Chapter 7

Non-parametric Density Estimation

The objective is to estimate a probability density f_X over the real line (or a subset thereof) from a set of points $\mathcal{D} = \{x_1, x_2, \ldots x_n\}$.

7.1 ML density estimation

The likelihood of $\mathcal{D}$ is

$$L(f_X|\mathcal{D}) \;=\; \prod_{i=1}^{n} f_X(x_i) \tag{7.1}$$

and the log-likelihood

$$l(f_X|\mathcal{D}) \;=\; \sum_{i=1}^{n} \log f_X(x_i) \tag{7.2}$$

Maximizing the above over all functions yields (without proof)

$$\hat{f}_X^{ML} \;=\; \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i} \tag{7.3}$$

where $\delta_{\bar{x}}$ is the Dirac "function"

$$\delta_{\bar{x}} \;=\; \left\{ \begin{array}{ll} \infty & \text{for } \; x = \bar{x} \\ 0 & \text{otherwise} \end{array} \right. \tag{7.4}$$

By convention

$$\int_{-\infty}^{\infty} \delta_x(t)dt \;=\; 1 \tag{7.5}$$

$$\int_{-\infty}^{\infty} \delta_x(t) g(t) dt = g(x) \tag{7.6}$$

Hence, the ML estimate of f is a weighted sum of δ spikes placed at the sampled points. Such an estimate is counterintuitive - we know that most densities aren't spikes! It is also completely impractical: if we used the model $\hat{f}_X$ for prediction then we would predict that all the future samples from f_X will lie at the locations $x_1, x_2, \ldots x_n$ and nowhere else!

Therefore, instead of maximizing the likelihood over all possible density functions we will impose some restrictions corresponding to our intuition of a "realistic" density. One way to do that is to decide on a model class (e.g uniform, normal) and find the ML estimate in that class. This is called *parametric* density estimation. The alternative is the *non-parametric* way. We will study two non-parametric models: the *histogram* and the *kernel density estimator*.

7.2 Histograms

To construct a histogram, we partition the domain of the distribution into n_b *bins* of equal width h. Then we count the number of points $n_i,\ i = 1, \ldots n_b$ in each bin and we define f_X to be equal to the $\frac{n_i}{nh}$ over bin i. Note that this way f_X is a piecewise constant function that integrates to 1. The density is zero in all bins that contain no points.

Figure 7.1 shows examples of histograms. The choice of the *bin width h* influences the aspect of the histogram and its *variance* w.r.t to the sample. This is an illustration of the *bias-variance trade-off* that will be discussed further on. Another source of variation in a histogram is the choice of *bin origins*. If all bins are shifted by an amount $\Delta < h$, the numbers n_i may change, because bin boundaries are shifted. The latter variability is entirely due to artefacts - having nothing to do either with the data or with other reasonable assumptions about nature. It is an example of problem to be avoided by a "good" statistical model. The next section will show a class of models which is clearly superior to histograms in all respects. Therefore, histograms are not recommended and should not be trusted except with caution, for a very qualitative look at the data.

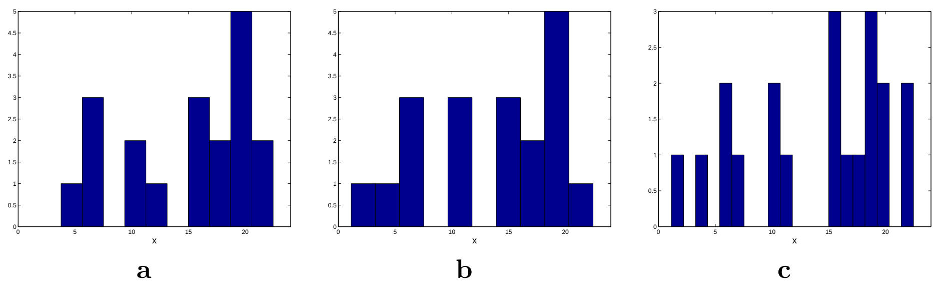


Figure 7.1: Three histograms (note that they are *unnormalized*, i.e don't sum to 1). The first two are over data sets that differ in only 1 point. The third is from the first data set but has twice as many bins.

