

Density Estimation

Abi Adams

Hilary 2018

You may be interested in the relationship between y and x .

Theory may include/exclude variables, imply monotonicity or concavity, or optimising behaviour.

It almost NEVER implies a functional form.

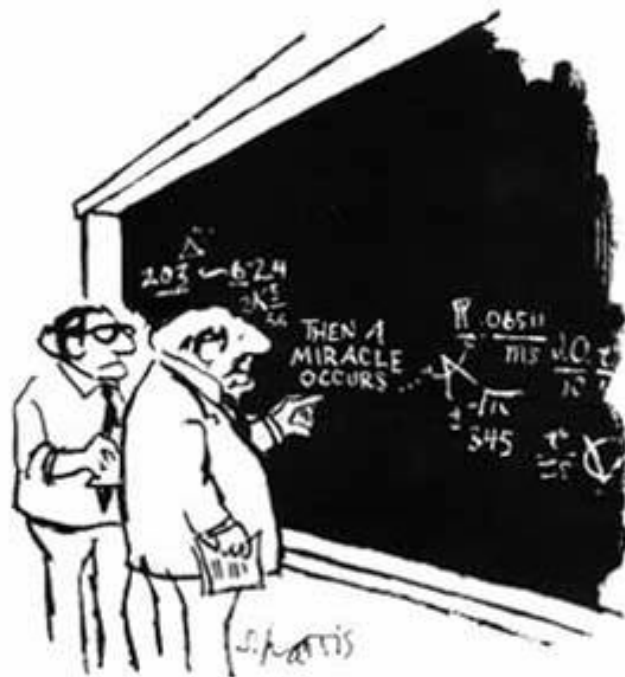
The implications of economic theory are generally *nonparametric*.

We are used in econometrics to supplying parametric functional forms to flesh out the theory.

Since, for the most part, the theory is silent about functional forms we use “fit” to distinguish between alternatives.

But this is not always easy if models are non-nested.

Nonparametric methods offer flexible approaches to modelling in which the functional form of the object of interest is not pre-specified.



"I think you should be more explicit here in step two."

The emphasis will be on **practicalities**. For the “big O, little o” stuff I will mostly refer you to.

Applied nonparametric regression Wolfgang Härdle, Cambridge : Cambridge University Press, 1990

Density estimation for statistics and data analysis B.W. Silverman, London : Chapman and Hall, 1986.

Semiparametric regression for the applied econometrician Adonis Yatchew, Cambridge : Cambridge University Press, 2003.

Nonparametric curve estimation : methods, theory and applications Sam Efromovich, New York : Springer, 1999.

1. Nonparametric Density Estimation

i. The basic idea and the naive estimator

ii. Kernel density estimation

iii. Properties of kernel density estimates

iv. Bandwidth selection, adaptive estimation and nearest-neighbour methods

v. Multivariate density estimation and the curse of dimensionality

vi. Other (fast) methods - series estimators.

Parametric Density Estimation

The probability density function is a fundamental concept in statistics.

It's a natural description of the distribution of an r.v. and allows probabilistic statements about r.v.'s to be made.

The parametric approach to density estimation assumes that the data are drawn from one of a known family of distributions, e.g. the normal distribution with mean μ and variance σ^2 .

The density underlying the data can then be estimated by finding estimates of these two parameters - I will not be talking about these methods at all; I will assume that they are completely familiar.

Nonparametric Density Estimation

Nonparametric estimation methods are less prescriptive

They assume that the data *has* a density but beyond that the data are “allowed to speak for themselves” in determining the estimate to a much greater extent than the parametric method allows.

They are also familiar.

In fact you have been using them ever since you started to study statistics.

The Histogram

The most widely used density estimator is the *histogram*.

You specify an origin (x_0) and a bin-width (h) - the latter mainly controls the number of bins/the smoothness of the estimator.

