

Day 3: Overfitting and Generalization

Summer STEM: Machine Learning

Department of Electrical Engineering
NYU Tandon School of Engineering
Brooklyn, New York

June 24, 2020

features: inputs to your model

labels: what you try to predict

Supervised learning: • we have labels

unsupervised learning: • no labels

- group data based on features

Regression / Classification

Regression: labels y are continuous real numbers

Classification labels y are discrete

- Ex: Two classes (binary)
 $\{0, 1\}$

Model: a function $f(x)$

in supervised. $\leftarrow \hat{y} = f(x)$

\hat{y} : prediction from the model

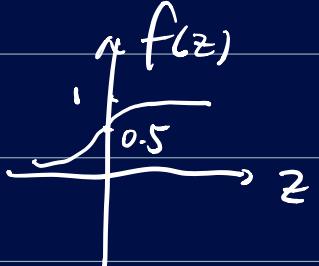
model parameters: coefficients of
the function $f(x)$

Ex: $f(x) = \frac{1}{e^{(w_1x+w_0)} + 1}$ w_0, w_1 are
the model params

w_0 is the coefficient of x^0

$$x^0 = 1$$

Sigmoid: $\frac{1}{e^{-z} + 1} = f(z)$



$$= \frac{e^z}{1 + e^z}$$

$$\hat{f}(x) = \underline{w_0} + \underline{w_1}x$$

$$f(x) = \underline{w_2}x^2 + \underline{w_1}x + \underline{w_0}$$

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

$f(x; w)$ this model is
parameterized by w

Cost functions : how much error

the hypothesis has given the dataset

↓ the model

MSE : mean square error

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

\hat{y} is the prediction from the model $f(x)$

$$\hat{y}_i = f(x_i)$$

$$MAE : \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

Outline

1 Leftovers from Day 2

2 Polynomial Regression

3 Train and Test Error, Overfitting

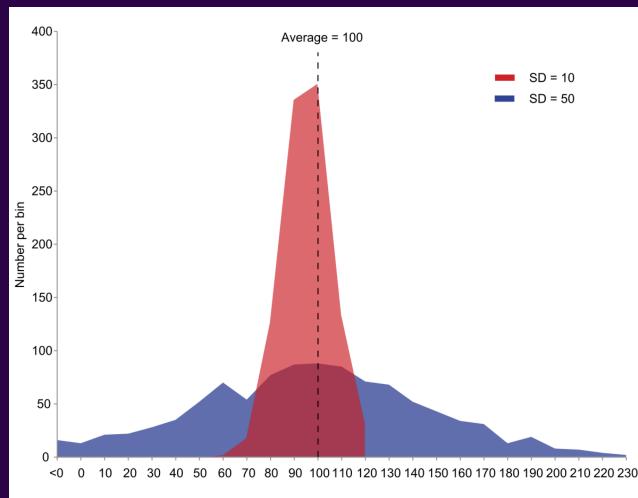
4 Regularization

Basic Concepts

- **Mean** (average value): $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$
- **Variance** describes the spread of the data with respect to the mean.
- **Covariance** describes the relationship between two variables.

Variance

$$\blacksquare \text{ Variance: } \sigma_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

