Introduction to Machine Learning

Lecture 3 Linear Regression - Introduction

Hongteng Xu

中国人民大學 RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院 Gaoling School of Artificial Intelligence

Outline

Review

- **Linear algebra:** Basic operations of matrix and linear space
- **Statistics:** Expectation, variance, and the method of moments
- **Probability:** MLE v.s. MAP, two statistical ML paradigms

Outline

Review

- **Linear algebra:** Basic operations of matrix and linear space
- **Statistics:** Expectation, variance, and the method of moments
- **Probability:** MLE v.s. MAP, two statistical ML paradigms

Today

- Linear regression: Take polynomial regression as an example
- Pre-processing, training, and evaluation
- Model selection: AIC v.s. BIC

Polynomial Regression

A polynomial with $N-1$ degrees:

$$p(x) = c_0 + c_1 x + \dots + c_{N-1} x^{N-1} = \sum_{j=1}^{N} \underbrace{c_{j-1}}_{\text{Coefficient}} x^{j-1}. \tag{1}$$

Polynomial Regression

A polynomial with $N-1$ degrees:

$$p(x) = c_0 + c_1 x + \ldots + c_{N-1} x^{N-1} = \sum_{j=1}^{N} \underbrace{c_{j-1}}_{\text{Coefficient}} x^{j-1}. \tag{1}$$

Given a sequence of numbers, *i.e.*, $x_1, \ldots, x_M$, the mapping

$$\underbrace{[c_0, \ldots, c_{N-1}]^\top}_{\boldsymbol{c}} \mapsto \underbrace{[p(x_1), \ldots, p(x_M)]^\top}_{p(\boldsymbol{x})}$$

is linear.

Polynomial Regression

A polynomial with $N-1$ degrees:

$$p(x) = c_0 + c_1 x + \ldots + c_{N-1}x^{N-1} = \sum_{j=1}^{N} \underbrace{c_{j-1}}_{\text{Coefficient}} x^{j-1}. \tag{1}$$

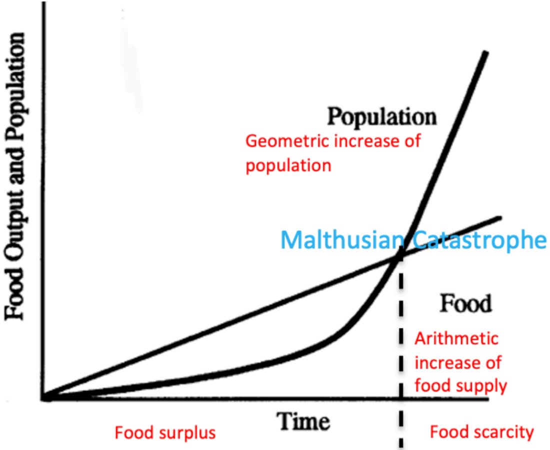
Given a sequence of numbers, *i.e.*, $x_1, \ldots, x_M$, the mapping

$$\underbrace{[c_0, \ldots, c_{N-1}]^\top}_{\boldsymbol{c}} \mapsto \underbrace{[p(x_1), \ldots, p(x_M)]^\top}_{p(\boldsymbol{x})}$$

is linear.

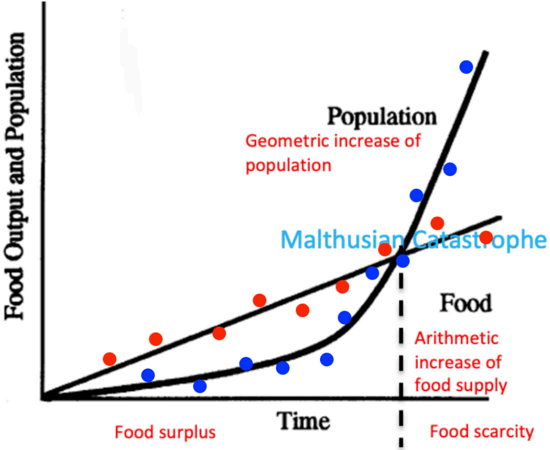
- $p(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{c}$
- Vandermonde matrix: $\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{N-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_M & x_M^2 & \cdots & x_M^{N-1} \end{bmatrix}$

Polynomial Regression: An Example

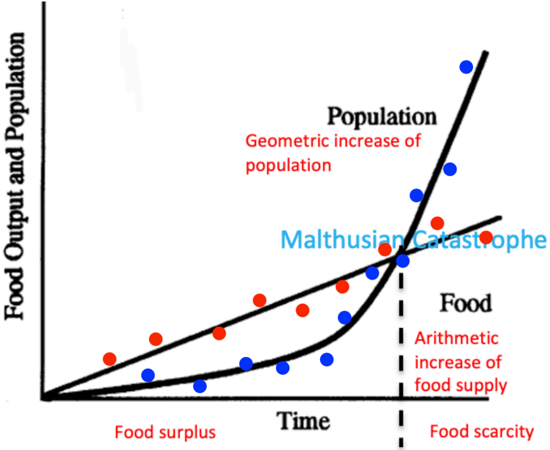


Malthus's principle of population.

