10 | Regularization

Ivan Corneillet

Data Scientist

Learning Objectives

After this lesson, you should be able to:

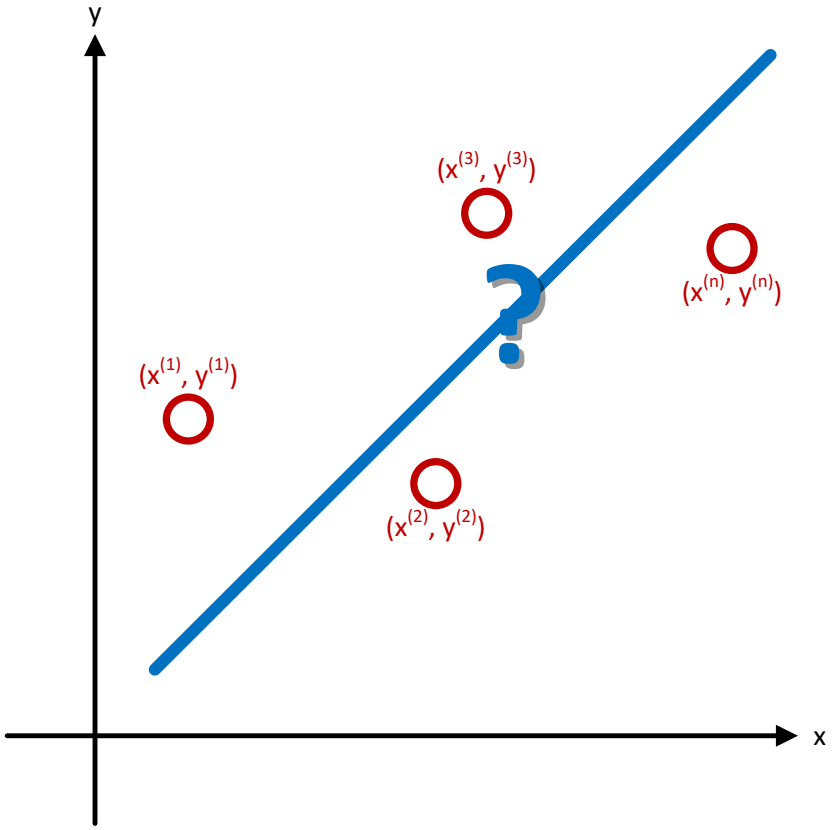
- Understand the closed-form solution of the regression coefficients for linear regression models
- Use Ordinary Least Squares (OLS) and Loss Functions to also derive estimations for the coefficients
- Understand the Regularization Bias-Variance Trade-Off

Here's what's happening today:

- How to fit a linear regression model on a dataset?
  - Closed-form solution for $\hat{\beta}$
  - Ordinary Least Squares (OLS) and Loss Functions
- Regularization

How to fit a linear regression model on a dataset?

How do we estimate $\hat{\beta}$?



Notations

$$y^{(i)} = \sum_{j=0}^{k} \beta_j \cdot x_j^{(i)} + \varepsilon^{(i)} \; \forall i, 1 \leq i \leq n$$

$$y = X \cdot \beta + \varepsilon$$

$$y = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{pmatrix}; \; X = \begin{pmatrix} | & | & & | \\ x_0 & x_1 & \cdots & x_k \\ | & | & & | \end{pmatrix}; \; \beta = \begin{pmatrix} | \\ \beta \\ | \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_k \end{pmatrix}; \; \varepsilon = \begin{pmatrix} \varepsilon^{(1)} \\ \vdots \\ \varepsilon^{(n)} \end{pmatrix}$$

$$x_0 = \begin{pmatrix} x_0^{(1)} = 1 \\ \vdots \\ x_0^{(n)} = 1 \end{pmatrix}; \; x_j = \begin{pmatrix} x_j^{(1)} \\ \vdots \\ x_j^{(n)} \end{pmatrix}$$

Matrix Multiplication ($X \cdot \beta$)

