Intro. to Supervised Learning

Zed

March 3, 2017

1 Statistical Decision Theory

1.1 Quantitative Dependent Variable

We want to firstly develop a general framework for supervised learning. We first consider quantitative output (label) $Y \in \mathbb{R}$ as a random variable. And $\boldsymbol{X} \in \mathbb{R}^p$ as a p-random column vector for input variables (features).
We place ourselves in probability space $(\mathbb{R}^p \times \mathbb{R}, \mathcal{F}, \mathbb{P})$. The pair $(\boldsymbol{X}, Y) \in \mathbb{R}^p \times \mathbb{R}$ has joint distribution $p_{\boldsymbol{X},Y}(\boldsymbol{x}, y)$, and we will also use similar notations for marginal and conditional distributions. Our goal is to find a function $\hat{Y} = f(\boldsymbol{X})$ to predict the value of Y corresponding to given input. We proceed as follows.

Def. **Loss Function**: $L(Y, \hat{Y})$ is constructed for penalizing errors in prediction. By far we choose a simple *squared error loss*:

$$L(Y, f(\boldsymbol{X})) = \|Y - f(\boldsymbol{X})\|_{\mathcal{L}^2} = (Y - f(\boldsymbol{X}))^2$$

Def. **Expected Prediction Error (EPE)**: We seek to find a function f that minimizes the expection of L over the probability space we defined, which is:

$$EPE(f) := \mathbb{E}\left[L(Y, \hat{Y})\right]$$

If we use the squared error loss,

$$\begin{aligned} EPE(f) &= \mathbb{E}\left[(Y - f(\boldsymbol{X}))^2\right] \\ &= \iint_{\mathbb{R}^p \times \mathbb{R}} (y - f(\boldsymbol{x}))^2 p_{\boldsymbol{X},Y}(\boldsymbol{x}, y) dy d\boldsymbol{x} \end{aligned} \tag{1}$$

We can split the joint distribution in (1) to conditional distribution times the marginal, and rewrite the integral as

$$\begin{aligned} EPE(f) &= \iint_{\mathbb{R}^p \times \mathbb{R}} (y - f(\boldsymbol{x}))^2 p_{\boldsymbol{X},Y}(\boldsymbol{x}, y) dy d\boldsymbol{x} \\ &= \int_{\boldsymbol{X}} p_{\boldsymbol{X}}(\boldsymbol{x}) \left(\int_Y (y - f(\boldsymbol{x}))^2 p_{Y|\boldsymbol{X}}(y|\boldsymbol{x}) dy \right) d\boldsymbol{x} \\ &= \mathbb{E}_{\boldsymbol{X}}\left[\mathbb{E}_{Y|\boldsymbol{X}}\left[(Y - f(\boldsymbol{X}))^2 | \boldsymbol{X} = \boldsymbol{x}\right]\right] \end{aligned} \tag{2}$$

And minimizing this expectation suffices to minimizing pointwise for any $\boldsymbol{x}$:

$$f(\boldsymbol{x}) = \underset{\xi}{\operatorname{argmin}}\, \mathbb{E}_{Y|\boldsymbol{X}}\left[(Y - \xi)^2 | \boldsymbol{X} = \boldsymbol{x}\right] \tag{3}$$

Take FOC:

$$\begin{aligned} \frac{\partial}{\partial \xi}\mathbb{E}_{Y|\boldsymbol{X}}\left[(Y-\xi)^2|\boldsymbol{X}=\boldsymbol{x}\right] = \mathbb{E}_{Y|\boldsymbol{X}}\left[\frac{\partial}{\partial \xi}(Y-\xi)^2|\boldsymbol{X}=\boldsymbol{x}\right] = 0 \\ \Rightarrow f(\boldsymbol{x}) = \xi = \mathbb{E}_{Y|\boldsymbol{X}}\left[Y|\boldsymbol{X}=\boldsymbol{x}\right] \end{aligned} \tag{4}$$

So the solution to pointwise minimization problem of EPE is $f(\boldsymbol{x}) = \mathbb{E}_{Y|\boldsymbol{X}}\left[Y|\boldsymbol{X}=\boldsymbol{x}\right]$. Thus the best prediction of Y at point $\boldsymbol{x}$ is the the conditional expectation of Y, when ths 'best' is measured by square error. This is referred to as *regression function.*

