Linear Regression

CS115 - Math for Computer Science

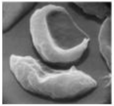
TS. Lương Ngọc Hoàng
TS. Dương Việt Hằng

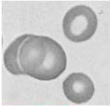
September 9, 2023

From Discrete to Continuous Labels

Classification

Sports
Science
News





Anemic cell
Healthy cell

X = Document | Y = Topic | X = Cell Image | Y = Diagnosis

Regression

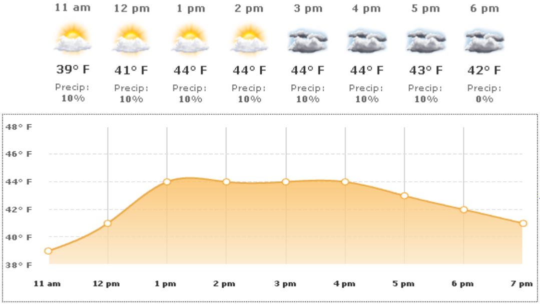
Stock Market Prediction

Y = ?

X = Feb01

Regression Tasks

Weather Prediction



Y = Temp

X = 7 pm

Estimating Contamination

X = new location

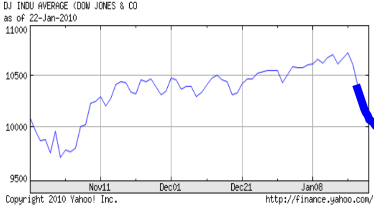
Y = sensor reading

Supervised Learning

Goal: Construct a **predictor** $f: X \rightarrow Y$ to minimize a risk (error measure) err(f).

Typical Error Measures

Sports
Science
News



Y = ?
X = Feb01

Classification:

$$\mathrm{err}(f) = P(f(X) \neq Y)$$

Probability of Error

Regression:

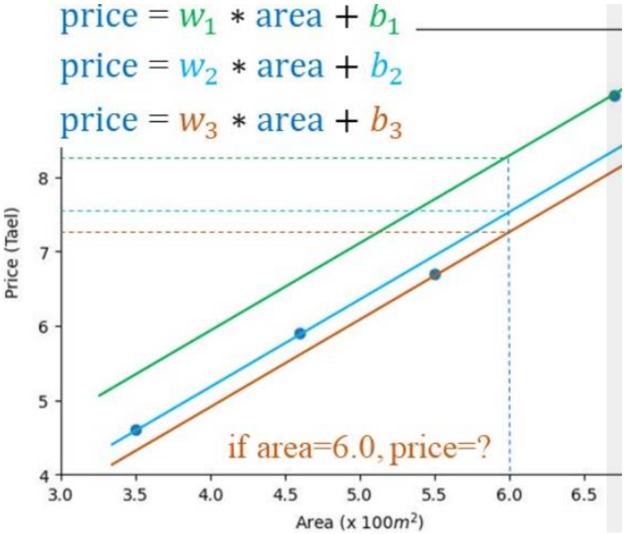
$$\mathrm{err}(f) = E[(f(X) - Y)^2]$$

Mean Squared Error

Linear Regression

House Price Prediction

Feature	Label
area	**price**
6.7	8.1
4.6	5.6
3.5	4.3
5.5	6.7

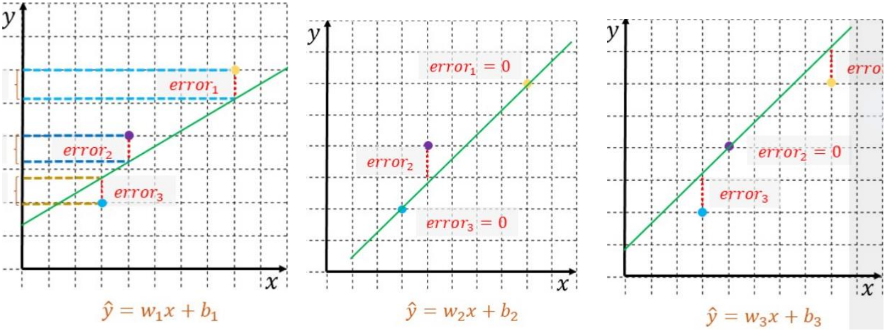


Linear Regression

House Price Prediction

Training data

$error_i = distance(\hat{y}_i, y_i)$



Find w and b whose models has the smallest error

$$error = \sum_i error_i$$

Linear Regression

House Price Prediction

Training data

$error_i = distance(\hat{y}_i, y_i)$

$$predicted\_price = w * area + b$$

$$error = (predicted\_price - real\_price)^2$$

$$\hat{y}_i = wx_i + b$$

$$L(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2$$

Find w and b whose models has the smallest error

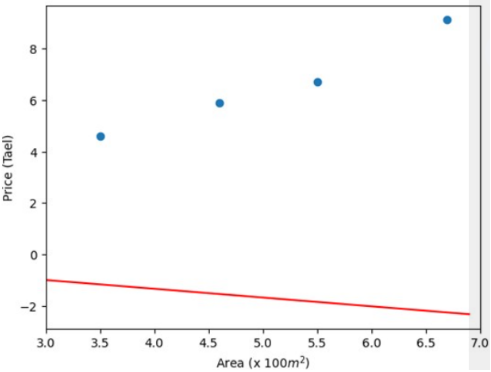
$$error = \sum_i error_i$$

House Price Prediction

area	price	predicted	error
6.7	9.1	-2.238	128.55
4.6	5.9	-1.524	55.11
3.5	4.6	-1.15	33.06
5.5	6.7	-1.83	72.76

$$w = -0.34$$

$$b = 0.04$$

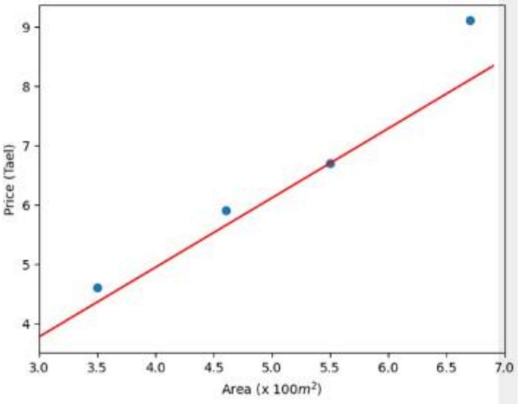


House Price Prediction

$w = 1.17$

$b = 0.26$

area	price	predicted	error
6.7	9.1	8.099	1.002
4.6	5.9	5.642	0.066
3.5	4.6	4.355	0.06
5.5	6.7	6.695	0.00002



How to change w and b so that $L(\hat{y}_i, y_i)$ reduces

House Price Prediction

Understand Loss Function

Linear equation

$$\hat{y} = wx + b$$

where $\hat{y}$ is a predicted value,

w and b are parameters

and x is input feature

Error (loss) computation

Idea: compare predicted values $\hat{y}$ and label values y

Squared loss

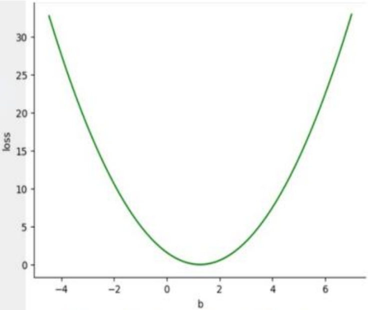
$$L(\hat{y}, y) = (\hat{y} - y)^2$$

House Price Prediction

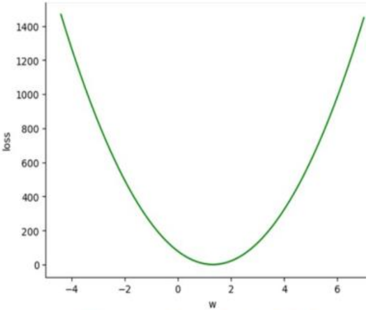
Understand Loss Function

$$\hat{y}_i = wx_i + b$$

$$L(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2$$

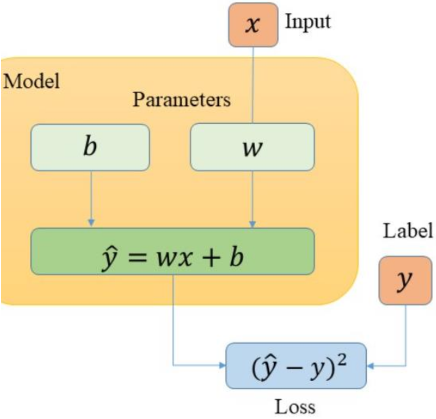


Different b values with a fixed w value



Different w values with a fixed b value

Gradient Descent-Based Optimization



1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

Definition: Function Gradient

- Gradient of a function indicates how strong the function increases.
  - For 1-dimension function: $f(x) = x^2$

