CS731 Spring 2011 Advanced Artificial Intelligence

Nonparametric Density Estimation and Regression

Lecturer: Xiaojin Zhu *jerryzhu@cs.wisc.edu*

The methods in this lecture are nonparametric.

1 Kernel Density Estimation

Let f be a probability density function. Given $x_1 \ldots x_n \sim f$, the goal is to estimate f.

Let us introduce the concept of *smoothing kernel*, not to be confused with the Mercer kernels used in the Reproducing Kernel Hilbert Space sense. A smoothing kernel K is any smooth function satisfying

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

Some common smoothing kernels are

- The Gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$
- The Epanechnikov kernel $K(x) = \frac{3}{4}(1-x^2),\ x \in [-1,1]$, 0 otherwise

Given a kernel K and a positive *bandwidth* h, the *kernel density estimator* is defined to be

$$\hat{f}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - x_i}{h}\right) \quad (4)$$

where the subscript n in $\hat{f}_n(x)$ denotes the training sample size. The intuition is to put a little bump on each training point and sum them up. It turns out that the choice of K is not crucial, but the choice of h is important. In general, we let the bandwidth depend on sample size with the notation h_n.

Theorem 1 *Assume that f is continuous at x, $h_n \to 0$, and $nh_n \to \infty$ as $n \to \infty$. Then $\hat{f}_n(x) \xrightarrow{P} f(x)$.*

Notice that $\hat{f}_n(x)$ is a random variable. Let $R_x = \mathbb{E}(\hat{f}_n(x) - f(x))^2$ be the risk at point x (with squared loss), and $R = \int R_x dx$ be the integrated risk. Then the asymptotically optimal bandwidth is

$$h_n^* = cn^{-1/(4+d)}, \quad (5)$$

and the risk decreases as

$$R = O(n^{-4/(4+d)}), \quad (6)$$

where d is the dimensionality of x. However, the constant c in the optimal bandwidth depends on the unknown density f, rending this theoretical result useless in practice. One typically find the optimal bandwidth by cross validation, as follows.

We will work with the loss function called the *integrated squared error*

$$L(h) = \int (\hat{f}_n(x) - f(x))^2 dx \tag{7}$$

$$= \int \hat{f}_n^{\,2}(x)dx - 2\int \hat{f}_n(x)f(x)dx + const(h). \tag{8}$$

Let

$$J(h) = \int \hat{f}_n^{\,2}(x)dx - 2\int \hat{f}_n(x)f(x)dx \tag{9}$$

be the part of the loss that depends on h. The *cross-validation estimator of risk* is

$$\hat{J}(h) = \int \hat{f}_n^{\,2}(x)dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i}(x_i) \tag{10}$$

where $\hat{f}_{-i}(x_i)$ is the kernel density estimator obtained on the training data excluding x_i. This is leave-one-out cross validation. It turns out that there is a short cut to computing $\hat{J}(h)$ without the need to do leave-one-out:

Theorem 2 *For any $h > 0$,*

$$\mathbb{E}[\hat{J}(h)] = \mathbb{E}[J(h)]. \tag{11}$$

Furthermore,

$$\hat{J}(h) = \frac{1}{n^2 h}\sum_{i,j=1}^{n}\left(G\left(\frac{x_i - x_j}{h}\right) - 2K\left(\frac{x_i - x_j}{h}\right)\right) + \frac{2}{nh}K(0) + O\left(\frac{1}{n^2}\right), \tag{12}$$

where $G(z) = \int K(z-y)K(y)dy$.

For example, when $K = N(0,1)$, $G = N(0,2)$.

2 Nonparametric Regression

Let

$$y_i = r(x_i) + \epsilon_i \tag{13}$$

for $i = 1 \ldots n$, $\mathbb{E}[\epsilon_i] = 0, \mathbb{V}[\epsilon_i] = \sigma^2$. The goal is to estimate $r(x)$ from $(x_1, y_1) \ldots (x_n, y_n)$.

