Chapter 7

Density Estimation

Density estimation deals with the problem of estimating probability density functions based on some data sampled from the PDF. It may use assumed forms of the distribution, parameterized in some way (parametric statistics), or it may avoid making assumptions about the form of the PDF (non-parametric statistics). We have already discussed parametric statistics, now we are concerned more with the non-parametric case. As we'll see, in some ways these aren't such distinct concepts. In either case, we are trying to learn about the sampling distribution.

Non-parametric estimates may be useful when the form of the distribution (up to a small number of parameters) is not known or readily calculable. They may be useful for comparison with models, either parametric or not. For example, the ubiquitous histogram is a form of non-parametric density estimator (if normalized to unit area). Non-parametric density estimators may be easier or better than parametric modeling for efficiency corrections or background subtraction. More generally, they may be useful in "unfolding" experimental effects to learn about some distribution of more fundamental interest. These estimates may also be useful for visualization (again, the example of the histogram is notable). Finally, we note that such estimates can provide a means to compare two sampled datasets.

The techniques of density estimation may be useful as tools in the context of parametric statistics. For example, suppose we wish to fit a parametric model to some data. It might happen that the model is not analytically calculable. Instead, we simulate the expected distribution for any given set of parameters. For each set of parameters, we need a new simulation. However, simulations are necessarily performed with finite statistics, and the resulting fluctuations in the prediction may lead to instabilities in the fit. Density estimation tools may be helpful here as "smoothers", to smooth out the fluctuations in the predicted distribution.

We'll couch discussion in terms of a set of observations (dataset) from some "experiment". This dataset consists of the values $x_i;\ i = 1, 2, \ldots, n$. Our dataset consists of repeated samplings from a (presumed unknown) probability distribution. The samplings are here assumed to be IID, although we'll note generalizations here and there. Order is not important; if we are discussing a time series, we could introduce ordered pairs $\{(x_i, t_i), i = 1, \ldots, n\}$, and call it two-dimensional, but this case may not be IID, due to correlations in time. In general, our quantities can be multi-dimensional; no special notation will be used to distinguish one- from multi-variate cases. We'll discuss where issues enter with dimensionality.

When we discussed point estimation, we introduced the hat notation for an estimator. Here we'll be concerned with estimators for the density function itself, hence $\widehat{p}(x)$ is a random variable giving our estimate for density $p(x)$.

7.1 Empirical Density Estimate

The most straightforward density estimate is the **Empirical Probability Density Function**, or **EPDF**: Place a delta function at each data point. Explicitly, this estimator is:

$$\widehat{p}(x) = \frac{1}{n}\sum_{i=1}^{n}\delta(x - x_i).$$

Figure 7.1 illustrates an example. Note that x could be multi-dimensional here. We'll later find the EPDF used in practice as the sampling density for a procedure known as the bootstrap.

7.2 Histograms

Probably the most familiar density estimator is based on the histogram:

$$h(x) = \sum_{i=1}^{n} B(x - \tilde{x}_i; w),$$

where $\tilde{x}_i$ is the center of the bin in which observation x_i lies, w is the bin width, and

$$B(x; w) = \begin{cases} 1 & x \in (-w/2, w/2) \\ 0 & \text{otherwise.} \end{cases}$$

We have already seen this function, called the indicator function, in our proof of the Central Limit Theorem. Figure 7.2 illustrates the idea in the histogram context.

This is written for uniform bin widths, but may be generalized to differing widths with appropriate relative normalization factors. Figure 7.3 provides an example of a histogram, for the same sample as in Fig. 7.1.

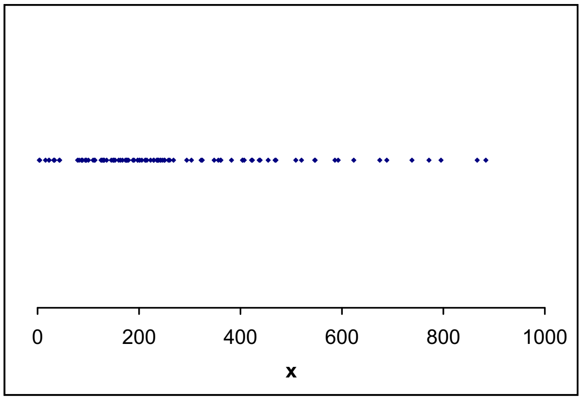


Figure 7.1: Example of an empirical probability density function. The vertical axis is such that the points are at infinity, with the "area" under each infinitely narrow point equal to $1/n$. In this example, $n = 100$. The actual sampling distribution for this example is a $\Delta(1232)$ Breit-Wigner (Cauchy; with pion and nucleon rest masses subtracted) on a second-order polynomial background. The probability to be background is 50%.

Given a histogram, the estimator for the probability density function (PDF) is:

$$\widehat{p}(x) = \frac{1}{nw} h(x).$$

There are some drawbacks to the histogram:

- Discontinuous even if PDF is continuous.
- Dependence on bin size and bin origin.
- Information from location of datum within a bin is ignored.

7.3 Kernel Estimation

Take the histogram, but replace "bin" function B with something else:

$$\widehat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} k(x - x_i; w),$$

where $k(x, w)$ is the "kernel function", normalized to unity:

$$\int_{-\infty}^{\infty} k(x; w)\, dx = 1.$$

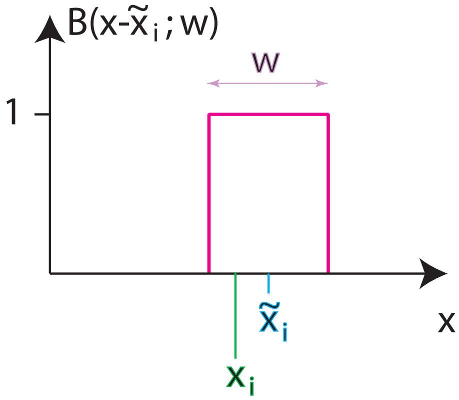


Figure 7.2: An indicator function, used in the construction of a histogram.

Usually interested in kernels of the form

$$k(x - x_i; w) = \frac{1}{w} K\left(\frac{x - x_i}{w}\right),$$

indeed this may be used as the definition of "kernel". The kernel estimator for the PDF is then:

$$\widehat{p}(x) = \frac{1}{nw} \sum_{i=1}^{n} K\left(\frac{x - x_i}{w}\right),$$

The role of parameter w as a "smoothing" parameter is apparent. The delta functions of the empirical distribution are spread over regions of order w.

Often, the particular form of the kernel used doesn't matter very much. This is illustrated with a comparison of several kernels (with commensurate smoothing parameters) in Fig. 7.4.

7.3.1 Multi-Variate Kernel Estimation

Explicit multi-variate case, $d = 2$ dimensions:

$$\widehat{p}(x, y) = \frac{1}{nw_x w_y} \sum_{i=1}^{n} K\left(\frac{x - x_i}{w_x}\right) K\left(\frac{y - y_i}{w_y}\right).$$

This is a "product kernel" form, with the same kernel in each dimension, except for possibly different smoothing parameters. It does not have correlations. The kernels we have introduced are classified more explicitly as "fixed kernels": The smoothing parameter is independent of x.

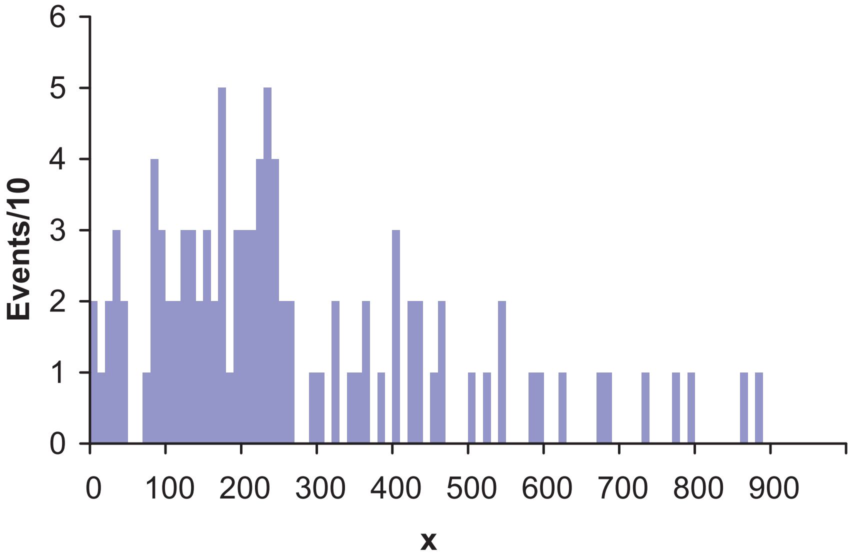


Figure 7.3: Example of a histogram for the data in Fig. 7.1, with binwidth $w = 10$.