7.3 Kernel density estimation

This method constructs the estimate of f_X by placing a "bump" at each data point and then summing them up.

$$f_X(x) \;=\; \frac{1}{nh}\sum_{i=1}^{n} k(\frac{x-x_i}{h}) \qquad (7.7)$$

The "bump" function $k(.)$ is called a *kernel* and the parameter h is the *kernel width*. Figure 7.3 shows three typical kernels. A kernel should always be non-negative and satisfy the following conditions

1. $\int_{-\infty}^{\infty} k(x)dx \;=\; 1$ integrate to 1
2. $\int_{-\infty}^{\infty} xk(x)dx \;=\; 0$ "centered" at 0
3. $\int_{-\infty}^{\infty} x^2k(x)dx \;<\; \infty$ "finite variance"

Usual kernels are also symmetric around 0, have a maximum at 0 and decrease monotonically away from the origin. If a kernel is 0 outside a neighborhood of the origin, then we say that it has *compact support*. The uniform and the Epanechnikov kernel have compact support, while the Gaussian kernel doesn't. The Epanechnikov kernel has optimal variance (something we'll discuss next).

Sometimes, the last condtion is replaced with $\int_{-\infty}^{\infty} x^2k(x)dx \;=\; 1$. This condition insures that different kernels are comparable w.r.t *width*.

Note that $\hat{f}_X$ defined above is a ML estimate. If we model f_X by summing n bumps of fixed shape and width and maximize the likelihood of the data w.r.t the bumps positions, then, if the width of the bumps is small enough, the optimal placement centers each bump on a data point.

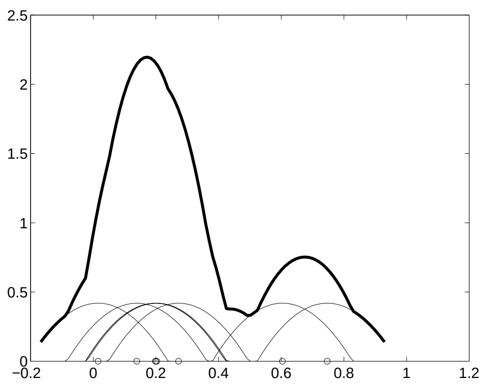


Figure 7.2: Kernel density estimation. A kernel is placed on each data point; the density is (proportional to) the sum of all kernels.

What happens if we also allow the kernel width to vary? Decreasing h will have the effect of increasing the likelihood. It will also make the estimated density look "spikier". The "optimal" h will be zero in which case the original, unconstrained ML solution with its n δ spikes is recovered. This shows that kernel density estimation is ML estimation with a restriction on how "spiky" we allow our solution to be.

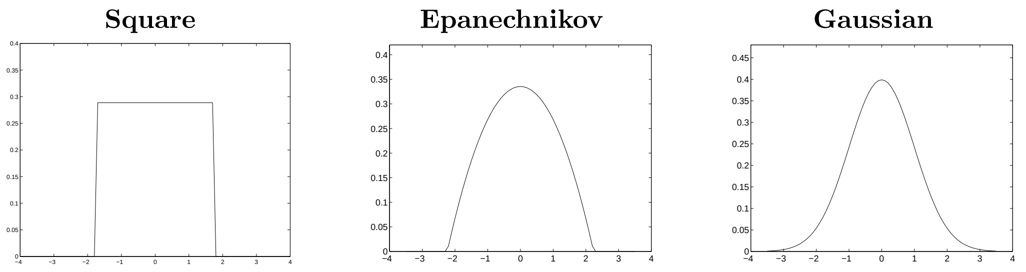
Another way of looking at kernel density estimation is as a convolution: the kernel density estimator represents the convolution of the kernel with a set of spikes placed at the data points.

$$\hat{f}_X = \frac{1}{h}\hat{f}_X^{ML} * k \tag{7.8}$$

Choosing a kernel A compactly supported kernel has computational advantages: $k(x)$ being zero outside a finite interval we will only need to compute the non-zero terms in 7.7. If we assume that the original density is defined only on an interval of the real axis (such an f_X is called *compactly supported*), then it also makes sense to choose a kernel with compact support.

On the contrary, the Gaussian kernel assures that $\hat{f}_X$ is non-zero everywhere. To compute such an $\hat{f}_X$ at one point x we have to evaluate k in n points, which can be quite a burden if the data set is large.