Polynomial Regression: An Example



Polynomial Regression: An Example



Could you enumerate more cases suitable for polynomial regression?

The Rationality behind Polynomial Regression

The Rationality behind Polynomial Regression

- **Taylor Expansion:** For $f(x) : \mathbb{R} \mapsto \mathbb{R} \in \mathbb{C}^\infty$ and $a \in \mathbb{R}$,

$$\begin{aligned} f(x) &= f(a) + \frac{f^{(1)}(a)}{1!}(x-a) + \frac{f^{(2)}(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n. \end{aligned} \tag{2}$$

The Rationality behind Polynomial Regression

- **Taylor Expansion:** For $f(x) : \mathbb{R} \mapsto \mathbb{R} \in \mathbb{C}^{\infty}$ and $a \in \mathbb{R}$,

$$\begin{aligned} f(x) &= f(a) + \frac{f^{(1)}(a)}{1!}(x-a) + \frac{f^{(2)}(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n. \end{aligned} \tag{2}$$

- Polynomial function can cover and approximate a large set of functions.

The Rationality behind Polynomial Regression

- **Taylor Expansion:** For $f(x) : \mathbb{R} \mapsto \mathbb{R} \in \mathbb{C}^{\infty}$ and $a \in \mathbb{R}$,

$$\begin{aligned} f(x) &= f(a) + \frac{f^{(1)}(a)}{1!}(x-a) + \frac{f^{(2)}(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n. \end{aligned} \tag{2}$$

- Polynomial function can cover and approximate a large set of functions.
- Could you enumerate the functions that cannot be fit well by polynomials?

A Naïve Learning Strategy

Given labeled data $\{(x_n, y_n)\}_{i=1}^{N}$, train a D-th order polynomial regression model:

$$y = \sum_{d=1}^{D} w_d x^{d-1} + \epsilon. \tag{3}$$

- The Vandermonde matrix $\boldsymbol{X} = [x_n^{d-1}] \in \mathbb{R}^{N \times D}$ and the label vector $\boldsymbol{y} = [y_n] \in \mathbb{R}^N$.
- Learning the model via:

$$\min_{\boldsymbol{w}} \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_p^p}_{L(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y})} \tag{4}$$

A Naïve Learning Strategy

Given labeled data $\{(x_n, y_n)\}_{i=1}^{N}$, train a D-th order polynomial regression model:

$$y = \sum_{d=1}^{D} w_d x^{d-1} + \epsilon. \tag{3}$$

- The Vandermonde matrix $\boldsymbol{X} = [x_n^{d-1}] \in \mathbb{R}^{N \times D}$ and the label vector $\boldsymbol{y} = [y_n] \in \mathbb{R}^N$.
- Learning the model via:

$$\min_{\boldsymbol{w}} \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_p^p}_{L(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y})} \tag{4}$$

- When $p = 2$, we have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \mathbf{0}$$

A Naïve Learning Strategy

Given labeled data $\{(x_n, y_n)\}_{i=1}^N$, train a D-th order polynomial regression model:

$$y = \sum_{d=1}^{D} w_d x^{d-1} + \epsilon. \tag{3}$$

- The Vandermonde matrix $\boldsymbol{X} = [x_n^{d-1}] \in \mathbb{R}^{N\times D}$ and the label vector $\boldsymbol{y} = [y_n] \in \mathbb{R}^N$.
- Learning the model via:

$$\min_{\boldsymbol{w}} \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_p^p}_{L(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y})} \tag{4}$$

- When $p = 2$, we have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \mathbf{0} \quad \Rightarrow \quad 2\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = \mathbf{0}$$