Bins are intervals $[x_0 + mh, x_0 + (m + 1)h)$ $m = 0, 1, 2, \dots$

Given an i.i.d. sample X_1, \dots, X_n the histogram is

$$\hat{f}(x) = \frac{1}{nh} [\text{No. of observations in the same bin as } x]$$

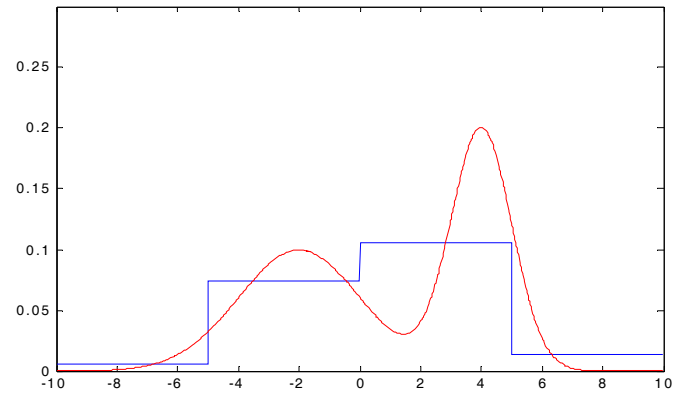
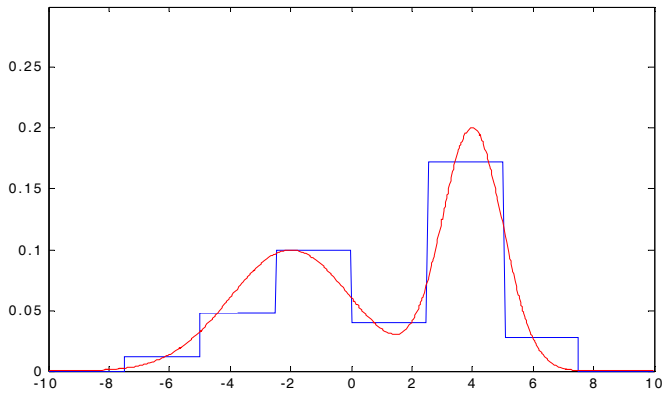
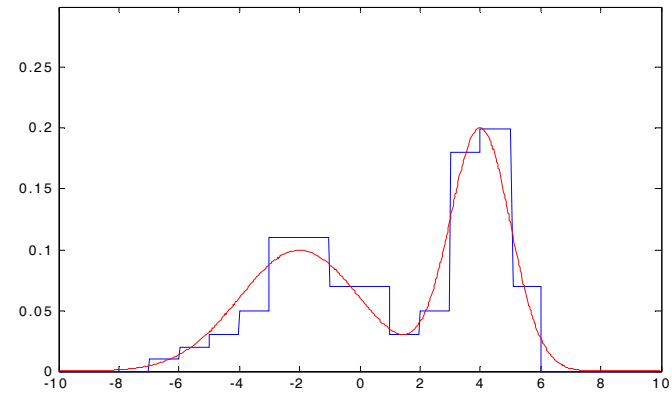
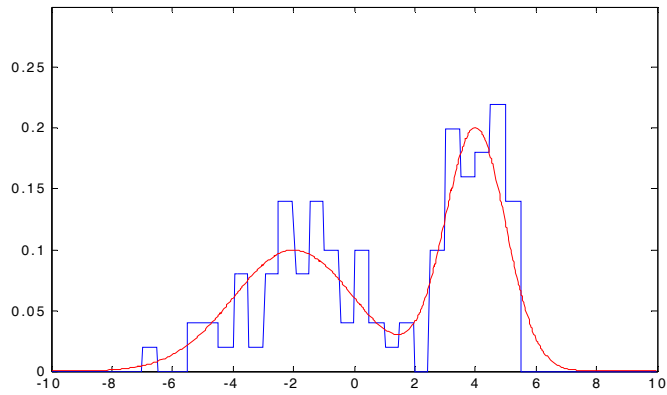
In what follows we have an i.i.d. sample of $n = 100$ from a bimodal distribution:

$$f(x) = 0.5d_{-2,2}(x) + 0.5d_{4,1}(x)$$

where

$$d_{\mu,\sigma} = (2\pi\sigma^2)^{-1/2} e^{-(x-\mu)/2\sigma^2}$$

Histograms with $h = \{0.25, 1, 2.5, 5\}$, $x_0 = -10$



In general

- under-smoothing produces a noisy, wiggly picture with many artificial and confusing modes,

while

- over-smoothing hides modes and obscures the fine structure.

The patterns in these figures reflect the most important issue for any nonparametric estimator, namely, how much to smooth the data.

The Empirical CDF

Suppose that we observed an i.i.d. sample X_1, \dots, X_n drawn from a distribution F .

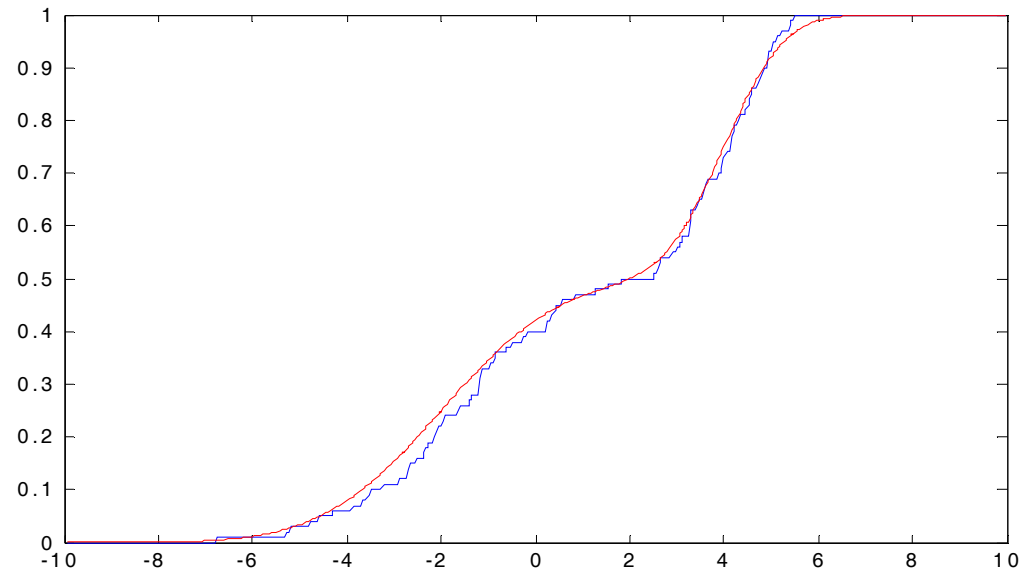
We can estimate the c.d.f. by

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(X_i \leq x)$$

where $\mathbf{1}(\cdot)$ is the indicator function.

Lets see how it does ...

The Empirical CDF



This is a step function. The heights of the jumps are n^{-1} in general (assuming no ties)

Unfortunately you can't use the empirical c.d.f. to estimate the p.d.f. by taking derivatives and using

$$f(x) = \frac{dF_n(x)}{dx}$$

This is because the c.d.f. estimator is discontinuous at the sample points and constant elsewhere.

