STATS205: Introduction to Nonparametric Statistics

Lecturer: Tengyu Ma
Scribe: Abigail VanderPloeg

Lecture # 7
May 14th, 2021

7.1 Review and overview

In the previous few lectures, the class have covered neural network methods. While topics such as local linear regression and splines are considered classical methods in nonparametric studies (having been developed 30 years ago), most recently covered lecture topics of kernel methods and neural networks are part of modern nonparametric methods.

This lecture will conclude the class's coverage of neural networks for use on the smaller datasets often seen in nonparametric statistics. Next, the lecture begins discussion of unsupervised learning, which does not have an output or response variable for the input data. Instead, we wish to learn about the underlying distribution of the input data, which can be achieved with CDF and density estimation methods.

7.2 Neural networks: few-shot learning

7.2.1 Transfer learning

Here we continue the lectures' coverage of transfer learning, which is one way in which we can utilize neural networks on small datasets.

Transfer learning involves first training a model on a big data set (as can be done by other researchers such as those at Google, who have released pre-trained models), then fine-tuning the model on the smaller data set. One way in which to fine-tune the model is to remove only the last layer of the model and replace it with a new, randomly initialized last layer that will be fine-tuned. Another approach is to fine-tune the entire network by again replacing the last layer of the model with a new, randomly initialized last layer, and then to fine-tune the parameters of the entire model with a small number of passes over the data.

7.2.2 Few-shot learning

Few-shot learning is utilized in an even more extreme case than transfer learning, when the dataset is even smaller. The learning setting involves training data of, say, N examples and l classes, where both N and l are very big (ImageNet [DDS$^+$09], a standard example, has $N = 1.2$M and $l = 10^3$).

At test time, however, we are only given a small number of examples $(\tilde{x}^{(1)}, \tilde{y}^{(1)}), ..., (\tilde{x}^{(nk)}, \tilde{y}^{(nk)})$ drawn independently and identically distributed from a distribution P_{test}. There are k new labels or classes and n images per each new class. In this scenario, n is very small (such as $n = 5$), and k could be bigger. Such a setting is called a "k-way n-shot setting." With this limited data for each new class, the goal is to classify the examples from P_{test} with one of the k labels. In few-shot learning settings, the feature dimension is typically large (on the order of 10^3), which makes it difficult to fine-tune a model to the small test dataset, as was done in transfer learning, because overfitting becomes likely.

7.2.3 Nearest neighbor algorithms using features

A simple but competitive algorithm in few-shot learning settings is nearest neighbor methods using features, with steps as follows:

1. Pretrain neural networks on the large pretraining dataset, which results in an output of $a^\top \phi_W(x)$.

   (a) In this case, we enforce $\phi_W(x)$ to have a norm of 1 during training. (This is helpful in order to not have dramatically different norms for different examples, and is likely applied by research teams such as Google who release pretrained neural networks.)

   This can be done in one of two ways by changing the parameterization

   $$\phi_W(x) = \text{normalize}(NN_W(x)) = \frac{NN_W(x)}{\|NN_W(x)\|_2}.$$

   for $NN_W(x)$ as the standard feed-forward NN. This normalization is a sequence of elementary operations, which can be done efficiently (such as computing $\|NN_W(x)\|_2$) and allows for efficient gradient calculations with auto-differentiation in backpropagation. Performing this operation then implies that $\|\phi_W(x)\|_2 = 1$.

2. At test time, we utilize an one-nearest neighbor algorithm. (Here, we predict based on the single nearest neighbor rather than a combination of the k nearest as used in k-nearest neighbors.) Generally, given an example x, we wish to predict the output label y. The steps are as follows:

   (a) Compute $\phi_w(x)$.

   (b) Find nearest neighbor in $\{\phi_w(\tilde{x}^{(1)}), ..., \phi_w(\tilde{x}^{(nk)})\}$.

   The "nearness" is quantified according to ℓ_2 distance or cosine distance. ℓ_2 distance calculates $d(a,b) = \|a-b\|_2$. Squaring this calculation results in

   $$\|a-b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 - 2\langle a, b\rangle. \tag{7.1}$$

   which, given the unit norms enforced on outputs $\phi_w(\tilde{x}^{(i)})$, can be simplified to

   $$\|a-b\|_2^2 = 2 - 2\langle a, b\rangle. \tag{7.2}$$

   which is a constant shift from the cosine distance $2\langle a, b\rangle$, the cosine angle between two vectors a, b.

