Lecture 3: Statistical Decision Theory (Part II)

Hao Helen Zhang

Fall, 2020
Outline of This Note

- Part I: Statistics Decision Theory (Classical Statistical Perspective - “Estimation”)
  - loss and risk
  - MSE and bias-variance tradeoff
  - Bayes risk and minimax risk

- Part II: Learning Theory for Supervised Learning (Machine Learning Perspective - “Prediction”)
  - optimal learner
  - empirical risk minimization
  - restricted estimators
Math Background

Linear Algebra
- matrix (positive, symmetric, invertible), vector, norms
- linear systems, condition number

Functional analysis
- linear and Euclidean spaces, scalar product, orthogonality
- orthonormal bases, norms, semi-norms, Cauchy sequences
- Hilbert spaces, function spaces, complete spaces, linear functional, Riesz representation theorem
- convex functions, functional calculus

Probability Theory
- random variables, random vectors
- law of large numbers, convergence, concentration inequalities
Any supervised learning problem has three components:

- Input vector $\mathbf{X} \in \mathbb{R}^d$.
- Output $Y$, either discrete or continuous valued.
- There is unknown probability distribution on the product space $\mathcal{X} \times \mathcal{Y}$.

$$(\mathbf{X}, Y) \sim P(\mathbf{X}, Y).$$

The training set $\{(x_1, y_1), \ldots, (x_n, y_n)\} = \{z_1, \ldots, z_n\}$.

The training set $(x_1, y_1), \ldots, (x_n, y_n)$ are generated from $P(x, y)$, so they carry information about $P(x, y)$. 

Hao Helen Zhang Lecture 3: Statistical Decision Theory (Part II) 4 / 1
The goal is to estimate the underlying (unknown) relationship between $X$ and $Y$, $f : X \rightarrow Y$, for future prediction.

- For regression, $f : \mathbb{R}^d \rightarrow \mathbb{R}$
- For $K$-class classification, $f : \mathbb{R}^d \rightarrow \{1, \cdots, K\}$
Learning Loss Function

Similar to the learning theory, we use a learning loss function $L$ to measure the discrepancy $Y$ and $f(X)$ to penalize the errors for predicting $Y$. Examples:

- squared error loss (used in mean regression)
  \[
  L(Y, f(X)) = (Y - f(X))^2
  \]

- absolute error loss (used in median regression)
  \[
  L(Y, f(X)) = |Y - f(X)|
  \]

- 0-1 loss function (used in binary classification)
  \[
  L(Y, f(X)) = I(Y \neq f(X))
  \]
The risk of $f$ is
\[ R(f) = E_{X,Y} L(Y, f(X)) = \int L(y, f(x)) dP(x, y) \]

In machine learning, $R(f)$ is called the expected loss, or the Expected Prediction Error (EPE).

- For squared error loss,
  \[ R(f) = EPE(f) = E_{X,Y} [Y - f(X)]^2 \]

- For 0-1 loss
  \[ R(f) = EPE(f) = E_{X,Y} I(Y \neq f(X)) = Pr(Y \neq f(X)), \]
  which is the probability of making a mistake.
We seek \( f \) which minimizes the average (long-term) loss over the entire population.

**Optimal Learner**: The minimizer of \( R(f) \) is

\[
f^* = \arg \min_f R(f).
\]

- The solution \( f^* \) depends on the loss \( L \).
- \( f^* \) is not achievable without knowing \( P(x, y) \) or the entire population

**Bayes Risk**: The optimal risk, or the risk of the optimal learner

\[
R(f^*) = E_{X,Y} (Y - f^*(X))^2.
\]
Note that

\[ R(f) = E_{X,Y} L(Y, f(X)) = \int L(y, f(x))dP(x, y) \]

\[ = E_X \left\{ E_{Y|X} L(Y, f(X)) \right\} \]

One can do the following,

- for each fixed \( x \), find \( f^*(x) \) by solving

\[ f^*(x) = \arg \min_c E_{Y|X=x} L(Y, c) \]
Example 1: Regression with Squared Error Loss

For regression, consider $L(Y, f(X)) = (Y - f(X))^2$. The risk is

$$R(f) = E_{X,Y}(Y - f(X))^2 = E_X E_{Y|X}((Y - f(X))^2|X).$$

- For each fixed $x$, the best learner solves the problem

  $$\min_c E_{Y|X}[(Y - c)^2|X = x].$$

  The solution is $E(Y|X = x)$, so

  $$f^*(x) = E(Y|X = x) = \int y dP(y|x).$$

- So the optimal learner is $f^*(X) = E(Y|X)$. 
For any learner $f$, its EPE under the squared error loss is decomposed as

$$EPE(f) = E_{X,Y}[Y - f(X)]^2 = E_X E_{Y|X}([Y - f(X)]^2|X)$$
$$= E_X E_{Y|X}([Y - E(Y|X) + E(Y|X) - f(X)]^2|X)$$
$$= E_X E_{Y|X}([Y - E(Y|X)]^2|X) + E_X [f(X) - E(Y|X)]^2$$
$$= E_X, Y [Y - f^*(X)]^2 + E_X [f(X) - f^*(X)]^2$$
$$= EPE(f^*) + MSE$$
$$= \text{Bayes Risk} + \text{MSE}$$

The Bayes risk (gold standard) gives a lower bound for any risk.
In binary classification, assume $Y \in \{0, 1\}$. The learning rule $f : \mathbb{R}^d \rightarrow \{0, 1\}$.