Ex. The k**-Nearest Neighbour** methods attempt to implement this recepie directly, with

$$\hat{f}_{knn}(\boldsymbol{x}) = \frac{1}{k}\sum_{\boldsymbol{x}_j \in N_k(\boldsymbol{x})} y_j$$

Where $N_k(\boldsymbol{x}) = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_k; \|\boldsymbol{x}-\boldsymbol{x}_j\| \leq \|\boldsymbol{x}-\boldsymbol{z}\| \; j = 1, 2, ..., k, \forall \boldsymbol{z} \notin N_k(\boldsymbol{x})\}$. It uses averaging to approximate expectation, and conditioning on 1 point is relaxed to conditioning on $N_k(\boldsymbol{x})$. It can be shown that with large training set of size N, $\hat{f}_{knn}(\boldsymbol{x}) \to \mathbb{E}_{Y|\boldsymbol{X}}\left[Y|\boldsymbol{X}=\boldsymbol{x}\right]$ as $N, k \to \infty$ with $k/N \to 0$. However, the rate of convergence decrease when dimension p increases.

Ex. **OLS Linear Regression** asssumes the function is linear in $\boldsymbol{x}$, i.e.

$$\hat{f}_{ols}(\boldsymbol{x}) = \boldsymbol{x}^\top \beta$$

This is a model-based approach. We plug in this functional form into EPE:

$$EPE(\hat{f}_{ols}) = \mathbb{E}\left[(Y - \boldsymbol{X}^\top\beta)^2\right] \tag{5}$$

Take FOC:

$$\begin{aligned} \frac{\partial}{\partial \beta}\mathbb{E}\left[(Y-\boldsymbol{X}^\top\beta)^2\right] &= \mathbb{E}\left[\frac{\partial}{\partial \beta}(Y-\boldsymbol{X}^\top\beta)^2\right] \\ &= \mathbb{E}\left[2\boldsymbol{X}(Y-\boldsymbol{X}^\top\beta)\right] = 0 \\ \Rightarrow \beta &= \mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^\top\right]^{-1}\mathbb{E}\left[\boldsymbol{X}Y\right] \end{aligned} \tag{6}$$

And then by replacing the expection by averaging over the dataset we obtain the familiar OLS estimator solution.

We have seen that both KNN and the OLS end up approximating conditional expectations by averages, but they differ in terms of model assumptions.

- $\hat{f}_{ols}(\boldsymbol{x})$ is assumed to be a globally linear function.
- $\hat{f}_{knn}(\boldsymbol{x})$ is assumed to be a locally constant function.

1.2 Categorical Dependent Variable

In this setting our dependent (random) varibale $G \in \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_K\} =: \mathcal{G}$, i.e. it has K types in total. $\hat{G} = \hat{G}(\boldsymbol{X})$ is the prediction. We can use a $K \times K$ matrix $\boldsymbol{L}$ to represent the loss, with L_{kl} being the loss of classifying a $\mathcal{G}_k$ observation as $\mathcal{G}_l$. Usually we will use *zero-one* loss function, which charges 1 unit for all misclassifications uniformly, i.e.

$$L(G, \hat{G}(\boldsymbol{X})) = \begin{cases} 0 & G = \hat{G}(\boldsymbol{X}), \\ 1 & G \neq \hat{G}(\boldsymbol{X}) \end{cases} = \mathbb{1}_{\{G \neq \hat{G}(\boldsymbol{X})\}} \tag{7}$$

We proceed in the same fashion

$$\begin{aligned} EPE(\hat{G}) &= \mathbb{E}\left[L(G, \hat{G}(\boldsymbol{X}))\right] \\ &= \int_X \left(\sum_{k=1}^{K} L(\mathcal{G}_k, \hat{G}(\boldsymbol{X}))\mathbb{P}\left(G = \mathcal{G}_k, \boldsymbol{X} = \boldsymbol{x}\right)\right) d\boldsymbol{x} \\ &= \int_X p_{\boldsymbol{X}}(\boldsymbol{x}) \left(\sum_{k=1}^{K} L(\mathcal{G}_k, \hat{G}(\boldsymbol{X}))p_{G|\boldsymbol{X}}(\mathcal{G}_k|\boldsymbol{x})\right) d\boldsymbol{x} \\ &= \mathbb{E}_X\left[\sum_{k=1}^{K} L(\mathcal{G}_k, \hat{G}(\boldsymbol{X}))p_{G|\boldsymbol{X}}(\mathcal{G}_k|\boldsymbol{x})\right] \end{aligned} \tag{8}$$