   Let's suppose that the nearest neighbor is $\phi_w(\tilde{x}^{(j)})$.

   (c) Assign the output label of $\tilde{y}^{(j)}$, or the label of the "nearest neighbor," to the example x.

7.3 Unsupervised learning: estimating the CDF

Now we return to classical methods, with one-dimensional problems that can be described by CDFs and PDFs (rather than the high-dimensional feature sets of machine learning).

7.3.1 Setup of CDF estimation

Let F be the CDF of some distribution over $\mathbb{R}$. Additionally, let us observe n examples from this distribution

$$X_1, ..., X_n \overset{\text{iid}}{\sim} F.$$

Our goal is to estimate the underlying function $F(x)$.

From earlier lectures, we can recall that a property of the CDF is that if $X \sim F$, then $\boxed{F(x) = \Pr[X \leq x]} \in [0, 1]$ (the probability that we observe an output less than or equal to x). $F(x)$ is monotonically increasing, and an example of CDF is shown in Fig. 7.1.

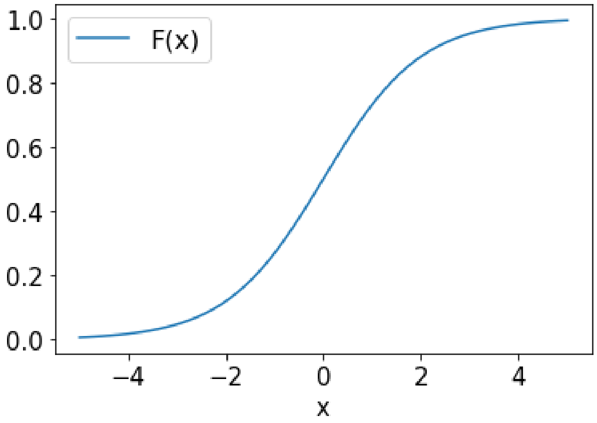

Figure 7.1: Example of CDF, the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$.

7.3.2 Empirical estimators

Given the empirical examples $X_1, ..., X_n \overset{\text{iid}}{\sim} F$, we can estimate the underlying CDF, $F(x)$, by evaluating how often $X_i \leq x$ for a given x and $1 \leq i \leq n$ (how often our examples are less than or equal to the input value). Thus, we can consider the empirical estimator $\hat{F}_n(x)$ defined as

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(X_i \leq x). \tag{7.3}$$

We can make the following observations on the function $\hat{F}_n(\cdot)$:

- $\hat{F}_n(\cdot)$ as a function is called an **empirical distribution function**.
- $\hat{F}_n(\cdot)$ is a step function which only takes values in $\left[0, \frac{1}{n}, ..., 1\right]$. This is because $\hat{F}_n(\cdot)$ multiplies $\frac{1}{n}$ by a sum of n integers with values $\in \{0, 1\}$, which then implies that $0 \leq \hat{F}_n(x) \leq 1$.
- $\hat{F}_n(\cdot)$ is a CDF itself, and is in fact the CDF of the uniform distribution over $\{X_1, ..., X_n\}$.

Using the previous example of $F(x)$ (the sigmoid function), we can illustrate what form the estimator will take for an example with $n = 4$ data points in Fig. 7.2.

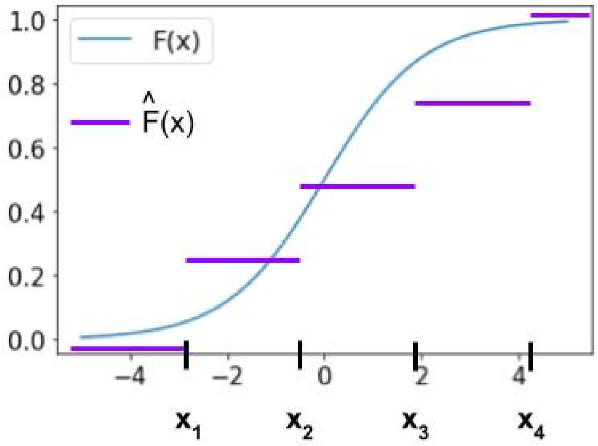

Figure 7.2: Example of empirical CDF.