The exact shape of the kernel is not critical in practice. Therefore in the next examples we shall only use the Gaussian kernel. Far more important than the kernel shape is the choice kernel width h that controls the *bias-variance trade-off*.



$k(x) = \frac{1}{2\sqrt{3}}$ for $|x| \leq \sqrt{3}$ $\quad$ $k(x) = \frac{3}{4\sqrt{5}}(1 - x^2/5)$ for $|x| \leq \sqrt{5}$ $\quad$ $k(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$

a **b** **c**

Figure 7.3: Examples of kernel functions: (a) the square kernel; (b) the Epanechnikov kernel[2]; (c) the Gaussian kernel. The area under each kernel equals 1. Note that they have different widths and different maximum heights; therefore we expect different amounts of smoothing for the same h.

7.4 The bias-variance trade-off

Bias. The bias refers to the capacity of a family of functions (in this case the family of kernel density estimators with a given kernel k and a given h) to fit the data. The better the fit to the data, the lower the bias. For example, estimating the density with delta spikes models the data perfectly, hence has 0 bias. On the other hand, if we use a kernel density estimator with h large, then the bumps are wide and their peaks are flat. No matter if the original density was flat or not, the estimator will look flat. Hence densities that have sharp peaks can't be approximated well with a large h. We say that an estimator with large h is *biased* toward slowly varying densities. In the case of the kernel density estimators, the bias increases with h.

Because h controls the smoothness of the resulting density estimate, is also called a *smoothing parameter*. Large bias toward some kind of solution implies potentially large estimation errors, i.e large differences between our solution and the "true" density that generated the data. Therefore we usually want to have low bias.

Variance measures how much the estimated density changes due to the randomness of the data set. The maximum variance is attained for $h = 0$ - the unconstrained ML estimate. Indeed, if the data set contains a data point at a then the density there is ∞; if we draw another sample which doesn't contain a data point exactly at a, then the density in a will be 0. A variation from infinite to 0 due to an infinitesimal change in the data! As h becomes larger, the density becomes less sensitive to small perturbations in the data, therefore the variance of the estimate will decrease with h.

Since we want an estimated density that fits well *all* the possible data sets, a low variance is what we should aim for.

Considering now what we know about bias, we see that minimizing bias (which means reducing h) and minimizing variance (by increasing h) are conflicting goals. This is the *bias-variance* trade-off: finding a value of the kernel width h that is reasonable both for bias and for variance.

The effect of the sample size n. Intuitively, it is harder to fit more data points than less data points. Thus, the bias will in general not decrease when the sample size n increases. For the case of kernel density estimates, the bias doesn't change with n. The variance however will decrease with n, therefore it is at our advantage to obtain as much data as possible. With enough data to compensate for the variance, we can afford using a small h to reduce the bias as well. In conclusion, a larger sample size n has a beneficial effect on the overall quality of the estimate.

How should h be changed with n? Theoretical studies show that the optimal kernel width should be

$$h \propto \frac{1}{n^{\frac{1}{5}}} \tag{7.9}$$

Example 7.1 *Traffic on the I-90 bridge*

Assume that we have placed a sensor on the I-90 bridge that records the moment a car passes in front of it. The data file `fig_h7_traffic.dat` *is a (***fictitious!***) recording of such data over 24 hours. The same data is plotted in figure 7.5 on the time axis (from 0 to 24 hrs). We will visualize the it by constructing a kernel density estimator.*

The figure 7.6 shows the density estimate using 3 different kernels with the same width $h = 1$. The rectangular kernel is easy to recognize by its ruggedness, the other two plots that are very close together are the Gaussian kernel and the Epanechnikov (call it E.!) kernel. Note two things: First, the Gaussian and E. kernels give almost indistinguishable estimates. It doesn't really matter which one we use. The rectangular kernel, at this h, produces a more rugged picture. While for the two other kernels $h = 1$ seems a good kernel width, for the rectangular kernel we may want to use a larger h.

