Machine Learning

From the Perspective of Statistics

Huyi Chen

Latest Update: September 7, 2019

Chapter 1

Introduction

1.1 Terminology and Framework

- **Data generating process**: X is a $p-$dimensional random vector with joint distribution $\mathrm{P}(x)$ and $Y = f(X)$ is a random variable.
  - Input vector: $X \in D \subset \mathbb{R}^p$.
  - Output vector: $Y \in G \subset \mathbb{R}$.
  - Data: Given the sample $S = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_{N+M}, Y_{N+M})\}$ where $(X_1, X_2, \cdots, X_{N+M})$ follows the distribution $\mathrm{P}(x)$, both the training data $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ and text data $Q = \{(x_{N+1}, y_{N+1}), (x_{N+2}, y_{N+2}), \cdots, (x_{N+M}, y_{N+M})\}$ consist of the realization values of the sample.

  Considering that in general Y cannot be totally determined by X, the data generating process of the learning model can be extended: (X, Y) is a $(p+1)-$dimensional random vector with joint distribution $\mathrm{P}(x, y)$.
  - Input vector: $X \in D \subset \mathbb{R}^p$.
  - Output vector: $Y \in G \subset \mathbb{R}$.
  - Data: Given the sample $S = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_{N+M}, Y_{N+M})\}$ following the distribution $\mathrm{P}(x, y)$, both the training data T and text data Q consist of the realization values of the sample.
- **Objective**: Given the sample S, estimate a decision function $\hat{f} \in \mathscr{F}$ trying to minimize the expected prediction error (EPE)
$$\mathrm{E}[L(Y, f(X))].$$
  - Decision function: $f : \mathbb{R}^p \supset D \to \mathbb{R}, x \mapsto f(x)$ serves to produce the prediction of Y, provided a specified value x of X.
  - Loss function: $L(Y, f(X))$ normally has the form of
$$L_2 = (Y - f(X))^2 \text{ or } L_1 = |Y - f(X)| \text{ or } L_I = 1_{Y \neq f(X)}.$$
  - Hypothesis space: $\mathscr{F}$ is a collection of all potential decision functions f to be selected. In some cases, we suppose that f as a candidate can be specified by several parameters. Thus $\mathscr{F} = \{f_\theta | Y = f_\theta(X),\ \theta \in \mathbb{R}^n\}$ can be described by the parametric space $\Theta = \{\theta : Y = f_\theta(X),\ \theta \in \mathbb{R}^n\}$.

  If the probability distribution of (X, Y) was known to us, it might succeed to find the optimal solution $\tilde{f}$ of the following minimization problem
$$\min_{f \in \mathscr{F}} \mathrm{E}[L(Y, f(X))]$$

in virtue of the optimization theory. In other words, what need to be settled is purely an optimization problem rather than a statistical problem. Unfortunately, the specific distribution of (X, Y) is inaccessible in reality. Thus we can only exploit the training data to estimate an acceptable decision function $\hat{f}$, with acceptance of the fact that $\hat{f}$ in general has a greater EPE than $\tilde{f}$.

- **Optimization strategies**: Since the EPE minimization problem is ill-formed, we have developed two major strategies to produce a tractable optimization problem.
  - Empirical risk minimization: according to the law of large numbers, if some regular conditions[1] hold, then as $N \to \infty$, the solution of the following minimization problem will converge to the theoretically optimal solution $\tilde{f}$.

$$\min_{f \in \mathscr{F}} \frac{1}{N} \sum_{i=1}^{N} L(Y_i, f(X_i))$$

  - Structural risk minimization: In reality, the size of training set N is limited and accordingly the method of empirical risk minimization may not generate a function $\hat{f}$ which is sufficiently close to $\tilde{f}$. Later we will elaborate this phenomenon named "overfitting". However, if we add a regularizer or penalty term $\lambda J(f)$ to penalize the complexity of the decision function f as follows

$$\min_{f \in \mathscr{F}} \frac{1}{N} \sum_{i=1}^{N} L(Y_i, f(X_i)) + \lambda J(f),$$

    it is possible to lead to a better result.

