36-700: Probability and Mathematical Statistics I **Fall 2016**

Lecture 17: October 7

Lecturer: Siva Balakrishnan

17.1 Review and Outline

In the last class we discussed linear regression:

1. Fixed versus random design
2. Statistical properties of simple regression: bias, variance, (asymptotic normality) confidence intervals
3. Multiple regression

Today we will discuss some statistical properties of multiple linear regression, the uses of models in statistics and begin our discussion of non-parametric regression (this is in Chapter 20 of the Wasserman book).

17.2 General Linear Regression

In the general setting the covariate is a d dimensional vector, so we observe $(Y_1, X_1), \ldots, (Y_n, X_n)$ where each $X_i \in \mathbb{R}^d$.

The regression model is written succinctly as:

$$y = X\beta + \epsilon,$$

where $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times d}$, $\epsilon \in \mathbb{R}^n$, and $\beta \in \mathbb{R}^d$.

At this point I will assume you are familiar with matrix calculus. If not some of the review aids at the bottom of this page: `http://www.stat.cmu.edu/~ryantibs/convexopt/` might be helpful.

The least squares estimate:

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2$$

Again we take the derivative with respect to β and set it to 0, in this case this gives:

$$-X^T(y - X\widehat{\beta}) = 0,$$

i.e. that:

$$X^T y = (X^T X)\widehat{\beta},$$

from which we obtain the LS estimator:

$$\widehat{\beta} = (X^T X)^{-1} X^T y.$$

An exercise: You should be able to recover our previous estimates for β_0, β_1 by taking X to be the matrix where the first column is always 1 and the second column is the single observed covariate.

We will do the rest of our analysis under the Gaussian noise assumption, i.e. we assume that

$$\epsilon \sim N(0, \sigma^2 I).$$

In this case, we obtain that:

$$\widehat{\beta} = (X^T X)^{-1} X^T y = (X^T X)^{-1} X^T (X\beta + \epsilon) = \beta + (X^T X)^{-1} X^T \epsilon.$$

So as before we can calculate that the least squares estimator is unbiased, and its variance is simply the variance of the second term, i.e.:

$$\mathrm{Var}(\widehat{\beta} | X_1, \ldots, X_n) = \mathrm{Var}((X^T X)^{-1} X^T \epsilon).$$

Now, you will need a fact about multivariate Gaussians: if $u \sim N(0, \sigma^2 I)$ then for any matrix M, $Mu \sim N(0, \sigma^2 M M^T)$. Using this you can calculate that:

$$\mathrm{Var}(\widehat{\beta} | X_1, \ldots, X_n) = \sigma^2 (X^T X)^{-1},$$

and furthermore that:

$$\widehat{\beta} \sim N(\beta, \sigma^2 (X^T X)^{-1}),$$

which you can use to construct confidence intervals. When the noise is not Gaussian, the bias and variance calculations are still correct, and the distributional result is true asymptotically, i.e. as $n \to \infty$ with d fixed.

17.3 Models in Statistics

Throughout our discussions so far, and in a lot of the remainder of the course we often hypothesize a model and then study estimation and inference. When you analyze a sufficiently complex real dataset a model will inevitably be a simplification of the "true" generative process.

Statistics in general has a complicated relationship with models: particularly, eminent statisticians like George Box and David Cox have said things like: "All models are wrong but some are useful", and "The very word model implies simplification and idealization."

This of course leads to several questions: why are models useful at all? What happens when the model is mis-specified? Do we really need strong modeling assumptions in statistics?

Here are a few answers:

1. **Why are models useful?**

   - Models even when mis-specified can be a useful lens through which we can view data. As a very simple example: even if the data I have does not have a Gaussian distribution, fitting a Gaussian to the data gives me a sense of the mean and variance.

     More generally, even when the model is wrong, the estimated parameters or model are much easier to understand/summarize than the raw data.

   - Models often force us to think of generative mechanisms (which can be useful in itself): i.e. (at least approximately) how did the data come about?

     Once we have a model, we are naturally led to formalize quantities of interest (i.e. parameters) and think about estimating them (i.e. solving the inverse problem).

     If you prefer a more Computer Science centric view of things:

     $$\text{model} + (\text{generic estimation method}) \implies \text{algorithm}.$$

     By this I mean, it might be quite difficult to come up with an algorithm to solve a certain (abstract) problem: I would like to understand a large collection of documents.

     A powerful way to try to solve a problem of this form is to hypothesize a generative model: in this context this is called a "topic model". We suppose that the collection is about a small number of topics, and that each topic has a different distribution over words: i.e. a topic like sports will be more likely to involve words like "bat", "ball", "soccer" etc. So maybe we could write down a very simple generative model: we first sample a topic for a document (from say a multinomial) and then for each we sample a bunch of words (the text of the document) also from a multinomial.