7.4 Ideogram

A simple variant on the kernel idea is to permit the kernel to depend on additional knowledge in the data. Physicists call this an **ideogram**. Most common is the **Gaussian ideogram**, in which each data point is entered as a Gaussian of area one and standard deviation appropriate to that datum. This addresses a way that the IID assumption might be broken.

The Particle Data Group has used ideograms as a means to convey information about possibly inconsistent measurements. Figure 7.5 shows an example of this. Another example of the use of a Gaussian ideogram is shown in Fig. 7.6.

7.5 Parametric vs non-Parametric Density Estimation

The distinction between parametric and non-parametric is actually fuzzy. A histogram is non-parametric, in the sense that no assumption about the form of the sampling distribution is made. Often an implicit assumption is made that the distribution is "smooth" on a scale smaller than bin size. For example, we might know something about the resolution of our apparatus and adjust the bin size to be commensurate. But the estimator of the parent distribution made with a histogram is parametric – the parameters are populations (or frequencies) in each bin. The estimators for those parameters are the observed histogram populations. There are even more parameters than in a typical parametric fit!

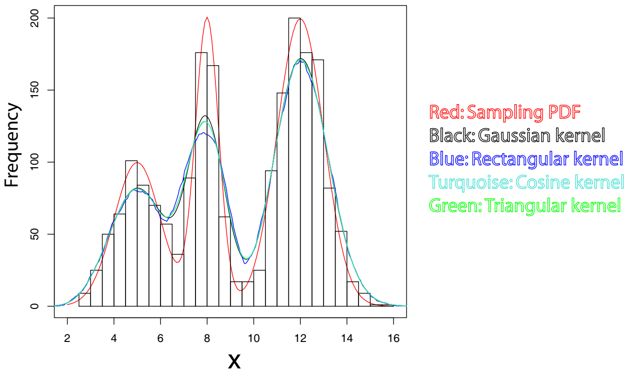


Figure 7.4: Comparison of different kernels.

The essence of the difference may be captured in notions of "local" and "non-local": If a datum at x_i influences the density estimator at some other point x this is non-local. A non-parametric estimator is one in which the influence of a point at x_i on the estimate at any x with $d(x_i, x) > \epsilon$ vanishes, asymptotically.[1] For example, for a kernel estimator, the bigger the smoothing paramter w, the more non-local the estimator,

$$\widehat{p}(x) = \frac{1}{nw} \sum_{i=1}^{n} K\left(\frac{x - x_i}{w}\right).$$

7.6 Optimization

We would like to make an optimal density estimate from our data. But we need to know what this means. We need a criterion for "optimal". In practice, the choice of criterion may be subjective; it depends on what you want to achieve.

As a plausible starting point, we may compare the estimator ($\widehat{f}(x)$) for a quantity ($f(x)$, the value of the density estimator at x) with the true value:

$$\Delta(x) = \widehat{f}(x) - f(x).$$

This is illustrated in Fig. 7.7. For a good estimator, we aim for small $\Delta(x)$, called the **error** in the estimator at point x.

We have seen that a common choice in point estimation is to minimize the sum of the squares of the deviations, as in a least-squares fit. We may take this

[1] As we'll discuss, the "optimal" choice of smoothing parameter depends on n.

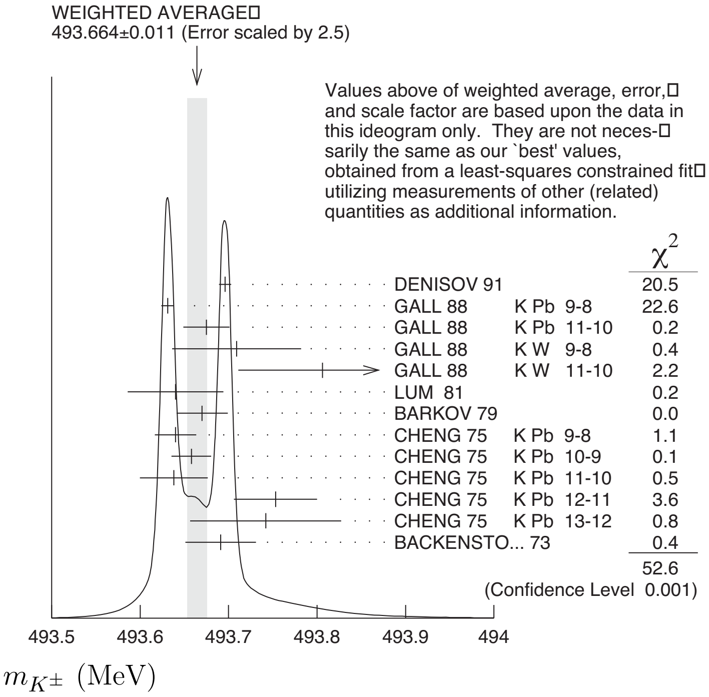


Figure 7.5: Example of a gaussian ideogram, from the particle data listings of the Particle Data Group's Review of Particle Properties [1].

idea over here, and form the **Mean Squared Error** (MSE):

$$\mathrm{MSE}[\widehat{f}(x)] \equiv \left\langle \left[\widehat{f}(x) - f(x)\right]^2 \right\rangle = \mathrm{Var}[\widehat{f}(x)] + \mathrm{Bias}^2[\widehat{f}(x)], \tag{7.1}$$

where

$$\mathrm{Var}[\widehat{f}(x)] \equiv E\left[\left(\widehat{f}(x) - E[\widehat{f}(x)]\right)^2\right] \tag{7.2}$$

$$\mathrm{Bias}[\widehat{f}(x)] \equiv E[\widehat{f}(x)] - f(x) \tag{7.3}$$

Since this isn't quite our familiar parameter estimation, let's take a little time to make sure it is understood. Suppose $\widehat{f}(x)$ is an estimator for the PDF $f(x)$, based on data $\{x_i; i = 1, \ldots, n\}$, IID from $f(x)$. Then

$$E[\widehat{f}(x)] = \int \cdots \int \widehat{f}(x; \{x_i\}) \mathrm{Prob}(\{x_i\}) d^n(\{x_i\}) \tag{7.4}$$

$$= \int \cdots \int \widehat{f}(x; \{x_i\}) \prod_{i=1}^{n} [f(x_i) dx_i] \tag{7.5}$$

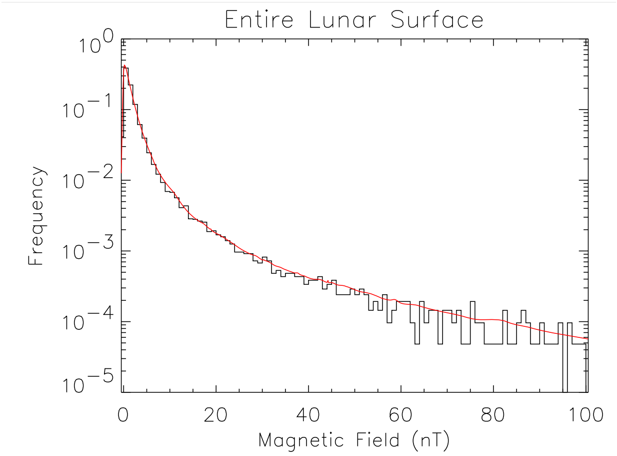


Figure 1. A histogram of magnetic field values (black), compared with a smoothed frequency distribution constructed using a Gaussian ideogram technique (red).

Figure 7.6: Example of a Gaussian ideogram overlaid on a histogram [2].

As an exercise, we'll derive Eqn. 7.1:

$$
\begin{aligned}
\mathrm{MSE}[\widehat{f}(x)] &= \langle(\widehat{f}(x) - f(x))^2\rangle \\
&= \int\cdots\int \left[\widehat{f}(x;\{x_i\}) - f(x)\right]^2 \prod_{i=1}^{n}[f(x_i)dx_i] \\
&= \int\cdots\int \left[\widehat{f}(x;\{x_i\}) - E(\widehat{f}(x)) + E(\widehat{f}(x)) - f(x)\right]^2 \prod_{i=1}^{n}[f(x_i)dx_i] \\
&= \int\cdots\int \Big\{ \left[\widehat{f}(x;\{x_i\}) - E(\widehat{f}(x))\right]^2 + \left[E(\widehat{f}(x)) - f(x)\right]^2 \\
&\qquad -2\left[\widehat{f}(x;\{x_i\}) - E(\widehat{f}(x))\right]\left[E(\widehat{f}(x)) - f(x)\right]\Big\} \prod_{i=1}^{n}[f(x_i)dx_i] \\
&= \mathrm{Var}[\widehat{f}(x)] + \mathrm{Bias}^2[\widehat{f}(x)] + 0. \qquad (7.6)
\end{aligned}
$$

In typical treatments of parametric statistics, we assume unbiased estimators, hence the "Bias" term is zero. However, this is not a good assumption here, as we now demonstrate. This is a fundamental problem with smoothing.

