Lecture 10 Introduction to Machine Learning and Linear Regression

Overview of the whole picture

Possible hierarchies of machine learning concepts:

- **Problems**: Supervised Learning(Regression,Classification), Unsupervised Learning (Dimension Reduction, Clustering), Reinforcement Learning (Not covered in this course)
- **Models**:
  - (Supervised) Linear Regression, Logistic Regression, K-Nearest Neighbor (kNN) Classification/Regression, Decision Tree, Random Forest, Support Vector Machine, Ensemble Method, Neural Network...
  - (Unsupervised) K-means,Hierachical Clustering, Principle Component Analysis, Manifold Learning (MDS, IsoMap, Diffusion Map, tSNE), Auto Encoder...
- **Algorithms**: Gradient Descent, Stochastic Gradient Descent (SGD), Back Propagation (BP),Expectation–Maximization (EM)...

For the same **problem**, there may exist multiple **models** to discribe it. Given the specific **model**, there might be many different **algorithms** to solve it.

Why there is so much diversity? The following two fundamental principles of machine learning may provide theoretical insights.

Bias-Variance Trade-off (https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229): Simple models -- large bias, low variance. Complex models -- low bias, large variance

No Free Lunch Theorem (https://analyticsindiamag.com/what-are-the-no-free-lunch-theorems-in-data-science/#:~:text=Once%20Upon%20A%20Time,that%20they%20brought%20a%20drink): (in plain language) There is no one model that works best for every problem. (more quantitatively) Any two models are equivalent when their performance averaged across all possible problems. --Even true for optimization algorithms (https://en.wikipedia.org/wiki/No_free_lunch_in_search_and_optimization).

Linear Regression

Recall the basic task of **supervised learning**: given the *training dataset* $(x^{(i)}, y^{(i)}), i = 1, 2, \ldots, N$ with $y^{(i)} \in \mathbb{R}^q$ (for simplicity, assume $q = 1$) denotes the *labels*, the supervised learning aims to find a mapping $y \approx \mathbf{f}(x) : \mathbb{R}^p \to \mathbb{R}$ that we can use it to make predictions on the test dataset.

Model Setup

Model assumption 1: Linear Mapping Assumption.

$$y \approx \mathbf{f}(x) = \beta_0 + \beta_1 x_1 + .. + \beta_p x_p = \tilde{x}\beta,$$

$$\tilde{x} = (1, x_1, .., x_p) \in \mathbb{R}^{1\times(p+1)}, \beta = (\beta_0, \beta_1, .., \beta_p)^T \in \mathbb{R}^{(p+1)\times 1}.$$

Here β is called regression coefficients, and β_0 specially referred to intercept.

Using the whole training dataset, we can write as

$$Y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \cdots \\ y^{(N)} \end{pmatrix} \approx \begin{pmatrix} \mathbf{f}(x^{(1)}) \\ \mathbf{f}(x^{(2)}) \\ \cdots \\ \mathbf{f}(x^{(N)}) \end{pmatrix} = \begin{pmatrix} \tilde{x}^{(1)}\beta \\ \tilde{x}^{(2)}\beta \\ \cdots \\ \tilde{x}^{(N)}\beta \end{pmatrix} = \begin{pmatrix} \tilde{x}^{(1)} \\ \tilde{x}^{(2)} \\ \cdots \\ \tilde{x}^{(N)} \end{pmatrix} \beta = \tilde{X}\beta,$$

where

$$\tilde{X} = \begin{pmatrix} 1 & \tilde{x}_1^{(1)} & \cdots & \tilde{x}_p^{(1)} \\ 1 & \tilde{x}_1^{(2)} & \cdots & \tilde{x}_p^{(2)} \\ \cdots & & & \\ 1 & \tilde{x}_1^{(N)} & \cdots & \tilde{x}_p^{(N)} \end{pmatrix}$$

is also called the augmented data matrix.

Model assumption 2: Gaussian Residual Assumption (L^2 loss assumption)

$$y^{(i)} = \tilde{x}^{(i)}\beta + \epsilon^{(i)}, i = 1, 2, .., N$$

The residuals or errors $\epsilon^{(i)}$ are **assumed** as independent Gaussian random variables with identical distribution $\mathcal{N}(0, \sigma^2)$ which has mean 0 and standard deviation σ.

From the density function of Gaussian distribution, the prabability to observe $\epsilon^{(i)}$ within the small interval $[z, z + \Delta z]$ is roughly

$$\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{z^2}{2\sigma^2})\Delta z.$$

From the data, we know indeed $z = y^{(i)} - \tilde{x}^{(i)}\beta$. Therefore, the probability density (likelihood) to observe $(x^{(i)}, y^{(i)})$ is roughly

$$l(x^{(i)}, y^{(i)}, \beta) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y^{(i)} - \tilde{x}^{(i)}\beta)^2}{2\sigma^2}).$$

Using the *independence* assumption, the overall likelihood to observe the data is

$$\mathcal{L}(\beta; x^{(i)}, y^{(i)}, 1 \leq i \leq N) = \prod_{i=1}^{N} l(x^{(i)}, y^{(i)}, \beta)$$

The famous **Maximum Likelihood Estimation** theory in statistics **assumes** that we aim to find the unknown parameter β that maximizes the $\mathcal{L}(\beta; x^{(i)}, y^{(i)}, 1 \leq i \leq N)$ by treating $x^{(i)}$ and $y^{(i)}$ as fixed numbers.

Equivalently, as the function of β, we can maximize $\ln \mathcal{L}(\beta; x^{(i)}, y^{(i)}, 1 \leq i \leq N) = \sum_{i=1}^{N} \ln l(x^{(i)}, y^{(i)}, \beta)$.

By removing the constants, we finally arrives at the **minimization** problem of L^2 loss function

$$L(\beta) = \sum_{i=1}^{N}(y^{(i)} - \tilde{x}^{(i)}\beta)^2 = ||Y - \tilde{X}\beta||_2^2.$$

The optimal parameter

$$\hat{\beta} = \text{argmin} L(\beta)$$

is also called the ordinary least square (OLS) estimator in statistics community.