From a statistical point of view, this is a simple mixture model, we can write down the likelihood of the text we observed as a function of the multinomial parameters, and then maximize the likelihood to figure out the topic distributions.

At the end of the day we will group the documents into topics, and further say for this collection here are the most likely words (useful for identifying the topics).

More abstractly we hypothesized a generative model, and then used standard statistical tools to devise an algorithm for our problem.

- Models can help us think rigorously about inference: roughly, you observe some pattern in the data (or an extreme value of some parameter) and you would like to know is this really surprising or just due to the usual stochastic fluctuations.

  This will be clearer when we discuss hypothesis testing, but the take away is just that we models in order to formalize questions of inference.

2. **What happens when the model is mis-specified?** Basically, one can wonder if maximum likelihood which is tailored to a particular model, is complete garbage if the model is wrong. Further, since the model is always wrong does maximum likelihood make any sense?

   This is a somewhat subtle question: most often the answer is yes, maximum likelihood is still a good idea even when you do not really completely believe the model.

   Roughly, the intuition you should have is that maximum likelihood will often estimate the projection of the true generative mechanism onto your model. This is some strong sense the best approximation of the true generative mechanism, i.e. maximum likelihood is just trying to fit the best approximation of the true model.

3. **Do we really need strong modeling assumptions?** Concretely, suppose we are doing regression. Do we really need to make the assumption that the relationship between X and Y is linear? This is a pretty strong modelling assumption.

   Often in statistics we use non-parametric models. They make much weaker modelling assumptions.

   However, as we will see there is natural bias variance tradeoff. You can make the bias small by making weaker assumptions but the variance of your estimate will get larger.

   In effect, maybe one answer is: yes you can make very weak modelling assumptions provided you have lots of data!

   We will explore this in more detail in the next couple of lectures.

17.4 Non-parametric Regression

Recall that broadly in regression our goal is to estimate the regression function $r(x) = \mathbb{E}[Y|X = x]$. Unlike the CDF which we could estimate with no assumptions about the

distribution, here we will need *smoothness* assumptions, i.e. we will need to assume that $r(x)$ is a smooth function of x. This allows us to gain statistical strength by averaging near by points.

Suppose we construct an estimate $\widehat{r}(x)$. Then a natural measure of how well we do is the squared loss, except since these are functions this is called the *integrated* squared loss, i.e.:

$$L(\widehat{r}, r) = \int (\widehat{r}(x) - r(x))^2 dx.$$

The risk is then just the expected loss, i.e.:

$$R(\widehat{r}, r) = \mathbb{E}\left(\int (\widehat{r}(x) - r(x))^2 dx\right).$$

As in the case of point estimation we have a bias variance decomposition. First we define the point-wise bias:

$$b(x) = \mathbb{E}(\widehat{r}(x)) - r(x),$$

and the point-wise variance:

$$v(x) = \mathbb{E}(\widehat{r}(x) - \mathbb{E}(\widehat{r}(x)))^2.$$

Now, as before we can verify that:

$$R(\widehat{r}, r) = \int b^2(x) dx + \int v(x) dx.$$

A natural strategy in non-parametric regression is to locally average the data, i.e. our estimate of the regression function at any point will be the average of the Y values in a small neighborhood of the point.

The width of this neighborhood will determine the bias and variance. Too large a neighborhood will result in high bias and low variance (this is called oversmoothing) and too small a neighborhood will result in low bias but large variance (this is known as undersmoothing).

17.4.1 Kernel Regression

One of the most basic ways of doing non-parametric regression is called kernel regression. We will analyze kernel regression when we only have one covariate. The general case is not very different.

Here the estimator is defined as:

$$\widehat{r}(x) = \sum_{i=1}^{n} w_i(x) Y_i,$$

where the weights assign more importance to points near x. This is called a kernel regressor when the weights are chosen according to a kernel, i.e. we have weights:

$$w_i(x) = \frac{K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)},$$

where h controls the amount of smoothing. It is called the bandwidth.

We will choose kernels that satisfy a few conditions:

1. $K(x) \geq 0$
2. $\int K(x) = 1$
3. $\int xK(x) = 0.$

These will be useful when we analyze this estimator. As a typical common example you should consider the Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2).$$

Finally, in order to analyze the kernel regression estimator we will need to assume something about $r(x)$. We will assume that $r(x)$ is Lipschitz, i.e. that there is some constant L such that:

$$\left| \frac{d}{dx} r(x) \right| \leq L.$$

More generally, we could assume things about higher-derivatives (more smoothness), i.e. for instance that the β-th derivative is bounded. In general, more smoothness will imply lower MSE.

In the next class we will bound the bias and variance of the kernel regressor as a function of the bandwidth and try to concretely pin down the bias-variance tradeoff.