$$Grad(x) = \frac{\partial f(x)}{\partial(x)} = 2x$$

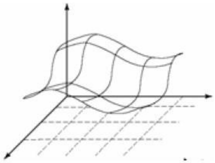
  - Grad(2)=4 indicates the the increasing direction of the function is to the right.
  - Grad(-1)=-2 indicates the increasing direction of the function is to the left.

Definition: Function Gradient

- Let $f(x,y)$ be a 2D function
- **Gradient:** Vector whose direction is in direction of maximum rate of change of f and whose magnitude is maximum rate of change of f

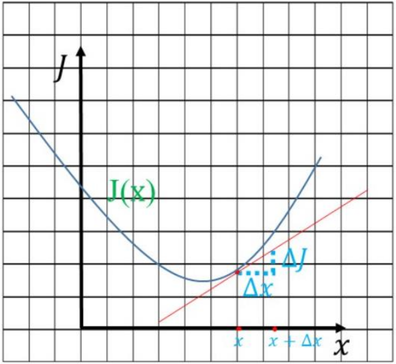
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^T$$

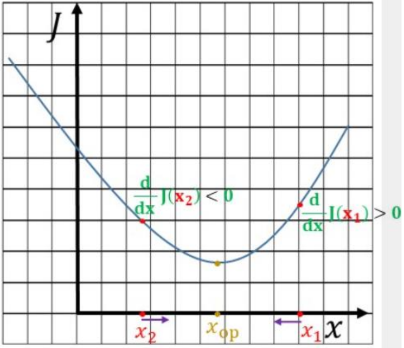
- magnitude $= [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{1/2}$
- direction $= \tan^{-1}(\frac{\partial f / \partial y}{\partial f / \partial x})$



Optimization

Gradient Descent





$$\frac{d}{dx}J(x) = \lim_{\Delta x \to 0} \frac{J(x + \Delta x) - J(x)}{\Delta x}$$

$$\mathbf{x}_{new} = \mathbf{x}_{old} - \eta \frac{\mathbf{d}}{\mathbf{dx}} \mathbf{J}(\mathbf{x}_{old})$$

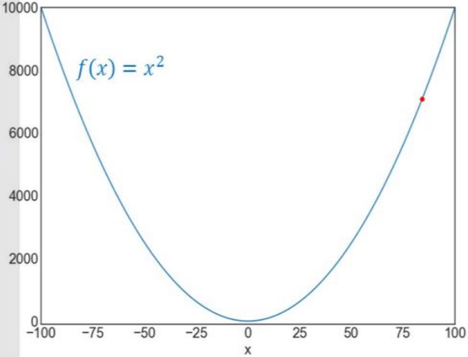
Derivate at x_{old}

learning rate

Optimization

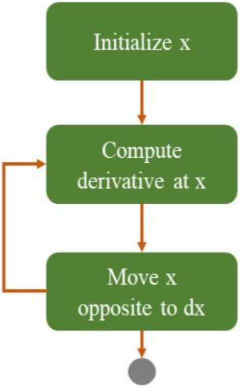
Gradient Descent

❖ **Square function**



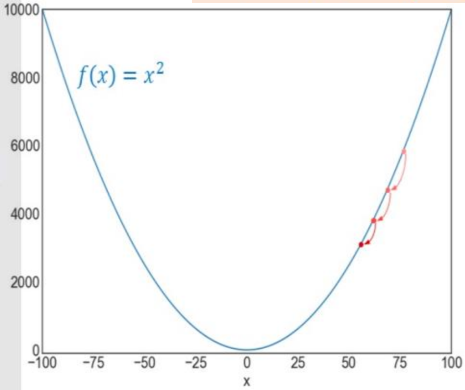
$$-100 \leq x \leq 100$$
$$x \in \mathbb{N}$$