Using 0-1 loss

$$L(Y, f(X)) = I(Y \neq f(X)),$$

the expected prediction error is

$$EPE(f) = EI(Y \neq f(X)) = \Pr(Y \neq f(X))$$

$$= E_X [P(Y = 1|X)I(f(X) = 0) + P(Y = 0|X)I(f(X) = 1)]$$
For each fixed $x$, the minimizer of $EPE(f)$ is given by

$$f^*(x) = \begin{cases} 
1 & \text{if } P(Y = 1|X = x) > P(Y = 0|X = x) \\
0 & \text{if } P(Y = 1|X = x) < P(Y = 0|X = x) 
\end{cases}$$

It is called the Bayes classifier, denoted by $\phi_B$.

The Bayes rule can be written as

$$\phi_B(x) = f^*(x) = I[P(Y = 1|X = x) > 0.5].$$

To obtain the Bayes rule, we need to know $P(Y = 1|X = x)$, or whether $P(Y = 1|X = x = x)$ is larger than 0.5 or not.
Assume the training set

\[(X_1, Y_1), \ldots, (X_n, Y_n) \sim \text{i.i.d } P(X, Y)\].

**Learning algorithm**: use the training data to construct a function \(f : X \rightarrow Y\) such that \(f(x) = \hat{y}\) for future prediction.

Goal: Find the *best* (optimal) \(\hat{f} \in F\) for decision making.

- However, the optimal learner \(f^*\) is not attainable without knowledge of \(P(x, y)\).
- We approximate the integral over \(P(x, y)\) by the average over the training samples.
Key Idea: We can not compute $R(f) = E_{X,Y} L(Y, f(X))$, but can compute its empirical version using the data!

**Empirical Risk**: also known as the training error

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)).$$

- A reasonable estimator $f$ should be close to $f^*$, or converge to $f^*$ when more data are collected.

Using the law of large numbers, $R_{\text{emp}}(f) \rightarrow EPE(f)$ as $n \rightarrow \infty.$
Learning Based on ERM

We construct the estimator of $f^*$ by

$$\hat{f} = \arg \min_f R_{\text{emp}}(f).$$

This principle is called **Empirical Risk Minimization**. (ERM)

For regression,

- the empirical risk is known as Residual Sum of Squares (RSS)

$$R_{\text{emp}}(f) = RSS(f) = \sum_{i=1}^{n} [y_i - f(x_i)]^2.$$

- ERM gives the least mean squares:

$$\hat{f}_{\text{ols}} = \arg \min_f RSS(f) = \arg \min_f \sum_{i=1}^{n} [y_i - f(x_i)]^2.$$
Minimizing $R_{\text{emp}}(f)$ over all functions is not proper.

- Any function passing through all $(x_i, y_i)$ has $RSS = 0$.

Example:

$$f_1(x) = \begin{cases} y_i & \text{if } x = x_i \text{ for some } i = 1, \ldots, n \\ 1 & \text{otherwise.} \end{cases}$$

It is easy to check that $RSS(f_1) = 0$.

- However, it does not amount to any form of learning. For the future points not equal to $x_i$, it will always predict $y = 1$. This phenomenon is called “overfitting”.

- Overfitting is undesired, since a small $R_{\text{emp}}(f)$ does not imply a small $R(f) = \int L(y, f(x))dP(x, y)$. 
What is generalization?
- estimating values of the function where there are no data

What is good generalization?
- predicting the function well overall (on future unseen data)
We need to restrict the estimation process within a set of functions $\mathcal{F}$, to control *complexity* of functions and hence avoid over-fitting.

- Choose a simple model class $\mathcal{F}$.

$$
\hat{f}_{ols} = \arg \min_{f \in \mathcal{F}} \text{RSS}(f) = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^{n} [y_i - f(x_i)]^2.
$$

- Choose a large model class $\mathcal{F}$, but use a roughness penalty to control model complexity: Find $f \in \mathcal{F}$ to minimize

$$
\hat{f} = \arg \min_{f \in \mathcal{F}} \text{Penalized RSS} \equiv \text{RSS}(f) + \lambda J(f).
$$

The solution is called regularized empirical risk minimizer.

How should we choose the model $\mathcal{F}$? Simple one or complex one?