The Naive Estimator

From the definition of a probability density, if x has density $f(x)$, then

$$f(x) = \lim_{h \rightarrow 0} \frac{1}{2h} P(x - h < X < x + h)$$

The naive density estimator mimics this.

$$\hat{f}(x) = \frac{1}{2nh} [\text{No. of observations falling in } (x - h, x + h)]$$

or

$$\hat{f}(x) = \frac{1}{2nh} \sum_{i=1}^n \mathbf{1}(|X_i - x| \leq h)$$

The Naive Estimator

$$\hat{f}(x) = \frac{1}{2nh} \sum_{i=1}^n \mathbf{1}(|X_i - x| \leq h)$$

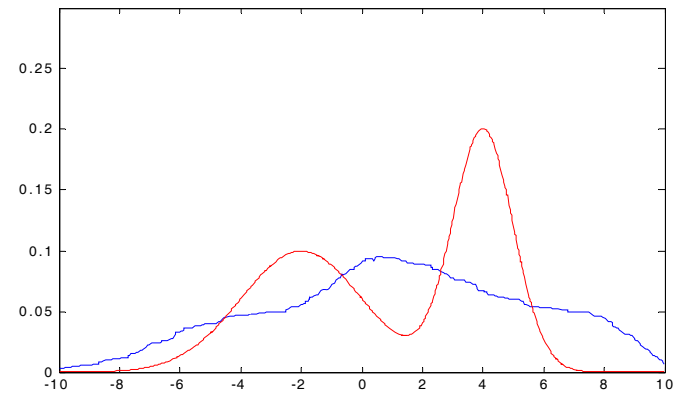
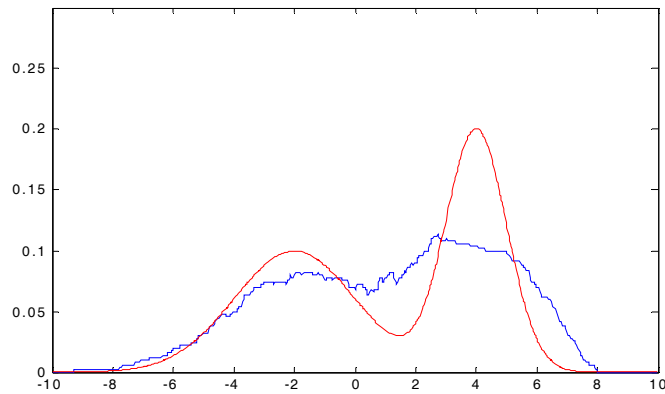
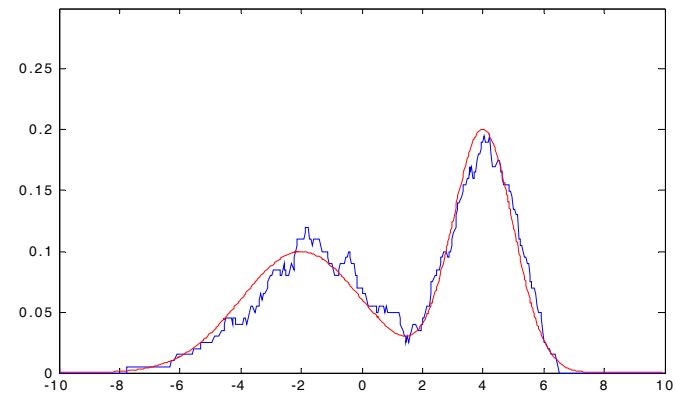
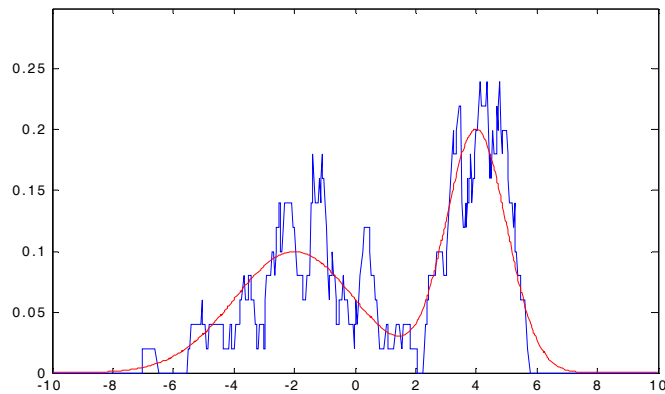
This can be written more succinctly (and more naturally as we shall see) as

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n w\left(\frac{X_i - x}{h}\right)$$

where $w(\cdot)$ is a weight function

$$w(x) = \frac{1}{2} \mathbf{1}(|x| \leq 1)$$

The Naive Estimator with $h = \{0.25, 1, 2.5, 5\}$



The Naive Estimator

The fundamental smoothness vs retention of features trade-off is still evident.

The ragged appearance comes from the fact that the estimator is a step-wise constant with jumps at the points $X_i + h$

The main difference from the histogram are

1. There is no origin
2. The centre of the bin is an observation.

The Kernel Estimator

The naive estimator is a particular example of a large class of estimators called *kernel* estimators.

Kernel smoothing can be used for any statistical model (we're interested in density estimation at the moment).

The naive estimator is an example but it has the drawback of being "steppy".

But it's easy to generalise this estimator to avoid this problem.

Namely, replace the rectangular weight function with a smooth weight function.

The Kernel Estimator

Kernel theory refers to the weight function as a “*kernel function*” and it is denoted $K(v)$. By assumption the kernel function:

$$\begin{array}{ll} \int K(v) dv = 1 & \text{integrates to one} \\ K(v) = K(-v) & \text{is symmetric} \\ \int vK(v) dv = 0 & \text{is mean zero (implied by sym)} \\ \int v^2 K(v) dv = \kappa_2 > 0 & \text{has positive variance} \end{array}$$

The Kernel Density Estimator

Given some choice of kernel function, the kernel density estimator is defined by

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

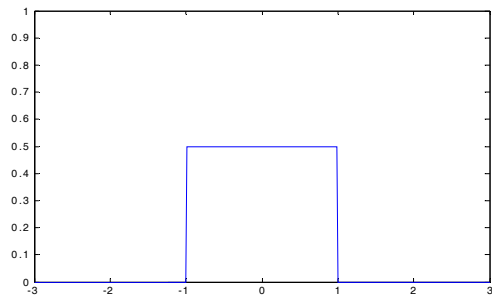
where h is referred to as the *bandwidth* or *smoothing parameter*.