A Naïve Learning Strategy

Given labeled data $\{(x_n, y_n)\}_{i=1}^{N}$, train a D-th order polynomial regression model:

$$y = \sum_{d=1}^{D} w_d x^{d-1} + \epsilon. \tag{3}$$

- The Vandermonde matrix $\boldsymbol{X} = [x_n^{d-1}] \in \mathbb{R}^{N \times D}$ and the label vector $\boldsymbol{y} = [y_n] \in \mathbb{R}^N$.
- Learning the model via:

$$\min_{\boldsymbol{w}} \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_p^p}_{L(\boldsymbol{w};\boldsymbol{X},\boldsymbol{y})} \tag{4}$$

- When $p = 2$, we have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{0} \quad \Rightarrow \quad 2\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{5}$$

A Frequentist Viewpoint: Maximum Likelihood Estimation (MLE)

- Recall the model: $y = \underbrace{\sum_{d=1}^{D} w_d x^{d-1}}_{\boldsymbol{x}^T \boldsymbol{w}} + \epsilon$
- Assume noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$

A Frequentist Viewpoint: Maximum Likelihood Estimation (MLE)

- Recall the model: $y = \underbrace{\sum_{d=1}^{D} w_d x^{d-1}}_{\boldsymbol{x}^T\boldsymbol{w}} + \epsilon$
- Assume noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$y - \boldsymbol{x}^T\boldsymbol{w} \sim \mathcal{N}(0, \sigma^2) \quad \Rightarrow \quad \underbrace{p(y - \boldsymbol{x}^T\boldsymbol{w})}_{p(y|\boldsymbol{w},\boldsymbol{x})} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \boldsymbol{x}^T\boldsymbol{w})^2}{2\sigma^2}\right) \tag{6}$$

A Frequentist Viewpoint: Maximum Likelihood Estimation (MLE)

- Recall the model: $y = \underbrace{\sum_{d=1}^{D} w_d x^{d-1}}_{\boldsymbol{x}^T\boldsymbol{w}} + \epsilon$
- Assume noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$y - \boldsymbol{x}^T\boldsymbol{w} \sim \mathcal{N}(0, \sigma^2) \quad \Rightarrow \quad \underbrace{p(y - \boldsymbol{x}^T\boldsymbol{w})}_{p(y|\boldsymbol{w},\boldsymbol{x})} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \boldsymbol{x}^T\boldsymbol{w})^2}{2\sigma^2}\right) \tag{6}$$

- Given $\{(x_n, y_n)\}_{n=1}^{N}$, maximum likelihood estimation:

$$\max_{\boldsymbol{w}} \prod_{n=1}^{N} p(y_n - \boldsymbol{x}_n^T\boldsymbol{w}) \qquad \text{(i.i.d. Assumption)}$$

A Frequentist Viewpoint: Maximum Likelihood Estimation (MLE)

- Recall the model: $y = \underbrace{\sum_{d=1}^{D} w_d x^{d-1}}_{\boldsymbol{x}^T\boldsymbol{w}} + \epsilon$
- Assume noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$y - \boldsymbol{x}^T\boldsymbol{w} \sim \mathcal{N}(0, \sigma^2) \quad \Rightarrow \quad \underbrace{p(y - \boldsymbol{x}^T\boldsymbol{w})}_{p(y|\boldsymbol{w},\boldsymbol{x})} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \boldsymbol{x}^T\boldsymbol{w})^2}{2\sigma^2}\right) \tag{6}$$

- Given $\{(x_n, y_n)\}_{n=1}^{N}$, maximum likelihood estimation:

$$\max_{\boldsymbol{w}} \prod_{n=1}^{N} p(y_n - \boldsymbol{x}_n^T\boldsymbol{w}) \qquad \text{(i.i.d. Assumption)}$$

$$\Rightarrow \min_{\boldsymbol{w}} -\sum_{n=1}^{N} \log p(y_n - \boldsymbol{x}_n^T\boldsymbol{w}) \qquad \text{(Negative Log-likelihood)}$$

A Frequentist Viewpoint: Maximum Likelihood Estimation (MLE)

- Recall the model: $y = \underbrace{\sum_{d=1}^{D} w_d x^{d-1}}_{\boldsymbol{x}^T \boldsymbol{w}} + \epsilon$
- Assume noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$y - \boldsymbol{x}^T \boldsymbol{w} \sim \mathcal{N}(0, \sigma^2) \quad \Rightarrow \quad \underbrace{p(y - \boldsymbol{x}^T \boldsymbol{w})}_{p(y|\boldsymbol{w},\boldsymbol{x})} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \boldsymbol{x}^T \boldsymbol{w})^2}{2\sigma^2}\right) \tag{6}$$