Theorem 7.1 (Rosenblatt (1956)) *A uniform minimum variance unbiased estimator for $f(x)$ does not exist.*

To be unbiased, we require:

$$E[\widehat{f}(x)] = f(x),\ \forall x.$$

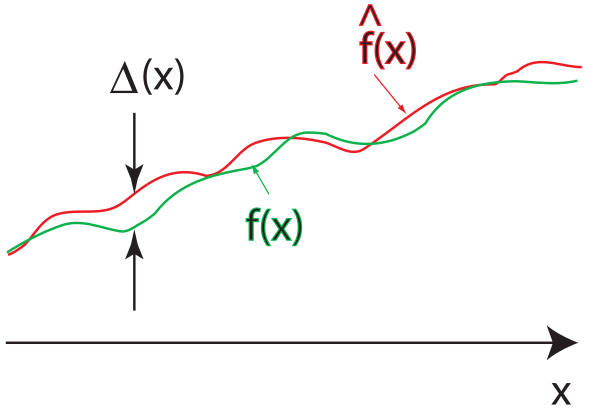


Figure 7.7: The difference (error) between a density and its estimator.

To have uniform minimum variance, we require:

$$\text{Var}\left[\widehat{f}(x)|f(x)\right] \leq \text{Var}\left[\widehat{g}(x)|f(x)\right], \ \forall x,$$

for all $f(x)$, where $\widehat{g}(x)$ is any other estimator of $f(x)$.

To illustrate this theorem, suppose we have a kernel estimator:

$$\widehat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} k(x - x_i; w),$$

Its expectation is:

$$\begin{aligned} E[\widehat{f}(x)] &= \frac{1}{n}\sum_{i=1}^{n}\int k(x - x_i; w) f(x_i) dx_i \\ &= \int k(x - y) f(y) dy. \qquad (7.7) \end{aligned}$$

Unless $k(x-y) = \delta(x-y)$, $\widehat{f}(x)$ will be biased for some $f(x)$. But $\delta(x-y)$ has infinite variance.

Thus, the nice properties we strive for in parameter estimation (and sometimes achieve) are beyond reach. The intuition behind this limitation may be understood as the effect that smoothing lowers peaks and fills in valleys. We see this effect at work in Fig. 7.8, where both a histogram and a Gaussian kernel estimator smooth out some of the structure in the original sampling distribution. Figure 7.9 shows how the effect may be mitigated, though not eliminated, with choice of binning or smoothing parameter.

The MSE for a density is a measure of uncertainty at a point. It is useful to somehow summarize the uncertainty over all points in a single quantity. We wish to establish a notion for the "distance" from function $\widehat{f}(x)$ to function

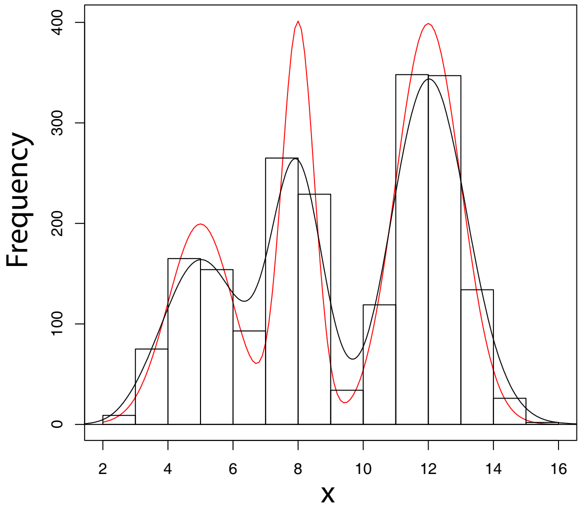


Figure 7.8: Red curve: PDF Histogram: Sampling from PDF Black curve: Gaussian kernel estimator for PDF

$f(x)$. This is a familiar subject, we are just dealing with normed vector spaces. Choice of norm is a bit arbitrary; obvious extremes are:

$$\|\widehat{f}(x) - f(x)\|_{L_\infty} \equiv \sup_x |\widehat{f}(x) - f(x)| \tag{7.8}$$

$$\|\widehat{f}(x) - f(x)\|_{L_1} \equiv \int |\widehat{f}(x) - f(x)|\, dx. \tag{7.9}$$

As is commonly done, we'll use the L_2 norm, or more precisely, the **Integrated Squared Error** (ISE):

$$\text{ISE} \equiv \int \left[\widehat{f}(x) - f(x)\right]^2 dx. \tag{7.10}$$

In fact, the ISE is still difficult object, as it depends on the true density, the estimator, and the sampled data. We may remove this latter dependence by evaluating the **Mean Integrated Squared Error** (MISE), or equivalently, the “integrated mean square error” (IMSE):

$$\text{MISE} \equiv E\,[\text{ISE}] = E\left[\int \left[\widehat{f}(x) - f(x)\right]^2 dx\right] \tag{7.11}$$

$$= \int E\left[\left(\widehat{f}(x) - f(x)\right)^2\right] dx = \int \text{MSE}\left[\widehat{f}(x)\right] dx \equiv \text{IMSE}. \tag{7.12}$$

A desirable property of an estimator is that the error decreases as the number of samples increases. This is a familiar notion from parametric statistics.

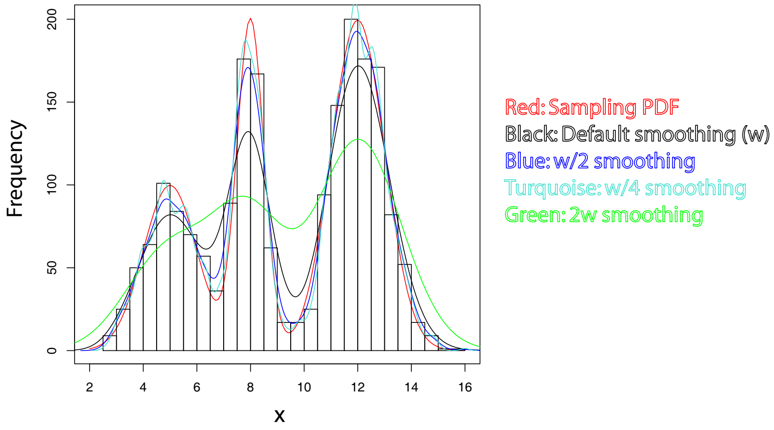


Figure 7.9: Plot showing effect of histogram binning and choice of smoothing parameter, for the same PDF as in Fig. 7.8.

Definition 7.1 *A density estimator $\widehat{f}(x)$ is consistent if:*

$$MSE\left[\widehat{f}(x)\right] \equiv E\left[\widehat{f}(x) - f(x)\right]^2 \to 0$$

as $n \to \infty$.

7.6.1 Choosing Histogram Binning

Considerations such as minimizing the MSE may be used to choose an "optimal" bin width for a histogram. Noting that the bin contents of a histogram are binomial-distributed, we could show (exercise) that, for the histogram density estimator $\widehat{f}(x) = h(x)/nw$:

$$\begin{aligned} \text{Var}\left[\widehat{f}(x)\right] &\leq \frac{f(x_j^*)}{nw} \\ \left|\text{Bias}\left[\widehat{f}(x)\right]\right| &\leq \gamma_j w, \end{aligned} \qquad (7.13)$$

where:

- $x \in \text{bin } j$,
- x_j^* is defined (and exists by mean value theorem) by:

$$\int_{\text{bin } j} p(x)\, dx = wp(x_j^*),$$

- γ_j is a positive constant (existing by assumption) such that

$$|f(x) - f(x_j^*)| < \gamma_j |x - x_j^*|, \quad \forall x \in \text{bin } j,$$

- equality is approached as the probability to be in bin j decreases (e.g., by decreasing bin size).

Thus, we have a bound on the MSE for a histogram:

$$\mathrm{MSE}\left[\widehat{f}(x)\right] = E\left[\widehat{f}(x) - p(x)\right]^2 \leq \frac{f(x_j^*)}{nw} + \gamma_j^2 w^2.$$

Theorem 7.2 *The MSE of the histogram estimator $\widehat{f}(x) = h(x)/nw$ is consistent if the bin width $w \to 0$ as $n \to \infty$ such that $nw \to \infty$.*

Note that the $w \to 0$ requirement insures that the bias will approach zero, according to our earlier discussion. The $nw \to \infty$ requirement ensures that the variance asymptotically vanishes.

