

MOAI Theoretical Part

Jimmy Choi, May, 2025

Part I. Machine Learning / Scikit-learn

1. Supervised learning: Linear regression 線性回歸

- (1) 主要目的：尋找最佳擬合數據的直線（一元簡單線性回歸）或者超平面（多元線性回歸）
- (2) 模型表示： $y = wx + b$ （簡單線性回歸）或者 $y = \sum_{i=1}^n w_i x_i + b$ （多元線性回歸）
- (3) Loss function 損失函數：

- ① Mean Squared Error, MSE

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

- ② Root Mean Squared Error, RMSE

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

- ③ Mean Absolute Error, MAE

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

- ④ R-squared, R^2

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

(4) Optimization Techniques

- ① Least Squares Method 最小二乘法

- 1) 使用 MSE 作為 loss function，完全不需要進行迭代，只需要一次計算就可以得到最優的權重向量 w

- 2) 計算方法：

- a) 將公式用矩陣形式表示，即 $y = Xw + \epsilon$ ，其中 y 表示真實值向量， X 表示輸入的特徵矩陣， w 表示權重矩陣， ϵ 表示所有樣品的誤差向量

- b) 將損失函數用矩陣表示，即 $J(w) = (y - Xw)^T (y - Xw)$

- c) 為了最小化 $J(w)$ ，需要找到使 $J(w)$ 對 w 的導數為 0 的 w ，即

$$\frac{\partial J(w)}{\partial w} = -2X^T(y - Xw) = 0$$

，解得 $w = (X^T X)^{-1} X^T y$ ，就是最小二乘法的解析解 (Analytical Solution)，也即閉合形式解 (Closed-form Solution)

- ② Gradient Descent 梯度下降：見 Part II

2. Supervised Learning: Logistic regression 邏輯回歸

- (1) 主要思想：用於分類問題，通過 Sigmoid 函數將線性回歸的輸出映射到 0 和 1 之間，表示屬於某個類別的概率。

- (2) 二分類

- ① 使用 Sigmoid 函數作為激活函數： $\sigma(z) = \frac{1}{1+e^{-z}}$ ，其中 $z = w^T x + b$

- ② Decision Threshold 決策閾值：通常設定為 0.5，大於 0.5 屬於類別 1，小於 0.5 屬

於類別 0

③ Loss function 損失函數: Binary Cross-Entropy 二元交叉熵

④ 優化方法: 梯度下降

(3) 多分類:

① 使用 softmax 函數作為激活函數, 將 k 個綫性組合映射到 k 個概率值, 這些概率值

$$\text{softmax}(z)_k = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}$$
 的和為 1, 選擇概率最高的類別作為預測結果

② Loss function 損失函數: Categorical Cross-Entropy 多類別交叉熵

③ 優化方法: 梯度下降

3. Supervised Learning: KNN (K nearest neighbor algorithm) K 近鄰算法

(1) 主要思想: 找到距離目標樣本最近的 K 個鄰居, 然後根據這些鄰居的類別來預測目標樣本的類別

(2) 算法步驟:

① 選擇 k 的大小: 過小發生 overfitting, 過大發生 underfitting

② 計算距離: Euclidean Distance, Manhattan Distance, Minkowski Distance, Cosine Similarity (一般使用前兩種方法)

③ 尋找 k 個最近的鄰居, 選擇頻率最高的類別分類

4. Supervised Learning: Decision Tree 決策樹

(1) 主要思想: 通過一系列的決策規則, 將數據集逐步劃分成更小的子集, 直到每個子集中的樣本都屬於同一類別 (或具有相似的數值)。其中, 每個節點表示一個特徵, 每個分支表示該特徵的一個取值, 每個葉節點表示一個類別 (或一個數值)。

(2) 算法步驟:

① 計算數據集的 Gini impurity: $Gini(D) = 1 - \sum_{k=1}^K p_k^2$, 其中 K 為類別數, p_k 是數據集中第 k 類樣本的所占比例