Although this class will not overview the in-depth theory, it is possible to show for a given x that as the number of examples $n \to \infty$, this estimator $\hat{F}_n(x)$ converges to the underlying distribution function $F(x)$.

In the extreme lower and/or upper range of inputs x, the density of data points is close to 0 (since $F(x)$ is flat), and the estimator does not change to transition to the next"increased step" often given the relative lack of examples in these regions. In the opposite scenario in a region where $F(x)$ increases sharply, there are more examples in this region, and the CDF will increase (transition to the next step) more quickly.

The following section involves analysis of simple theorems related to the estimator $\hat{F}_n(x)$.

Theorem 7.1. *For any fixed value of x, the expectation of the empirical estimator, $\mathbb{E}\left[\hat{F}_n(x)\right]$, satisfies*

$$\mathbb{E}\left[\hat{F}_n(x)\right] = F(x), \tag{7.4}$$

with randomness over the choice of $X_1, ..., X_n$. This means that $\hat{F}_n(x)$ is an unbiased estimator of $F(x)$.

Proof This result can be seen by evaluating the $\mathbb{E}\left[\hat{F}_n(x)\right]$, since

$$\begin{aligned}
\mathbb{E}\left[\hat{F}_n(x)\right] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(X_i \leq x)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\mathbb{1}(X_i \leq x)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \Pr(X_i \leq x) \\
&= F(x).
\end{aligned} \tag{7.5}$$

We can also evaluate the variance of $\hat{F}_n(x)$ as

$$\text{Var}\left[\hat{F}_n(x)\right] = \frac{F(x)(1 - F(x))}{n}. \tag{7.6}$$

and observe that the numerator is bounded in the range $[0, 1]$ while the denominator becomes larger (approaching infinity) with more and more examples. This means that, with more examples, the variance becomes smaller, and the (unbiased) estimate becomes more accurate. Therefore, we see that

$$\text{Var}\left[\hat{F}_n(x)\right] \xrightarrow{p} F(x). \tag{7.7}$$

or that our estimator converges in probability to the true underlying function (with more and more examples).

Theorem 7.2 (Glivenko-Cantelli)**.**

$$\sup_x |\hat{F}_n(x) - F(x)| \xrightarrow{a.s.} 0. \tag{7.8}$$

This means that the supremum of $|\hat{F}_n(x) - F(x)|$ almost surely converges to 0.

Expressed in words, the Glivenko-Cantelli theorem ensures that the estimator converges to the true underlying distribution over the entire function.

Remark 7.3. While smoothing may produce an estimator that looks more similar to a true CDF, the step-wise estimator described is optimal to ensure the convergence of the estimator to the underlying CDF, and smoothing is therefore not necessary. However, this estimator has a zero derivative at most places and no derivative in others, which makes it inapplicable when trying to estimate the density of the data (which is the derivative of the CDF).

7.3.3 Estimating functionals of the CDF

Consider $T(F)$, which is a function of the CDF F. For examples, $T(F)$ could be any of the following functions of F that represent a property of the CDF:

- Mean of the distribution F.
- Variance of F.
- Skewness of F (measuring the asymmetry of the CDF about the mean).
- Quantile of F.

A **plug-in estimator** uses $T(\hat{F}_n)$ as the estimator (directly plugging in the estimator for F into the functional). Under certain conditions (satisfied with the functionals listed above), $T(\hat{F}_n) \to T(F)$. Note that more "abnormal" functions of F (such as the derivative) do not satisfy these conditions.

7.4 Unsupervised learning: density estimation

As before, we assume that we have data points

$$X_1, ..., X_n \overset{iid}{\sim} F.$$

with F as the CDF of the underlying data distribution.

The PDF, or density, is $f = F'$. In other words, it is the derivative of the CDF. In density estimation, the goal is to estimate the underlying density function f from the data $X_1, ..., X_n$.

In order to do so, we utilize technical ideas similar to regression problems. In this case, instead of predicting over an output y, we aim to predict $f(x)$. However, we do not directly observe $f(x)$ in any of the data points. This problem is separate from the problem of empirical CDF estimates, since we cannot simply take $\hat{f} = \hat{F}_n(x)'$ for an empirical CDF estimator $\hat{F}_n(x)$ because this CDF estimator is a step function which has a derivative of 0 at most inputs and infinity at the location of the data points.

