Statistical Learning, Fall 2022-23

Class notes 1

Our course is focused on **Supervised learning** or **Predictive modeling**, where we have two types of entities:

- A *response* or *dependent varaible* y that we are interested to model. We differentiate *regression* where $y \in \mathbb{R}$ from *classification* where $y \in \mathcal{G}$ an unordered set.
- A vector of *explanatory variables* or *covariates* or *features* $x \in \mathcal{X}$ that encode information about y. We will often but not always assume $\mathcal{X} = \mathbb{R}^p$ a vector of measurements.

Treating x, y as random variables, there is an (unknown) joint distribution $Pr(x, y)$. In the predictive modeling problem, however, our interest is focused on the conditional distribution $Pr(y|x)$. For regression we can write:

$$y = f(x) + \epsilon,\ \mathbb{E}(\epsilon) = 0 \ \ \Rightarrow f(x) = \mathbb{E}(y|x),$$

so we often focus our attention on modeling $f(x)$ which summarizes the dependence of y on x.

In the most standard setting, we assume we have a *training set* $T = \{(x_1, y_1), \ldots, (x_n, y_n)\} = (\mathbb{X}_{n\times p}, \mathbb{Y}_{n\times 1})$ sampled i.i.d from $Pr(x, y)$, and we want to use this data to learn a model $\hat{f}(x)$ that will summarize what we can learn about the true $f(x)$. We call $\hat{f}(x)$ our prediction model, imagining a situation in the future where we will get x only, then apply $\hat{f}$ to get a prediction $\hat{y} = \hat{f}(x)$. A good prediction model would tend to give $y \approx \hat{y}$ and/or $f(x) \approx \hat{f}(x)$.

A secondary goal, beyond predicting well, is often to "learn" things about the true function f, for example:

- Which coordinates of x are actually informative about y (equivalently, have a role in f)?
- More generally, what kind of function is f? For example: Is it linear, or can it be well approximated by linear functions?

A generic description of the predictive modeling process is given in Fig. 1.

A topic often mentioned in the context of predictive modeling and beyond is *big data.* In our context, this could mean one or both of the following:

- *Tall* data: large number of observations in the training data, n is very big
- *Wide* data: large dimensionality, p is very big

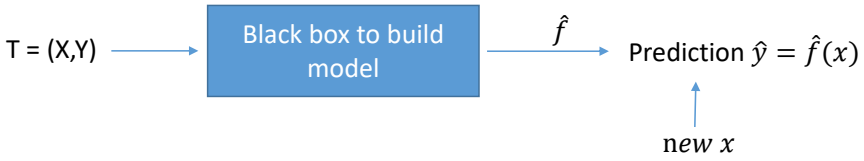

Figure 1: generic description

As we will see, the two modes tend to have very different implications for the predictive modeling process, affecting: which approaches are relevant, what are the difficulties (statistical vs computational), etc.

Basic predictive modeling approaches: least squares regression and nearest neighbors

We will start from describing two important and fundamental methods for regression: least squares regression and nearest neighbors regression. In addition to being important in themselves, they will represent for us conceptual approaches that are shared among many methods and approaches that we will study during the course:

- Parametric global methods, which build $\hat{f}(x)$ as a parametric model that applies to the whole space
- Non-parametric local methods, which don't explicitly build a function $\hat{f}(x)$, but describe it in terms of local behavior of the training data in the region near x

Least squares regression (quick reminder)

We are dealing with regression $y \in \mathbb{R}$ with covariates $x \in \mathbb{R}^p$. We want to build a linear model $\hat{y} = \hat{\beta}_0 + x^t\hat{\beta}$. Since we want to also include an intercept $\hat{\beta}_0$ in our model, we typically define

$$\tilde{x} = \begin{pmatrix} 1 \\ x \end{pmatrix} \in \mathbb{R}^{p+1}, \text{ and similarly } \tilde{\mathbb{X}} \ ,$$

and also assume $\hat{\beta} \in \mathbb{R}^{p+1}$ to now write simply $\hat{y} = \tilde{x}^t\hat{\beta}$.
Note: In practice we will often neglect the tildes and simply write x, implicitly assuming the added intercept when relevant.

For any given β vector, we have $\mathbb{X}\beta \in \text{span}(\mathbb{X})$ (column space), so we can think of least squares regression as using the training data T to find a good vector $\mathbb{X}\hat{\beta}$ in the column space of $\mathbb{X}$. This is typically done via least squares regression:

$$\hat{\beta} = \arg\min \|\mathbb{Y} - \mathbb{X}\beta\|_2^2,$$

meaning we are looking for the closest vector to Y in the column space of $\mathbb{X}$, and this gives us the least squares solution $\hat{\beta}$. This is an *orthogonal projection*.