1.2 The Common Form of Optimal Decision Function

1.2.1 Loss function for quantitative output variables: squared error loss

Let $Y \in \mathbb{R}$ be a quantitative variable. And we take the most common and convenient loss function, squared error loss

$$L(Y, f(X)) = (Y - f(X))^2.$$

In this case, the problem of minimizing the expected prediction error becomes

$$\min_{f \in \mathscr{F}} \mathrm{E}[(Y - f(X))^2] = \min_{f \in \mathscr{F}} \int (y - f(x))^2 \, \mathrm{dP}(x, y).$$

Note that

$$\mathrm{E}[(Y - f(X))^2] = \mathrm{E}[\mathrm{E}[(Y - f(X))^2 | X]] = \int \mathrm{E}[(Y - f(x))^2 | X = x] \, \mathrm{dP}(x).$$

It suffices to minimize EPE pointwise, that is,

$$\begin{aligned}
&\min_{f(x) \in \mathbb{R}} \mathrm{E}[(Y - f(x))^2 | X = x] \\
&= \min_{c \in \mathbb{R}} \mathrm{E}[(Y - c)^2 | X = x] \\
&= \min_{c \in \mathbb{R}} \mathrm{Var}[Y - c | X = x] + (\mathrm{E}[(Y - c) | X = x])^2 \\
&= \min_{c \in \mathbb{R}} \mathrm{Var}[Y | X = x] + (\mathrm{E}[Y | X = x] - c)^2.
\end{aligned}$$

[1] For example, the uniform law of large numbers can be applied here. The details are consigned to the appendix

We see the optimal solution is

$$\tilde{f}(x) = \mathrm{E}[Y \mid X = x],$$

Thus the best prediction of Y at any point $X = x$ is the conditional expectation, when best is measured by average squared error.
Later in this book we are to develop effective methods to estimate the conditional expectation $\mathrm{E}[Y \mid X = x]$.

1.2.2 Loss function for categorical output variable: 0-1 indicator

Assume that $Y \in G$ is a categorical variable and that the set of possible classes is $G = \{G_1, G_2, \cdots, G_K\}$. This time the 0–1 loss function

$$L(Y, f(X)) = 1_{Y \neq f(X)} = \begin{cases} 0, & Y = f(X), \\ 1, & Y \neq f(X), \end{cases}$$

is adopted for simplification. Likewise it suffices to minimize EPE pointwise.

$$\begin{aligned} &\min_{f(x) \in G} \mathrm{E}[1_{Y \neq f(x)} \mid X = x] \\ &= \min_{g \in G} \mathrm{E}[1 - 1_{Y=g} \mid X = x] \\ &= \min_{g \in G} 1 - \mathrm{P}(Y = g \mid X = x) \end{aligned}$$

And the optimal solution is

$$\tilde{f}(x) = \max_{g \in G} \mathrm{P}(Y = g \mid X = x)$$

1.3 Generalization Error Bound

As is mentioned before, we always hope that $\hat{f}$ has as small EPE as possible. EPE is also called the generalization error, indicating it gauges the performance of the selected function $\hat{f}$ in a general sense. We also emphasize that without knowing the probability distribution of (X, Y), there is no way to calculate the expectation of $L(Y, f(X))$. In practice, it is typical to analyze the upper bound of generalization error to describe the generalization ability of the selected function $\hat{f}$.

In the case of binary classifications, deriving the generalization error bound is relatively simple. Let's denote the generalization error and the empirical error by

$$R(f) = \mathrm{E}[L(Y, f(X))]$$

and

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^{N} L\left(Y_i, f\left(X_i\right)\right),$$

respectively. Actually, We have the following result.