NB this is **exactly** the same general form as the naive estimator.

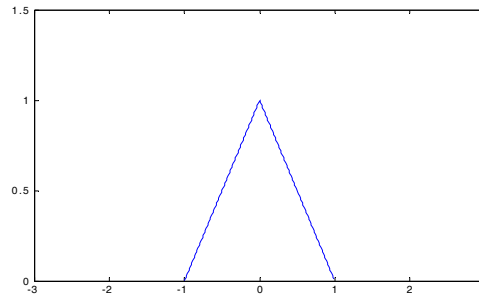
Some Kernel Functions

Kernel	$K(v)$	
Rectangular	$\frac{1}{2}$	for $ v < 1$, 0 otherwise
Triangular	$1 - v $	for $ v < 1$, 0 otherwise
Biweight	$\frac{15}{16} (1 - v^2)^2$	for $ v < 1$, 0 otherwise
Triweight	$\frac{35}{32} (1 - v^2)^3$	for $ v < 1$, 0 otherwise
Epanechnikov	$\frac{3}{4} \left(1 - \frac{1}{5}v^2\right) 5^{-0.5}$	for $ v < 5^{0.5}$, 0 otherwise
Gaussian	$(2\pi)^{-1/2} \exp(-0.5v^2)$	

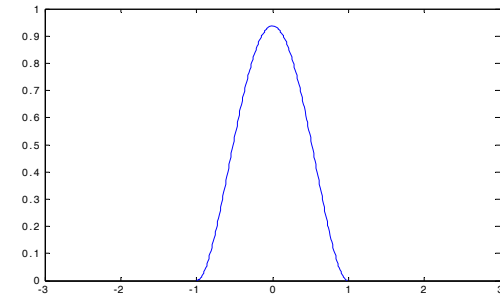
Rectangular



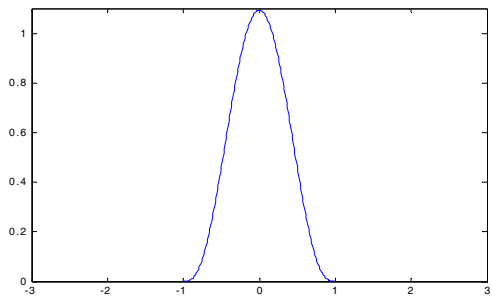
Triangular



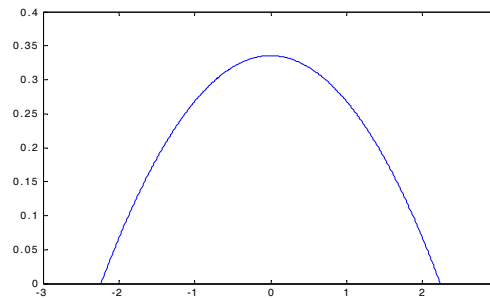
Biweight



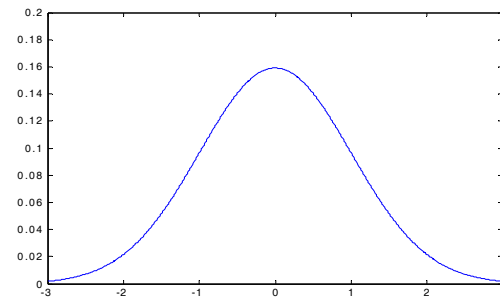
Biweight



Epanechnikov

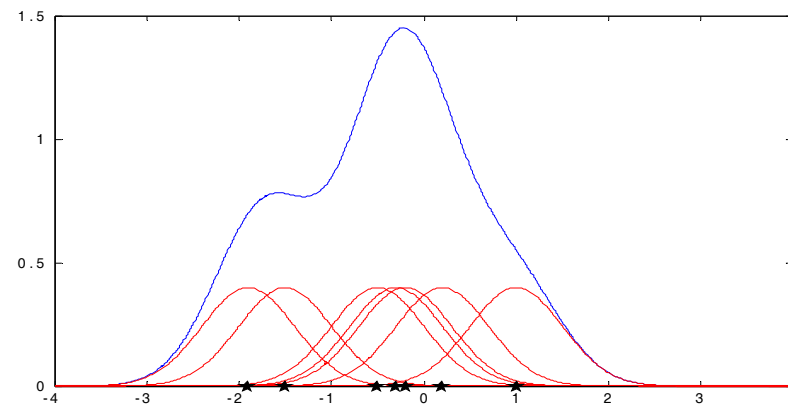


Gaussian



How Kernels Work

Just as the naive estimator can be considered the “sum of boxes”, Kernel estimators can be considered the “sum of bumps”.



Each kernel is centred at an X_i and the kernel estimator is constructed as the average of the kernel ordinates at that point.

Implementation - Gaussian kernel density estimator

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

```
n = length(X); x = -10:0.01:10; J = length(x);
```

```
h = 1; fx = NaN*ones(1,J);
```

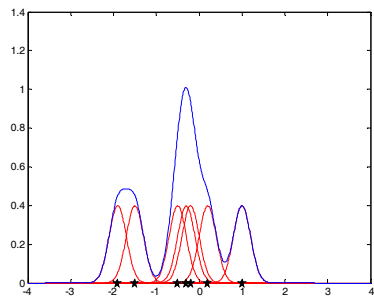
```
for j=1:J;
```

```
fx(j)=(1/(n*h))*sum((1/sqrt(2*pi))*exp(-0.5*((X-x(j))/h).^2));
```