- Given $\{(x_n, y_n)\}_{n=1}^{N}$, maximum likelihood estimation:

$$\max_{\boldsymbol{w}} \prod_{n=1}^{N} p(y_n - \boldsymbol{x}_n^T \boldsymbol{w}) \qquad \text{(i.i.d. Assumption)}$$

$$\Rightarrow \min_{\boldsymbol{w}} - \sum_{n=1}^{N} \log p(y_n - \boldsymbol{x}_n^T \boldsymbol{w}) \qquad \text{(Negative Log-likelihood)}$$

$$\Rightarrow \min_{\boldsymbol{w}} \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^T \boldsymbol{w})^2 - \text{Const.}$$

A Frequentist Viewpoint: Maximum Likelihood Estimation (MLE)

- Recall the model: $y = \underbrace{\sum_{d=1}^{D} w_d x^{d-1}}_{\boldsymbol{x}^T\boldsymbol{w}} + \epsilon$
- Assume noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$

$$y - \boldsymbol{x}^T\boldsymbol{w} \sim \mathcal{N}(0, \sigma^2) \quad \Rightarrow \quad \underbrace{p(y - \boldsymbol{x}^T\boldsymbol{w})}_{p(y|\boldsymbol{w},\boldsymbol{x})} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \boldsymbol{x}^T\boldsymbol{w})^2}{2\sigma^2}\right) \tag{6}$$

- Given $\{(x_n, y_n)\}_{n=1}^{N}$, maximum likelihood estimation:

$$\begin{aligned}
&\max_{\boldsymbol{w}} \prod_{n=1}^{N} p(y_n - \boldsymbol{x}_n^T\boldsymbol{w}) && \text{(i.i.d. Assumption)} \\
\Rightarrow &\min_{\boldsymbol{w}} -\sum_{n=1}^{N} \log p(y_n - \boldsymbol{x}_n^T\boldsymbol{w}) && \text{(Negative Log-likelihood)} \\
\Rightarrow &\min_{\boldsymbol{w}} \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^T\boldsymbol{w})^2 - \text{Const.} \\
\Rightarrow &\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{w}\|_2^2
\end{aligned} \tag{7}$$

When Facing Big Data: First-order Optimization

- Closed form solution: $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- How many operations does it involve?

When Facing Big Data: First-order Optimization

- Closed form solution: $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- How many operations does it involve? $\mathcal{O}(D^2N + D^3)$

When Facing Big Data: First-order Optimization

- Closed form solution: $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- How many operations does it involve? $\mathcal{O}(D^2N + D^3)$

Stochastic gradient descent:

- Initialize $\boldsymbol{w}_0$, randomly.

When Facing Big Data: First-order Optimization

- Closed form solution: $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- How many operations does it involve? $\mathcal{O}(D^2N + D^3)$

Stochastic gradient descent:

- Initialize $\boldsymbol{w}_0$, randomly.
- At the t-th iteration:
  - Sample a batch of data $\boldsymbol{y}_B, \boldsymbol{X}_B$ randomly.
  - Compute the gradient $\frac{\partial L}{\partial \boldsymbol{w}} = 2\boldsymbol{X}_B^T(\boldsymbol{X}_B\boldsymbol{w}_{t-1} - \boldsymbol{y}_B)$.

When Facing Big Data: First-order Optimization

- Closed form solution: $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- How many operations does it involve? $\mathcal{O}(D^2N + D^3)$

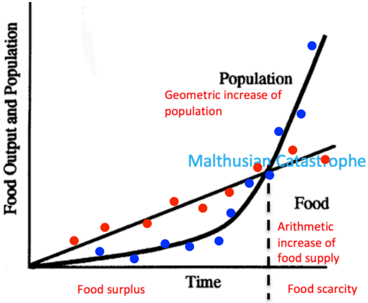
Stochastic gradient descent:

- Initialize $\boldsymbol{w}_0$, randomly.
- At the *t*-th iteration:
  - Sample a batch of data $\boldsymbol{y}_B, \boldsymbol{X}_B$ randomly.
  - Compute the gradient $\frac{\partial L}{\partial \boldsymbol{w}} = 2\boldsymbol{X}_B^T(\boldsymbol{X}_B\boldsymbol{w}_{t-1} - \boldsymbol{y}_B)$.
  - $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \tau \frac{\partial L}{\partial \boldsymbol{w}}$.