Theorem 1.3.1 Suppose that the training set $T = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_N, Y_N)\}$ with $X_i \in \mathbb{R}^p$ and $Y_i \in \{+1, -1\}$ is independently generated from the distribution $\mathrm{P}(x, y)$. If the hypothesis space is a finite set $\mathscr{F} = \{f_1, f_2, \cdots, f_d\}$ and the 0–1 loss function is taken, then for any $f \in \mathscr{F}$, with a probability of not less than $1 - \delta$ we have

$$R(f) \leq \hat{R}(f) + \varepsilon(d, N, \delta)$$

where

$$\varepsilon(d, N, \delta) = \sqrt{\frac{1}{2N}\left(\log d + \log \frac{1}{\delta}\right)}.$$

Proof. The Hoeffding's inequality states that if $X_1, \cdots, X_n$ are independent random variables bounded by the interval $[0,1]$: $0 \le X_i \le 1$, then $\overline{X} = \sum_{i=1}^{n} X_i/n$ satisfies

$$\mathrm{P}(\mathrm{E}[\overline{X}] - \overline{X} \ge t) \le e^{-2nt^2}.$$

for all $t > 0$. Applying this inequality we get

$$\mathrm{P}(R(f_i) - \hat{R}(f_i) \ge t) \le e^{-2Nt^2} \quad (i = 1, 2, \cdots, d).$$

Let A_i be the event which refers to $R(f_i) - \hat{R}(f_i) \ge t$. We have

$$\mathrm{P}\left(\bigcup_{i=1}^{d} A_i\right) \le \sum_{i=1}^{d} \mathrm{P}(A_i) \le de^{-2Nt^2} \implies \mathrm{P}\left(\bigcap_{i=1}^{d} A_i^c\right) \ge 1 - de^{-2Nt^2}.$$

In other words, for any $f \in \mathscr{F}$, with a probability of not less than $1 - de^{-2Nt^2}$ we have

$$R(f) - \hat{R}(f) < t \quad \text{or} \quad R(f) < \hat{R}(f) + t.$$

Take

$$t = \varepsilon(d, N, \delta) = \sqrt{\frac{1}{2N}\left(\log d + \log\frac{1}{\delta}\right)}$$

and we shows that for any $f \in \mathscr{F}$, with a probability of not less than $1 - \delta$ we have

$$R(f) \le \hat{R}(f) + \varepsilon(d, N, \delta).$$

□

The generalization error bound $\hat{R}(f) + \varepsilon(d, N, \delta)$ helps up understand what is "overfitting" and why we can use structural risk minimization to get over it.

Assume N is relatively small and

$$\hat{f} = \arg\min_{f \in \mathscr{F}} \hat{R}(f).$$

By increasing d endlessly, we can extend the hypothesis space $\mathscr{F}$ and consequently reduce or at least maintain the empirical error $\hat{R}(\hat{f})$ all along. However, the cost it brings is a greater $\varepsilon(d, N, \delta)$, since $\varepsilon(d, N, \delta)$ is strictly increasing in d. Noticing $\hat{R}(\hat{f})$ cannot be less than 0 and $\varepsilon(d, N, \delta) \to \infty$ as $d \to \infty$, we can assert when d is sufficiently large, generalization error bound will be also increasing in d. As a result, although we can find a function $\hat{f}$ with a perfect performance on the training set, the generalization ability of $\hat{f}$ can be very poor. That is to say, $\hat{f}$ is likely to perform badly on the sample out of the training set. The term "overfitting" exactly refers to such a case in which the selected function $\hat{f}$ has a quite small empirical error along with a tremendous generalization error.

Now it is clear to see why a penalty term is introduced into the structural risk minimization. It takes the size of hypothesis space $\mathscr{F}$ or equivalently the complexity of candidate functions into consideration. Thus when N is limited, this method promisingly leads to a smaller generalization error.

Chapter 2

Linear Model

2.1 Finite Sample Linear Model

2.1.1 Statistic model setup

Linear model supposes the data generating process is

$$Y = X^T\beta + \varepsilon,$$

where $X = (1, X^{(1)}, \cdots, X^{(p)}) \in \mathbb{R}^{p+1}$, $Y \in \mathbb{R}$, $\beta \in \mathbb{R}^{p+1}$ is an unknown parameter and ε is an error term which cannot be directly observed. Without loss of generality, we can always assume that $\mathrm{E}[\varepsilon|\, X] = 0$. Given a finite sample $(\mathbf{X}, \mathbf{y})$ of size n, the linear Model indicates

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon,$$

where

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \cdots & x_{Np} \end{pmatrix} = \begin{pmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_N^T \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}.$$