After experimenting with the three kernel types, we decide to use one of the smooth kernels, and the choice falls onto the E. kernel. The next plots show the density obtained with this kernel for various kernel widths. At the smallest kernel width, $h = 0.3$ the density has many peaks and valleys. Even without having seen the true f, we may assume that traffic doesn't vary that wildly. The estimate for $h = 1$ is much smoother and on it two peaks - corresponding

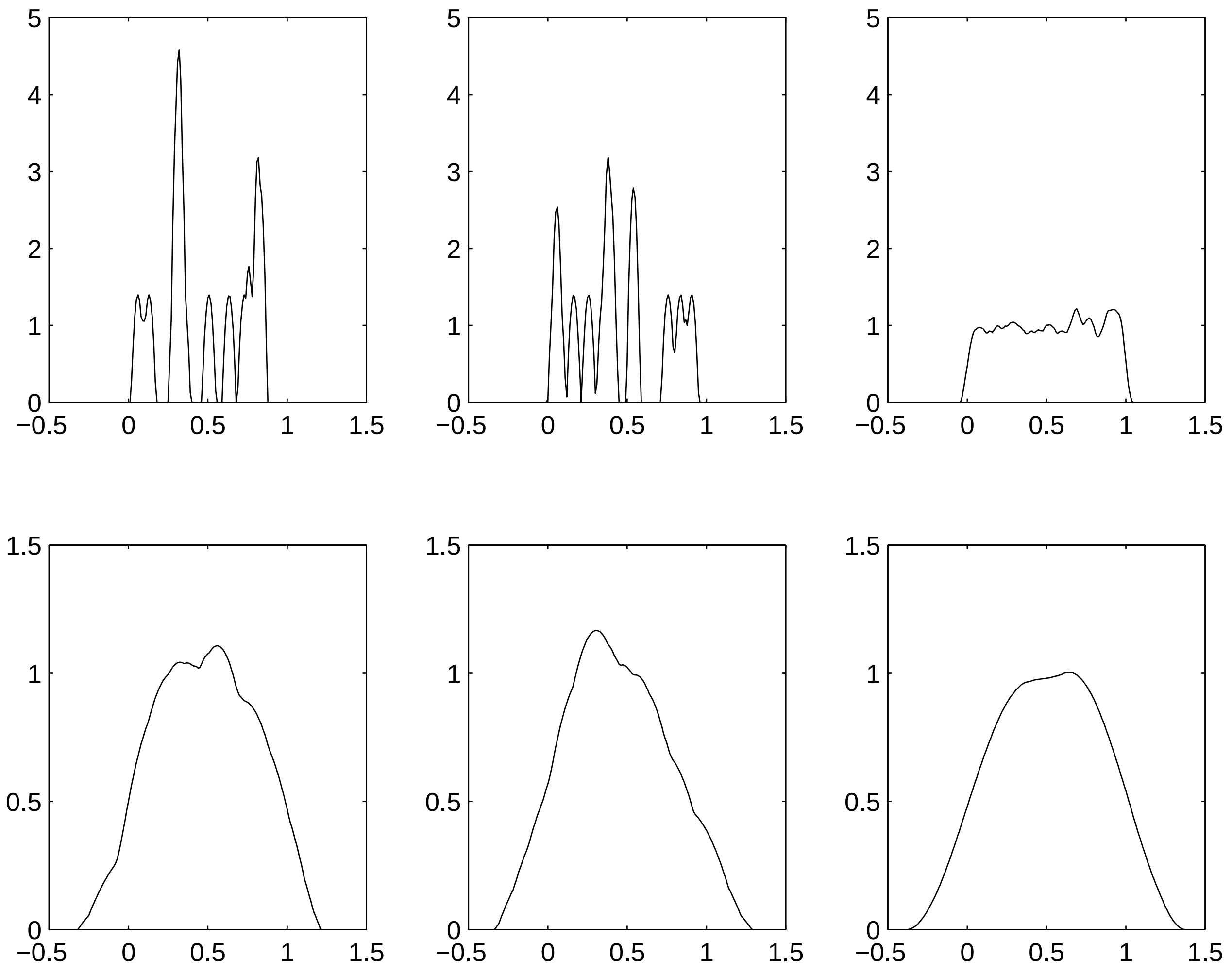


Figure 7.4: The effect of bias, variance and the sample size. The first row plots density estimates with $h = 0.02$ for 3 samples from the same distribution (a uniform over $[0, 1]$). The first two samples have size $n = 12$, the third has $n = 1200$. The density estimate is concentrated at the data points (thus the bias is low); this is beneficial for the large sample, but produces high variance for small samples. The second row shows density estimates from the same three data sets for $h = 0.17$. Now the three curves look very similar – the variance is low. The estimates obtained from the small data sets are much closer to the true distribution now. But this h is too large for the large data set, resulting in a worse estimate then previously. Last, note also that more data points are better than less data points in both cases.