Keypoints of The Learning Problem

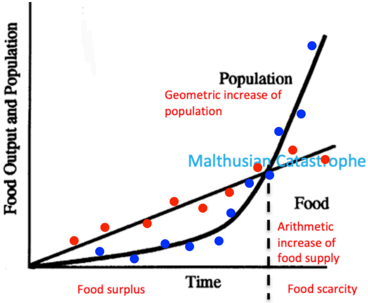
- **Data preprocessing**
  - Suppress the unfairness of features
- **Evaluation**
  - Key criteria
  - Data splitting and cross-validation
- **Training**
  - Model selection

Why Do We Need Data Preprocessing?



- The x here is time (e.g., year).
- Consider a 3rd-order polynomial $y = w_1 + w_2 x + w_3 x^2 + w_4 x^3$

Why Do We Need Data Preprocessing?



- The x here is time (e.g., year).
- Consider a 3rd-order polynomial $y = w_1 + w_2 x + w_3 x^2 + w_4 x^3$
- $x = \mathcal{O}(10^3)$, while $x^d \mathcal{O}(10^{3d})$.
- **Numerical issue**

Data Preprocessing: Normalization

Motivation:

- Make each feature comparable on their ranges.

Data Preprocessing: Normalization

Motivation:

- Make each feature comparable on their ranges.

Principle: Given $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_D]$, for each $\boldsymbol{x}_d$

- $\|\boldsymbol{x}_d\|_2 = 1 \quad \Rightarrow \quad$ Normalization energy
- $\|\boldsymbol{x}_d\|_1 = 1 \quad \Rightarrow \quad$ Normalization absolute sum
- $\|\boldsymbol{x}_d\|_\infty = 1 \quad \Rightarrow \quad \max\{|x_{nd}|\}_{n=1}^{N} = 1$

Data Preprocessing: Shifting and Scaling

Motivation:

- Same range does not mean same statistics.
- Make each feature with zero mean and normalized variance.

Data Preprocessing: Shifting and Scaling

Motivation:

- Same range does not mean same statistics.
- Make each feature with zero mean and normalized variance.

Principle: Given $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D]$, for each $\boldsymbol{x}_d$

- Estimate expectation $\hat{\mu}_d = \frac{1}{N} \sum_{n=1}^{N} x_{nd}$
- Estimate variance $\hat{\sigma}_d = \frac{1}{N-1} \sum_{n=1}^{N} (x_{nd} - \hat{\mu}_d)^2$.
- $\tilde{x}_{nd} = \frac{x_{nd} - \hat{\mu}_d}{\hat{\sigma}_d}$

Data Preprocessing: Shifting and Scaling

Motivation:

- Same range does not mean same statistics.
- Make each feature with zero mean and normalized variance.

Principle: Given $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_D]$, for each $\boldsymbol{x}_d$

- Estimate expectation $\hat{\mu}_d = \frac{1}{N}\sum_{n=1}^{N} x_{nd}$
- Estimate variance $\hat{\sigma}_d = \frac{1}{N-1}\sum_{n=1}^{N}(x_{nd} - \hat{\mu}_d)^2$.
- $\tilde{x}_{nd} = \frac{x_{nd}-\hat{\mu}_d}{\hat{\sigma}_d}$

$$x \sim \mathcal{N}(\mu, \sigma^2) \quad \Rightarrow \quad \frac{x-\mu}{\sigma} \sim \mathcal{N}(0,1) \tag{8}$$

Data Preprocessing: Whitening

Motivation:

- Shifting and scaling assumes uncorrelated features, which is questionable.
- Make each feature with zero mean, normalized variance, and uncorrelated.

Data Preprocessing: Whitening

Motivation:

- Shifting and scaling assumes uncorrelated features, which is questionable.
- Make each feature with zero mean, normalized variance, and uncorrelated.

Principle: Given $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D]$, for each $\boldsymbol{x}_d$

- Estimate expectation $\hat{\mu}_d = \frac{1}{N}\sum_{n=1}^{N} x_{nd}$ for $d = 1, \ldots, D$.
- Estimate covariance matrix

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N-1}(\boldsymbol{X} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^T)^T(\boldsymbol{X} - \mathbf{1}_N\hat{\boldsymbol{\mu}}^T) \in \mathbb{R}^{D\times D} \tag{9}$$

- Whitening: $\tilde{\boldsymbol{X}} = \boldsymbol{X}\widehat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}$.

