Statistical Learning Notes
For Intuition

Luis Calderon

March 2020

1 Introduction

Take some relationship between X and Y, where X could be a multi-dimensional object containing inputs or **features** that *could* explain Y, a 1-dimensional object. Y is sometimes called the **label** or response variable among others. In statistical learning, one might be interested in trying to *predict* Y given the multi-dimensional object X or looking for the individual effects a feature might *infer* on Y.

1.1 Modeling

To model a relationship, lets use the following form:

$$Y = f(X) + \varepsilon$$

where $f(X)$ is a function of our features, X, and ε is the error term. In words, we wish to estimate Y given some function of X and some unobservable. $f(X)$ may be linear, quadratic, or some higher-order polynomial. As a researcher, it is our duty to use theory and statistics to help us select the appropriate functional form of $f(X)$ in our estimations, with the idea of reducing **reducible error**. Reducible error is error that arises from $\hat{f}(X)$ not perfectly predicting the true $f(X)$. It is called reducible because we can *reduce* the error through appropriate model selection or statistical learning technique. Since Y is also a function of ε, there is some irreducible error. Meaning, given the true $f(X)$, Y can still be inaccurate because of the variability of the noise, ε. This can come from omitted-variable bias, other endogeneity biases, and measurement error.

To contextualize this, suppose you're a home buyer and a Realtor. As the home buyer, you want to see if the house is over-valued or under-valued given some inputs (e.g., bedrooms, square footage, floors, location, etc). This is **prediction** — only interested in the outcome. The Realtor might want to know, using data, how would living next to a great school affect the price of a home or if the home added one more bedroom. He's more preoccupied with how a change in the inputs changes the price. This is **inference**.

1.2 Parametric vs. Non-parametric approaches

The goal of statistical learning is to use **training data** to estimate $f(X)$. In parametric methods, we make an assumption on the form of $f(X)$, for example, assuming $f(X)$ is linear and pick a procedure to estimate the parameters. For example, look at the following equation:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

. This assumes f is a linear combination of X. To estimate, we could use ordinary least squares, which takes our data and tries to fit the model we've just put together, reducing the problem to just finding a vector of parameters which reduces are reducible error.

Non-parametric approaches do not make an assumption on the functional form of $f(X)$. It estimates the best possible $f(X)$ given the data, without being too "wiggly". Wiggly here refers to Runge's phenomenon, when the model is too flexible from specifying a higher-order polynomial to estimate $f(X)$. The researcher must still then select the level of smoothness of the functional form to lessen this wiggliness. This approach also requires many observations to obtain the best possible shape of $f(X)$.

1.3 General notes

- **Supervised vs. Unsupervised learning**: Supervised learning is like this: suppose I have a child, and I am trying to teach him what a pig looks like. I show him images of the pig, and sometimes other animals, and ask whether it is a pig or not. He provides a response "Yes" or "No". Meaning, for each observation, there is an associated response. Unsupervised learning has no response variable. Its purpose is more about "What are the groups or clusters that can be derived from this information."

- **Regression vs. Classification**: it's just quantitative vs. qualitative (linear regression equals continuous output variable, classification equals discrete).

- **Assessing model accuracy**: From a regression perspective, minimize mean-squared error (MSE). YET, this is only informative given YOUR training data. Our goal is the following: We want to fit some statistical learning method (i.e. linear regression) on our training observations to obtain a $\hat{f}(X)$, WHICH we then try on previously **unseen** data. If our function can take an observation from an outside dataset and predict the correct outcome, then we've reached model accuracy. In particular, these observations could be from the future. For example, a good model should be able to predict tomorrow's stock price using 6 months of prior data. Thus, the training MSE is not the same as the test MSE.

From a classification perspective, we are interested in the **error rate**. The error rate is how many incorrect classifications were made divided by observations. A good classifier will minimize this test rate.

- **Bayes Classifier**: This is just classifying labels given some vector of features. For example, given the color of the fruit, its shape and taste, the classifier will say it's an apple or a banana. The classifier relies on independence of the features, however. It is considered the gold-standard since it expects the true conditional distribution of Y given X. **K-nearest neighbors** attempts to estimate the conditional distribution by classifying the observation to the class with the highest probability.
- **K-nearest neighbors**: The KNN approach is really simple. Suppose I have my training data and we have 6 points. 3 blue and 3 red. Now I have a test observation. What KNN does is it checks its K nearest neighbors and sees what is their label or "color" in this case. Suppose around this test observation, we look at $K = 3$ neighbors. 1 is blue and 2 are red. Thus, we label our test observation as red. That's it. Just kidding. It is more complicated than that. Specifically, how many neighbors K should I check? and what is the goal? This is where the researcher has to test and see. Just compare training errors and test errors across different K amounts.

2 Linear Regression

The focus of this section is not on the concept of linear regression, but the method in a machine learning environment. Machine learning is about predicting and we wish to assess how accurate our trained model is on some test data.