(row i/column j of $X \cdot \beta$ is the dot product of row i of X and column 1 of β)

$$\beta = \begin{pmatrix} \boldsymbol{\beta_0} \\ \boldsymbol{\beta_1} \\ \vdots \\ \boldsymbol{\beta_k} \end{pmatrix}$$

$$X = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_k^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{1} & \boldsymbol{x_1^{(i)}} & \cdots & \boldsymbol{x_k^{(i)}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \cdots & x_k^{(n)} \end{pmatrix} \begin{pmatrix} \beta_0 \cdot 1 + \beta_1 \cdot x_1^{(1)} + \cdots + \beta_k \cdot x_k^{(1)} \\ \vdots \\ \boldsymbol{\beta_0 \cdot 1 + \beta_1 \cdot x_1^{(i)} + \cdots + \beta_k \cdot x_k^{(i)}} \\ \vdots \\ \beta_0 \cdot 1 + \beta_1 \cdot x_1^{(n)} + \cdots + \beta_k \cdot x_k^{(n)} \end{pmatrix} = X \cdot \beta$$

$$y = X \cdot \beta + \varepsilon$$

$$\underbrace{\begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} \beta_0 + \beta_1 \cdot x_1^{(1)} + \cdots + \beta_k \cdot x_k^{(1)} \\ \vdots \\ \beta_0 + \beta_1 \cdot x_1^{(n)} + \cdots + \beta_k \cdot x_k^{(n)} \end{pmatrix}}_{X \cdot \beta} + \underbrace{\begin{pmatrix} \varepsilon^{(1)} \\ \vdots \\ \varepsilon^{(n)} \end{pmatrix}}_{\varepsilon}$$

How to fit a linear regression model on a dataset?

Closed-form solution for $\hat{\beta}$

Closed-form solution for $\hat{\beta}$ – Take 1

$$y_{train} = X_{train} \cdot \hat{\beta}$$

- We would like to left multiply both sides by X^{-1} and get:

$$X_{train}^{-1} \cdot y_{train} = X_{train}^{-1} \cdot \left(X_{train} \cdot \hat{\beta}\right) = \left(X_{train}^{-1} \cdot X_{train}\right) \cdot \hat{\beta} = \hat{\beta}$$

- However, X_{train} is usually not invertible (it would need to be a square matrix in the first place which would mean having as many features as samples; not good, right?)

$$\hat{\beta} = X_{train}^{-1} \cdot y_{train}$$

Closed-form solution for $\hat{\beta}$ – Take 2

- Let's start over and this time, we left multiply both sides by X_{train}^T:

$$X_{train}^T \cdot y_{train} = X_{train}^T \cdot \left(X_{train} \cdot \hat{\beta}\right) = \left(X_{train}^T \cdot X_{train}\right) \cdot \hat{\beta}$$

- $X^T \cdot X$ is a symmetric matrix and is usually invertible; if not we can slightly reformulate the problem to make it invertible

$$\left(X_{train}^T \cdot X_{train}\right)_{i,j} = \sum_{l=0}^{k} \left(x_{train}^{(i)}\right)_l \cdot \left(x_{train}^{(j)}\right)_l = \left(X_{train}^T \cdot X_{train}\right)_{j,i}$$

- We can now left multiply both sides by $(X_{train}^T \cdot X_{train})^{-1}$:

$$\left(X_{train}^T \cdot X_{train}\right)^{-1} \cdot X^T \cdot y_{train} = \left(X_{train}^T \cdot X_{train}\right)^{-1} \cdot \left(\left(X_{train}^T \cdot X_{train}\right) \cdot \hat{\beta}\right)$$

$$= \left(\left(\left(X_{train}^T \cdot X_{train}\right)^{-1}\right) \cdot \left(\left(X_{train}^T \cdot X_{train}\right)\right)\right) \cdot \hat{\beta} = \hat{\beta}$$

Closed-form solution for $\hat{\beta}$ – Take 2 (cont.)

$$\hat{\beta} = \left(X_{train}^{T} \cdot X_{train}\right)^{-1} \cdot X_{train}^{T} \cdot y_{train}$$

$$\hat{y}_{predict} = X_{predict} \cdot \hat{\beta}$$

Closed-form solution for $\hat{\beta}$ – Take 2 (cont.)