And again it suffices to minimize EPE pointwise w.r.t. $\boldsymbol{x}$,

$$\begin{aligned} \hat{G}(\boldsymbol{x}) &= \underset{g\in\mathcal{G}}{\operatorname{argmin}} \sum_{k=1}^{K} L(\mathcal{G}_k, g)p_{G|\boldsymbol{X}}(\mathcal{G}_k|\boldsymbol{x}) \\ &= \underset{g\in\mathcal{G}}{\operatorname{argmin}} \sum_{k=1}^{K} \mathbb{1}_{\{g\neq\mathcal{G}_k\}}\mathbb{P}\left(G = \mathcal{G}_k|\boldsymbol{X} = \boldsymbol{x}\right) \\ &= \underset{g\in\mathcal{G}}{\operatorname{argmin}} \left(1 - \mathbb{P}\left(G = g|\boldsymbol{X} = \boldsymbol{x}\right)\right) \end{aligned} \tag{9}$$

In another word, $\hat{G}(\boldsymbol{x}) = \mathcal{G}_k \iff$

$$\mathcal{G}_k = \underset{g\in\mathcal{G}}{\operatorname{argmax}}\, \mathbb{P}\left(G = g|\boldsymbol{X} = \boldsymbol{x}\right) \quad \iff$$

$$\mathbb{P}\left(G = \mathcal{G}_k|\boldsymbol{X} = \boldsymbol{x}\right) = \max_{g\in\mathcal{G}} \mathbb{P}\left(G = g|\boldsymbol{X} = \boldsymbol{x}\right)$$

which says that, given $\boldsymbol{X} = \boldsymbol{x}$, $G = \mathcal{G}_k$ has the greatest conditional probability. This solution is known as **Bayes Classifier**. And the error rate is called the *Bayes Rate*. If we know the generating distribution of dataset, the Bayes classifier decision boundary can be specifed exactly.
The dummy variable regression approach fits in this framework, and is just another way of representing the Bayes classifier. Because we use $\hat{Y}(\boldsymbol{X}) = \mathbb{E}\left[Y|\boldsymbol{X}\right] = \mathbb{P}\left(\mathcal{G}_1|\boldsymbol{X}\right)$ if $\mathcal{G}_1$ corresponds to $Y = 1$.

2 Local Methods in High Dimensions

The increase dimensionality p cast shadow on our established intuition, that we could always approximate the conditional expectation with k-nearest neighbour averaging.

2.1 Curse of Dimensionality

- **Neighbourhood not-so-local**: Consider inputs $\boldsymbol{X} \sim \text{Uniform}([0,1]^p)$, uniformly distributed in p-dim hypercube. We want to use hypercubical neighbourhood to capture a fraction r of all obs. The expected length of edge is $e(p) = r^{\frac{1}{p}}$, increases with p exponentially. In high dimension, we need to look at a wide range on each input variable to capture a desirable fraction of data. Such neibourhood is not "local" any more.
- **Sample close to boundary**: Each sample point becomes closer to boundary of the sample space. And the prediction is more difficult near the edges.

· **Sparsity of sample**: The sampling density is proportional to $N^{1/p}$. The number of obs required to form a desirably dense sample grows exponentially with dimensionality. Thus in high dimensions training samples *sparsely* populate the input space.