Evaluation: Loss Functions and Key Criteria

Mean-square error (MSE)

- $|y - \hat{y}|^2$

Evaluation: Loss Functions and Key Criteria

Mean-square error (MSE)

- $|y - \hat{y}|^2$

Mean absolute error (MAE)

- $|y - \hat{y}|$

Evaluation: Loss Functions and Key Criteria

Mean-square error (MSE)

- $|y - \hat{y}|^2$

Mean absolute error (MAE)

- $|y - \hat{y}|$

The loss function actually is designed based on the evaluation measurement.

- MSE $\Rightarrow$ Gaussian noise $\Rightarrow$ ℓ_2-norm as loss function.

Evaluation: Loss Functions and Key Criteria

Mean-square error (MSE)

- $|y - \hat{y}|^2$

Mean absolute error (MAE)

- $|y - \hat{y}|$

The loss function actually is designed based on the evaluation measurement.

- MSE $\Rightarrow$ Gaussian noise $\Rightarrow$ ℓ_2-norm as loss function.
- MAE?

How To Evaluate The Stability of Learning Methods?

- When training data change, will we obtain the model with same/similar parameters?
- When training data change, will we train the model and make it achieve similar performance on the same testing data?

How To Evaluate The Stability of Learning Methods?

- When training data change, will we obtain the model with same/similar parameters?
- When training data change, will we train the model and make it achieve similar performance on the same testing data?

Which one is harder?

Confidence Interval and Bootstrapping

Confidence Interval

- Let X be a random sample from a probability distribution with parameter θ.
- A confidence interval of θ with confidence level α, is an interval with random endpoints $(l(X), u(X))$, such that

$$P_{\theta,\psi}(l(X) < \theta < u(X)) = \alpha, \quad \forall\,(\theta, \psi). \tag{10}$$

Confidence Interval and Bootstrapping

Confidence Interval

- Let X be a random sample from a probability distribution with parameter θ.
- A confidence interval of θ with confidence level α, is an interval with random endpoints $(l(X), u(X))$, such that

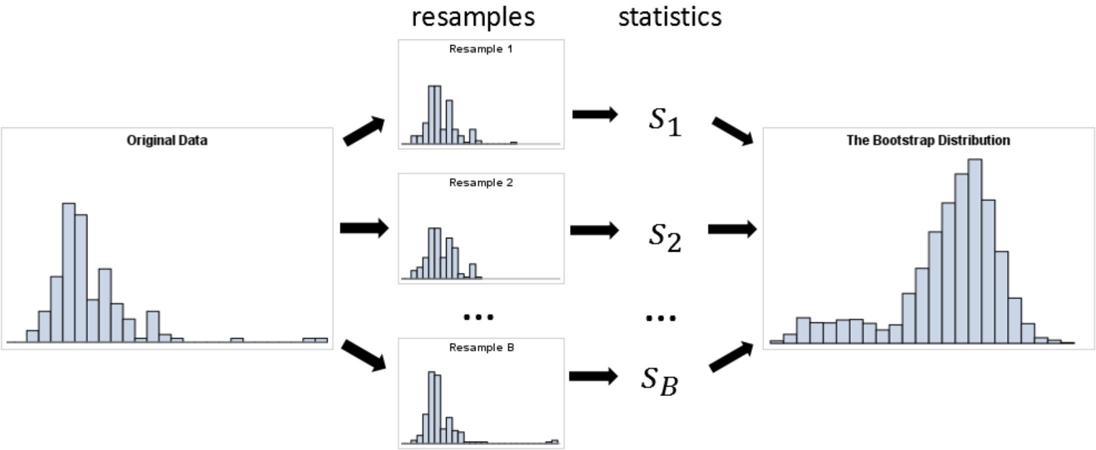
$$P_{\theta,\psi}(l(X) < \theta < u(X)) = \alpha, \quad \forall\,(\theta,\psi). \tag{10}$$

Note:

- The random interval covers the unknown θ with probability α, no matter what the true θ is.
- The true value can be out of the range.

Confidence Interval and Bootstrapping

Bootstrapping has been widely used to estimate confidence interval.



Confidence Interval and Bootstrapping

- Given bootstrapped parameters $\{\theta_n^*\}_{n=1}^N$ derived by bootstrapping
- **Percentile bootstrap:**

$$\left(\theta^*_{(\alpha/2)},\ \theta^*_{(1-\alpha/2)}\right) \tag{11}$$

where $\theta^*_{(1-\alpha/2)}$ denote the $1-\alpha/2$ percentile of the bootstrapped parameters.