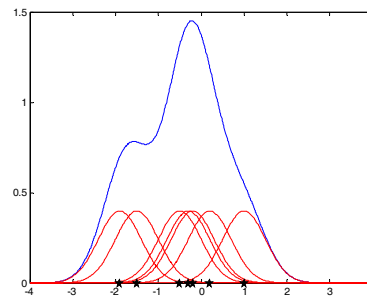
```
end;
```

How Kernels Work - The effect of bandwidth

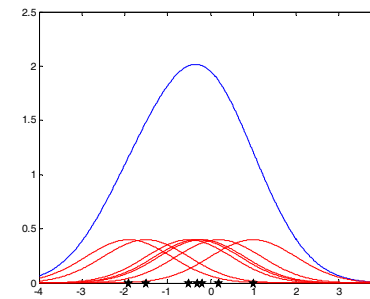
$h = 0.2$



$h = 1$



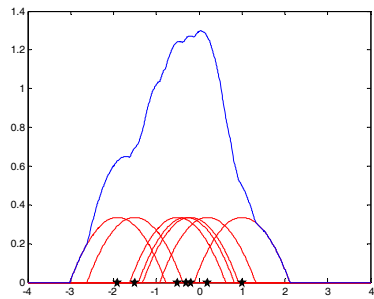
$h = 2$



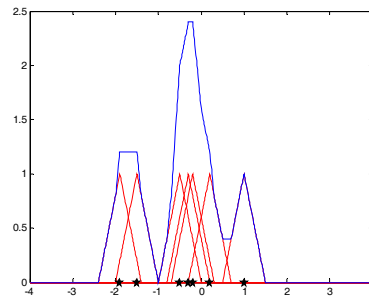
How Kernels Work - The effect of Kernel function

(NB - 7 observations!!!)

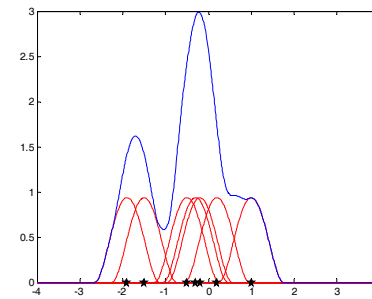
Epanechnikov



Triangular

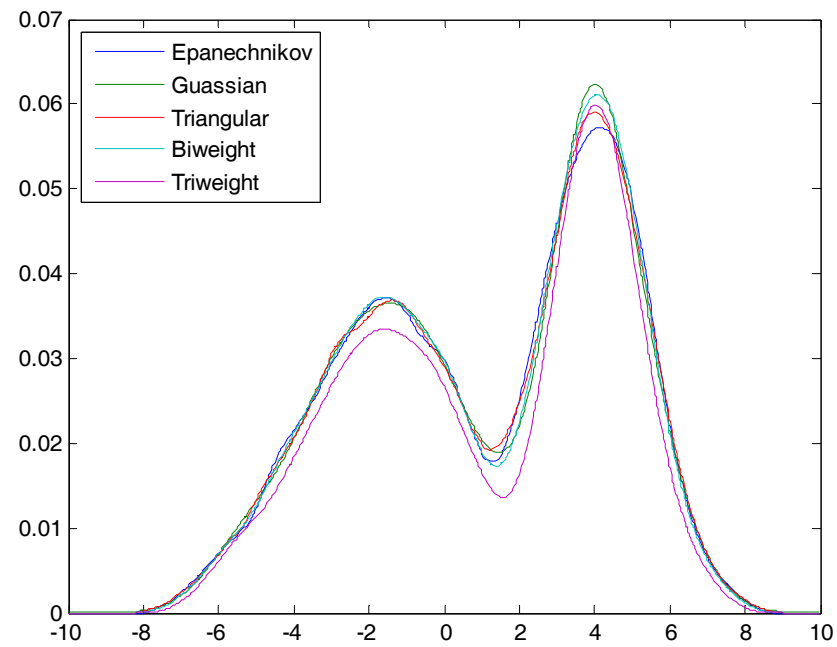


Biweight



How Kernels Work - The effect of Kernel function

(500 observations)



How Kernels Work

Given a reasonable number of observations (and we shall return to this point).

Choice of bandwidth/smoothing-parameter is much more crucial than choice of the kernel function

Except, perhaps, at the “edge” of the data.

Kernel Density Estimators - Properties

Some elementary properties of the kernel estimator follow immediately from the definition

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

1. Provided the kernel is non-negative and integrates to one then $\hat{f}(x)$ will itself be a p.d.f.
2. $\hat{f}(x)$ inherits all of the continuity and differentiability properties of the kernel.

Kernel Density Estimators - Elementary Properties

The key questions (as it is with any estimator) are

1. "is it any good?"
2. "how can I make it better?"

Both relate to how close the estimator \hat{f} is to the true density f .

Mean Square Error

When considering estimation at a single point (x) a natural measure is the mean square error (MSE)

$$\begin{aligned}MSE(\hat{f}(x)) &= E[\hat{f}(x) - f(x)]^2 \\ &= [E\hat{f}(x) - f(x)]^2 + \text{var}\hat{f}(x) \\ &= \text{squared bias plus the variance at } x\end{aligned}$$

Mean Integrated Squared Error

The most widely used global measure of the accuracy of $\hat{f}(x)$ adds up the MSE over the domain of the function. This is the mean integrated square error (MISE)

$$\begin{aligned} MISE(\hat{f}(x)) &= E \int [\hat{f}(x) - f(x)]^2 dx \\ &= \int [E\hat{f}(x) - f(x)]^2 dx + \int Var\hat{f}(x) dx \\ &= \text{integrated sq bias and variance} \end{aligned}$$

MSE and MISE

$$E\hat{f}(x) = \int \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) f(x) dx$$
$$\text{Var}\hat{f}(x) = \int \left[\frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) \right]^2 f(x) dx$$
$$- \left[\int \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) f(x) dx \right]^2$$

Exact expressions for MSE/MISE can be obtained by plugging these expression into the definitions (in principle). In general plugged these expression for the

mean and variance into the expressions for MSE/MISE will leave you with an intractable mess to deal with.