- Was the matrix X_{train}^{T} the only matrix possible we could use for the left-multiply operation?
  - No. But it will because clear in the next section

Closed-form solution for $\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix}$

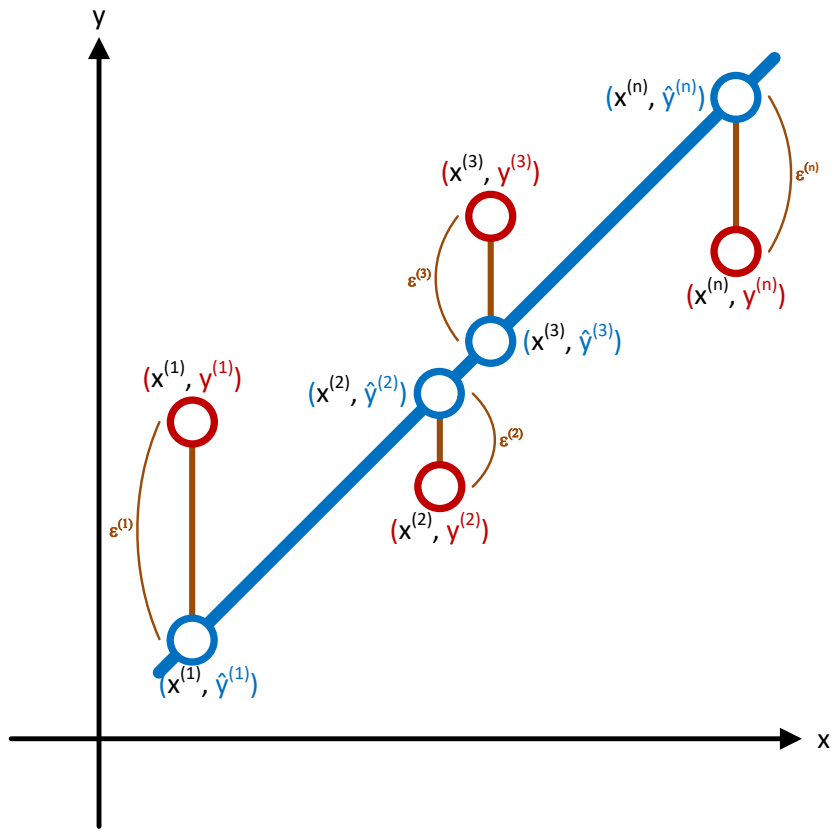
$$\hat{\beta}_1 = \frac{cov(x_{train}, y_{train})}{var(x_{train})}$$

$$\hat{\beta}_0 = \bar{y}_{train} - \bar{x}_{train} \cdot \hat{\beta}_1$$

How to fit a linear regression model on a dataset?

Ordinary Least Squares (OLS) and Loss Functions

We can also estimate $\hat{\beta}_0$ and $\hat{\beta}_1$ with Ordinary Least Squares (OLS)