An estimator $\hat{r}$ of r is a *linear smoother* if, for each x, there exists a vector $\gamma(x) = (\gamma(x), \ldots, \gamma(x))^\top$ such that

$$\hat{r}(x) = \sum_{i=1}^{n} \gamma(x) y_i. \tag{14}$$

That is, $\gamma(x)$ is the weight given to y_i in forming the estimate $\hat{r}(x)$.

★ *This does not mean $\hat{r}(x)$ is necessarily linear in x!*

Example 1 *Linear regression is a special case of linear smoother:*

$$\hat{r}(x) = \sum_{d=1}^{D} \beta_d x_d = \sum_{i=1}^{n} \gamma_i(x) y_i, \tag{15}$$

where

$$\gamma(x)^\top = x^\top (X^\top X)^{-1} X^\top. \tag{16}$$

2.1 The Nadaraya-Watson Kernel Estimator

Let $h > 0$ be the bandwidth, and K a smoothing kernel. The *Nadaraya-Watson kernel estimator* is a linear smoother

$$\hat{r}(x) = \sum_{i=1}^{n} \gamma_i(x) y_i \tag{17}$$

where

$$\gamma_i(x) = \frac{K\left(\frac{x-x_i}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right)}. \tag{18}$$

To select the bandwidth in practice, we use cross-validation. The risk under squared loss is

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{r}(x_i) - r(x_i))^2\right). \tag{19}$$

The corresponding leave-one-out score is

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{r}(x_i) - \hat{r}_{-i}(x_i))^2. \tag{20}$$

For each point x_i, the leave-one-out estimator is

$$\hat{r}_{-i}(x) = \sum_{j=1}^{n} \gamma_{-i,j}(x) y_j \tag{21}$$

where

$$\gamma_{-i,j}(x) = \begin{cases} \frac{\gamma_j(x)}{\sum_{k \neq i} \gamma_k(x)} & j \neq i \\ 0 & j = i. \end{cases} \tag{22}$$

That is, $\gamma_{-i,j}(x)$ is a renormalized version of $\gamma_j(x)$ after removing the i-th weight. Again, there is no need to actually compute n different estimates $\hat{r}_{-i}$, because the leave-one-out score can be computed in closed-form.

Theorem 3 *The leave-one-out score can be written as*

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{r}(x_i)}{1 - \gamma_i(x_i)}\right)^2. \tag{23}$$

One then selects the optimal bandwidth by minimizing the score above (could have multiple local minima).

2.2 Local Linear Regression

First, consider the best constant function fit $hatr(x) = a$ to training data:

$$\min_a \frac{1}{n}\sum_i (a - y_i)^2. \tag{24}$$

The solution is simply $a = \frac{1}{n}\sum_i y_i$. Now, consider the weighted version "centered" at x where the i-th training point is associated with a weight $\gamma_i(x) = K((x - x_i)/h)$. The constant fit to this weighted training data is

$$\min_a \frac{1}{n}\sum_i \gamma_i(x)(a - y_i)^2. \tag{25}$$

The solution turns out to be

$$a = \frac{\sum_{i=1}^n \gamma_i(x) y_i}{\sum_{i=1}^n \gamma_i(x)}. \tag{26}$$

Because it is a constant function, in particular at x we have $\hat{r}(x) = a$. This recovers the Nadaraya-Watson kernel estimator.

More importantly, this suggests a way to improve upon the Nadaraya-Watson kernel estimator: instead of assuming a constant function $\hat{r}(u) = a$ in (25), we may assume a family of linear functions, one of each x's neighborhood:

$$\hat{r}_x(u) = a_0(x) + a_1(x)(u - x). \tag{27}$$

We now minimize the following objective:

$$\min_{a_0(x), a_1(x)} \frac{1}{n} \sum_i \gamma_i(x)(a_0(x) + a_1(x)(u - x_i) - y_i)^2. \tag{28}$$

Once the solution $\hat{a}_0(x)$ and $\hat{a}_1(x)$ are found, we have

$$\hat{r}_x(u = x) = \hat{a}_0(x). \tag{29}$$

This is called *local linear regression*. Even though this is the constant term, it is different from a local constant fit (which would be Nadaraya-Watson). See AoS Theorem 5.57 for the closed-form solution.