An exception is the case of the Gaussian kernel for a true density which is also normal.

*MSE and MISE**

$$MISE = \frac{1}{2\pi^{1/2}} n^{-1} \left\{ h^{-1} - (\sigma^2 + h^2)^{-1/2} \right\} + \sigma^{-1} + (\sigma^2 + h^2)^{-1/2} - 2(2)^{1/2} (2\sigma^2 + h^2)^{-1/2}$$

This can be minimised w.r.t. h to find the optimal (MISE-minimising) bandwidth.

*Fryer (1976)

Typically it is easier to obtain approximations to $E\hat{f}(x)$ and $Var\hat{f}(x)$ and use these to investigate the bias/variance trade-off..

The basic issue though is as follows:

Optimal estimation is based on a balance between bias and variance.

The key parameter is the bandwidth.

$h \uparrow$	Increases bias	Decreases variance
$h \downarrow$	Decreases bias	Increases variance

Asymptotic Properties

It can be shown that

$$\begin{aligned} \text{bias} \hat{f}(x) &\approx \frac{1}{2} h^2 f''(x) \kappa_2 \\ \text{var} \hat{f}(x) &\approx n^{-1} h^{-1} f(x) \kappa_1 \\ \int \text{bias} \hat{f}(x)^2 dx &\approx \frac{1}{4} h^4 \kappa_2^2 \int f''(x)^2 dx \\ \int \text{var} \hat{f}(x) &\approx n^{-1} h^{-1} \kappa_1 \end{aligned}$$

where $\kappa_1 = \int K(v)^2 dv$ and $\kappa_2 = \int v^2 K(v) dv$

Suppose we want to minimise MISE.

$$\int \text{bias} \hat{f}(x)^2 dx \approx \frac{1}{4} h^4 \kappa_2^2 \int f''(x)^2 dx$$

$$\int \text{Var} \hat{f}(x) \approx n^{-1} h^{-1} \kappa_1$$

$$\text{MISE} \approx \frac{1}{4} h^4 \kappa_2^2 \int f''(x)^2 dx + n^{-1} h^{-1} \kappa_1$$

A small value of h will eliminate the integrated squared bias, but the integrated variance becomes large. And *vice versa*.

NB this does not just apply to kernel estimators - the trade-off should be familiar.

The asymptotically ideal value of h (MISE-minimising) is found by minimising

$$MISE(h) \approx \frac{1}{4}h^4 \kappa_2^2 \int f''(x)^2 dx + n^{-1}h^{-1} \kappa_1$$

which gives

$$h^* = c_0 n^{-1/5}$$

where

$$c_0 = \kappa_2^{-2/5} \kappa_1^{1/5} \left[\int f''(x)^2 dx \right]^{-1/5} > 0$$

What does this mean?

First note that as $h^* \rightarrow 0$ bias disappears: the model converges to the truth.

1. h^* goes to zero as the sample size increases, but slowly
2. h^* depends on the density being estimated
3. h^* depends on the roughness of the density - rougher densities imply narrower bandwidths.

Plug the formula for h^* back into $MISE(h^*)$ and you get

$$MISE(h^*) \approx \frac{1}{4} \left(\kappa_2^{-2/5} \kappa_1^{1/5} \left[\int f''(x)^2 dx \right]^{-1/5} n^{-1/5} \right)^4 \kappa_2 \int f''(x)^2 dx \\ + n^{-1} \left(\kappa_2^{-2/5} \kappa_1^{1/5} \left[\int f''(x)^2 dx \right]^{-1/5} n^{-1/5} \right)^{-1} \kappa_1$$

$$MISE(h^*) \approx \frac{5}{4} C \left(\int f''(x)^2 dx \right)^{1/5} n^{-4/5}$$

$$C = \kappa_2^{2/5} \kappa_1^{4/5}$$

We should choose a kernel with a small value of C – this should make it possible to get a small value of $MISE$ given we choose the bandwidth parameter well. The Epanechnikov kernel has something to recommend it on this criteria.

Other approaches

1. It's not very scientific but aesthetics play a role - choose an estimator which looks right.
2. Given perhaps a pilot estimate have a look at it and then choose a standard distribution which fits to compute the $f''(x)$ term.
3. Cross-validation

Cross-validation

Suggested by Rudemo (1982) and Bowman (1984), it's an automated method for bandwidth selection.

The idea is the the integrated squared error is

$$\int [\hat{f}(x) - f(x)]^2 dx = \int \hat{f}^2(x) dx - 2 \int \hat{f}(x) f(x) dx + \int f(x)^2 dx$$

Only the first two terms depend on h so we just need to choose h to minimise

$$\int \hat{f}^2(x) dx - 2 \int \hat{f}(x) f(x) dx$$

For the 1st term we use

$$\int \hat{f}^2(x) dx = \frac{1}{n^2 h} \sum_{i=1}^n \sum_{j=1}^n K^* \left(\frac{X_i - X_j}{h} \right)$$

where $K^*(.)$ is the convolution of the kernel with itself (in the case of the Gaussian kernel this is just a normal with a variance of 2 (rather than 1)).

For the second term we use

$$2 \int \hat{f}(x) f(x) dx = \frac{2}{n(n-1)h} \sum_{i=1}^n \sum_{j=1, j \neq i}^n K \left(\frac{X_i - X_j}{h} \right)$$

The second term is the mean of the leave- i -out estimate of the density evaluated at observation i . NB if you don't do this the method breaks down and you get $h = 0$ as the value which minimises the MISE

You need to evaluate n different kernel estimates to apply this.