2.1.2 Ordinary least square estimate

If we take squared error loss, then the best prediction of Y is

$$\tilde{f}(X) = \mathrm{E}[Y|\, X] = X'\beta.$$

We can use least square method to estimate the parameter β in the linear model, by minimizing the residual sum-of-squares

$$\hat{\beta} = \arg\min_{\theta \in \mathbb{R}^{p+1}} RSS(\theta) = \arg\min_{\theta \in \mathbb{R}^{p+1}} \sum_{i=1}^{N} (Y_i - X_i^T\theta)^2.$$

If $\mathbf{X}$ is of full column rank, the optimization problem has a unique solution

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

The statistic properties of the OLS estimator $\hat{\beta}$ is remarkable. According to Gauss-Markov theorem, Under Assumptions 1.1-1.4

1.1 linearity: $Y_i = X_i^T \beta + \varepsilon_i$, $(i = 1, 2, \cdots, N)$,

1.2 strict exogeneity: $\mathrm{E}[\varepsilon | \mathbf{X}] = 0$,

1.3 no multicollinearity: $\mathrm{P}(\mathrm{rank}(\mathbf{X}) = p + 1) = 1$,

1.4 spherical error variance: $\mathrm{Var}[\varepsilon | \mathbf{X}] = \sigma^2 > 0$,

$\hat{\beta}$ is the best linear unbiased estimator(BLUE). That is, for any unbiased estimator $\hat{\theta}$ that is linear in $\mathbf{Y}$,

$$\mathrm{Var}[\hat{\theta} | \mathbf{X}] \geq \mathrm{Var}[\hat{\beta} | \mathbf{X}]$$

in the matrix sense.

Chapter 3

SVM

3.1 linear SVM

$$\max_{\beta,\beta_0,\|\beta\|=1} M$$
$$\text{subject to } y_i\left(x_i^T\beta+\beta_0\right)\geq M, i=1,\ldots,N$$

Chapter 4

KNN

4.1 Nearest-neighbor methods

Nearest-neighbor methods use those observations in the training set T closest in input space to x to estimate the aforementioned conditional expectation $\mathrm{E}[Y|\ X = x]$. Specifically, the k-nearest neighbor fit for $\hat{Y}$ is defined as follows:

- Quantitative output
$$\hat{Y} = \hat{f}(x) = \frac{1}{k} \sum_{i:x_i \in N_k(x)} y_i,$$
- Categorical output
$$\hat{Y} = \hat{f}(x) = \arg\max_{g \in G} \sum_{i:x_i \in N_k(x)} 1_{y_i = g},$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample. Closeness implies a metric, which for the moment we assume is Euclidean distance

$$\|x_1 - x_2\|_2 = (x_1 - x_2)^2.$$

.

Chapter 5

Decision Tree

Tree-based methods partition the feature space into a set of rectangles, and then fit a simple model (like a constant) in each one. We first describe a popular tree-based method called CART (classification and regression tree), and later contrast it with C4.5, a major competitor. CART requires a specific hypothesis space $\mathscr{F}$ which consists of functions of such forms:

$$f(x) = \sum_{m=1}^{M} c_m 1[x \in R_m],$$

where p-dimensional rectangular regions $R_1, R_2, \cdots, R_M$ constitute a recursive partition of the feature space $\mathbb{R}^p$. That is, if a partition $P_J = (R_1, R_2, \cdots, R_J)$ has been obtained, we just cut some rectangular region $R_m (1 \le m \le J)$ into two parts by a $p-1$-dimensional hyperplane $x^{(i)} = \ell_J (1 \le i \le p)$ to produce a new partition $P_{J+1} = (R'_1, R'_2, \cdots, R'_{J+1})$