② 計算每個特徵劃分後的 weighted Gini impurity: $Gini_{weighted}(D, A) = \sum_{v=1}^V \frac{|D^v|}{|D|} Gini(D^v)$, A 表示一種特徵, 有 V 種可能的取值, 將 D 劃分成 V 個子集, 計算每個子集的 weighted Gini impurity 和。

③ 就計算每個特徵的信息增益: $Gain_{Gini}(D, A) = Gini(D) - Gini_{weighted}(D, A)$, 選擇信息增益最大的為當前節點的劃分特徵

④ 重複上述步驟, 直到滿足下述條件之一:

- 1) 當前節點的所有數據屬於同一類別;
- 2) 沒有更多的特徵可用於分割;
- 3) 達到預定義的樹的最大深度;
- 4) 當前節點的數據量小於預定義的閾值。

5. Supervised Learning: Random Forest 隨機森林

(1) 主要思想: 構建多個決策樹, 組合他們的預測。“隨機”體現在隨機選擇樣本和隨機選擇特徵 (並不選擇所有特徵) 上。

(2) 算法步驟:

① Bootstrap 抽樣: 從原始訓練集中隨機有放回地抽取 N 個樣本, 生成一個新的訓練集。重複此步驟 T 次, 生成 T 個訓練集。

- ② 訓練決策樹：隨機選擇 m 個特徵，通常 $m=M^{(1/2)}$ ， M 是原始特徵的數量，再依照決策樹的方法訓練決策樹
- ③ 預測：classification 取最多預測結果，regression 取平均值

6. Unsupervised Learning: K-means

- (1) 主要思想：將數據集無監督地分成 k 個不同的 cluster，使得每個不同的樣本都屬於離它最近的 centroid 對應的 cluster。
- (2) 算法步驟：
 - ① 選擇一個 k ，並且隨機選擇 k 個樣本/ k 個點作為初始的 centroid。
 - ② 對於每個樣本，計算與 k 個 centroid 的距離，並且分配到最近的 centroid 對應的 cluster。
 - ③ 重新計算每一個 cluster 的 centroid。
 - ④ 重複上述步驟，直到滿足下述條件之一：
 - 1) Centroid 不再變化；
 - 2) Cluster 的分配不再變化；
 - 3) 達到預定義的最大迭代次數。
- (3) 損失函數：Within-Cluster Sum of Squares, WCSS（簇內平方和）

$$WCSS(k) = \sum_{j=1}^k \sum_{\mathbf{x}_i \in \text{cluster } j} \|\mathbf{x}_i - \bar{\mathbf{x}}_j\|^2,$$

where $\bar{\mathbf{x}}_j$ is the sample mean in cluster j

其中 \mathbf{x}_j 就是第 j 個 cluster 的 centroid。用

Euclidean distance。

- (4) K 值的選擇：(Elbow Method) 繪製 k 與 WCSS 的關係圖，選擇 WCSS 下降變緩的“肘部”。

7. Unsupervised Learning: PCA (Principle Component Analysis) 主成分分析

8. Model Evaluation 模型分析指標

9. Overfitting & Underfitting 過擬合和欠擬合

Part II. Deep Learning and Neural Network / PyTorch

- 1. Gradient Descent 梯度下降
- 2. BP (Backpropagation) 反向傳播
- 3. Activation function 激活函數
- 4. Loss function 損失函數
- 5. MLP (Multilayer perceptron) 多層感知機
- 6. SGD (Stochastic Gradient Descent) 隨機梯度下降
- 7. Adam
- 8. 學習率

Part III. Computer Vision / PyTorch

- 1. CNN (Convolution Neural Network) 卷積神經網絡
- 2. Image Classification 圖像分類

Part IV. NLP (Natural Language Processing) 自然語言處理

- 1. Transformer and Attention