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

Lecture 20: October 17

Lecturer: Siva Balakrishnan

20.1 Review and Outline

In the last class we discussed histograms:

1. Density estimation
2. Histograms
3. Bias and Variance of histograms for Lipschitz densities

Today we will discuss kernel density estimation (KDE). This material is also in Chapter 20 of the Wasserman book.

At a high-level there are two main reasons why we often prefer KDEs to histograms:

1. KDEs are usually smooth density estimators, unlike histograms which are piecewise constant.
2. KDEs with a well-chosen kernel can give optimal mean-squared error rates even for higher-order smoothness. Histograms on the other hand are only optimal for estimating Lipschitz densities. They cannot exploit more smoothness.

20.2 Kernel Density Estimator

First lets recall the definition of a kernel. For density estimation a kernel will be any function that satisfies:

1. $K(x) \geq 0$,
2. $\int K(x)dx = 1$,
3. $\int xK(x)dx = 0$,
4. $\int x^2K(x)dx = \sigma_K^2 > 0$.

A standard example (that we will use today) is the Gaussian kernel:

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

Now, the KDE with bandwidth h, is defined to be:

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

It is easy to see that:

$$\int \frac{1}{h} K\left(\frac{x - X_i}{h}\right) dx = \int K(x) dx = 1,$$

by a simple change of variables. Roughly, the kernel density estimator is just a smoothed version of the empirical measure, i.e. we just place a smoothed point mass of weight $1/n$ at each data point. The bandwidth controls the amount of smoothing. Let us try to understand the two extremes:

1. **As $h \to 0$:** The KDE will tend to a collection of spikes at the data points. The height of the spikes will tend to ∞.
2. **As $h \to \infty$:** The KDE will tend to a uniform density over the range of the observed data (with some tapering at the ends).

As with kernel regression, the precise choice of the kernel will not matter much, but the bandwidth will be critical.

20.3 Bias and Variance of KDE

Today we will consider the case when the unknown density has a bounded *second* derivative, i.e., at every x we have that

$$\left|\frac{d^2 f}{dx^2}\right| \leq L.$$

As usual we will assume the density is supported on $[0, 1]$. It is not too hard to see that the histogram density estimator's analysis from last lecture cannot be improved, i.e., the histogram still achieves the rate $n^{-2/3}$ even though the density is twice differentiable. We will see today that the kernel density estimator achieves the optimal $n^{-4/5}$ rate. This is the rate $n^{-2\beta/(2\beta+d)}$ with $\beta = 2$ and $d = 1$.

In this case, we will show that:

1. **Bias at x:** $b(x) \approx \frac{1}{2}\sigma_K^2 L h^2$.
2. **Variance at x:** $v(x) \approx \frac{f(x)\int K^2(u)du}{nh}$.

Before we prove this lets examine the consequences. It is easy to check that the MSE satisfies:

$$R(\widehat{f}, f) \leq \frac{L^2\sigma_K^4 h^4}{4} + \frac{\int K^2(u)du}{nh},$$

we will treat $\int K^2(u)du, L, \sigma_K$ all as constants. In this case if we choose, $h \asymp n^{-1/5}$, we obtain the desired rate:

$$R(\widehat{f}, f) \asymp n^{-4/5}.$$

It is important to note that we are still getting a "non-parametric rate", i.e. a rate slower than $1/n$. However, it is also worth noting that this is a better rate than the $n^{-2/3}$ rate for estimating a Lipschitz density. This should be intuitive: estimating a smoother density is easier.

Let us develop some basic properties of the bias and variance of the KDE. Defining:

$$K_h(x, X) = \frac{1}{h}K\left(\frac{x - X}{h}\right),$$

we have that the bias is given as:

$$b(x) = \mathbb{E}[\widehat{f}(x)] - f(x) = \mathbb{E}_{X\sim f}(K_h(x, X)) - f(x),$$

and the variance is:

$$\begin{aligned} v(x) &= \mathbb{E}(\widehat{f}(x) - \mathbb{E}[\widehat{f}(x)])^2 \\ &= \frac{1}{n}\mathrm{Var}\left(K_h(x, X)\right). \end{aligned}$$