Theorem 7.3 *The MSE(x) bound above is minimized when*

$$w = w^*(x) = \left[\frac{p(x_j^*)}{2\gamma_j^2 n}\right]^{1/3}.$$

This theorem suggests that the optimal bin size decreases as $1/n^{1/3}$. The $1/n$ dependence of the variance is our familiar result for Poisson statistics. The optimal bin size depends on the value of the density in the bin. This suggests an “adaptive binning” approach with variable bin sizes. However, according to Scott [4]: “…in practice there are no reliable algorithms for constructing adaptive histogram meshes.”

Alternatively, the MISE error is minimized (Gaussian kernel, asymptotically, for normally distributed data) when

$$w^* = \left(\frac{4}{3}\right)^{1/5} \sigma n^{-1/5}.$$

An early and popular choice is **Sturges’ rule**, which says that the number of bins should be

$$k = 1 + \log_2 n,$$

where n is the sample size. This is the rule that was used in making the histogram in Fig. 7.8. It is the default choice when making a histogram in R.

However, the argument behind this rule has been criticized [3]. Indeed we see in our example that we probably would have “by hand” selected more bins; our histogram is “over-smoothed”. There are other rules for optimizing the number of bins. For example, **Scott’s rule** [4] for the bin width is:

$$w = 3.5 s n^{-1/3},$$

where s is the sample standard deviation. Physicists typically ignore these rules, and make explicit choices for the bin widths, often based on experimental resolution. The standard rulesoften leave the visual impression that the binning could usefully be finer, see Fig. 7.10.

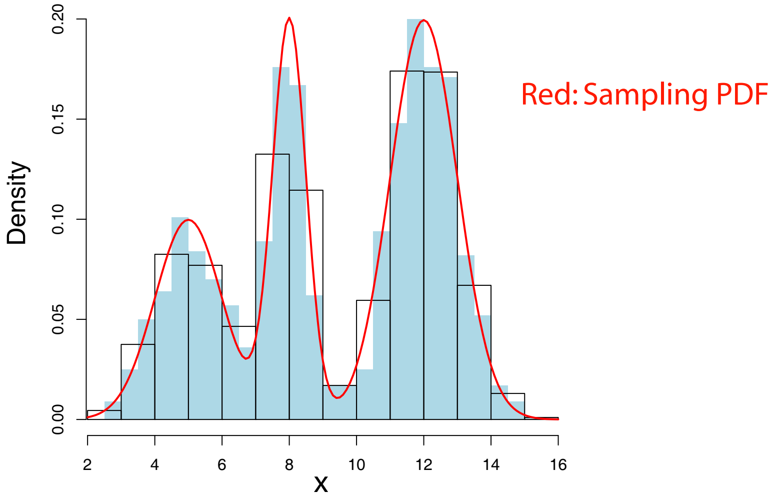


Figure 7.10: Illustration of histogram binning. The red curve is the sampling PDF. The "standard rules" (Sturges, Scott, Freedman-Diaconis) correspond roughly to the coarser binning above. The shaded histogram seems like a better choice.

7.7 The Curse of Dimensionality

While in principle problems with multiple dimensions in the sample space are not essentially different from a single dimension, there are very important practical differences. This is referred to as the **curse of dimensionality**. The most obvious difficulty is in displaying and visualizing as the number of dimensions increases. Typically one displays projections in lower dimensions, but this loses the correlations that might be present in the un-displayed dimensions.

Another difficulty is that "all" the volume (of a bounded region) goes to the boundary (exponentially!) as the dimensions increases. This is illustrated in Fig. 7.11. A unit cube in d dimensions is illustrated as d increases. A central cube with edges $1/2$ unit in length is shown. The fraction of the volume contained in this central cube decreases as 2^{-d}. As a consequence, data becomes "sparse". To obtain a suitable density of statistics, for example in a simulation tends to require exponentially growing computation as the dimensionality increases.

7.8 The Bootstrap

A statistical technique known as the **bootstrap** provides a means to evaluate how much to trust our density estimate. The **bootstrap algorithm** in this context is as follows:

1. Form density estimate $\widehat{p}$ from data $\{x_i; i = 1, \ldots, n\}$.

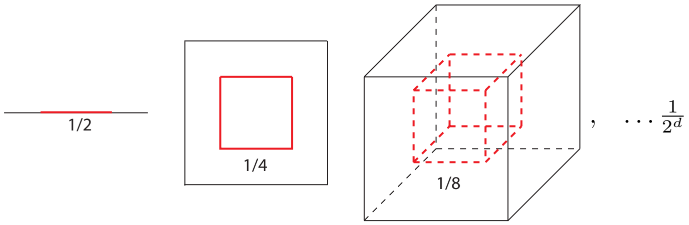


Figure 7.11: Demonstration of the "curse of dimensionality".

2. Resample (uniformly) n values from $\{x_i; i = 1, \ldots, n\}$, with replacement, obtaining $\{x_i^*; i = 1, \ldots, n\}$ (bootstrap data). Note that in resampling with replacement our bootstrap dataset may contain the same x_i multiple times.

3. Form density estimate $\widehat{p}^*$ from data $\{x_i^*; i = 1, \ldots, n\}$.

4. Repeat steps 2&3 many (N) times to obtain a family of bootstrap density estimates $\{\widehat{p}_i^*; i = 1, \ldots, N\}$.

5. The distribution of $\widehat{p}_i^*$ about $\widehat{p}$ mimics the distribution of $\widehat{p}$ about p.

Consider, for a kernel density estimator, the expectation of the boostrap dataset:

$$E\left[\widehat{p}^*(x)\right] = E\left[K(x - x_i^*; w)\right] = \widehat{p}(x),$$

where the demonstration is left as an exercise. Thus, the bootstrap distribution about $\widehat{p}$ does not reproduce the bias which may be present in $\widehat{p}$ about p. However, it does properly reproduce the variance of $\widehat{p}$, hence the bootstrap is a useful tool for estimating the variance of our density estimator. An illustration of the distribution of bootstrap samples is shown in Fig. 7.12.

7.9 Estimating Bias: The Jackknife

We have seen that we may use the bootstrap to evaluate the variance of a density estimator, but not the bias. Both properties are needed for a complete understanding of our estimator. A method that can be used to estimate bias is the **jackknife**. The idea behind this method is that bias depends on sample size. If we can assume that the bias vanishes asymptotically, we may use the data to estimate the dependence of the bias on sample size.

The jackknife algorithm is as follows:

1. Divide the data into k random disjoint subsamples.

2. Evaluate estimator for each subsample.

3. Compare average of estimates on subsamples with estimator based on full dataset.

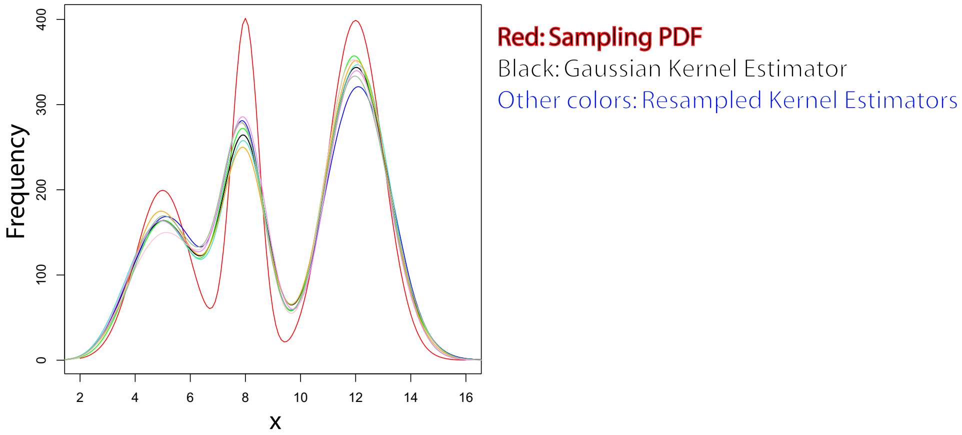


Figure 7.12: Use of the bootstrap to determine the spread of a Gaussian kernel density estimator.

Figure 7.13 illustrates the jackknife technique. Besides estimating the bias, jackknife techniques may also be used to reduce bias (see Scott [4]).

7.10 Cross-validation

Similar to the jackknife, but different in intent, is the **cross-validation** procedure. In density estimation, cross-validation is used to optimize smoothing bandwidth selection. It can improve on "theoretical" optimizations by making use of the actual sampling distribution, via the available samples.

The basic method ("leave-one-out cross-validation") is as follows: Form n subsets of the dataset, each one leaving out a different datum. Use subscript $-i$ to denote subset omitting datum x_i. For density estimator $\widehat{f}(x; w)$ evaluate the following average over these subsets:

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

The reader is encouraged to show that the expectation of this quantity is the MISE of $\widehat{f}$ for $n-1$ samples, except for a term which is independent of w. Thus, we may evaluate the dependence of this quantity on smoothing parameter w, and select the value w^* for which it is minimized. Figure 7.14 illustrates the procedure.

