Reading Summary for The Elements of Statistical Learning Section 2.4 and 2.6

Yufeng Yuan(Arthur)

In section 2.4, frameworks for models with quantitative outputs and categorical outputs are given. In these frameworks, generally speaking, $X \in R^p$ is the random input vector and $Y \in R$ is the random output scalar value, L is the loss function and f is the function to learning.

In models with quantitative outputs, *squared error loss* is used as the loss function for simplicity. The *expected prediction error* can be written as $EPE(f) = E(Y - f(X))^2$ and the solution can be obtained by minimizing the previous formula: $f(x) = E(Y|X = x)$. This conditional expectation is also known as the regression function. Two methods discussed that can be fitted into this framework are nearest-neighbors and linear regression. Nearest-neighbor methods can be formalized as $\hat{f}(x) = Ave(y_i|x_i \in N_k(x))$, where Ave denotes average and $N_k(x)$ is the neighborhood containing the k points in the dataset closest to x. Linear models $f(x) = x^T\beta$, with β as the parameters, can be theoretically solved as $\beta = [E(XX^T)]^{-1}E(XY)$.

Things to be noted about these two methods are the approximation and assumptions behind them. They all use the sample average to replace the expectation and nearest-neighbors conditions on certain regions instead of a specific point. Moreover, these two methods assume that $f(x)$ can be well approximated by a global linear function and a locally constant function respectively.

In models with categorical outputs G, the framework is similar except now the loss function L should be matrix-shaped. The error can be written as $EPE = E[L(G, \hat{G}(X))]$ where $\hat{G}$ is the estimate. If 0-1 loss function is used for simplicity, the solution can be simplified as $\hat{G}(X) = \arg\max_{g \in G} Pr(g|X = x)$, which is known as *Bayes classifier*. Though it seems similar to nearest-neighbors, Bayes classifiers condition on a point instead a region.

In section 2.6, an ideal model where our data is generated from is proposed as $Y = f(X) + \epsilon$ where ϵ is the random error with $E(\epsilon) = 0$. In the setting of supervised learning, a training set $T = (x_i, y_i), i = 1, ..., N$ is used as a teacher to learn f. In terms of function approximation, data pairs x_i, y_i are viewed as points in $(p+1)$-dimensional Euclidean space where x_i p-dimensional and y_i is 1-dimensional, and to learn the $\hat{f}$, different kinds of models can be applied such as linear basis expansions and neural networks.

In the discussion above, least squares is widely used, however, a more general approach is *maximum likelihood estimation*. The idea behind it is to maximize the log-probability of observed random sample $y_i, i = 1, ..., N$ from density $Pr_\theta(y)$ with parameters θ and it can be written as $L(\theta) = \sum_{i=1}^{N} logPr_\theta(y_i)$. In this case, the most reasonable θ would make the probability of the observed samples largest. Least squares for the additive error model $Y = f_\theta(X) + \epsilon$, with $\epsilon \sim N(0, \sigma^2)$, is equivalent to maximum likelihood using the conditional likelihood $Pr(Y|X, \theta) = N(f_\theta(X), \sigma^2)$.