$$x_t = x_{t-1} - \eta f'(x_{t-1})$$



$$x_t = x_{t-1} - \eta f'(x_{t-1})$$

$x_0 = 70.0$ $\eta = 0.1$



$$-100 \leq x \leq 100$$
$$x \in \mathbb{N}$$

$$f'(x_0) = 140.0$$
$$x_1 = x_0 - \eta f'(x_0) = 56.0$$

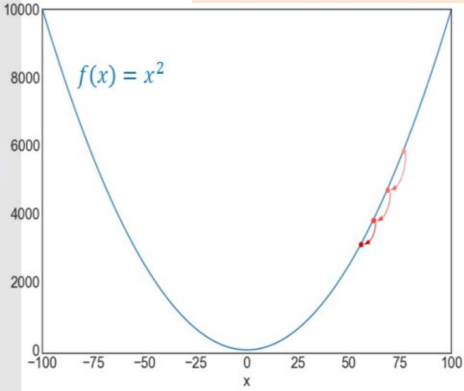
$$f'(x_1) = 112.0$$
$$x_2 = x_1 - \eta f'(x_1) = 44.8$$

$$f'(x_2) = 89.6$$
$$x_3 = x_2 - \eta f'(x_2) = 35.84$$

$$f'(x_3) = 71.68$$
$$x_4 = x_3 - \eta f'(x_3) = 28.672$$

$$x_t = x_{t-1} - \eta f'(x_{t-1})$$

$x_0 = 70.0$ $\eta = 0.1$



$$-100 \le x \le 100$$
$$x \in \mathbb{N}$$

$x_{10} = 6.012$ $\eta = 0.1$

$$f'(x_{10}) = 12.02$$
$$x_{11} = x_{10} - \eta f'(x_{10}) = 4.81$$

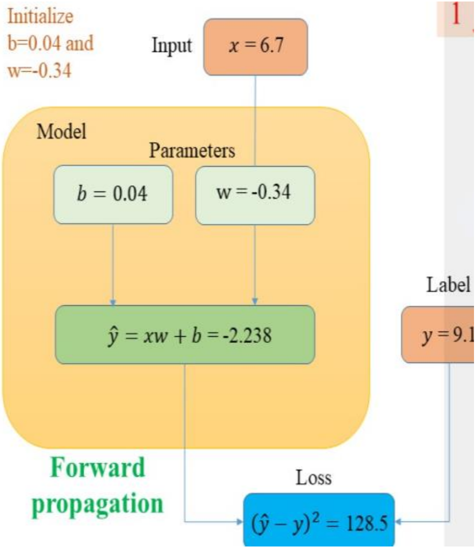
$$f'(x_{11}) = 9.62$$
$$x_{12} = x_{11} - \eta f'(x_{11}) = 3.84$$

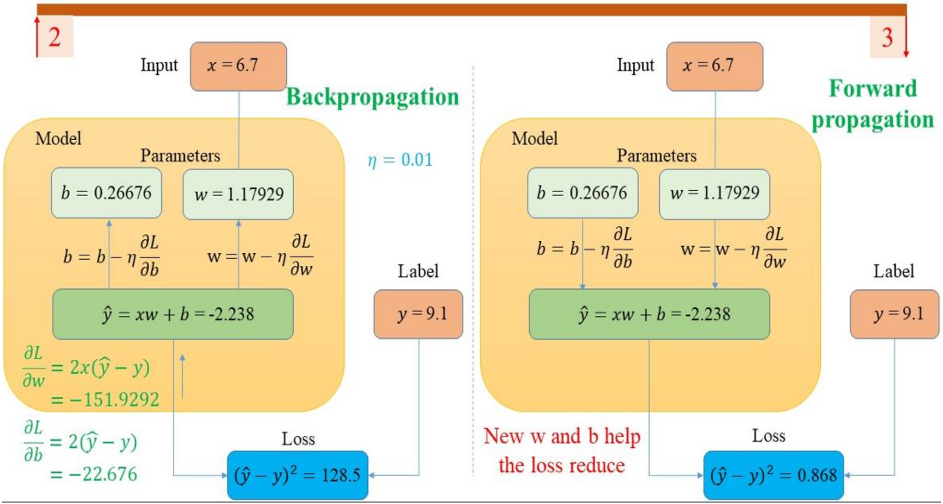
$$f'(x_{12}) = 7.69$$
$$x_{13} = x_{12} - \eta f'(x_{12}) = 3.078$$