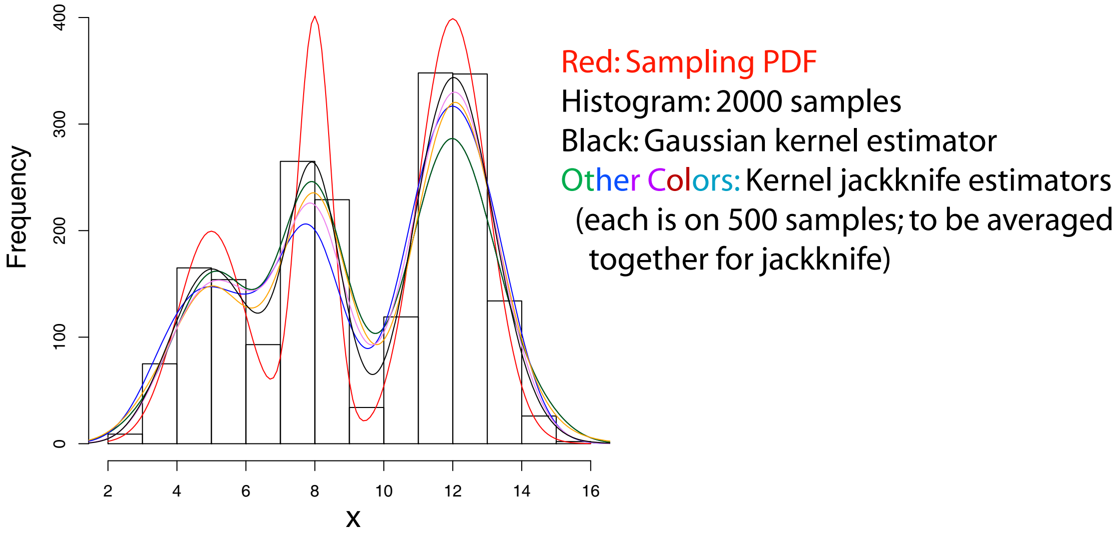


Figure 7.13: Jackknife estimate of bias.

7.11 Adaptive Kernel Estimation

We saw in our discussion of histograms that it is probably more optimal to use variable bin widths. This applies to other kernels as well. Indeed, the use of a fixed smoothing parameter, deduced from all of the data introduces a non-local, hence parametric, aspect into the estimation. It is more consistent to look for smoothing which depends on data locally. This is **adaptive kernel estimation**.

We argue that the more data there is in a region, the better that region can be estimated. Thus, in regions of high density, we should use narrower smoothing. In Poisson statistics (e.g., histogram binning), the relative uncertainty scales as

$$\frac{\sqrt{N}}{N} \propto \frac{1}{\sqrt{p(x)}}.$$

Thus, in the region containing x_i, the smoothing parameter should be:

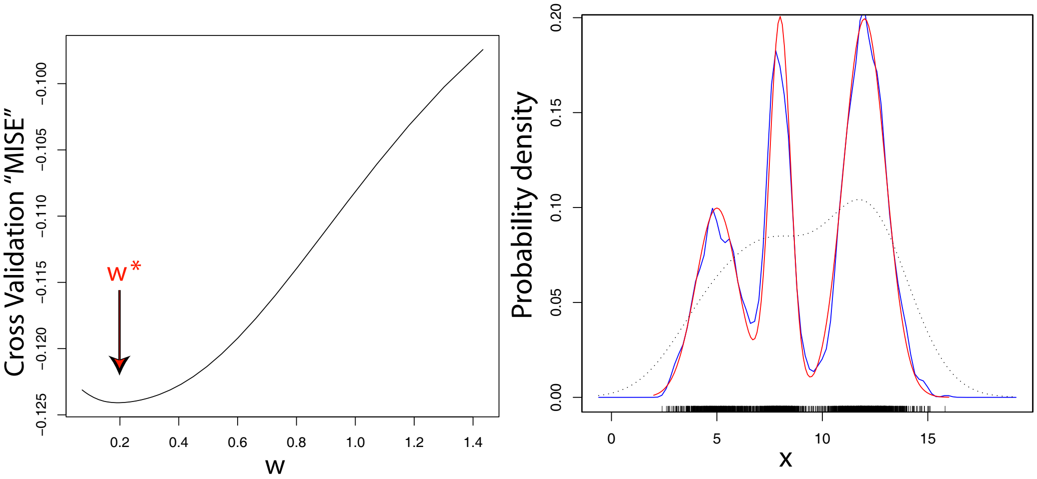
$$w(x_i) = w^*/\sqrt{p(x_i)}.$$

There are two issues with implementing this:

- What is w^*?
- We don't know $p(x)$.

For $p(x)$, we may try substituting our fixed kernel estimator, call it $\widehat{p}_0(x)$. For w^*, we use dimensional analysis:

$$D\left[w(x_i)\right] = D[x]; \quad D\left[p(x)\right] = D[1/x] \quad \Rightarrow \quad D[w^*] = D\left[\sqrt{x}\right] = D\left[\sqrt{\sigma}\right].$$



Choosing smoothing parameter which minimizes estimated MISE.

Red: Actual PDF

Blue: Kernel estimate, using smoothing from cross-validation optimization

Dots: Kernel estimate with a default value for smoothing

Figure 7.14: Cross validation. Left: Determining the optimal w^*; Right: Applying the optimized smoothing.

Then, for example, using the "MISE-optimized" choice earlier, we iterate on our fixed kernel estimator to obtain:

$$\widehat{p}_1(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}K\left(\frac{x-x_i}{w_i}\right),$$

where

$$w_i = w(x_i) = \left(\frac{4}{3}\right)^{1/5}\sqrt{\frac{\rho\sigma}{\widehat{p}_0(x_i)}}n^{-1/5}.$$

ρ is a factor which may be further optimized, or typically set to one.

The iteration on the fixed-kernel estimator nearly removes the dependence on our intitial choice of w. The boundaries pose some complication in carrying this out.

There are packages for adaptive kernel estimation, for example, the KEYS ("Kernel Estimating Your Shapes") package [5]. Figure 7.15 illustrates the use of this package.

7.12 Multivariate Kernel Estimation

Besides the curse of dimensionality, the multi-dimensional case introduces the complication of covariance. When using a product kernel, the local estimator

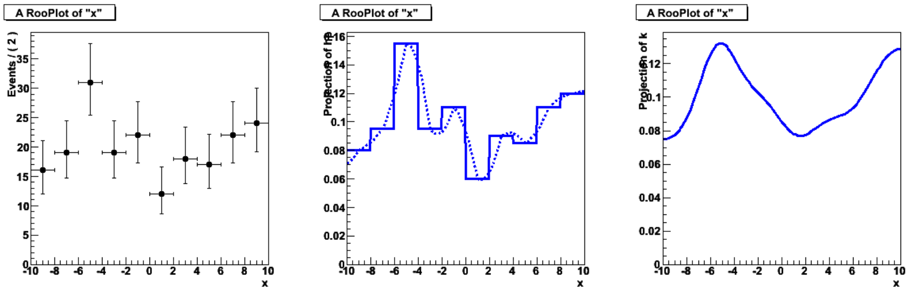


Figure 9 – Non-parametric p.d.f.s: Left: histogram of unbinned input data, Middle: Histogram-based p.d.f (2nd order interpolation), Right: KEYS p.d.f from original unbinned input data.

Figure 7.15: Example of the KEYS adaptive kernel estimation [6]

has diagonal covariance matrix. In principle, we could apply a local linear transformation of the data to a coordinate system with diagonal covariance matrices. This amounts to a non-linear transfomation of the data in a global sense, and may not be straightforward. However, we can at least work in the system for which the overall covariance matrix of the data is diagonal.

If $\{y_i\}$ is the suitably diagonalized data, the product fixed kernel estimator in d dimensions is:

$$\widehat{p}_0(y) = \frac{1}{n}\sum_{i=1}^{n}\left[\prod_{j=1}^{d}\frac{1}{w_j}K\left(\frac{y^{(j)} - y_i^{(j)}}{w_j}\right)\right],$$

where $y^{(j)}$ denotes the j-th component of the vector y. The asymptotic, normal MISE-optimized smoothing parameters are now:

$$w_j = \left(\frac{4}{d+2}\right)^{1/(d+4)}\sigma_j n^{-1/(d+4)}.$$

The corresponding adaptive kernel estimator follows the discussion as for the univariate case. An issue in the scaling for the adaptive bandwidth arises when the multivariate data is approximately sampled from a lower dimensionality than the dimension d.

Fig. 7.16 shows an example in which the sampling distribution has diagonal covariance matrix (locally and globally). Applying kernel estimation to this distribution yields the results in Fig. 7.17, for two different smoothing parameters.