Confidence Interval and Bootstrapping

- Given bootstrapped parameters $\{\theta_n^*\}_{n=1}^N$ derived by bootstrapping
- **Percentile bootstrap:**

$$\left(\theta^*_{(\alpha/2)},\ \theta^*_{(1-\alpha/2)}\right) \tag{11}$$

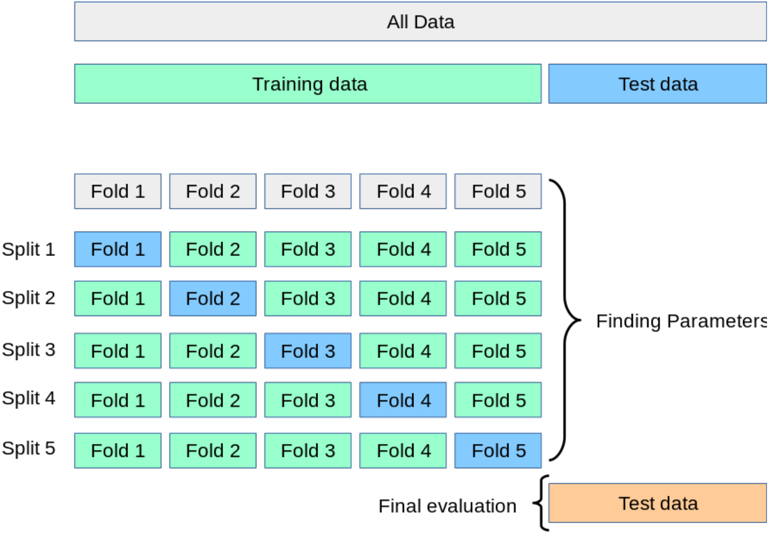
where $\theta^*_{(1-\alpha/2)}$ denote the $1-\alpha/2$ percentile of the bootstrapped parameters.

- **Basic bootstrap:**

$$\left(2\hat{\theta}-\theta^*_{(1-\alpha/2)},\ 2\hat{\theta}-\theta^*_{(\alpha/2)}\right) \tag{12}$$

The difference between Frequentist statistics and Bayesian statistics is not so obvious as they claimed:)

Cross-validation



How To Compare Models and Select The Best?

- Good-of-fitness v.s. simplicity of the model
- Overfitting v.s. Underfitting

Akaike Information Criterion (AIC)

Motivation:

- Estimate the relative amount of information lost by a given model.
- Achieve a trade-off between good-of-fitness and model simplicity.

Akaike Information Criterion (AIC)

Motivation:

- Estimate the relative amount of information lost by a given model.
- Achieve a trade-off between good-of-fitness and model simplicity.

Principle: Suppose that we have a statistical model of some data.

- Let K be the number of model parameters.
- Let $\widehat{L} = \max p(\boldsymbol{X}|\hat{\boldsymbol{\theta}})$ be the maximum likelihood for the model.
- The AIC value of the model:

$$\mathrm{AIC} = 2K - 2\log\widehat{L} \tag{13}$$

Akaike Information Criterion (AIC)

Given M models and their AIC values $\{\mathrm{AIC}_m\}_{m=1}^{M}$

- The relative likelihood of model m:

$$\exp\left(\frac{\mathrm{AIC}_{\min} - \mathrm{AIC}_m}{2}\right) \tag{14}$$

- It is proportional to the probability that the model m minimizes the (estimated) information loss.

Akaike Information Criterion (AIC)

Given M models and their AIC values $\{\mathrm{AIC}_m\}_{m=1}^{M}$

- The relative likelihood of model m:

$$\exp\left(\frac{\mathrm{AIC}_{\min} - \mathrm{AIC}_m}{2}\right) \tag{14}$$

- It is proportional to the probability that the model m minimizes the (estimated) information loss.

Any drawbacks?

Bayesian Information Criterion (BIC)

Motivation:

- BIC also penalizes model complexity, and with larger penalty term.
- Consider the influence of data size.

Bayesian Information Criterion (BIC)

Motivation:

- BIC also penalizes model complexity, and with larger penalty term.
- Consider the influence of data size.

Principle: Suppose that we have a statistical model of some data.

- Let K be the number of model parameters.
- Let N be the number of data points (samples)
- Let $\widehat{L} = \max p(\boldsymbol{X}|\hat{\boldsymbol{\theta}})$ be the maximum likelihood for the model.
- The BIC value of the model:

$$\mathrm{BIC} = K \log N - 2 \log \widehat{L} \tag{15}$$

What is it reasonable?