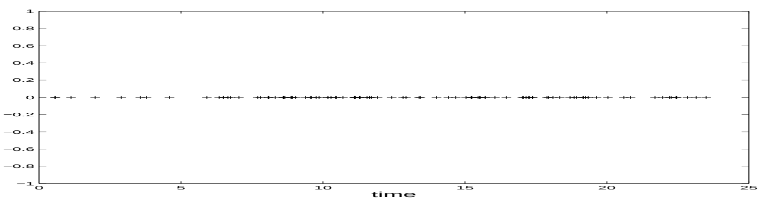


Figure 7.5: The traffic data set.

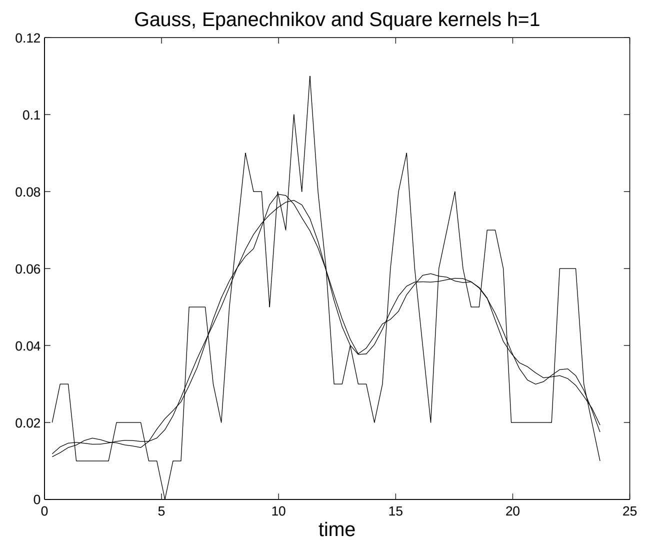


Figure 7.6: Density estimates of the traffic data with three different kernels: square, Epanechnikov and Gaussian. The kernel width is $h = 1$ in all cases.

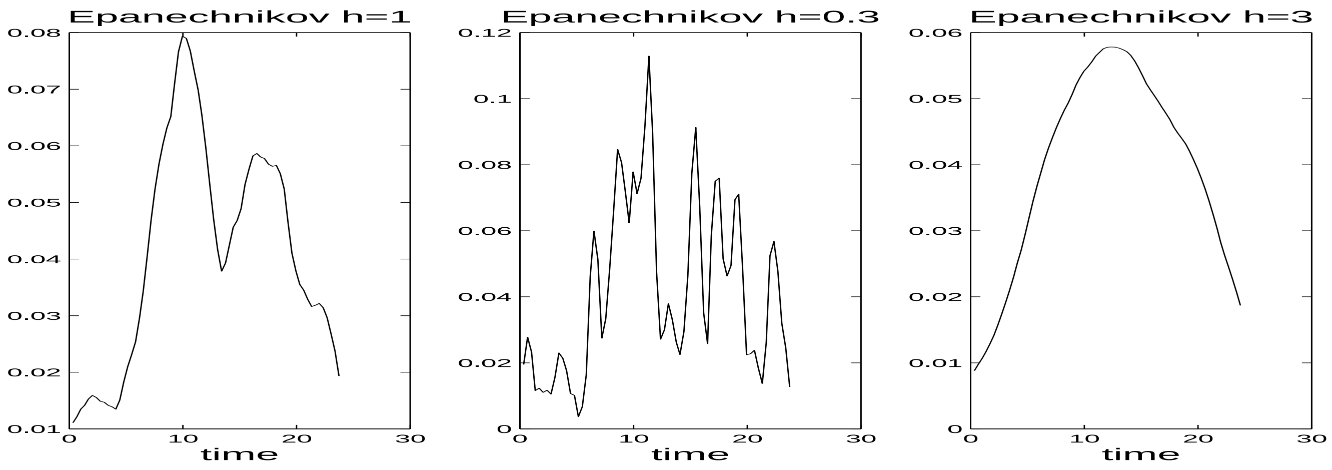


Figure 7.7: Density estimates of the traffic data with the same kernel at three different kernel widths.