7.4.1 Measuring performance of density estimators

There are several different ways in which to measure and evaluate the performance of density estimators. The most common way in which we can measure the performance of a density estimator $\hat{f}$ in accurately estimating the true density f is by calculating the **integrated mean square error**:

$$R(\hat{f}, f) = \int \left(\hat{f}(x) - f(x)\right)^2 dx.$$

For a one-dimensional problem, this can be seen as a natural extension the mean squared error.

Another way to calculate the risk in order to measure performance is to calculate the ℓ_1 **integrated risk**, also known as the **total variation (TV) distance** between the two distributions f and $\hat{f}$:

$$R_{\ell_1}(\hat{f}, f) = \int \left|\hat{f}(x) - f(x)\right| dx. \tag{7.9}$$

Throughout the rest of the lecture, we will utilize the mean squared error as a metric to evaluate density estimator performance.

Remark 7.4. The mean squared error is not very useful in high dimensions. (If $f = \hat{f}$, then the mean squared error will evaluate to 0, but this error generally does not scale well in higher dimensions.)

7.4.2 Mean squared error in high-dimensional spaces

Consider the d-dimension problem as follows. We assume that f is a spherical Gaussian $\sim \mathcal{N}(0, I)$. It follows that

$$f(x) = \frac{1}{\left(\sqrt{2\pi}\right)^d} \cdot \exp\left(-\frac{1}{2}\|x\|_2^2\right). \tag{7.10}$$

Some key observations we can make about this density function are:

- f is a density and therefore

$$f(x) \geq 0. \tag{7.11}$$

- We can evaluate the point with the largest density as

  $$\sup_x f(x) = f(0) \leq \frac{1}{\left(\sqrt{2\pi}\right)^d}. \quad \text{(an inverse exponential)} \tag{7.12}$$

  This means that, in high dimensional-spaces, we are aiming to predict very small values, which becomes an issue that is exacerbated in the integrated mean squared error calculation.

Now, consider some $\hat{f}$ that approximates f reasonably well. Because we have shown the output of $f(x)$ to be less than the inverse exponential $\frac{1}{\left(\sqrt{2\pi}\right)^d}$, we can reasonably expect that $\hat{f} \leq \frac{1}{\left(\sqrt{2\pi}\right)^d}$ for most x.

We can evaluate the integrated mean squared error between the described f and $\hat{f}$ as follows:

$$\begin{aligned}
R(\hat{f}, f) &= \int \left(\hat{f}(x) - f(x)\right)^2 dx \\
&\leq \int \left|\hat{f}(x) - f(x)\right| \cdot \left(|\hat{f}(x)| + |f(x)|\right) dx \\
&\lesssim \frac{2}{\left(\sqrt{2\pi}\right)^d} \int \left|\hat{f}(x) - f(x)\right| dx \\
&\lesssim \frac{2}{\left(\sqrt{2\pi}\right)^d} \int \left(\hat{f}(x) + f(x)\right) dx \qquad (f \text{ and } \hat{f}(x) \text{ are positive}) \\
&\leq \frac{4}{\left(\sqrt{2\pi}\right)^d}.
\end{aligned}$$

In conclusion, if we have an estimator $\hat{f}$ such that $\hat{f}(x) \leq \frac{1}{\left(\sqrt{2\pi}\right)^d}$ $\forall x$, then

$$R(\hat{f}, f) \leq \frac{4}{\left(\sqrt{2\pi}\right)^d}. \tag{7.13}$$

and thus f and $\hat{f}$ need not be close by any means for the error to be proportionally very small as an inverse exponential. Note that the TV distance can also not be very meaningful in high dimensions. Generally, the distance between two distributions in high-dimensional space is non-trivial. There are, however, alternatives used that offer slightly better method of measuring the performance of density estimators in high dimensions. One such alternative is the KL divergence metric. However, the KL divergence can still result in a large error for very similar distributions. Take, for example, two distributions $P_1 = \mathcal{N}(0, I)$ and $P_1 = \mathcal{N}(\mu, I)$, where μ is a small vector. Then, the KL divergence can become very large. Wassertein distance is another alternative method that incorporates the geometry into the calculation, and performs better for examples such as distributions which are two point masses that are very close to one another, such as pictured below:

7.4.3 Mean squared error and other errors in low-dimensional spaces

Suppose that $d = 1$ and the situation is thus low-dimensional In this case, use of the mean-squared error is ok (as well as other distance metrics discussed). Going forward in lecture, we will primarily focus on discussing one-dimensional scenarios.

