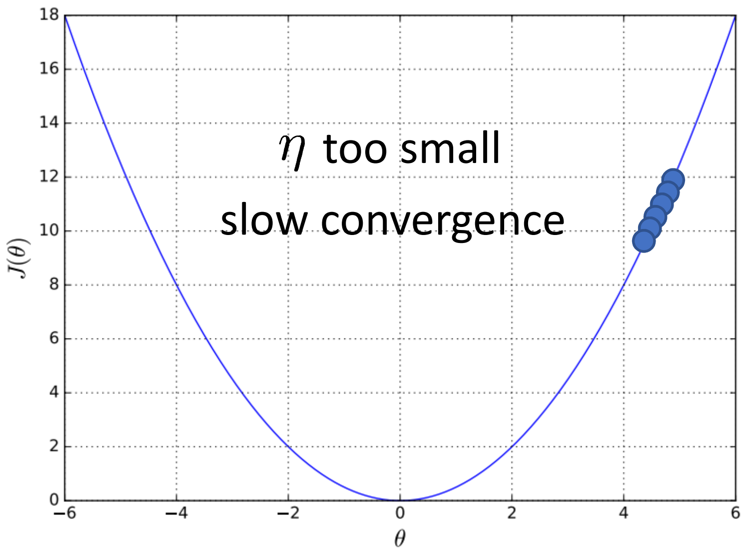
Basic Search Procedure

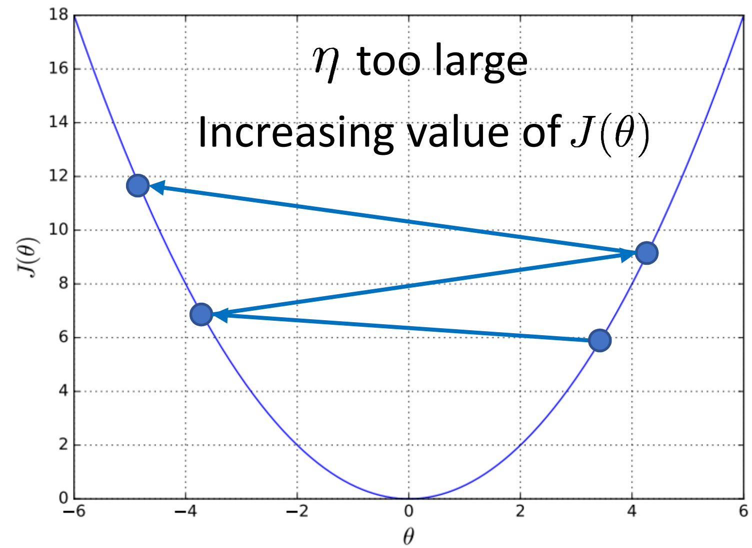
- Choose a new initial value for θ
- Update θ iteratively with the data
- Until we research a minimum



Choosing Learning Rate

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \frac{\partial J(\theta)}{\partial \theta}$$





- The initial point may be too far away from the optimal solution, which takes much time to converge

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

- To see if gradient descent is working, print out $J(\theta)$ for each or every several iterations. If $J(\theta)$ does not drop properly, adjust η

Slide credit Eric Eaton

Advanced Content

课程难度:

掌握程度:

Algebra Perspective

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \vdots \\ \boldsymbol{x}^{(n)} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_d^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(n)} & x_2^{(n)} & x_3^{(n)} & \dots & x_d^{(n)} \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- Prediction $\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{x}^{(1)}\boldsymbol{\theta} \\ \boldsymbol{x}^{(2)}\boldsymbol{\theta} \\ \vdots \\ \boldsymbol{x}^{(n)}\boldsymbol{\theta} \end{bmatrix}$

- Objective $J(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{y} - \hat{\boldsymbol{y}})^\top(\boldsymbol{y} - \hat{\boldsymbol{y}}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})$

Matrix Form

- Objective

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) \qquad \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

- Gradient

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})$$

- Solution

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{0} \;\Rightarrow\; \boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) = \boldsymbol{0}$$

$$\Rightarrow\; \boldsymbol{X}^\top\boldsymbol{y} = \boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\theta}$$

$$\Rightarrow\; \hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

http://dsp.ucsd.edu/~kreutz/PEI-05%20Support%20Files/ECE275A_Viewgraphs_5.pdf

Matrix Form

$$\frac{\partial X^T B X}{\partial X} = (B + B^T)X$$

$$\frac{\partial \theta^T x}{\partial x} = \theta \quad \frac{\partial A\theta}{\partial \theta} = A^T$$

$$\frac{\partial X^T X}{\partial X} = 2X$$

- Objective

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) \qquad \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