5.1 Regression Trees

Assume the output Y is a continuous variable. Then it leads to the concept of regression trees. Conventionally we choose the squared error loss and accordingly obtain the minimization problem

$$\begin{aligned}
\min_{f\in\mathscr{F}} \mathrm{E}[(Y - f(X))^2] &= \min_{f\in\mathscr{F}} \mathrm{E}\left[\left(\sum_{m=1}^{M}(Y-c_m)1_{X\in R_m}\right)^2\right] \\
&= \min_{f\in\mathscr{F}} \mathrm{E}\left[\sum_{m=1}^{M}(Y-c_m)^2 1_{X\in R_m}\right] \\
&= \min_{f\in\mathscr{F}} \int \mathrm{E}\left[\sum_{m=1}^{M}(Y-c_m)^2 1_{x\in R_m}\middle| X = x\right]\mathrm{dP}(x) \\
&= \min_{f\in\mathscr{F}} \sum_{m=1}^{M}\int_{R_m} \mathrm{E}\big[(Y-c_m)^2\big| X = x\big]\mathrm{dP}(x).
\end{aligned}$$

It is easy to see whichever partition is specified, the optimal c_m remains

$$\tilde{c}_m = \mathrm{E}[Y \mid X \in R_m].$$

Now let's consider the empirical risk minimization problem

$$\min_{f\in\mathscr{F}} \frac{1}{N}\sum_{i=1}^{N} L(Y_i, f(X_i)) = \min_{f\in\mathscr{F}} \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{M}(Y_i - c_m)^2 1_{X_i\in R_m}.$$

Thus the proper estimate of $\hat{c}_m$ is just the average of Y_i in region R_m

$$\hat{c}_m = \text{ave}(Y_i | X_i \in R_m).$$

However, finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible. Hence we proceed with a greedy algorithm. Starting with all of the data, consider a splitting variable indexed by $r \in \{1, 2, \cdots, p\}$ and split point $s \in \mathbb{R}$, and define the pair of half-planes

$$R_1(r,s) = \left\{X \in \mathbb{R}^p | X^{(r)} \leq s\right\} \text{ and } R_2(r,s) = \left\{X \in \mathbb{R}^p | X^{(r)} > s\right\}.$$

Then we seek the splitting variable r and split point s that solve

$$\begin{aligned} &\min_{r,s} \left[\min_{c_1} \sum_{X_i \in R_1(r,s)} (Y_i - c_1)^2 + \min_{c_2} \sum_{X_i \in R_2(r,s)} (Y_i - c_2)^2 \right] \\ = &\min_{r,s} \sum_{X_i \in R_1(r,s)} (Y_i - \hat{c}_1)^2 + \sum_{X_i \in R_2(r,s)} (Y_i - \hat{c}_2)^2. \end{aligned}$$

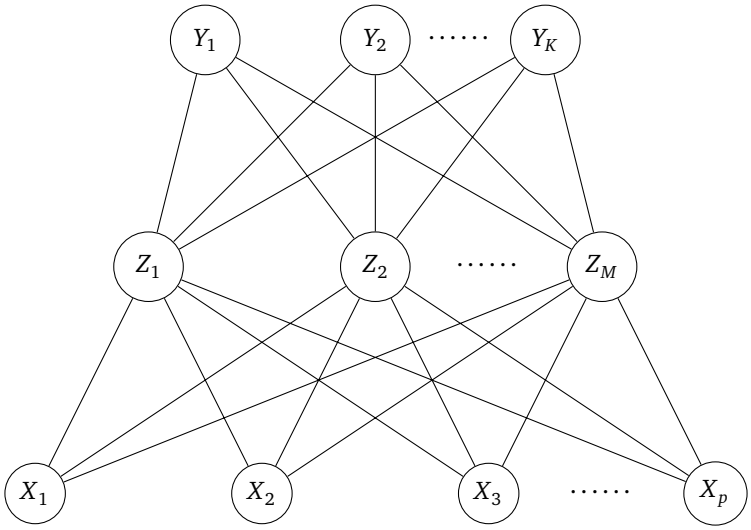
Having found the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions. Then this process is repeated on all of the resulting regions.

Clearly a very large tree might overfit the data until some minimum node size (say 5) is reached.

Chapter 6

Neural Networks

6.1 Single hidden layer back-propagation network



$$Z_m = \sigma\left(\alpha_{0m} + \alpha_m^T X\right), m = 1, \ldots, M$$
$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, \ldots, K$$
$$f_k(X) = g_k(T), k = 1, \ldots, K$$

Appendix

1. Hoeffding's inequality