This is just an application of the idea that out-of-sample prediction is a useful criterion for estimation and testing (e.g. the Chow test in time series analysis)

It turns out that under very mild assumptions, asymptotically cross-validation achieves the best possible choice of smoothing parameter (Stone 1984).

So far we've been worrying about the best choice of bandwidth.

Even so the optimal bandwidth stays fixed across the whole estimation range.

Whilst this is globally best (in the MISE-minimising sense) it can lead to local problems.

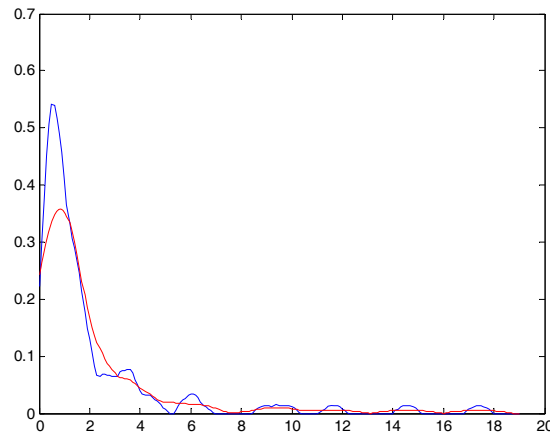
This is because, fundamentally, the intuition of the naive estimator still stands - we are measuring the expected number of observations in a box of fixed width.

A bandwidth which is optimal in regions in which the data are dense won't still be optimal in regions in which the data are sparse.

Eg. 100 observations from a log-normal.

Fixed bandwidth tends to give spurious bumps in the tail.

Increasing the bandwidth smooths the bumps but sacrifices features of the main part of the density.



Nearest Neighbour Methods

Recall that the basic idea was based on the ideas

1. $f(x) = \lim_{h \rightarrow 0} \frac{1}{2h} P(x - h < X < x + h)$

2. The expected number of observations falling in the interval $(x - h, x + h)$ is $n(2h) f(x) = k$

The naive estimator was based on the idea of counting the number of observations in the box $(x - h, x + h)$, calling this number \hat{k} then setting

$$\hat{f}(x) = \frac{\hat{k}}{2nh}$$

Nearest Neighbour Methods

The nearest neighbour (NN) method takes k as fixed and varies the box width.

It calculates the distance \hat{h} from x to the k th observation and this gives the *k th nearest neighbour kernel estimator*.

$$\tilde{f}_n(x) = \frac{k}{2n\hat{h}}$$

The bandwidth at x is calculated as the absolute difference between x and the k th nearest neighbour and so depends on the density of the data in that neighbourhood.

This makes the dependence of the bandwidth on the data look fairly data-driven and it is, except for the fact that the investigator must choose k

Applied to kernel estimators this gives

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

Adaptive Kernels

The nearest-neighbour method is pretty easy to apply, but it has some undesirable features relating to the differentiability and integrability of the resulting estimator at points where the bandwidth changes (see Silverman, 2.5, p.21).

Nevertheless the idea of allowing the bandwidth to go up in regions where the data are sparse is a good one.

The idea of adaptive kernel estimation uses this idea by smoothing things out.

Adaptive Kernels

It is a two stage procedure

1. Firstly decide where the area of low density are.
2. Secondly use this information to control the bandwidth as you move through the data.

Adaptive Kernels

Step 1. Find a *pilot estimate* $\tilde{f}(x)$ which satisfies $\tilde{f}(X_i) > 0$ for all i

Define the *local bandwidth factors* λ_i by

$$\lambda_i = \left(\frac{\tilde{f}(X_i)}{\prod_{i=1}^n \tilde{f}(X_i)^{1/n}} \right)^{-\alpha}$$

where $\alpha \in [0, 1]$ is a sensitivity parameter.

Step 2. Define the adaptive kernel estimate $\hat{f}_n(x)$

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

Adaptive Kernels

1. The local bandwidth factors scale the bandwidth inversely according to how dense the data are locally (as determined by the pilot estimate).
2. The first step requires a pilot kernel. The consensus in the literature (Bremain *et al* (1977), Abramson (1882)) is that the method is insensitive to the fine detail of the pilot estimate. Also, since this does not need to have any special smoothness properties (in fact it seems that under-smoothing the pilot might work best) and since time costs matter the Epanechnikov kernel is often used. A time-consuming approach like cross-validation is not generally considered worthwhile (see Silverman, 5.3.1, p. 102).

3. The sensitivity parameter controls the way in which the estimate responds to variations in the pilot estimate. Having $\alpha = 0$ simply returns the standard estimator as this will set $\lambda_i = 1$. Higher values increase the sensitivity.
4. The resulting estimate will be a *bona fide* probability density (unlike the NN method).

Multivariate Density Estimation

Multivariate d -dimensional density estimation is easy *in principle*.

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\det \mathbf{H}} \mathcal{K}(\mathbf{H}^{-1}(\mathbf{X}_i - \mathbf{x}))$$

where \mathbf{H} is a bandwidth *matrix* and $\mathcal{K}(\mathbf{x})$ is a multivariate kernel like the multivariate normal

$$\mathcal{K}(\mathbf{x}) = 2\pi^{-d/2} \exp\left(-\frac{1}{2}\mathbf{x}'\mathbf{x}\right)$$

or the multivariate Epanechnikov

$$\begin{aligned} \mathcal{K}(\mathbf{x}) &= \frac{1}{2}c_d^{-1}(d+2)\left(1-h^{-1}\mathbf{x}'\mathbf{x}\right) \text{ if } \mathbf{x}'\mathbf{x} < 1 \\ &= 0 \text{ otherwise} \end{aligned}$$

where c_d is the volume of the unit d -dimensional sphere ($c_2 = \pi, c_3 = 4\pi/3, \dots$)

However, the easiest solution is choosing the form for $\mathcal{K}(\cdot)$ is to use a multiplicative kernel

$$\mathcal{K}(\mathbf{X}_i, \mathbf{x}) = K\left(\frac{X_{i1} - x_1}{h_1}\right) \dots K\left(\frac{X_{id} - x_d}{h_d}\right)$$

in which case

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d \frac{1}{h_j} K\left(\frac{X_{ij} - x_j}{h_j}\right)$$

(this is like having a bandwidth matrix which is $\mathbf{H} = \text{diag}(h_1, \dots, h_d)$)

We won't go through the asymptotics for the multivariate kernel. They are analogous to the univariate case but much more cumbersome (see Scott 1992).