The well known solution to this optimization solution is the least squares estimator:

$$\hat{\beta} = \left(\mathbb{X}^t\mathbb{X}\right)^{-1}\mathbb{X}^t\mathbb{Y}, \;\; \hat{Y} = \mathbb{X}\hat{\beta} = \mathbb{X}\left(\mathbb{X}^t\mathbb{X}\right)^{-1}\mathbb{X}^t\mathbb{Y} := H\mathbb{Y}.$$

Important note: We generally do not assume $f(x) = \mathbb{E}(y|x) = x\beta$, in other words there is no real connection between assuming that the linear model is correct and performing least squares regression. We will carefully analyze the *bias* component that arises in linear regression when the linear model assumption does not hold.

Nearest neighbor regression

Given a point we want to predict at x, we will simply predict y as an average over y of the k "nearest neighbors" of x in the training data:

$$\hat{y} = \frac{1}{k}\sum_{i \in N_k(x)} y_i,$$

where $N_k(x)$ contains the indexes for the "neighborhood" of size k of x in the training data.

The key to defining nearest neighbor methods is the definition of distance and hence neighborhood. When $x \in \mathbb{R}^p$, the simplest approach is simply to use Euclidean distance: $d(x, x_i) = \|x - x_i\|^2$, and $N_k(x)$ will be the indexes of the k observations with minimal distance. Different definitions of distance are possible, and of course affect model quality, as well as the selection of the number of neighbors k we consider.

An important option to consider is that of standardization (centering and rescaling columns of $\mathbb{X}$ to have norm 1), which makes all variables "similarly" influential in the distance function, as opposed to having columns on a different scale (milimeters vs kilometers).
Question: What effect does standardization have on least squares regression estimates, specifically on each of $\hat{\beta}$ and $\hat{\mathbb{Y}}$?

Application to classification problems

If we limit our discussion for now to two-class classification problems, where $y \in \mathcal{G}$ with $|\mathcal{G}| = 2$, $\mathcal{G} = \{g_0, g_1\}$ we can easily propose versions of linear regression and nearest neighbors that can also apply to this setting and build prediction models.

For linear regression, if we simply encode $y = g_0 \to y = 0$, $y = g_1 \to y = 1$, we can apply linear regression as it is, and we often then use it to predict by thresholding the resulting model:

$$\hat{y}(x) = g_1 \;\Leftrightarrow\; \hat{f}(x) > 0.5.$$

For nearest neighbors, we can simply replace the averaging in the regression setting to voting, so

$$\hat{y}(x) = \mathbb{I}\left\{\frac{1}{k}\sum_{i \in N_k(x)} \mathbb{I}\{y_i = g_1\} > 0.5\right\}.$$

Probability notations and laws - refresher

We use $(X, Y) \sim Pr(x, y)$ a general notation for discrete probabilities or continuous densities. We also use $\int$ for sums or integrals interchangeably.

Conditional distribution: $Pr(Y = y|X = x) = \frac{Pr(x,y)}{Pr(x)}$. Conditional expectation: $\mathbb{E}(Y|X = x) = \int yPr(y|X = x)$.

Note that $\mathbb{E}(Y|X)$ is a random variable, which attains a value for each value of X. We can define conditional variance in the same way.

Law of iterated expectation: $\mathbb{E}(g(X, Y)) = \mathbb{E}_X \left[\mathbb{E}\left(g(X, Y)|X\right)\right]$.

Law of total variation: $Var(Y) = \mathbb{E}_X Var(Y|X) + Var_X \mathbb{E}(Y|X)$.

Example: Assume $X \in \{Male, Female\}$ and $Y = height$. Assume $Y|X = M \sim N(175, 100)$, $Y|X = F \sim N(165, 100)$. (Note that height does not have a normal distribution but a mixture of normals.) Assume also $Pr(X = F) = 0.5$.
Then:

$$E(Y) = 0.5 \times 175 + 0.5 \times 165 = 170\ , \quad Var(Y) = 100 + Var(Z) = 125,$$

where

$$Z = E(Y|X) \begin{cases} 175 & \text{w.p. } 0.5 \\ 165 & \text{w.p. } 0.5 \end{cases}.$$