7.4.4 Bias-variance tradeoff

Just as we evaluated the bias-variance tradeoff in regression problems, we can calculate the bias-variance tradeoff with expectation of the integrated mean square error risk over the randomness of $X_1, ..., X_n$ as

$$\mathbb{E}\left[\int \left(f(x) - \hat{f}(x)\right)^2 dx\right] = \int \left(\mathbb{E}\left[(f(x) - \hat{f}(x))^2\right]\right) dx.$$

Because, for a random variable Z, $\mathbb{E}\left[Z^2\right] = \left(\mathbb{E}\left[Z\right]\right)^2 + \mathrm{Var}(Z)$, then we can decompose the bias-variance tradeoff as:

$$\begin{aligned}
\mathbb{E}\left[\left(f(x) - \hat{f}(x)\right)^2\right] &= \left(\mathbb{E}\left[f(x) - \hat{f}(x)\right]\right)^2 + \mathrm{Var}\left[f(x) - \hat{f}(x)\right] \qquad (7.14) \\
&= \left(f(x) - \mathbb{E}\left[\hat{f}(x)\right]\right)^2 + \mathrm{Var}\left[-\hat{f}(x)\right] \qquad (f(x) \text{ is a constant}) \\
&= \left(\mathbb{E}\left[\hat{f}(x)\right] - f(x)\right)^2 + \mathrm{Var}\left[\hat{f}(x)\right].
\end{aligned}$$

Thus, we can continue to evaluate $\mathbb{E}\left[\int \left(f(x) - \hat{f}(x)\right)^2 dx\right]$ as

$$\mathbb{E}\left[\int \left(f(x) - \hat{f}(x)\right)^2 dx\right] = \int \boxed{\left(\mathbb{E}\left[\hat{f}(x)\right] - f(x)\right)^2} dx + \int \boxed{\mathrm{Var}\left[\hat{f}(x)\right]} dx. \qquad (7.15)$$

where the term in the blue box is the bias term and the term in the red box is the variance (although sometimes each term including the integral is regarded as the bias and variance respectively). This clear distinction between bias and variance is a property of the integrated mean squared error loss.

7.4.5 Histograms

The first algorithm that we will discuss is the histogram algorithm, which is an analog of regressograms. Recalling from previous lectures, we remember that the process of solving a regressogram problem involves

1. binning the input domain
2. fitting constant density functions across each bin

These two steps are also used in the histogram algorithm.

If we assume $X \in [0, 1]$, then we can create bins $B_1, ..., B_m$ within the input range, and we generate a constant function across each bin, $z_1, ..., z_m$. To set up the notation we will utilize, we first define

$$\text{length of each bin} \triangleq h = \tfrac{1}{m}.$$

Furthermore, let Y_i equal the number of observations (data points) in each bin B_i. Then, we define $\hat{p}$ as

$$\hat{p}_i = \tfrac{Y_i}{n} = \text{the fraction of data points in bin } B_i.$$

The value of $z_i \propto \hat{p}_i$, but we need to normalize our $\hat{p}_i$ in order to achieve a proper density function.

In order to form a proper density from $z_1, ..., z_n$, we require that

$$\int \hat{f}(x)dx = \sum_{i=1}^{m} \int_{B_i} \hat{f}(x)dx = \sum_{i=1}^{m} h \cdot z_i = 1. \tag{7.16}$$

Suppose that $z_i = c \cdot \hat{p}_i$. Then, using the property that $\sum\limits_{i=1}^{m} \hat{p}_i = 1$, we see that

$$\int \hat{f}(x)dx = h \cdot c \sum_{i=1}^{m} \hat{p}_i = 1 \quad \Longrightarrow \quad c = \frac{1}{h \cdot \sum\limits_{i=1}^{m} \hat{p}_i} = \frac{1}{h} \quad \Longrightarrow \quad z_i = \frac{\hat{p}_i}{h}, \tag{7.17}$$

which tells us that each z_i is computed as the fraction of the points in the bin B_i normalized by the size of the bin. More succinctly, we can write

$$\hat{f}(x)dx = \sum_{j=1}^{n} z_j \mathbb{1}(x \in B_j) = \sum_{j=1}^{n} \frac{\hat{p}_j}{h} \mathbb{1}(x \in B_j). \tag{7.18}$$