$$f'(x_{13}) = 6.15$$
$$x_{14} = x_{13} - \eta f'(x_{13}) = 2.46$$

Given sample data

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7




2
Input
$x = 6.7$
Backpropagation
Model
Parameters
$b = 0.26676$
$w = 1.17929$
$b = b - \eta \frac{\partial L}{\partial b}$
$w = w - \eta \frac{\partial L}{\partial w}$
$\hat{y} = xw + b = -2.238$
$\eta = 0.01$
Label
$y = 9.1$
$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) = -151.9292$
$\frac{\partial L}{\partial b} = 2(\hat{y} - y) = -22.676$
Loss
$(\hat{y} - y)^2 = 128.5$
3
Input
$x = 6.7$
Forward propagation
Model
Parameters
$b = 0.26676$
$w = 1.17929$
$b = b - \eta \frac{\partial L}{\partial b}$
$w = w - \eta \frac{\partial L}{\partial w}$
$\hat{y} = xw + b = -2.238$
Label
$y = 9.1$
New w and b help the loss reduce
Loss
$(\hat{y} - y)^2 = 0.868$

Linear Regression

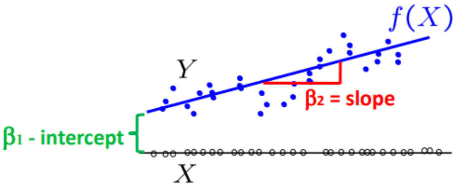
$$\hat{f}_n^L = \arg\min_{f \in F_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

Least Squares Estimator

F_L - Class of Linear functions

Uni-variate case:

$$f(X) = \beta_1 + \beta_2 X$$



Multi-variate case:

$$f(X) = f\left(X^{(1)}, \dots, X^{(p)}\right) = \beta_1 \cancelto{1}{X^{(1)}} + \beta_2 X^{(2)} + \cdots + \beta_p X^{(p)}$$

$$= X\beta \quad \text{where} \quad X = \left[X^{(1)} \dots X^{(p)}\right], \qquad \beta = \left[\beta_1 \dots \beta_p\right]^T$$

Least Squares Estimator

$$\hat{f}_n^L = \arg\min_{f \in F_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

$$\Downarrow$$

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (X_i \beta - Y_i)^2 \qquad \hat{f}_n^L(X) = X\hat{\beta}$$

$$= \arg\min_{\beta} \frac{1}{n} (\mathbf{A}\beta - \mathbf{Y})^T (\mathbf{A}\beta - \mathbf{Y})$$

$$\mathbf{A} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_1^{(1)} & \dots & X_1^{(p)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} & \dots & X_n^{(p)} \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_n \end{bmatrix}$$

Least Squares Estimator

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) = \arg\min_{\beta} J(\beta)$$

$$J(\beta) = (\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y})$$

$$\left.\frac{\partial J(\beta)}{\partial \beta}\right|_{\hat{\beta}} = 0$$

Normal Equations

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})\hat{\beta} = \mathbf{A}^{\mathrm{T}}\mathbf{Y}$$

p xp p x1 p x1

If $(\mathbf{A}^{\mathrm{T}}\mathbf{A})$ is invertible,

$$\boxed{\hat{\beta} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{Y} \qquad \hat{f}_n^L(X) = X\hat{\beta}}$$

Geometric Interpretation

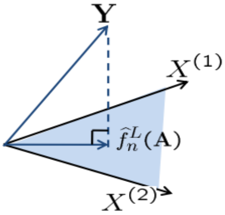
$$\hat{f}_n^L(X) = X\hat{\beta} = X\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{Y}$$

Difference in prediction on training set:

$$\hat{f}_n^L(\mathbf{A}) - \mathbf{Y} =$$

$$\mathbf{A}^T\left(\hat{f}_n^L(\mathbf{A}) - \mathbf{Y}\right) = 0$$

$\hat{f}_n^L(\mathbf{A})$ is the orthogonal projection of $\mathbf{Y}$ onto the linear subspace spanned by the columns of $\mathbf{A}$.



Revisiting Gradient Descent

Even when $(\mathbf{A}^{\mathrm{T}}\mathbf{A})$ is invertible, might be computationally expensive if $\mathbf{A}$ is huge.