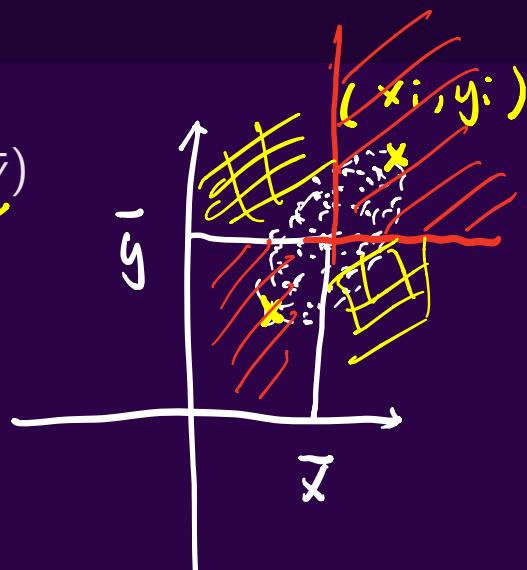
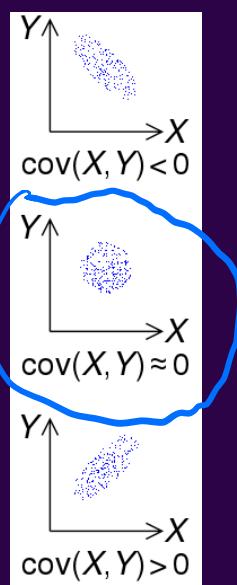
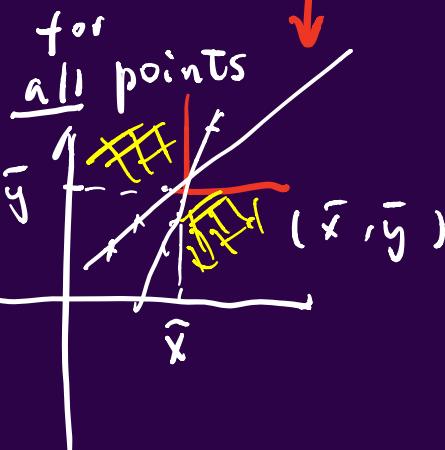


<https://en.wikipedia.org/wiki/Variance>

Covariance

■ Covariance: $\sigma_{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$

if $(x_i - \bar{x})(y_i - \bar{y})$ ←
are positive

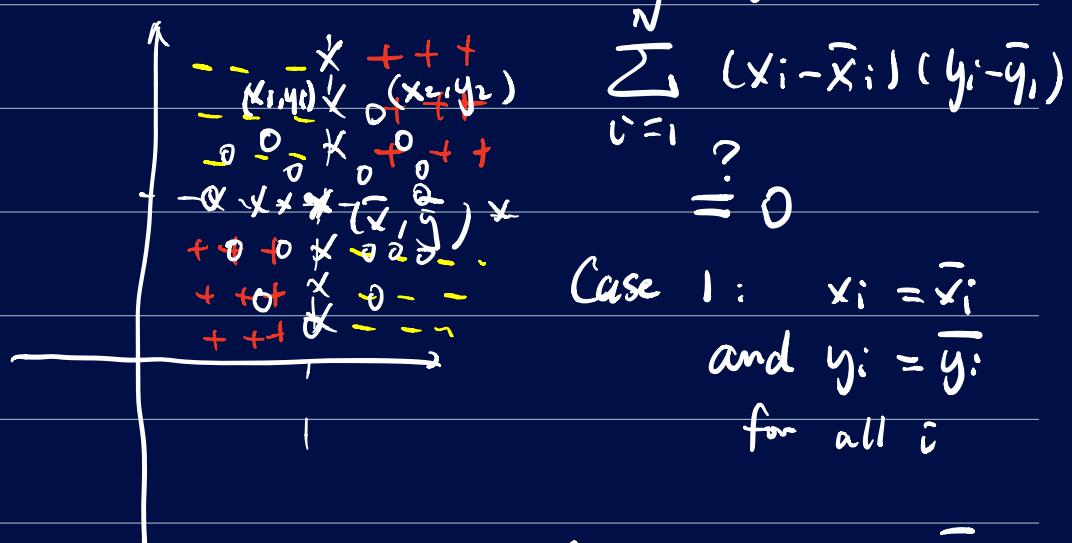


$$\sigma_{xy} > 0 \stackrel{?}{\Rightarrow}$$

for all i $(y_i - \bar{y})(y_i - \bar{y})$
one positive

<https://en.wikipedia.org/wiki/Covariance>

$$(x_1 - \bar{x})(y_1 - \bar{y}) = -(x_2 - \bar{x})(y_2 - \bar{y})$$



Case 1 : $x_i = \bar{x}_i$
and $y_i = \bar{y}_i$
for all i

Case 2 : $x_i = \bar{x}_i$
or $y_i = \bar{y}_i$

Case 3 :

$$\frac{1}{N} \left(\sum_{\text{positive}} (x_i - \bar{x}_i)(y_i - \bar{y}_i) + \sum_{\text{negative}} (x_i - \bar{x}_i)(y_i - \bar{y}_i) \right) = 0$$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data $(x_i, y_i), i = 1, 2, \dots, N$