- Hypothesis

$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$$

- Parameters

$$\hat{\beta}_0, \hat{\beta}_1$$

- Goal

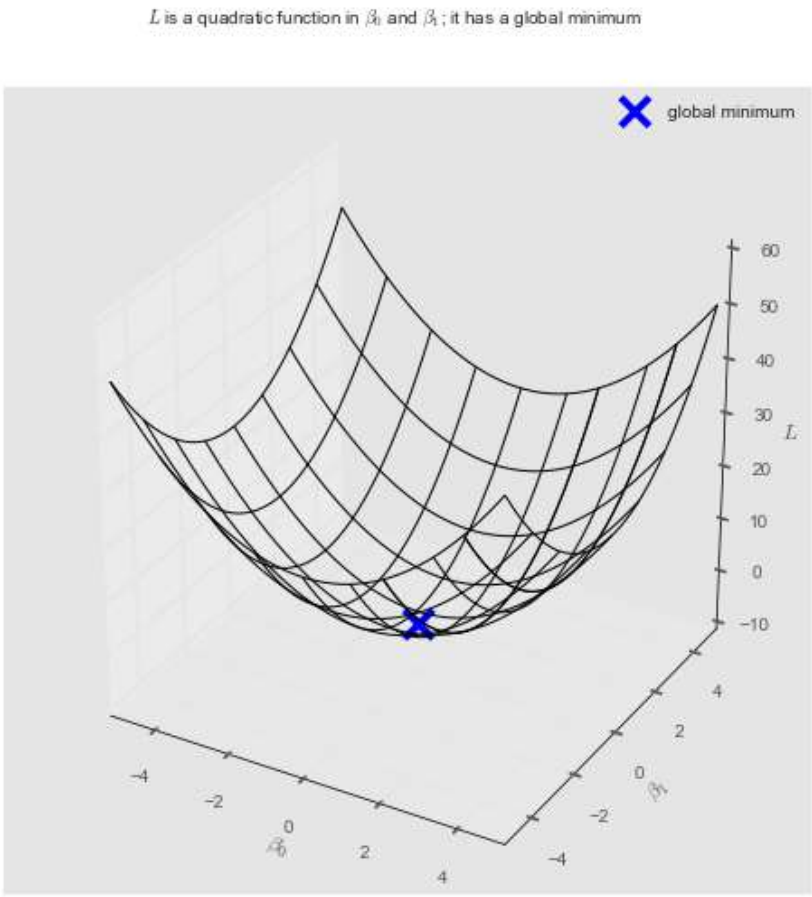
$$\min_{\hat{\beta}_0, \hat{\beta}_1} \underbrace{\sum_{i=1}^{n} \left(y_{train}^{(i)} - \hat{y}\left(x_{train}^{(i)} \right) \right)^2}_{L(\hat{\beta}_0, \hat{\beta}_1)}$$

(i.e., minimizing the least square errors)

$L\left(y_{train}^{(i)} - \hat{y}\left(x_{train}^{(i)}\right)\right)$ is a quadratic function in $\hat{\beta}_0$ and $\hat{\beta}_1$ in the form

$$A\hat{\beta}_0^2 + B\hat{\beta}_0\hat{\beta}_1 + C\hat{\beta}_1^2 + D\hat{\beta}_0 + E\hat{\beta}_1 + F$$

(A, B, C, D, E, and F constant)



How to fit a linear regression model on a dataset?

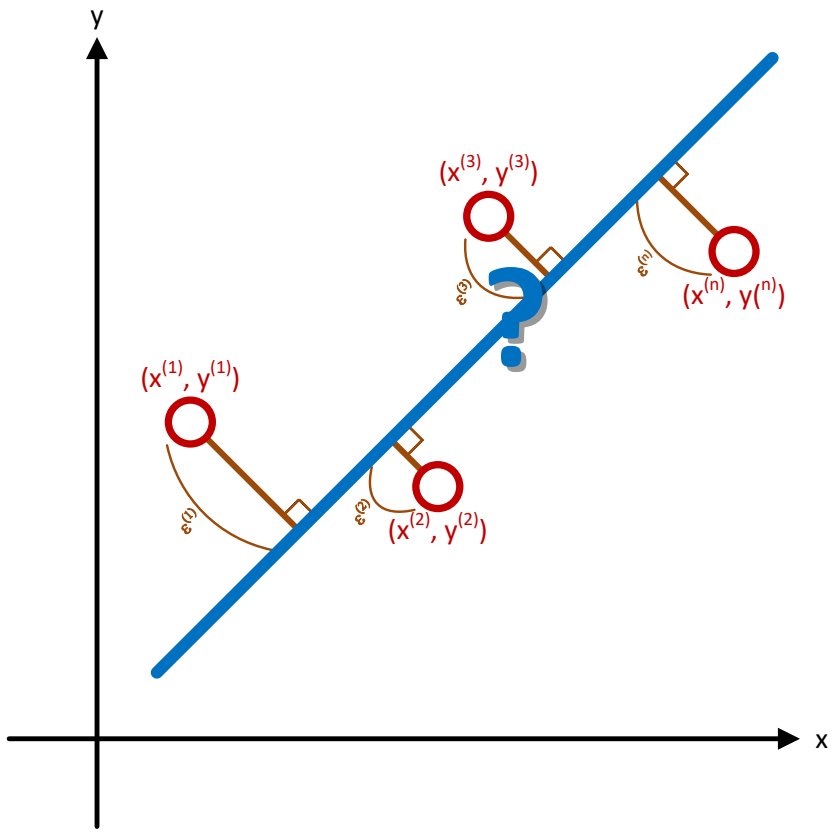
Ordinary Least Squares (OLS) and the closed-form solution for $\hat{\beta}$