7.4.6 Bias-variance of histogram

In the case of the histogram algorithm, we can explicitly compute the bias and variance. First, for use in our calculation of the bias and variance for $x \in B_j$, we can evaluate the expectation of the estimator on x as

$$\begin{aligned}
\mathbb{E}\left[\hat{f}(x)\right] &= \mathbb{E}\left[\frac{\hat{p}_j}{h}\right] \\
&= \frac{1}{h} \cdot \mathbb{E}\left[p_j\right] \\
&= \frac{1}{h} \Pr\left[X \in B_j\right] \\
&= \frac{1}{h} \cdot \int_{B_j} f(u)du \\
&= \frac{p_j}{h}.
\end{aligned}$$

with $p_j \triangleq \Pr\left[X \in B_j\right]$ defined as the probability of a random sample being in bin B_j.

Bias

Thus, we can evaluate the bias as

$$\text{Bias} = \left(f(x) - \mathbb{E}\left[\hat{f}(x)\right]\right)^2 = \left(f(x) - \frac{p_j}{h}\right)^2. \tag{7.19}$$

When h is infinitesimally small, each bin becomes a very very small window. Knowing that $\int_{B_j} f(u)du$ can thus be approximated as $h \cdot f(x)$ for any $x \in B_j$ allows us to evaluate

$$\frac{p_j}{h} = \frac{1}{h} \int_{B_j} f(u)du \approx \frac{1}{h} \cdot h \cdot f(x) = f(x), \tag{7.20}$$

and thus the bias goes to 0 as $h \to 0$.

Variance

When evaluating the variance of the estimator for $x \in B_j$,

$$\operatorname{Var}\left(\hat{f}(x)\right) = \frac{1}{h^2} \operatorname{Var}\left(\hat{p}_j\right) \tag{7.21}$$

We can note that, for the number of points that fall into a given bin B_j as $n\hat{p}_j$,

$$\begin{aligned} n\hat{p}_j &= Y_j \sim \text{Binomial}(n, p_j) \\ &= \sum_{j=1}^{n} \mathbb{1}(X_i \in B_j), \end{aligned} \tag{7.22}$$

where $\mathbb{1}(X_i \in B_j)$ follows the Bernoulli distribution $B(p_j)$.

Thus, we can calculate the variance of $n\hat{p}_j$ as

$$\operatorname{Var}(n\hat{p}_j) = \sum_{i}^{n} \operatorname{Var}(\mathbb{1}(X_i \in B_j)) = n \cdot p_j(1 - p_j), \tag{7.23}$$

and we can therefore evaluate $\operatorname{Var}(\hat{p}_j)$ as

$$\operatorname{Var}(\hat{p}_j) = \frac{1}{n^2} \operatorname{Var}(n\hat{p}_j) = \frac{p_j(1 - p_j)}{n}, \tag{7.24}$$

allowing us to evaluate $\operatorname{Var}(\hat{f}(x))$ as

$$\operatorname{Var}(\hat{f}(x)) = \frac{1}{h^2} \operatorname{Var}(\hat{p}_j) = \frac{1}{h^2 \cdot n} p_j(1 - p_j). \tag{7.25}$$

By analyzing the above result, we see that when $h \to 0$, then $\operatorname{Var}(\hat{f}(x)) \to \infty$, and when $n \to \infty$, then $\operatorname{Var}(\hat{f}(x)) \to 0$. (This is consistent with the results we saw for regression problems).

Theorem 7.5. *Suppose f' is absolutely continuous and that $\int f'(u)^2 du < \infty$. Then, for the histogram estimator $\hat{f}$,*

$$R(\hat{f}, f) = \boxed{\frac{h^2}{12} \int (f'(u))^2 du} + \boxed{\frac{1}{nh}} + \mathcal{O}(h^2) + \mathcal{O}(\frac{1}{n}), \tag{7.26}$$

where the term in the blue box is the bias term and the term in the red box is the variance. (The bias depends upon the Lipschizes of f, meaning that f must be smooth.)