- Mean:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

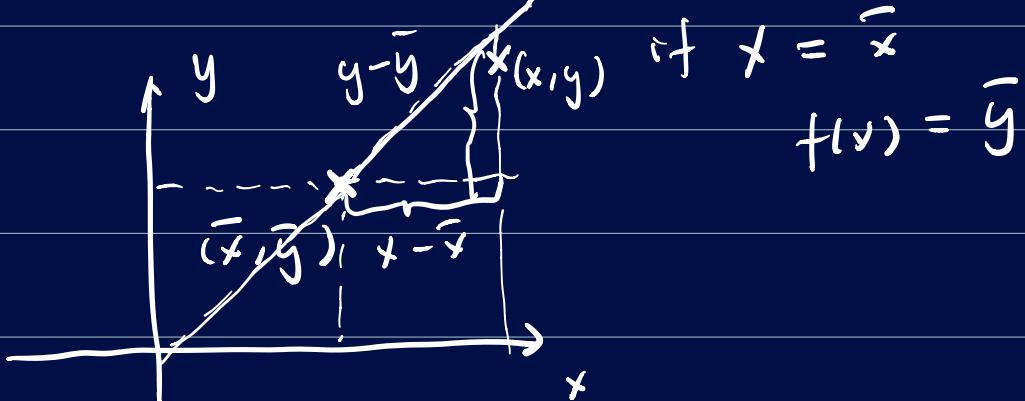
- Variance:

$$\sigma_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2$$

- **Covariance:**

$$\sigma_{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

$$f(x) = \bar{y} + k(x - \bar{x})$$



let's look at a point (x, y)

$$y = f(x) = \bar{y} + k(x - \bar{x})$$

$$y - \bar{y} = k(x - \bar{x})$$

$$\frac{y - \bar{y}}{x - \bar{x}} = k$$

Least Square Solution: Using Statistics

■ Solution:

$$f(x) = \bar{y} + \frac{\sigma_{xy}}{\sigma_x^2} (x - \bar{x})$$

slope $w_1 = \bar{y} + \left(\frac{\sigma_{xy}}{\sigma_x^2} \right) x - \frac{\sigma_{xy}}{\sigma_x^2} \bar{x}$

$$f(x) = \bar{y} + \left[\frac{\sigma_{xy}}{\sigma_x^2} \right] (x - \bar{x})$$

$\frac{\sigma_{xy}}{\sigma_x^2}$

$$w_1 = \frac{\sigma_{xy}}{\sigma_x^2}, \quad w_0 = \bar{y} - w_1 \bar{x}$$

■ Prediction:

$$f(x) = w_0 + w_1 x$$

w_0 w_1

$$w_0 = \bar{y} - \left[\frac{\sigma_{xy}}{\sigma_x^2} \right] \bar{x}$$

conditional expectation of normal distribution

Dimensions, shapes, sizes

Ex. $v = (1, 2, 3)$

vector 1D array $(3,)$
 ↑

$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ 2D array $(3, 2)$
 ↑

matrix x size of first dim = 3
 - - - second -- = 2

T_{ijk} multi-dimensional

Tensor arrays

• linear model : for scalar-valued

x and y $f(x) = w_1x + w_0$

Least Square Solution

$$\left(\text{Sample weights } \beta_i \quad \frac{1}{N} \sum_{i=1}^N \beta_i \|y_i - \hat{y}_i\|^2 \right)$$

- Model:

$$f(x) = w_0 + w_1 x$$

- Loss:

$$J(w_0, w_1) = \frac{1}{N} \sum_{i=1}^N \|y_i - f(x_i)\|^2$$

$(w_0 + w_1 x_i)$

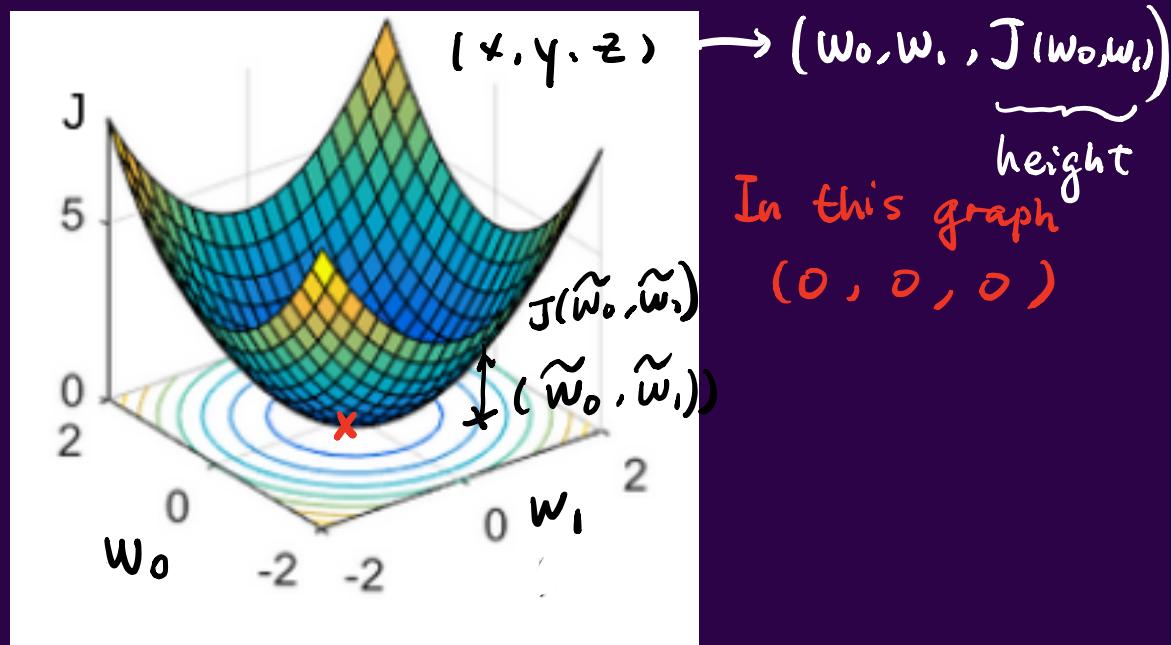
- Optimization: find w_0, w_1 such that $J(w_0, w_1)$ is the least possible value (hence the name “least square”).

$$J^* = \min_{w_0, w_1} J(w_0, w_1) \quad \underline{J(w_0^*, w_1^*) = J^*}$$

$$\rightarrow \hat{w}_0, \hat{w}_1^* = \arg \min_{w_0, w_1} J(w_0, w_1)$$

Loss Landscape

Plot the loss against the parameters:



Linear Regression

- Linear models: For scalar-valued feature x , this is
$$f(x) = w_1x + w_0$$
- One of the simplest machine learning model, yet very powerful.
- Two ways to get the solution, we will show them later.

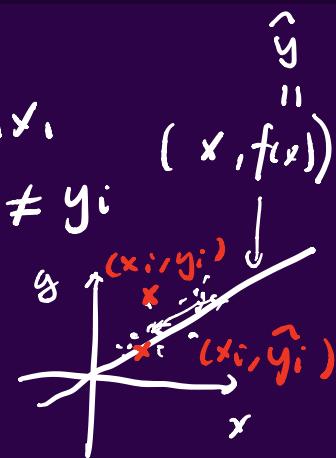
Least Square Solution: Using Pseudo-Inverse

■ For N data points (x_i, y_i) we have, $f(x) = \hat{y}_i \neq y_i$

$$\begin{aligned} y_1 &\approx w_0 + w_1 x_1 = \hat{y}_1 \\ y_2 &\approx w_0 + w_1 x_2 = \hat{y}_2 \end{aligned}$$

 \vdots

$$y_N \approx w_0 + w_1 x_N.$$



Linear Regression

$$\mathbf{v} = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix}$$

- In matrix form we have,

$$\left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \approx \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \right) \|\mathbf{v}\| = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \times \|\mathbf{v}\| = \sqrt{1+4+9}$$

- We can write it as $\mathbf{Y} \approx \mathbf{X}\mathbf{w}$. We call \mathbf{X} the design matrix.