2.2 Bias-Variance Decomposition

We first reconsider the familiar definition in the context of statistical learning. Consider random variables $\boldsymbol{X}, Y$, Y can be a *deterministic function* of $\boldsymbol{X}$, say

$$Y = f(\boldsymbol{X})$$

The **true** relationship between $\boldsymbol{X}$ and Y can also be *stochastic*, for example, in linear regression we have

$$Y = \boldsymbol{X}^\top \beta + \epsilon$$

where ϵ is some kind of Gaussian random variable.
In the first setting, $f(\cdot)$ is a deterministic function, but **unknown**, i.e. we always estimate the function with $\hat{f}(\cdot)$, with the help of some kind of supervised learning algorithm, and a traing set $\mathcal{T}$ that contains the notion of randomness, because it is essentially a random sample of $(\boldsymbol{X}, Y)$, to which we have no complete control.

Hence the estimation $\hat{f}(\cdot)$ relies on the random sample $\mathcal{T}$. At a given point $\boldsymbol{x}$, our estimator for Y is $\hat{Y}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x})$ is therefore also a random variable that relies on $\mathcal{T}$. When we take expectation of $\hat{Y}$, we are actually averaging over all possible training sets $\mathcal{T}$, which is the meaning of the notion

$$\mathbb{E}_{\mathcal{T}}\left[\hat{f}(\boldsymbol{x})\right] = \int_{\text{all } \mathcal{T}} \hat{f}(\boldsymbol{x}) d\mathbb{P}$$

Def. **Bias (Supervised Learning)**: at a given point $\boldsymbol{x}$, the bias of an estimator $\hat{f}(\boldsymbol{x})$ of the deterministic & true value $f(\boldsymbol{x})$ is

$$\text{Bias}(\hat{f}(\boldsymbol{x})) = \mathbb{E}_{\mathcal{T}}\left[\hat{f}(\boldsymbol{x})\right] - f(\boldsymbol{x})$$

Def. **Variance (Supervised Learning)**: at a given point $\boldsymbol{x}$, the variance of a predictor $\hat{f}(\boldsymbol{x})$ of the deterministic & true value $f(\boldsymbol{x})$ is the centerred second moment wrt randomly sampled training set

$$\mathbb{V}\text{ar}_{\mathcal{T}}\left[\hat{f}(\boldsymbol{x})\right] = \mathbb{E}_{\mathcal{T}}\left[\left(\hat{f}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{T}}\left[\hat{f}(\boldsymbol{x})\right]\right)^2\right]$$

Def. **Mean Squared Error (MSE)**: deterministic model $Y = f(\boldsymbol{X})$, given point $\boldsymbol{x}$,

$$MSE(\hat{f}(\boldsymbol{x})) := \mathbb{E}_{\mathcal{T}}\left[\left(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right]$$

For simplicity we denote $\hat{Y} = \hat{Y}(\boldsymbol{x})$, $y = f(\boldsymbol{x})$ at a given $\boldsymbol{x}$, the first is a random varible, the second is deterministic. We have

$$\begin{aligned} MSE(\hat{Y}) &= \mathbb{E}_{\mathcal{T}}\left[\left(\hat{Y} - y\right)^2\right] \\ &= \mathbb{E}_{\mathcal{T}}\left[\left(\hat{Y} - \mathbb{E}_{\mathcal{T}}[\hat{Y}] + \mathbb{E}_{\mathcal{T}}[\hat{Y}] - y\right)^2\right] \\ &= \mathbb{E}_{\mathcal{T}}\left[\left(\hat{Y} - \mathbb{E}_{\mathcal{T}}[\hat{Y}]\right)^2 + \left(\mathbb{E}_{\mathcal{T}}[\hat{Y}] - y\right)^2 + 2(\hat{Y} - \mathbb{E}_{\mathcal{T}}[\hat{Y}])(\mathbb{E}_{\mathcal{T}}[\hat{Y}] - y)\right] (\dagger) \end{aligned} \tag{10}$$

Note that the second term is a constant wrt. expectation, the cross term is 0. Hence

$$\begin{aligned}(\dagger) &= \mathbb{E}_{\mathcal{T}}\left[\left(\hat{Y} - \mathbb{E}_{\mathcal{T}}[\hat{Y}]\right)^2\right] + \left(\mathbb{E}_{\mathcal{T}}[\hat{Y}] - y\right)^2 \\ &= \mathbb{V}\mathrm{ar}_{\mathcal{T}}[\hat{Y}] + \mathrm{Bias}^2[\hat{Y}]\end{aligned} \tag{11}$$

which is called the bias-variance decomposition.