to the morning and afternoon rush hours - appear clearly. This plot can help anyone trying to learn something about the traffic see the global pattern (in this case two intervals of intense traffic) amids the "sampling noise" that the small h estimate failed to suppress. Thus a density estimator is a tool in data visualization. The last plot, for $h = 3$ shows only one large peak; the kernel width is too large, smoothing out not only the noise but also the structure in the data.

The density estimate can be used also for prediction: How many cars will cross the I-90 bridge tomorrow between noon and 1 pm, if the total number of cars that cross it in a day is 10,000? The answer is

$$10,000 \int_{12.00}^{13.00} f(t)dt \approx 535 \tag{7.10}$$

In the above example, $h = 1$ has been chosen by visually examining the plots. Although "the visual appeal" method is quite popular, one can do something more principled.

7.5 Cross-validation

The idea of cross-validation is to "test" the obtained model on "fresh" data, data that has not been used to construct the model. Of course, we need to have access to such data, or to set aside some data before building the model. In our imaginary example, we are lucky to be given "next day's data", another

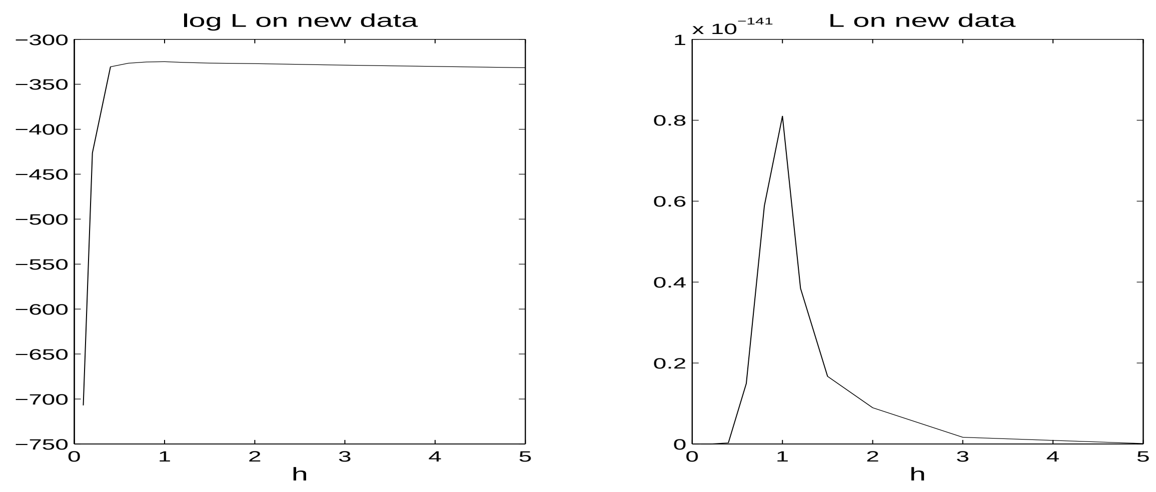


Figure 7.8: Likelihood (right) and log-likelihood of the test set of traffic data for different kernel sizes h. The optimum is at $h = 1$.

sample *from the same distribution* (this is the file `fig_h7_traffic_next.dat`). This data set is called *test data* or *hold out* data, in contrast to the data used to build the model which is called *training data*.

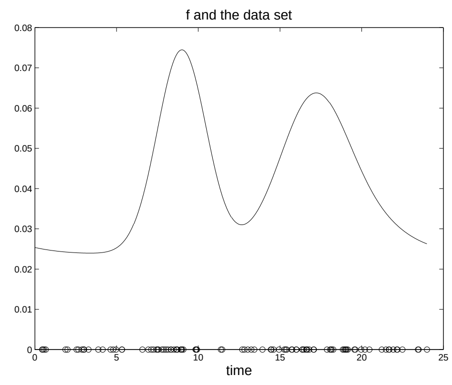
We will "test" the model on the holdout data. If the model is accurate, it must be able to predict well unseen data coming from the same distribution. In statistics terms, the unseen data should have high likelihood. Thus, the log-likelihood of the test data