For comparison, Fig. 7.19 shows an example in which the sampling distribution has non-diagonal covariance matrix. Applying the same kernel estimation to this distribution gives the results shown in Fig. 7.19. It may be observed (since this is the same data as in Fig. 7.16, just rotated to give a non-diagonal covariance matrix) that this is more difficult to handle.

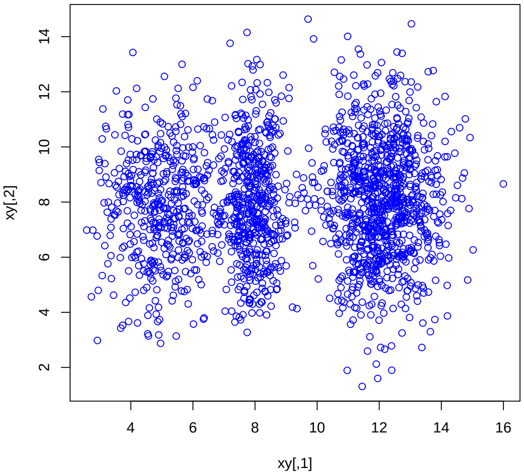


Figure 7.16: A two-dimensional distribution with diagonal covariance matrix.

7.13 Estimation Using Orthogonal Series

We may take an alternative approach, and imagine expanding the PDF in a series of orthogonal functions:

$$p(x) = \sum_{k=0}^{\infty} a_k \psi_k(x),$$

where

$$a_k = \int \psi_k(x)p(x)\rho(x)\,dx = E\left[\psi_k(x)\rho(x)\right],$$

and

$$\int \psi_k(x)\psi_\ell(x)\rho(x)\,dx = \delta_{k\ell},$$

where $\rho(x)$ is a “weight function”.

Since the expansion coefficients are expectation values of functions, it is natural to substitute sample averages as estimators for them:

$$\widehat{a}_k = \frac{1}{n}\sum_{i=1}^{n} \psi_k(x_i)\rho(x_i),$$

and thus:

$$\widehat{p}(x) = \sum_{k=1}^{m} \widehat{a}_k \psi_k(x),$$

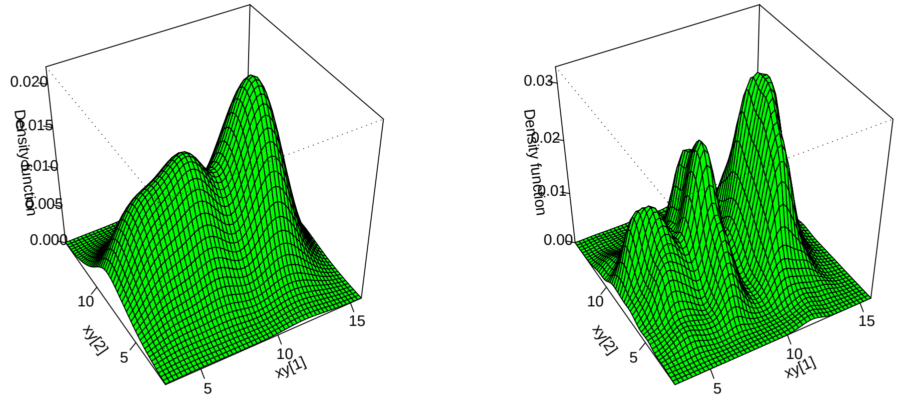


Figure 7.17: Kernel estimation applied to the two-dimensional data in Fig. 7.16. Left: Default smoothing parameter; Right: Using one-half of the default smoothing parameter.

where the number of terms m is chosen by some optimization criterion.

Note the analogy between choosing m and choosing smoothing parameter w in kernel estimators; and between choosing K and choosing $\{\psi_k\}$. We are actually rather familiar with estimation using orthogonal series: It is the method of moments, or substitution method, that we described in Chapter 3. An example of an application of this method is shown in Fig. 7.20. In particular, the left graph shows the estimated spectra broken up by orthogonal function, in this case $Y_{\ell m}$ spherical harmonics.

7.14 Using Monte Carlo Models

We often build up a data model using Monte Carlo computations of different processes, which are added together to get the complete model. This may involve weighting of events, if more effective time is simulated for some processes than for others. The overall simulated empirical density distribution is then:

$$\widehat{f}(x) = \sum_{i=1}^{n} \rho_i \delta(x - x_i),$$

where $\sum \rho_i = 1$ (or n to correspond with an event sample of some total size).

The weights must be included in computing the sample covariance matrix

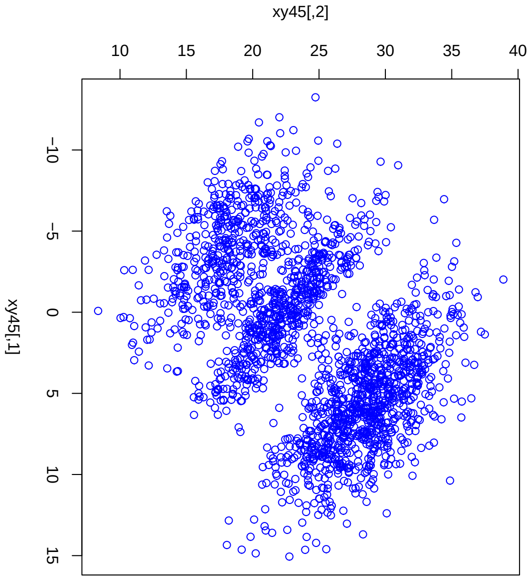


Figure 7.18: A two-dimensional distribution with non-diagonal covariance matrix.

(x_i has components $x_i^{(k)}, k = 1 \ldots d$):

$$V_{k\ell} = \sum_{i=1}^{n} \rho_i \frac{(x_i^{(k)} - \widehat{\mu}_k)(x_i^{(\ell)} - \widehat{\mu}_\ell)}{\sum_j \rho_j},$$

where $\widehat{\mu}_k = \sum_i \rho_i x_i^{(k)} / \sum_j \rho_j$ is the sample mean in dimension k.

Assuming we have transformed to a diagonal system using this covariance matrix, our product kernel density based on this simluation is then:

$$\widehat{f}_0(x) = \frac{1}{\sum_j \rho_j} \sum_{i=1}^{n} \rho_i \prod_{k=1}^{d} \frac{1}{w_k} K\left(\frac{x^{(k)} - x_i^{(k)}}{w_k}\right).$$

This may be iterated to obtain an adaptive kernel estimator as discussed earlier.

7.15 Unfolding

We may not be satisfied with merely estimating the density from which our sample $\{x_i\}$ was drawn. The interesting physics may be obscured by convolution with uninteresting functions, for example efficiency dependencies or radiative corrections. Thus, we wish to **unfold** the interesting distribution from the sampling distribution. Our treatment of this big subject will be cursory. See, for example, Ref. [8] for further discussion.

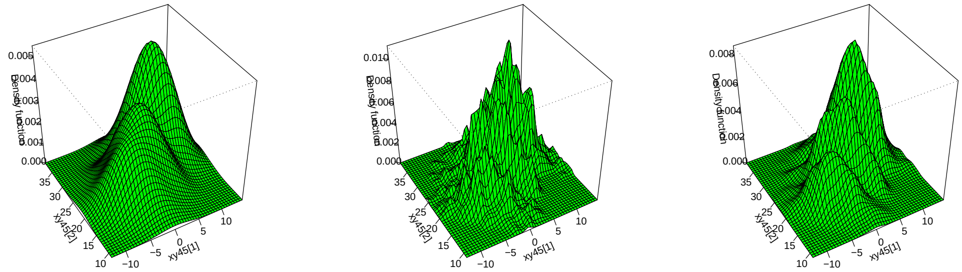


Figure 7.19: Kernel estimation applied to the two-dimensional data in Fig. 7.18. Left: Default smoothing parameter; Middle: Using one-half of the default smoothing parameter; Right: Intermediate smoothing.

We will assume the convolution function relating the two distribtuions is known. However, it often is also estimated via auxillary measurements. Because data fluctuates, unfolding usually also necessitates smoothing to control fluctuations. This is referred to as **Regularization** in this context.

To set up a typical problem: Suppose we ample from a distribution with some kernel function $K(x, y)$:

$$o(x) = \int K(x, y) f(y) dy.$$

We are given a sampling $\widehat{o}$, and wish to estimate f.

In principle, the solution is easy:

$$\widehat{f}(y) = \int K^{-1}(y, x) \widehat{o}(x) dx,$$

where

$$\int K^{-1}(x, y) K(y, x')\, dy = \delta(x - x').$$

In practice, our observations are discrete, and we need to interpolate/smooth.