Bayesian Information Criterion (BIC)

Suppose that $\boldsymbol{\theta}$ are specific parameters of a model $\mathcal{M}$.

- Consider the 2nd-order Taylor expansion of the log-likelihood $\log p(\boldsymbol{X}|\boldsymbol{\theta}, \mathcal{M})$ about the MLE $\hat{\boldsymbol{\theta}}$:

$$\log p(\boldsymbol{X}|\boldsymbol{\theta}, \mathcal{M}) \approx \log \underbrace{p(\boldsymbol{X}|\hat{\boldsymbol{\theta}})}_{\widehat{L}} - \frac{N}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \boldsymbol{I}(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \tag{16}$$

- Fisher Information Matrix:

$$\boldsymbol{I}(\boldsymbol{\theta}) = -\mathbb{E}\left[\frac{\partial^2 \log p(\boldsymbol{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] \tag{17}$$

Bayesian Information Criterion (BIC)

Suppose that $\boldsymbol{\theta}$ are specific parameters of a model $\mathcal{M}$.

- Consider the 2nd-order Taylor expansion of the log-likelihood $\log p(\boldsymbol{X}|\boldsymbol{\theta}, \mathcal{M})$ about the MLE $\hat{\boldsymbol{\theta}}$:

$$\log p(\boldsymbol{X}|\boldsymbol{\theta}, \mathcal{M}) \approx \log \underbrace{p(\boldsymbol{X}|\hat{\boldsymbol{\theta}})}_{\widehat{L}} - \frac{N}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \boldsymbol{I}(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \tag{16}$$

- Fisher Information Matrix:

$$\boldsymbol{I}(\boldsymbol{\theta}) = -\mathbb{E}\left[\frac{\partial^2 \log p(\boldsymbol{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] \tag{17}$$

- Why the 1st order derivation term $\frac{\partial \widehat{L}}{\partial \hat{\boldsymbol{\theta}}}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$ is ignored?

Bayesian Information Criterion (BIC)

- We have

$$p(\boldsymbol{X}|\mathcal{M}) = \int p(\boldsymbol{X}|\boldsymbol{\theta}, \mathcal{M}) p(\boldsymbol{\theta}|\mathcal{M}) \mathrm{d}\boldsymbol{\theta} \approx \left(\frac{2\pi}{N}\right)^{K/2} \widehat{L} \underbrace{|\boldsymbol{I}(\hat{\boldsymbol{\theta}})|^{-1/2} p(\hat{\theta})}_{\mathcal{O}(1) \text{ as } N \to \infty}$$

Bayesian Information Criterion (BIC)

- We have

$$
\begin{aligned}
p(\boldsymbol{X}|\mathcal{M}) &= \int p(\boldsymbol{X}|\boldsymbol{\theta},\mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})\mathrm{d}\boldsymbol{\theta} \approx \left(\frac{2\pi}{N}\right)^{K/2} \widehat{L}\,\underbrace{|\boldsymbol{I}(\hat{\boldsymbol{\theta}})|^{-1/2}p(\hat{\theta})}_{\mathcal{O}(1)\text{ as }N\to\infty} \\
&= \exp\left(\underbrace{\log\widehat{L} - \frac{K}{2}\log N}_{-0.5\mathrm{BIC}} + \mathcal{O}(1)\right)
\end{aligned} \tag{18}
$$

In Summary

- Introduction of linear regression model (Take polynomial regression as an example)
- The keypoints in the whole training and testing pipeline
- Model selection: AIC and BIC

Next...

- Generalized linear regression
- Bias v.s. variance
- Regularization

Homework 1: DDL — March 17, 2022, Midnight

Python Programming

- Lab # 1 (3 Pts, Done)
- Lab # 2 (5 Pts)

Questions for Tech Report (6 Pts, $\leq$ 3 Pages)

- Demonstrate the equivalence of the following four claims (3 Pts):

Theorem

$\boldsymbol{A}$ is of full column rank (Rank$(\boldsymbol{A}) = N$).

$\Leftrightarrow$ *$\boldsymbol{A}$ is injective.*

$\Leftrightarrow$ *$[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$ is linearly-independent.*

$\Leftrightarrow$ *Null$(\boldsymbol{A}) = \{\mathbf{0}\}$.*

- Demonstrate $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1 \leq \sqrt{N}\|\boldsymbol{x}\|_2 \leq N\|\boldsymbol{x}\|_\infty, \forall \boldsymbol{x} \in \mathbb{R}^N$, and provide an illustration of the principle in the case of $N = 2$ (3 Pts).