$$l_{test}(h) = \sum_{x\in\mathcal{D}_{test}} \log f_h(x) \tag{7.11}$$

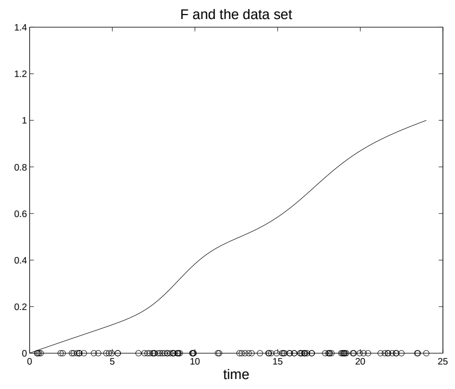
$$= \sum_{x\in\mathcal{D}_{test}} \log\left[\frac{1}{|\mathcal{D}|h}\sum_{y\in\mathcal{D}} k\left(\frac{x-y}{h}\right)\right] \tag{7.12}$$

is a measure of the goodness of our estimator. In the above equation, we have indexed the density by h the kernel width. Now all we need to do is to is to compute f_h and l_{test} for a sufficiently large range of h. This was done and the results, both as likelihood and as log-likelihood are shown in figure 7.8. The maximum value of the (log-)likelihood is attained for $h = 1$. This is the value that predicts the future data best, confirming our intuition.

Now at least having made all the choices we can allow ourselves to take a look at the "true" density that I used to generate this sample. Figure 7.9 depicts it along with the sample ($n = 100$).



a



b

Figure 7.9: The data and the true density (left) and cumulative distribution function (right).

7.5.1 Practical issues in cross-validation

1. **The range of h to be tested.** If the kernel is finitely supported, then, once h is smaller than the smallest distance between two data points, each point is under a separate bump and decreasing it further will only create larger 0 density regions between the data. So, this is a good lower limit for h. For the upper limit, a good choice is an kernel width of about the range of the data $x_{max} - x_{min}$, or a fraction of it, e.g $1/2(x_{max} - x_{min})$.

2. **The size of the validation set $\mathcal{D}_{test}$.** If the validation set is too small, then the value of l_{test} will have high variance (i.e will change much if we pick another validation set out of the original data set). So, our decision based on it will be prone to error. But if $n_{test} = |\mathcal{D}_{test}|$ is large, then we may be left with too little data for building the model.

   What is recommended depends on the amount of data available. If data is abundant (several thousands or more data points) then a n_{test} of about 1000 should suffice; the rest of the data should be used for constructing the model. If the available data set is medium (several hundreds), then it is recommende to split it into a ratio of $\frac{n_{test}}{n} \approx \frac{1}{3} \ldots \frac{1}{2}$.

   For smaller data sets, a procedure called **K-fold cross validation** is used. The whole data is divided at random into equal sized sets $\mathcal{D}_1, \ldots \mathcal{D}_K$. Then, for $k = 1, \ldots K$, $\mathcal{D}_k$ is used as a validation set, while the rest of the data is used as training set. The log-likelihood $l_k(h)$ of $\mathcal{D}_k$ for the k-th model is calculated. The final score for each h is equal to the arithmetic mean of l_k, $k = 1, \ldots K$. In practice, the values of K range from 3–5 to n. If $K = n$ the method is called **leave-one-out cross validation**. You can notice that, the larger the value of K, the more credible are the results of the procedure (why?). The downside is that computational costs also grow with K as follows. The number of kernel computations to evaluate a density estimate from n' points on n_{test} points is $n'n_{test}$. Therefore, to perform K-fold CV we need

$$N \;=\; K\left(\frac{n}{K} \times (K-1)\frac{n}{K}\right) \;=\; n^2(1-1/K) \qquad (7.13)$$

kernel evaluations.

3. Other sanity checks include looking at the shape of the density estimate for the chosen h, or even at how this shape changes with h.

Note that in spite of its conceptual elegance, cross-validation is not a completely error-proof method. For example, it can be shown that $h_{CV} \not\to 0$ if the target density f has infinite support and decays exponentially or slower. Also, outliers can cause problems in cross-validation.