If we don't know how (or are too lazy) to invert K, we may try an iterative solution. For example, consider the problem of unfolding radiative corrections in a cross section measurement. The observed cross section, $\sigma_E(s)$ is related to the "interesting" cross section σ according to:

$$\sigma_E(s) = \sigma(s) + \delta\sigma(s),$$

where

$$\delta\sigma(s) = \int K(s, s') \sigma(s')\, ds'.$$

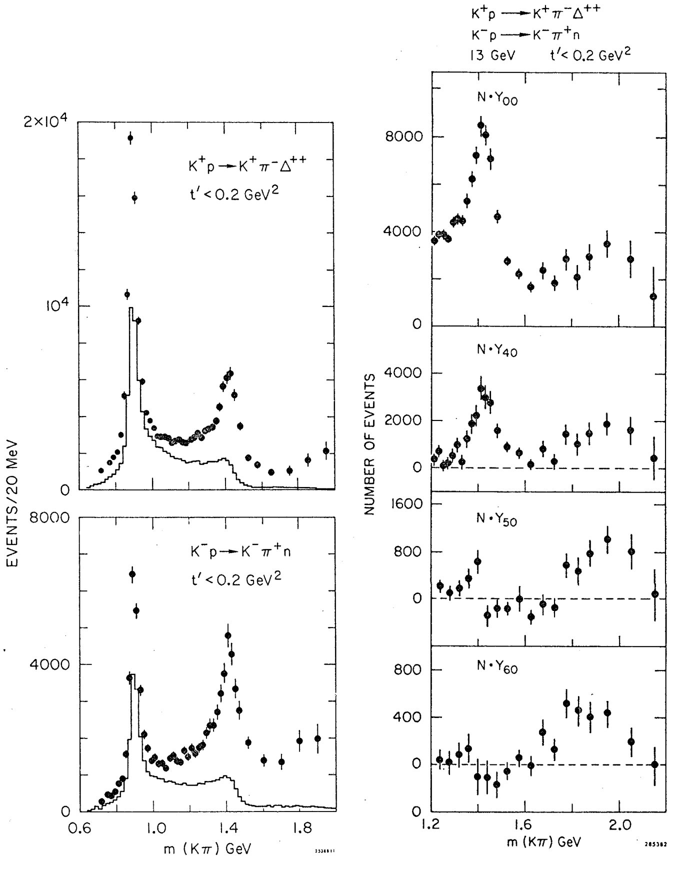


Figure 7.20: The $K\pi$ mass spectrum from Ref. [7]. Left: The measured spectrum. The line histogram is the raw measurement data; the points show the data corrected for acceptance. Left: The mass spectrum broken up by angular distribution.

We form an iterative estimate for $\sigma(s)$ according to:

$$\begin{aligned} \widehat{\sigma}_0(s) &= \sigma_E(s) & (7.14) \\ \widehat{\sigma}_i(s) &= \sigma_E(s) - \int K(s,s')\widehat{\sigma}_{i-1}(s')\,ds', \quad i=1,2,\ldots & (7.15) \end{aligned}$$

Notice that this is just the Neumann series solution to an integral equation.

Since $\sigma_E(s)$ is measured at discrete s values and with some statistical precision, some smoothing/interpolation is still required.

7.15.1 Unfolding: Regularization

If we know K^{-1} we can incorporate the smoothing/interpolation more directly. We could use the techniques already described to form a smoothed estimate $\widehat{o}$, and then use the transformation K^{-1} to obtain the estimator $\widehat{f}$. For simplicity, consider here the problem of unfolding a histogram. Then we restate the earlier integral formula as:

$$o_i = \sum_{j=1}^{k} K_{ij} f_j,$$

where K is a square matrix, assumed invertible.

A popular procedure is to form a likelihood (or χ^2), but add an extra term, a "regulator", to impose smoothing. The modified likelihood is maximized to obtain the estimate for $\{f_j\}$.

$$\ln\mathcal{L} \to \ln\mathcal{L}' = \ln\mathcal{L} + wS(\widehat{o}_i).$$

The regulator function $S(\widehat{o}_i)$ as usual gets its smoothing effect by being somewhat non-local. A popular choice is to add a "curvature" term to be minimized (hence smoothed):

$$S(\widehat{o}_i) = -\sum_{j=2}^{k-1}\left[(\widehat{o}_{i+1} - \widehat{o}_i) - (\widehat{o}_i - \widehat{o}_{i-1})\right]^2.$$

This is implemented, for example, in the RUN package [9].

Another implementation is GURU [10]. Further discussion may be found in [11]. This paper has a nice demonstration of the importance of smoothing: Note that the transformation K itself corresponds to a sort of smoother, as it acts non-locally. The act of "unfolding" a smoother can produce large variances.

7.16 Non-parametric Regression

Regression is the problem of estimating the dependence of some "response" variable on a "predictor" variable. Given a dataset of predictor-response pairs $\{(x_i, y_i), i = 1, \ldots, n\}$, we write the relationship as:

$$y_i = r(x_i) + \epsilon_i,$$

where the "error" ϵ_i might also depend on x through the parameters of the sampling distribution it represents.

We are used to solving this problem with parametric statistics, for example, the dependence of accelerator background on beam current, where we might try a power-law form. However, we may also bring our non-parametric methods to bear on this problem.

The sampling of the response-predictor pairs may be a **fixed design** in which the x_i values are deliberately selected, or a **random design**, in which (x_i, y_i) is drawn from some joint PDF. We'll work in the context of the random design here, and also will work in two dimensions.

The **regression function** r may be expressed as:

$$r(x) = E\left[y|x\right] = \int yp(y|x)\,dy = \frac{\int yf(x,y)\,dy}{\int f(x,y)dy}.$$

Let us construct an estimator for r by substituting a bivariate product kernel estimator for the unknown PDF $f(x,y)$:

$$\widehat{f}(x,y) = \frac{1}{nw_xw_y}\sum_{i=1}^{n} K\left(\frac{x-x_i}{w_x}\right) K\left(\frac{y-y_i}{w_y}\right).$$

Assuming a symmetric kernel, after a little algebra we find:

$$\widehat{r}(x) = \sum_{i=1}^{n} y_i K\left(\frac{x-x_i}{w_x}\right) / \sum_{i=1}^{n} K\left(\frac{x-x_i}{w_x}\right).$$

This is known as the **local mean** estimator. Note the absence of dependence on w_y, and the linearity in the y_i.

We may achieve better properties by considering **local polynomial** estimators, corresponding to local polynomial fits to the data. This may be achieved with a least-squares minimization (the local mean is the result for a fit to a zero-order polynomial). Thus, the **local linear** regression estimate is given by:

$$\widehat{r}(x) = \sum_{i=1}^{n} \frac{\left[S_2(x) - S_1(x)(x_i - x)\right] K\left((x_i - x)/w_x\right) y_i}{S_2(x)S_0(x) - S_1(x)^2},$$

where

$$S_\ell(x) \equiv \sum_{i=1}^{n} (x_i - x)^\ell K\left(\frac{x-x_i}{w_x}\right).$$

R provides a package `loess` for local polynomialregression fitting. Examples of its use are shown in Figs; 7.21 and 7.22.

7.17 sPlots

The use of the density estimation technique known as **sPlots** [12] has gained popularity in some physics applications. This is a multivariate technique that

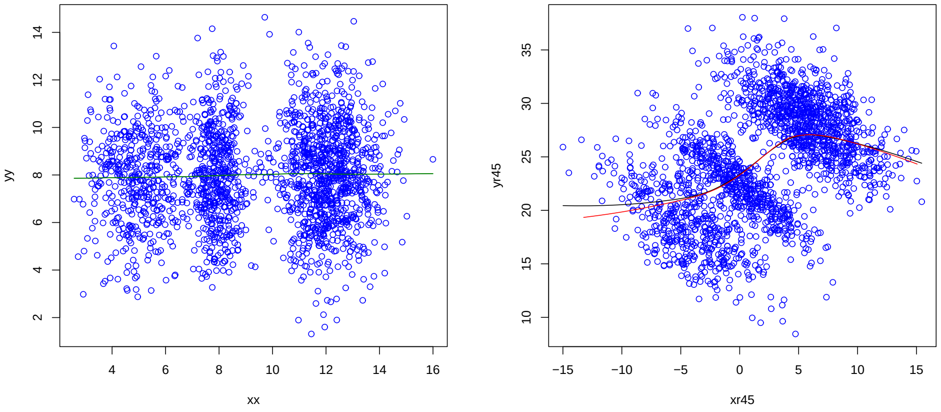


Figure 7.21: Illustration of local linear regression using the `loess` package. The dataset is the same in both plots, but the variables have been transformed by a 45 degree rotation in the second plot. It may be observed that the introduction of covariance influences the result.

uses the distribution on a subset of variables to predict the distribution in another subset. It is based on a (parametric) model in the predictor variables, with different categories (e.g., "signal" and "background"). It provides both a means to visualize agreement with the model for each category and an easy way to do "background subtraction".