The bottom line is that the multivariate density has a slower rate of convergence than the univariate especially as d gets big.

And the optimal bandwidth (in the diagonal case) has to be considerably bigger to make sure that the estimate is reasonably smooth.

It's a curse

The Curse of Dimensionality.

Suppose that $n = 100$ points are independently uniformly distributed within a 5-dimensional unit cube $[0, 1]^5$

What is the probability of having some points in a neighbourhood of a reasonable size, say a cube with side 0.2?

The volume of such a cube is $0.2^5 = 0.00032$

The expected number of observations in such a neighbourhood is 0.032.

To get an expected number of 5 would require a cube whose side was 0.55, i.e. more than half the range in each dimension (hardly “local”)

The Curse of Dimensionality.

Multivariate data is *extremely* sparse - much more so than we normally think when we are used to fitting parametric models.

In comparison to parametric estimation, nonparametric estimation imposes enormous data requirements. For example, to ensure that the relative MSE at $f(0)$ is less than 0.1 requires (for normal $f(\mathbf{x})$ and Gaussian $K()$)

d	1	2	3	4	5	6	7	8	9	10
n	4	19	67	233	768	2790	10700	43700	187000	842000

The Curse of Dimensionality.

Weirdly, in high dimensions a great deal of the density is in the “tails”!

In a 10D normal 99% of the mass of the distribution is further than 1.645 from the origin.

In the univariate case 90% of the mass lies within ± 1.645 .

Like the example discussed above this is an example of the *empty space* phenomenon (Scott and Thompson (1986)).

Series Estimation

Kernel methods are probably the most well-established approach to nonparametric estimation.

But it's not the only method and whilst the asymptotics are all worked out and fully enunciated it can be computationally slow.

One *quick* alternative is to use **series estimation**.

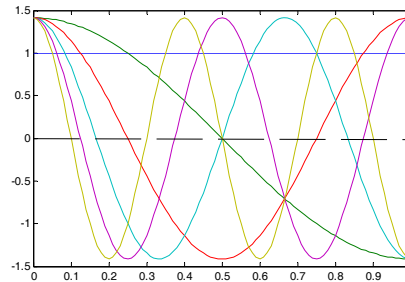
Unlike kernel methods which estimate the curve of the density function at each point x series estimators estimate the whole curve.

The speed with which one can recover the density makes this idea for bootstrapping.

Other Methods: Series Estimation

Under mild assumptions a function $f(x)$ [which is really just a curve] can be represented as the Fourier series $f(x) = \sum_{j=0}^{\infty} w_j \theta_j \varphi_j(x)$ where $\varphi_j(x)$ is an orthonormal basis function. For example the cosine basis function

$$\varphi_0(x) = 1, \quad \varphi_j(x) = 2^{1/2} \cos(\pi j x), \text{ for } j = 1, \dots$$



The density estimate is then constructed by "adding up the wiggles" of the basis functions.

The terms θ_j are known as Fourier coefficients and they are given by $\theta_j = \int \varphi_j(x) f(x) dx = E\varphi_j(x)$, and the terms w_j are weights.

Of course an infinite sum is clearly infeasible and in any case Fourier coefficients decline, so the aim is to find an acceptable cutoff \hat{J}

$$\hat{f}(x) = \sum_{j=0}^{\hat{J}} \hat{w}_j \hat{\theta}_j \varphi_j(x)$$

Series estimation methods then boil down to selection of

1. the cut-off J
2. the smoothing weights w_j
3. the estimate of the Fourier coefficients $\hat{\theta}_j$

Fourier coefficients: they are means

$$\hat{\theta}_j = \int \varphi_j(x) f(x) dx = E\varphi_j(x) = n^{-1} \sum_{i=1}^n \varphi_j(X_i)$$

Cut-off :

$$\hat{J} = \arg \min_{0 \leq J \leq J_n} \sum_{j=0}^J \left(2\hat{d}n^{-1} - \hat{\theta}_j^2 \right)$$

Weights :

$$\hat{w}_0 = 1, \quad \hat{w}_j = \left(1 - \hat{d}/n\hat{\theta}_j^2 \right)_+$$

where $\hat{d} = \hat{\theta}_0$ (see Efromovich, pp 60-63)

The series estimator is (very) quick compared to kernel estimation methods.

But except for somewhat special choices of weights the resulting density can't be guaranteed to be non-negative (!)

Somewhat *ad hoc* fixes are put in place to make sure that (i) it doesn't go negative (ii) extraneous wiggles are removed and (iii) it still integrates to one (see Efromovich chapter 3 for discussion)

Density Estimation: Summary

Nonparametric density estimation by kernel methods is essentially a generalisation of histogram/*local averaging*.

Choice of kernel function is unimportant if there are a reasonable number of observations (locally).

Choice of smoothing parameter is crucial - CV is a robust automated, but slow, method.

A fixed bandwidth (even if MISE-optimal) has certain drawbacks - adaptive or nearest-neighbour methods

Multivariate density estimation is a straightforward extension

But beware *The Curse*.

Nonparametric estimators converge slowly so demand large datasets.

The main alternative method is via orthogonal series estimators - less good in some ways but very fast.