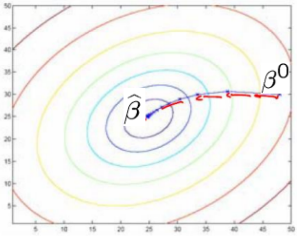
$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n}(\mathbf{A}\beta - \mathbf{Y})^{\mathrm{T}}(\mathbf{A}\beta - \mathbf{Y}) = \arg\min_{\beta} \mathrm{J}(\beta)$$

Gradient Descent since $J(\beta)$ is convex

Initialize: β^0

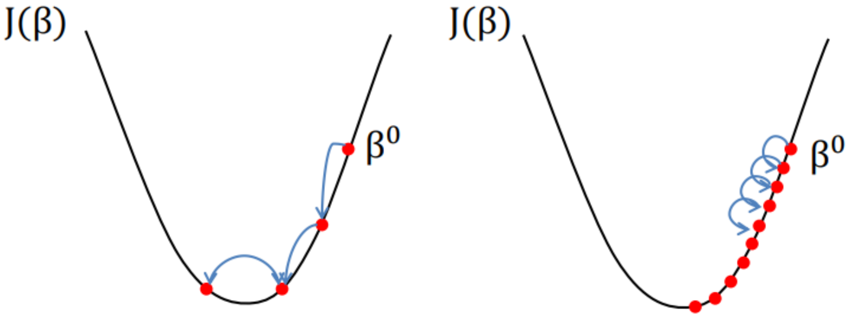
Update: $\beta^{t+1} = \beta^t - \frac{\alpha}{2}\frac{\partial \mathrm{J}(\beta)}{\partial \beta}\Big|_t$

$$= \beta^t - \alpha\mathbf{A}^{\mathrm{T}}\underbrace{(\mathbf{A}\beta^t - \mathbf{Y})}_{0 \text{ if } \beta^t = \hat{\beta}}$$



Stop: when some criterion met, e.g. fixed # iterations, or $\frac{\partial \mathrm{J}(\beta)}{\partial \beta}\Big|_{\beta^t} < \epsilon$

Effect of step-size α



Large $\alpha \Rightarrow$ Fast convergence but larger residual error
Also possible oscillations

Small $\alpha \Rightarrow$ Slow convergence but small residual error

Least Squares and MLE

Intuition: Signal plus (zero-mean) Noise model

$$Y = f^*(X) + \epsilon = X\beta^* + \epsilon \qquad\qquad \epsilon \sim N(0, \sigma^2 \mathbf{I})$$

$$Y \sim N(X\beta^*, \sigma^2 \mathbf{I})$$

$$\hat{\beta}_{MLE} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^{n} \mid \beta, \sigma^2, X)}_{\text{log likelihood}}$$

$$= \arg\min_{\beta} \sum_{i=1}^{n} (X_i\beta - Y_i)^2 = \hat{\beta}$$

Least Square Estimate is same as Maximum Likelihood Estimate under a Gaussian model !

13

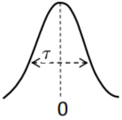
Regularized Least Squares and MAP

$$\hat{\beta}_{MAP} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^{n} \mid \beta\sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

I) Gaussian Prior

$$\beta \sim N(0, \tau^2 \mathbf{I}) \qquad p(\beta) \propto e^{-\beta^T\beta/2\tau^2}$$

$$\hat{\beta}_{MAP} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \underset{\substack{\downarrow \\ \text{constant}(\sigma^2, \tau^2)}}{\lambda} \|\beta\|_2^2$$



Ridge Regression

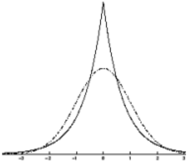
Prior belief that β is Gaussian with zero-mean biases solution to "small" β

Regularized Least Squares and MAP

$$\hat{\beta}_{MAP} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^{n} \mid \beta\sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

II) Laplace Prior

$\beta_i \sim \text{Laplace}(0, t)$ [iid] $\qquad p(\beta_i) \propto e^{-|\beta_i|/t}$



$$\hat{\beta}_{MAP} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \underset{\downarrow \atop \text{constant}(\sigma^2, t)}{\lambda} ||\beta||_1$$

Lasso

Prior belief that β is Laplace with zero-mean biases solution to "small" β