With these preliminaries in place we can attempt to bound the bias and variance. I will be a bit sloppy with Taylor series' but everything can be made precise without too much additional effort:

$$\begin{aligned} b(x) &= \mathbb{E}K_h(x, X) - f(x) \\ &= \int \frac{1}{h}K\left(\frac{x - X}{h}\right) f(X)dX - f(x) \\ &= \int K(u)f(x - uh)du - f(x) \\ &\approx \int K(u)[f(x) - uhf'(x) + \frac{1}{2}u^2h^2f''(c)]du - f(x) \\ &\quad \text{where } c \text{ is some point between } x \text{ and } x - uh \\ &\leq \frac{1}{2}Lh^2\int u^2K(u)du = \frac{\sigma_K^2 Lh^2}{2}. \end{aligned}$$

Similarly, we can bound the variance as:

$$\begin{aligned} v(x) &\leq \frac{1}{n}\mathbb{E}[K_h^2(x,X)] \\ &= \frac{1}{n}\int K_h^2(x,X)f(X)dX \\ &= \frac{1}{nh}\int K^2(u)f(x-uh)du \\ &\approx \frac{1}{nh}\int K^2(u)[f(x)-uhf^{'}(c)]du \\ &\text{where } c \text{ is some point between } x \text{ and } x-uh \\ &\leq \frac{f(x)\int K^2(u)du}{nh} + \frac{f^{'}(c)}{n}\int uK^2(u)du \\ &\approx \frac{f(x)\int K^2(u)du}{nh}. \end{aligned}$$

This completes the analysis.

20.4 Cross-validation

In both non-parametric regression and in density estimation we had an unknown bandwidth parameter. In practice, this tuning parameter is chosen using cross-validation. We will only go over this briefly in this class. Lets first discuss this in the context of regression, and then we will discuss density estimation.

Before we discuss cross-validation lets understand a train-test split, i.e. suppose we split the data into two parts we can estimate our regression function for a grid of bandwidths $\{h_1, \ldots, h_M\}$. Now, we want to pick one of these bandwidths.

In this case, we could simply check how well we can predict on the test set, i.e.,

$$\widehat{R}(\widehat{f}_{h_1}, f) = \frac{1}{n_{\text{test}}}\sum_{i=1}^{n_{\text{test}}}(Y_i - \widehat{f}_{h_1}(X_i))^2,$$

and repeat this for each of the bandwidths and then pick the bandwidth that minimizes this. Why does this work? Essentially, train-test splits allow us to *estimate* the risk. It is not hard to see that $\widehat{R}$ is an unbiased estimator of R, and so the procedure makes sense.

Practitioners might be concerned that this is wasteful of the data: you might need a pretty large test set to get a good estimate and this is data that you might have instead used to estimate the model. Also, train-test splits have a "lottery" effect: you might get unlucky in the way you split the data and this could affect results.

K-fold cross-validation tries to get around this by splitting the data into K pieces (think of K as a small number like 5). Now, we repeat the train-test split K times, each time we use $K-1$ pieces for training and the K-th piece for testing. In this way we get, K estimates of the error for each value of the bandwidth. We average these K numbers to get our risk estimate. Finally, we choose the value of the bandwidth that minimizes the risk. The extreme case of K-fold cross-validation is called leave-one-out or n-fold cross-validation. Here we leave out one observation, and try to predict it and then cycle through the observations.

Regression is a case when its relatively clear how to estimate the risk. It is much less clear in the context of density estimation. In density estimation, we want to estimate:

$$R(\widehat{f}_h, f) = \int (\widehat{f}_h(x) - f(x))^2 dx,$$

and pick an h that (approximately) minimizes this. We can expand this out:

$$R(\widehat{f}_h, f) = \int \widehat{f}_h^2(x)dx + \int f^2(x)dx - 2\int \widehat{f}_h(x)f(x)dx,$$

the second term does not depend on h so we can ignore it while selecting h, the first term of course we can calculate, but the third term depends on the unknown true density. The standard procedure is to estimate this using the samples in a leave-one-out fashion. The estimator of the risk is then:

$$\widehat{J}(\widehat{f}_h, f) = \int \widehat{f}_h^2(x)dx - \frac{2}{n}\sum_{i=1}^{n} \widehat{f}_{-i,h}(X_i).$$

Here $\widehat{f}_{-i,h}(X_i)$ is the KDE without the ith observation, with bandwidth h, evaluated at the ith observation. Finally, we choose h to minimize this quantity.