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. ↳ $\hat{y} = f(x)$

$\hat{y}$: prediction from the model

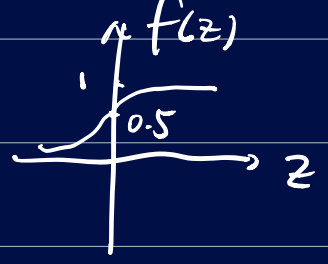
model parameters: coefficients of the function $f(\underline{x})$

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

w_0 is the coefficient of x^0

$x^0 = 1$

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



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

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

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

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

$f(x; w)$ 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: $\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$

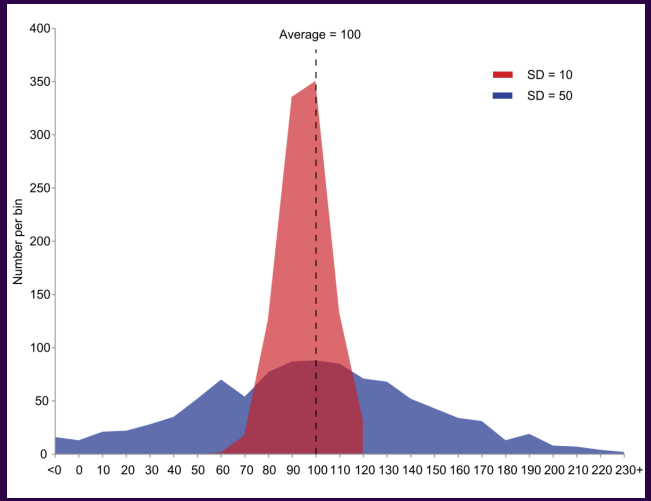
Outline

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

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

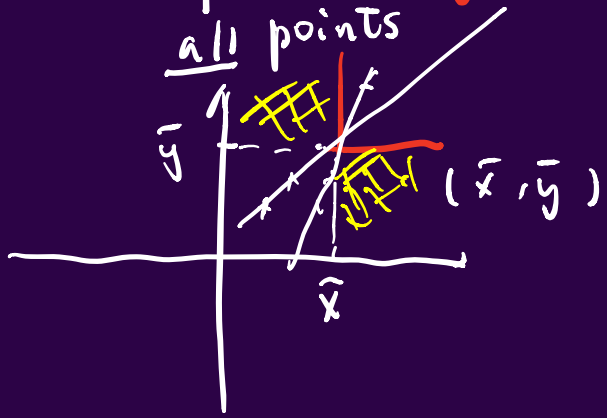


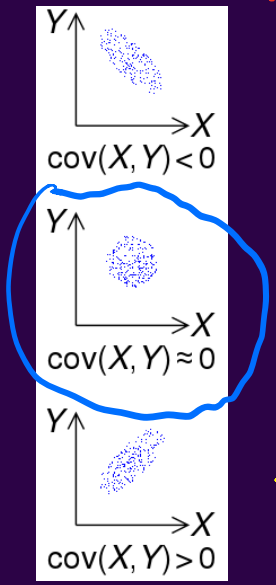
https://en.wikipedia.org/wiki/Variance

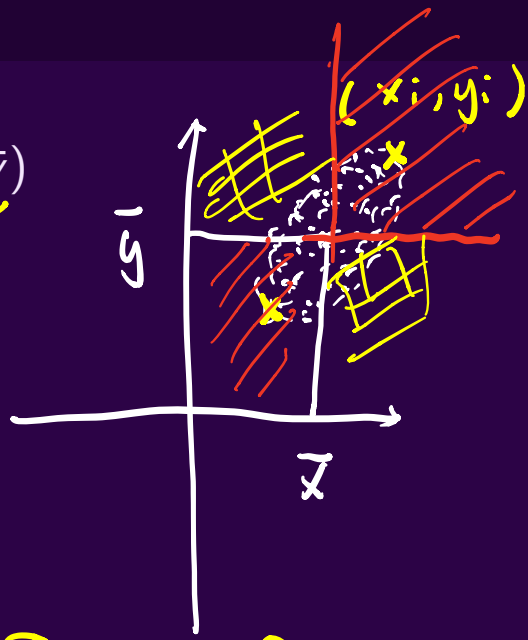
Covariance

- 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 for all points





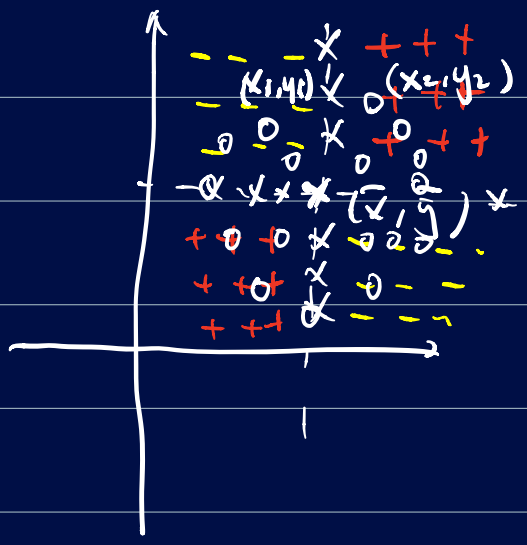


$\sigma_{xy} > 0$?

$\Rightarrow$ for all i $(x_i - \bar{x})(y_i - \bar{y})$ are positive

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

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