- Exercise: verify $\|\mathbf{Y} - \mathbf{X}\mathbf{w}\|^2 = \sum_{i=1}^N \|y_i - (w_0 + w_1 x_i)\|^2$

N. MSE

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} (1, x_1) \cdot (w_0, w_1) \\ \vdots \\ \vdots \end{bmatrix}$$

$\underbrace{\mathbf{X}}$

design matrix
(feature matrix \mathbf{x})

$$= \begin{bmatrix} w_0 + w_1 x_1 \\ w_0 + w_1 x_2 \\ \vdots \\ w_0 + w_1 x_N \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Normalize a vector $\frac{\mathbf{x}}{\|\mathbf{x}\|} \leftarrow \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|} \right\| = 1 \quad \underbrace{\mathbf{y}}$

Normalize the data : something different

~~X~~

$$\boxed{\begin{aligned} \mathbf{Y} &= \mathbf{XW} \\ \mathbf{X}^{-1} \mathbf{Y} &= \underbrace{\mathbf{X}^{-1} \mathbf{X}}_{\mathbf{I}} \mathbf{W} \\ \mathbf{X}^{-1} \mathbf{Y} &= \mathbf{W} \end{aligned}}$$

We can only do
matrix inverse
for square
matrices

Use pseudo-inverse $\mathbf{W} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$

Linear Least Square

- $\min_{\mathbf{w}} \frac{1}{N} \|Y - X\mathbf{w}\|^2$
- Using the psuedo-inverse (only square matrices have an inverse),

$$\begin{aligned}
 & X \in (\mathbb{R}^{m,n}) \quad X^T X \in (\mathbb{R}^{n,n}) \\
 & Y \approx X\mathbf{w} \quad X^T (n,n) \\
 & X^T Y \approx X^T X\mathbf{w} \\
 & (X^T X)^{-1} X^T Y \approx (X^T X)^{-1} X^T X\mathbf{w} \\
 & \underbrace{(X^T X)^{-1} X^T Y}_{\text{pseudo-inverse of } X} = \underbrace{\mathbf{w}}_{I}
 \end{aligned}$$

↓

the i -th sample has the feature $\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}$

for scalar-valued feature $f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

for vector features

$$f(\mathbf{x}_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} = \hat{y}_i$$

$$i=1 \quad (\mathbf{x}_1, y_1) \quad \hat{y}_1 = f(\mathbf{x}_1)$$

$$i=12 \quad (\mathbf{x}_{12}, y_{12}) \quad \hat{y}_{12} = f(\mathbf{x}_{12})$$

What would a linear model look like if

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{bmatrix}$$

$$f(\mathbf{x}_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3}$$

In general, for \mathbf{x}_i with D elements

$$\text{the linear model } f(\mathbf{x}_i) = w_0 + w_1 x_{i1} + \dots + w_D x_{iD}$$

Linear Regression

- What if we have multivariate data with x being a vector?

- ~~$\hat{y} = w^T x$, here both w and x are vectors.~~

~~Ex: $x_i = [1, x_{i1}, x_{i2}]^T$ and $w = [w_0, w_1, w_2]^T$~~

$$y_1 \approx w_0 + w_1 x_{11} + w_2 x_{12}, \dots$$

- In matrix-vector form

$\hat{y}_i = w^T \varphi(x_i)$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \approx \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_{n2} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = [w_0 \ w_1 \ w_2] \begin{bmatrix} 1 \\ x_{i1} \\ x_{i2} \end{bmatrix}$$

- Solution remains the same $(X^T X)^{-1} X^T Y = w = w_0 + w_1 x_{i1} + w_2 x_{i2}$
- Exercise: open `demo_multilinear.ipynb`

$$\begin{aligned}
 & \left[\begin{array}{ccc} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & & \\ 1 & x_{N1} & x_{N2} \end{array} \right] \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} \\
 = & \begin{bmatrix} w_0 + w_1 x_{11} + w_2 x_{12} \\ w_0 + w_1 x_{21} + w_2 x_{22} \\ \vdots \\ w_0 + w_1 x_{N1} + w_2 x_{N2} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}
 \end{aligned}$$