Assume there are a total of $r + p$ parameters in the overall fit to the data: (i) The expected number of events, $N_j, j = 1, \ldots, r$ in each category, and (ii) Distribution parameters, $\{\theta_i, i = 1, \ldots, p\}$. We use a total of N events to estimate these parameters via a maximum likelihood fit to the sample $\{x\}$.

We wish to find weights $w_j(x'_i)$, depending only on $\{x'\} \subseteq \{x\}$ (and implicitly on the unknown parameters), such that the asymptotic distribution in $y \notin \{x'\}$ of the weighted events is the sampling distribution in y, for any chosen category j. Assume that y and $\{x'\}$ are statistically independent within each category.

The weights that satisfy our criterion and produce minimum variance summed over the histogram are given by [12]:

$$w_j(e) = \frac{\sum_{k=1}^r V_{jk} f_k(x'_e)}{\sum_{k=1}^r \widehat{N}_k f_k(x'_e)},$$

where $w_j(e)$ is the weight for event e in category j V is the covariance matrix from a "reduced fit" (i.e., excluding y):

$$\left(V^{-1}\right)_{jk} \equiv \sum_{e=1}^N \frac{f_j(x'_e) f_k(x'_e)}{\left[\sum_{i=1}^r \widehat{N}_i f_i(x'_e)\right]^2},$$

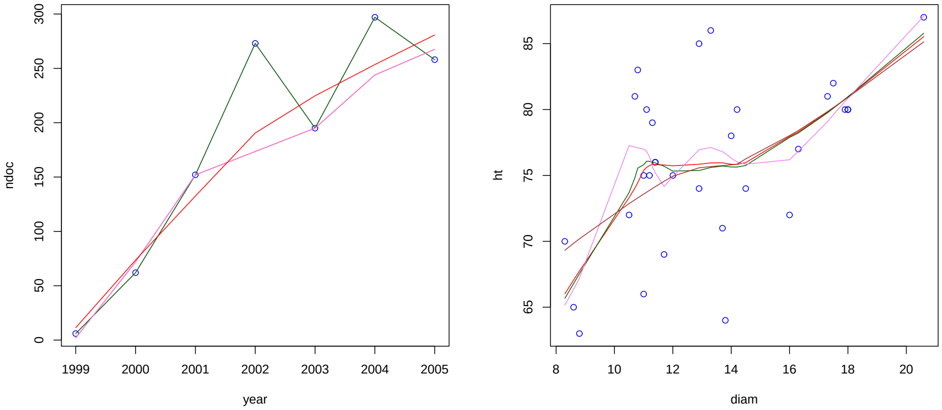


Figure 7.22: Illustration of local linear regression using the `loess` package. Left: Applied to a dataset of cumulative publications from an experiment by year. The green lines just connect the dots. The other curves show the results for variations on the regression. Right: Various regressions applied to a dataset of tree heights vs diameter (from R).

$\widehat{N}_k$ is the estimate of the number of events in category k, according to the reduced fit. $f_j(x'_e)$ is the PDF for category j evaluated at x'_e.

Finally, the sPlot is constructed by adding each event e with $y = y_e$ to the y-histogram (or scatter plot, etc, if y is multivariate), with weight $w_j(e)$. The resulting histogram is then an estimator for the true distribution in $y$for category j. Figures 7.23 and 7.24 provide illustrations of the use of sPlots.

Typically the sPlot error in a bin is estimated simply according to the sum of the squares of the weights. This sometimes leads to visually misleading impressions, due to fluctuations on small statistics. If the plot is being made for a distribution for which there is a prediction, then that distribution can be used to estimate the expected uncertainties, and these can be plotted. If the plot is being made for a distribution for which there is no prediction, it is more difficult, but a (smoothed) estimate from the empirical distribution may be used to estimate the expected errors.

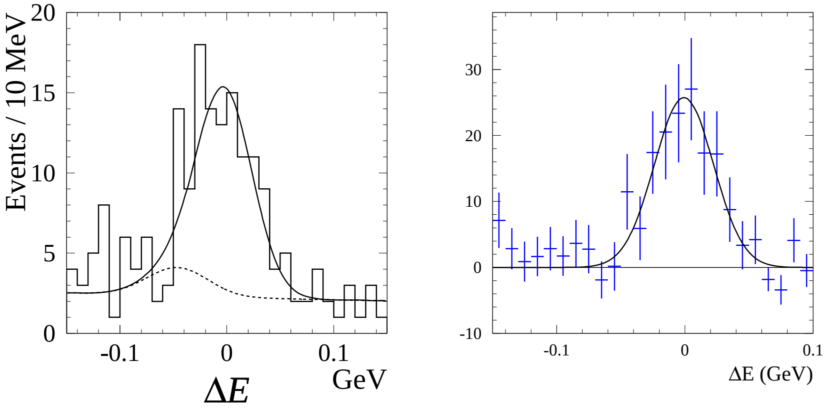


Figure 1. Signal distribution of the ΔE variable. The left figure is obtained applying a cut on the Likelihood ratio to enrich the data sample in signal events (about 60% of signal is kept). The right figure shows the ${}_s\mathcal{P}lot$ for signal (all events are kept).

Figure 7.23: Illustration of the sPlot technique. Left: A non-sPlot, which uses a subset of the data in an attempt to display signal behavior. Right: An sPlot for the signal category. The curve is the expected signal behavior. Note the excess of events at low values of ΔE. This turned out to be an unexpected portion of the signal distribution, which was found using the sPlot. From: M. Pivk, "sPlot: A Quick Introduction", arXiv:physics/0602023 (2006).

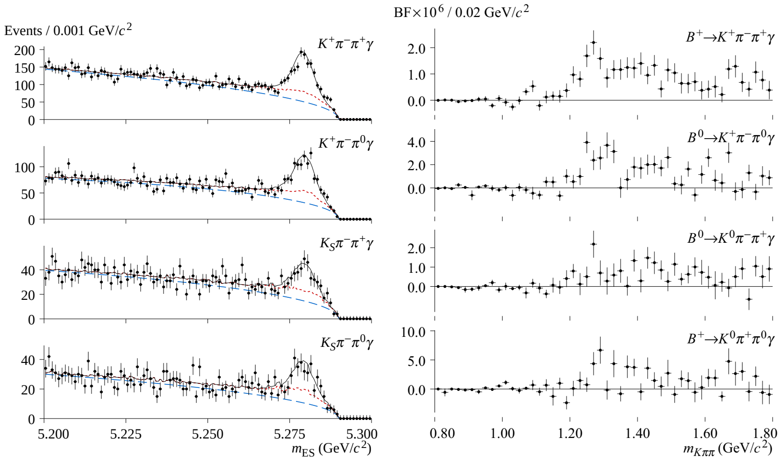


Figure 7.24: The sPlot technique used for background subtraction in mass spectra. Left: The spectra before background subtraction. Right: The spectra after background subtraction using sPlots. From hep-ex/0507031.

Bibliography

[1] W.-M. Yao et al, J. Phys. G 33, 1 (2006).

[2] J. S. Halekas et al., “Magnetic Properties of Lunar Geologic Terranes: New Statistical Results”, Lunar and Planetary Science XXXIII (2002), 1368.pdf)

[3] http://www-personal.buseco.monash.edu.au/∼hyndman/papers/sturges.pdf.

[4] David W. Scott, Multivariate Density Estimation, John Wiley & Sons, Inc., New York (1992).

[5] K. S. Cranmer, “Kernel Estimation in High Energy Physics”, Comp. Phys. Comm. **136**, 198 (2001) [hep-ex/0011057v1]; http://arxiv.org/PS_cache/hep-ex/pdf/0011/0011057.pdf; http://java.freehep.org/jcvslet/JCVSlet/diff/freehep/freehep/hep/aida /ref/pdf/NonParametricPdf.java/1.1/1.2.

[6] W. Verkerke and D. Kirkby, RooFit Users Manual V2.07: http://roofit.sourceforge.net/docs/RooFit_Users_Manual_2.07-29.pdf.

[7] G. Brandenburg et al., “Determination of the $K^*(1800)$ Spin Parity”, SLAC-PUB-1670 (1975).

[8] Glen Cowan, “Statistical Data Analysis”, Oxford University Press (1998).

[9] V. Blobel, http://www.desy.de/∼blobel/wwwrunf.html.

[10] A. Höcker & V. Kartvelishvili, “GURU”, NIM A bf 372, 469 (1996).

[11] Glen Cowan, http://www.ippp.dur.ac.uk/Workshops/02/statistics/proceedings//cowan.pdf.

[12] M. Pivk & F. R. Le Diberder, “sPlot: a statistical tool to unfold data distributions”, Nucl. Instr. Meth. A **555**, 356 (2005).