7.4.7 Finding the optimal h^*

The best value for h is the minimizer of $R(\hat{f}, f)$ over h, ignoring higher order terms. Using the results from Theorem 7.5, we see that

$$\begin{aligned} h^* &= \underset{h}{\operatorname{argmin}} \left[\frac{h^2}{12} \int (f'(u))^2 du + \frac{1}{nh}\right] \\ &= \frac{1}{n^{1/3}} \cdot \left(\frac{6}{\int (f'(u))^2 du}\right)^{1/3}, \end{aligned} \tag{7.27}$$

which, most importantly, informs us that the rate that $h \propto \frac{1}{n^{1/3}}$. Plugging in this choice of h^*, we get

$$R(\hat{f}, f) \sim \frac{c}{n^{2/3}}. \tag{7.28}$$

which shows that the convergence rate of the error as $n \to \infty$.

7.4.8 Proof sketch of Theorem 7.5

When working through linear regression problems in previous lectures, we never derived these theorems. However, we can prove Theorem 7.5 as shown in this proof sketch. We have shown that $\hat{f}(x) = \frac{\hat{p}_j}{h}$ if $x \in B_j$.

We also saw that we can evaluate the expectation of the estimator $\hat{f}(x)$ at a point $x \in B_j$ as

$$\mathbb{E}\left[\hat{f}(x)\right] = \frac{p_j}{h} = \frac{1}{h} \cdot \int_{B_j} f(u)du. \tag{7.29}$$

Before, we had roughly approximated $f(u) \approx f(x)$. However, we can more explicitly obtain an expression for $f(u)$ using a first order Taylor expansion:

$$f(u) = f(x) + (u - x)f'(x) + \mathcal{O}(h^2), \tag{7.30}$$

since $|u - x| \leq h$ we have that the higher-order terms scale as $\mathcal{O}(h^2)$.

Therefore, we can further simplify our calculation of $\mathbb{E}\left[\hat{f}(x)\right] = \frac{1}{h}\int_{B_j} f(u)du$ by evaluating

$$\begin{aligned}
\int_{B_j} f(u)du &= \int_{B_j} \left(f(x) + (u - x)f'(x) + \mathcal{O}(h^2)\right) dn \\
&= h \cdot f(x) + f'(x) \int (u - x)du + \mathcal{O}(h^2) \cdot h \\
&= h \cdot f(x) + f'(x) \cdot \mathcal{O}(h^2) + \mathcal{O}(h^3),
\end{aligned}$$

given that $|u - x| \leq h$ and the size of B_j is similarly bounded as $\leq h$. Thus, we can evaluate the expectation of the estimator as

$$\mathbb{E}\left[\hat{f}(x)\right] = \frac{1}{h} \cdot \int_{B_j} f(u)du = f(x) + f'(x) \cdot \mathcal{O}(h) + \mathcal{O}(h^2). \tag{7.31}$$

Bias

Given our previous calculation of $\mathbb{E}\left[\hat{f}(x)\right]$, we can evaluate the bias as

$$\begin{aligned}
\left(f(x) - \mathbb{E}\left[\hat{f}(x)\right]\right)^2 &= \left(f'(x) \cdot \mathcal{O}(h) + \mathcal{O}(h^2)\right)^2 \\
&= h^2 \left(f'(x) + \mathcal{O}(h)\right)^2 \\
&= \mathcal{O}(h^2) f'(x)^2 + \mathcal{O}(h^3).
\end{aligned} \tag{7.32}$$

And thus, the integrated bias can be calculated as

$$\int \left(f(x) - \mathbb{E}\left[\hat{f}(x)\right]\right)^2 dx = \mathcal{O}(h^2) \cdot \int f'(x)^2 dx + \mathcal{O}(h^3). \tag{7.33}$$

Variance

Given our previous calculation of $\mathbb{E}\left[\hat{f}(x)\right]$, we can evaluate the integrated vairance as

$$\begin{aligned}\int \operatorname{Var}(\hat{f}(x))dx &= \sum_{j=1}^{m} \int_{B_j} \operatorname{Var}(\hat{f}(x))dx \\ &= \sum_{j=1}^{m} \frac{p_j(1-p_j)}{h^2 \cdot n} \cdot h \\ &\le \sum_{j=1}^{m} \frac{p_j}{h^2 \cdot n} \cdot h \\ &= \frac{1}{nh} \sum_{j=1}^{m} p_j \\ &= \frac{1}{nh}.\end{aligned} \tag{7.34}$$

Notice that the variance does not depend on f.

Bibliography

[DDS$^+$09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, *Imagenet: A large-scale hierarchical image database*, 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.