Ordinary Least Squares (OLS) and the closed-form for $\hat{\beta}$

- Modeling just an intercept ($\hat{y} = \hat{\beta}_0$) or an intercept and a slope ($\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$) yield the same results for both the closed-form solution for $\hat{\beta}$ and the Ordinary Least Squares
  - In fact, minimizing $L(\hat{\beta}) = (y_{train} - X_{train} \cdot \hat{\beta})^T \cdot (y_{train} - X_{train} \cdot \hat{\beta})$ in the general case yields our previous closed-form solution for $\hat{\beta} = (X_{train}^T \cdot X_{train})^{-1} \cdot X_{train}^T \cdot y_{train}$

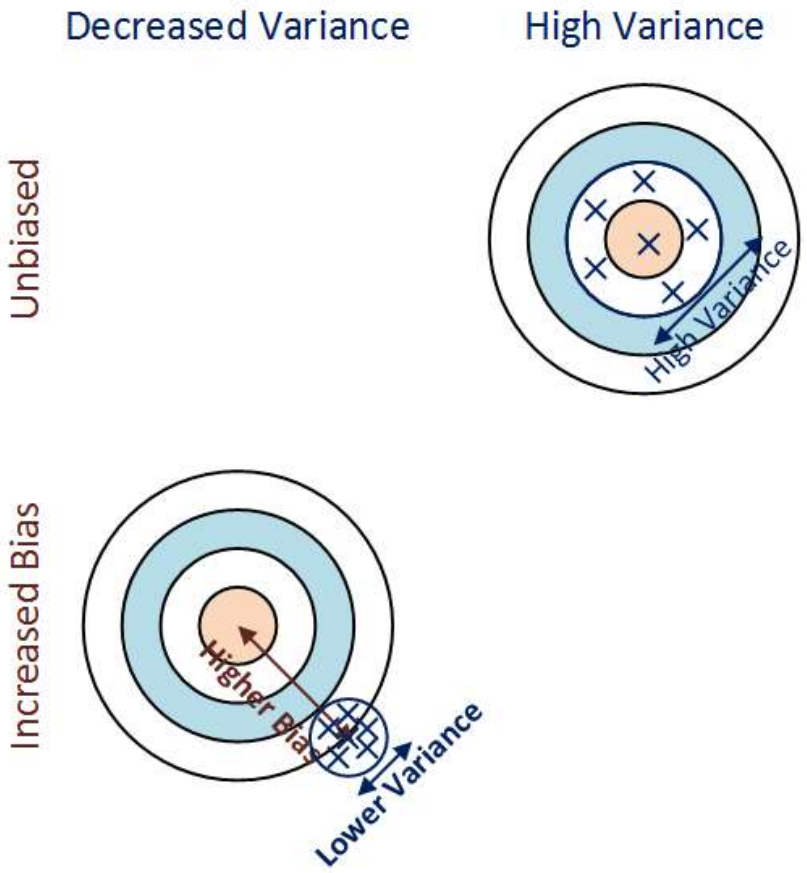
There are Many Ways to Fit a Line…

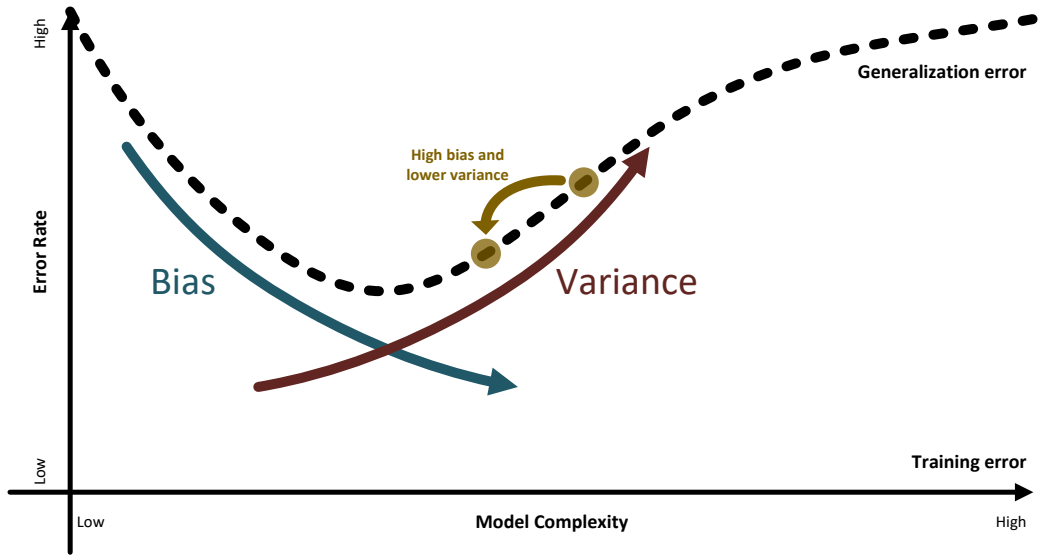
- It can be shown that $\hat{y}$ is unbiased for y
  - I.e., $E[\hat{y}] = y$
- E.g., if $\hat{\beta} = (\hat{\beta}_0)$, then $\hat{y} = \bar{y}$ is unbiased
  - In the SF housing dataset, if you don't anything about the houses (e.g., size, etc.) then the best estimation of the sale price you can give is the mean of the training set's sale price



Regularization

OLS yields Unbiased Estimators at the cost of High Variance. Can we trade some (Higher) Bias for Lower Variance and get ahead on the Bias-Variance Trade-off?





Revisiting Complexity

- E.g., as a function of the size of the coefficients