$$\sum_{i=1}^{N} (x_i-\bar{x}_i)(y_i-\bar{y}_i) \stackrel{?}{=} 0$$

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), $i = 1, 2, ..., N$
- Mean:

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

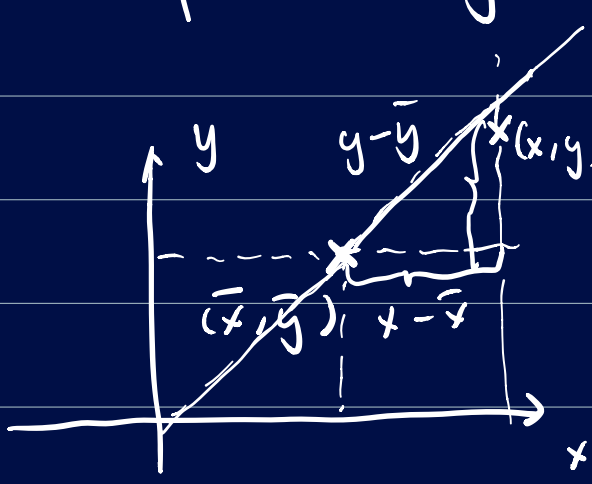
- Variance:

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

- **Covariance**:

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

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



if $x = \bar{x}$

$f(x) = \bar{y}$

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 k

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

w_1 w_1

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

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

w_1

- Prediction:

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

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 — $(3, 2)$

size of first dim = 3

... second ... = 2

matrix

T_{ijk}

Tensor

multi-dimensional arrays

- linear model: for scalar-valued x and y $\quad f(x) = w_1 x + w_0$

Least Square Solution

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

- Model:

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

- Loss:

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

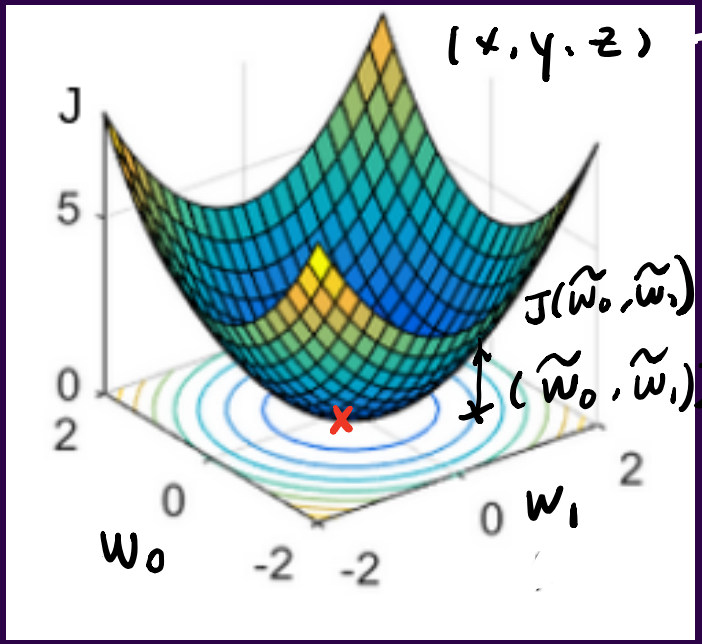
- Optimization: find w_0, w_1 such that $J(w_0, w_1)$ is the least possible value (hence the name "least square").