- Gradient

$$\begin{aligned} J(\theta) &= \tfrac{1}{2}(y - X\theta)^\top(y - X\theta) \\ &= \tfrac{1}{2}\left(y^T y - (X\theta)^T y - y^T X\theta + (X\theta)^T X\theta\right) \\ &= \tfrac{1}{2}y^T y - y^T X\theta + \tfrac{1}{2}\theta^T X^T X\theta \end{aligned}$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) \qquad \frac{\partial J(\theta)}{\partial \theta} = -X^T y + X^T X\theta = 0$$

- Solution

$$\begin{aligned} \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0} \;&\Rightarrow\; \boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) = \mathbf{0} \\ &\Rightarrow\; \boldsymbol{X}^\top\boldsymbol{y} = \boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\theta} \\ &\Rightarrow\; \hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y} \end{aligned}$$

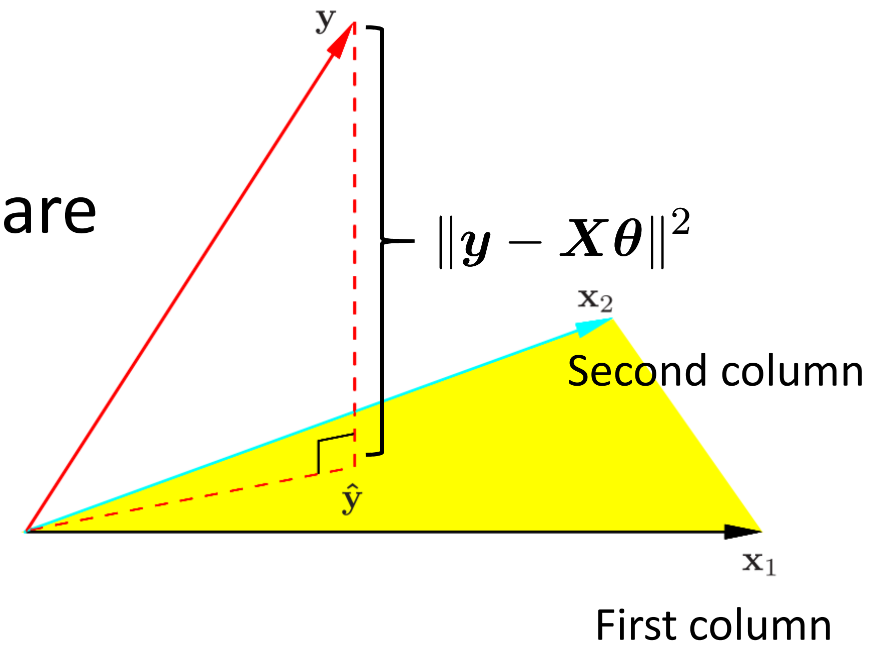
http://dsp.ucsd.edu/~kreutz/PEI-05%20Support%20Files/ECE275A_Viewgraphs_5.pdf

Matrix Form

- Then the predicted values are

$$\hat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$$
$$= \boldsymbol{H}\boldsymbol{y}$$

H: hat matrix



- Geometrical Explanation
  - The column vectors $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d]$ form a subspace of $\mathbb{R}^n$
  - *H* is a least square projection

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \cdots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \cdots & x_d^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(n)} & x_2^{(n)} & x_3^{(n)} & \cdots & x_d^{(n)} \end{bmatrix} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d] \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

More details refer to Sec 3.2. Hastie et al. The elements of statistical learning.

$\boldsymbol{X}^\top \boldsymbol{X}$ Might be Singular

- When some column vectors are not independent
  - For example, $\mathbf{x}_2 = 3\mathbf{x}_1$

then $\boldsymbol{X}^\top \boldsymbol{X}$ is singular, thus $\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ cannot be directly calculated.

- Solution: regularization

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$$

Matrix Form with Regularization

- Objective

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 \qquad \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

- Gradient

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}$$

- Solution

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0} \;\rightarrow\; -\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) + \lambda\boldsymbol{\theta} = \mathbf{0}$$

$$\rightarrow\; \boldsymbol{X}^\top\boldsymbol{y} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})\boldsymbol{\theta}$$

$$\rightarrow\; \hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

Summary

- Hypothesis of linear regression
- Loss function (MSE)
- Optimization (BGD, SGD)
- Algebra Perspective: the least square projection
- Extension: Generalized Linear Models