  - $\|\beta\|_p = \left(\sum_{j=0}^{k} |\beta_j|^p\right)^{1/p}$ (Lp-norm)
  - $\|\beta\|_1 = \sum_{j=0}^{k} |\beta_j|$ (L1-norm)
  - $\|\beta\|_2 = \left(\sum_{j=0}^{k} |\beta_j|^2\right)^{1/2}$ (L2-norm)

Regularization helps against Overfitting by explicitly controlling Model Complexity

- These definitions of complexity lead to the following regularization techniques
  - $min\left(\underbrace{\|y_{train} - X_{train} \cdot \beta\|^2}_{OLS\ term} + \underbrace{\lambda\left\|\hat{\beta}\right\|_1}_{regularization\ term}\right)$ (Lasso regularization using the L1 norm)
  - $min\left(\|y_{train} - X_{train} \cdot \beta\|^2 + \lambda\left\|\hat{\beta}\right\|_2^2\right)$ (Ridge regularization using the L2 norm)
    - (note that in the loss function the term $\hat{\beta}_0$ isn't regularized and is in fact excluded from the norm here)
- This formulation reflects the fact that there is a cost associated with regularization that we want to minimize

About Loss Functions

- Loss functions are a powerful tool to optimize the fit of machine learning algorithms
- Loss functions are not limited to linear regression- and regularization-based models.
  - E.g., training a logistic regression algorithm (while also leveraging linear regression) is also modeled and fitted with loss functions