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

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

Loss Landscape

Plot the loss against the parameters:



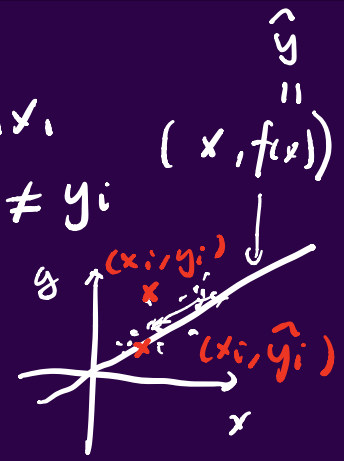
Linear Regression

- Linear models: For scalar-valued feature x, this is $f(x) = w_1 x + w_0$
- 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,

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



Linear Regression

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

$$|v| = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \times$$

$$\|v\| = \sqrt{1+4+9}$$

- In matrix form we have,

$$\left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \approx \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \right)$$

- We can write it as $Y \approx X\mathbf{w}$. We call X the design matrix.
- Exercise: verify $\|Y - X\mathbf{w}\|^2 = \sum_{i=1}^{N} \|y_i - (w_0 + w_1 x_i)\|^2$

$N \cdot MSE$

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

$$\underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{bmatrix}}_{X} = \begin{bmatrix} w_0 + w_1 x_1 \\ w_0 + w_1 x_2 \\ \vdots \\ w_0 + w_1 x_N \end{bmatrix} \approx \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{Y}$$

design matrix (feature matrix)

Normalize a vector $\frac{x}{\|x\|} \leftarrow \left\| \frac{x}{\|x\|} \right\| = 1$

Normalize the data : something different

$$Y = Xw$$

$$X^{-1} Y = \underbrace{X^{-1} X}_{I} w$$

$$I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$

$$X^{-1} Y = w$$

We can only do matrix inverse for square matrices

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

Linear Least Square

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

X (m,n) $\quad X^TX$ (n,n)

X^T (n,n)

$$Y \approx X\mathbf{w}$$
$$X^TY \approx X^TX\mathbf{w}$$
$$(X^TX)^{-1}X^TY \approx (X^TX)^{-1}X^TX\mathbf{w}$$
$$(X^TX)^{-1}X^TY = \mathbf{w}.$$

$(X^TX)^{-1}X^TX = I$

pseudo-inverse of X

Linear Regression

- What if we have multivariate data with $\mathbf{x}$ being a vector?
- Ex: $\mathbf{x}_i = [x_{i1}, x_{i2}]^T$

$$
\begin{aligned}
y_1 &\approx w_0 + w_1 x_{11} + w_2 x_{12} = \hat{y}_1 \\
y_2 &\approx w_0 + w_1 x_{21} + w_2 x_{22} = \hat{y}_2 \\
&\vdots \\
y_N &\approx w_0 + w_1 x_{N1} + w_2 x_{N2} = \hat{y}_N
\end{aligned}
$$

- The model can be written as $\hat{y}_i = \mathbf{w}^T \phi(\mathbf{x}_i)$, here both $\mathbf{w} = [w_0, w_1, w_2]^T$ and $\mathbf{x}$ are vectors. $\phi(\mathbf{x})$ is a feature transformation that transforms the original feature to $\phi(\mathbf{x}_i) = [1, x_{i1}, x_{i2}]^T$.

NYU TANDON SCHOOL OF ENGINEERING

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

for scalar-valued feature $f(x) = w_1 x + w_0$

for vector features

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

$i = 1$ (x_1, y_1) $\hat{y}_1 = f(x_1)$

$i = 12$ (x_{12}, y_{12}) $\hat{y}_{12} = f(x_{12})$

What would a linear model look like if

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{bmatrix}$$

$$f(x_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3}$$

In general, for x_i with D elements

the linear model $f(x_i) = w_0 + w_1 x_{i1} + \cdots + w_D x_{iD}$

$$\begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & & \\ 1 & x_{N1} & x_{N2} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

$$= \begin{bmatrix} w_0 + w_1 x_{11} + w_2 x_{12} \\ w_0 + w_1 x_{21} + w_2 x_{22} \\ \vdots \\ w_0 + w_1 x_{N1} + w_2 x_{N2} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$