Technical details of 'kerneval'

Gawain T Antell

2023-07-12

Approximating functions and integration

In R, probability density estimates are evaluated at discrete points. However, to work with the probability density (e.g. calculate the probability over an interval or the overlap between distributions), one usually needs a description of the probability density as a continuous function; point estimates on their own have little use besides plotting. One could use a linear (`stats::approxfun()`), spline (`stats::splinefun()`), or polynomial function for approximation. Due to the mostly concave-down, unimodal shape of many niches, linear interpolation tends to under-estimate the probability when integrated compared to spline interpolation. However, the difference is small, and splines can lead to negative probability values in niche tails. **kerneval** interpolates linearly between adjacent point estimates. The default discretisation ($N = 2^9$ for `base::density()`) gives good behaviour in the author's experience, but one could specify a different resolution. There is a trade-off between scale and computation time.

Analagously, one could use any of a variety of functions to test that the probability density function (PDF) integral sums to unity. Common approximations of the integral include rectangular, trapezoidal, Gauss-Kronrod, or Simpson adaptive numerical integration. In the author's experience with simulated and empirical data, all integration options are equivalent to at least 3 significant figures, so the default `stats::integrate()` implementation is sufficient for ecological analysis.

Comparison with other R packages

Both the **GoFKernel** and **ecospat** packages integrate PDFs to rescale the density estimate. The `density.reflected()` function in **GoFKernel** (Pavia 2015) approximates the PDF integral with rectangles, as illustrated in Figure 1. For comparison, a trapezoial approximation treats the point estimates of the PDF as the top vertices of adjacent trapezoids (Fig. 2). Where the curve is concave (in the middle), trapezoids under-estimate the integral. Where the curve is convex (in the tails), trapezoids over-estimate the integral. In general, over the entire distribution, trapezoids approximate functions well.

One can quickly calculate the true integral of the normal function on the interval $[-3, 3]$ (the green curve in Figs. 1 and 2):

```
pnorm(3) - pnorm(-3)
```

```
## [1] 0.9973002
```

The rectangular estimate (Fig. 1) is 0.9997294 and the trapezoidal estimate (Fig. 1) is 0.9952975. Increasing the discretisation from 7 points to 100, the estimates improve to 0.9975607 and 0.9972921, respectively. Recall that the default discretisation in `base::density()` is $N = 512$.

The **ecospat** package does not estimate integrals with polygons, but rather treats the PDF as a probability mass function (Di Cola et al. 2017; Broennimann, Di Cola, and Guisan 2020). (To review, a PDF describes probability along a continuous axis, whereas a PMF describes probability at discrete intervals. For a PDF, the probability of an event occurring in a given interval is the integral of the PDF over that interval, and the probability any single point is zero. For a PMF, probability is defined only at individual points, and the sum of the function across all categories/discrete points is unity.) The `ecospat.grid.clim.dyn()` function

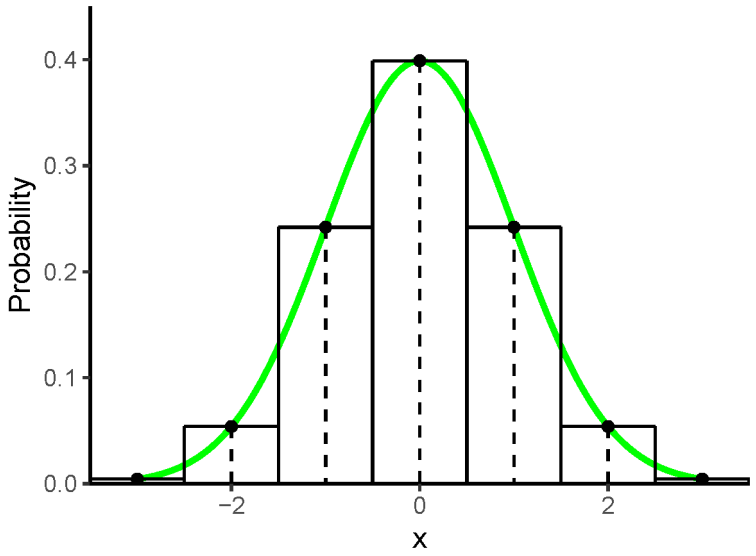

Figure 1: The green curve traces a normal distribution with $\mu = 0$ and $\sigma = 1$, over the interval $[-3, 3]$. The PDF is estimated at 7 equidistant points (dashed lines) for a rectangular approximation of the integral over the interval. Each rectangle is centred on a point estimate. In the **GoFKernel** implementation, the estimate includes both endpoints of the KDE, which in this example extends the integration to the interval $[-3.5, 3.5]$.

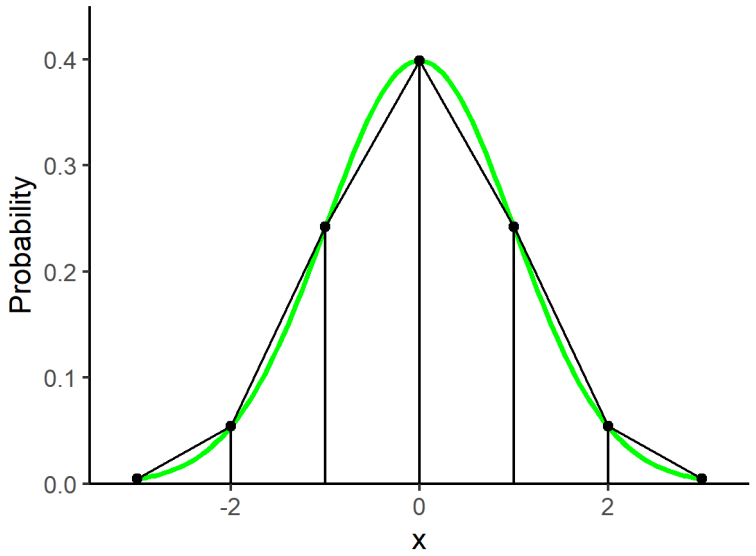

Figure 2: The green curve is as in Figure 1. The PDF is estimated at 7 equidistant points (vertical lines) for a trapezoidal approximation of the integral over the interval $[-3, 3]$.

estimates the density of a species' niche (`sp.dens$y`) at 100 discrete points, and re-scales the estimate at those points by the sum of all discrete density estimates. The result is scaled again by the number of observations for the species (`nrow(sp)`) and then by the maximum estimate.

```
# z <- sp.dens$y * nrow(sp)/sum(sp.dens$y)
# z.uncor <- z/max(z)
```

Boundary reflection

For some density estimation problems it may be desirable to reflect data across the estimation bounds so that the density does not drop towards zero as it approaches those bounds. In particular, if the range of sampled values is much narrower than the suspected true breadth of the distribution, then boundary reflection could improve KDE accuracy greatly. In Figure 3, observations are taken from only the middle region of a broad distribution; boundary reflection (blue) leads to an estimate much closer to the true shape of the PDF (black) than an unreflected kernel estimate (red). Note that the mode/peak of the KDE is similar regardless of reflection - the main differences are in the tails of the estimated distributions. **kerneval** allows boundary reflection with the argument `reflect=TRUE` and specified bounds `a` and `b`, which the `wdens()` and `transdens()` functions will pass on internally to `GoFKernel::density.reflected()`.

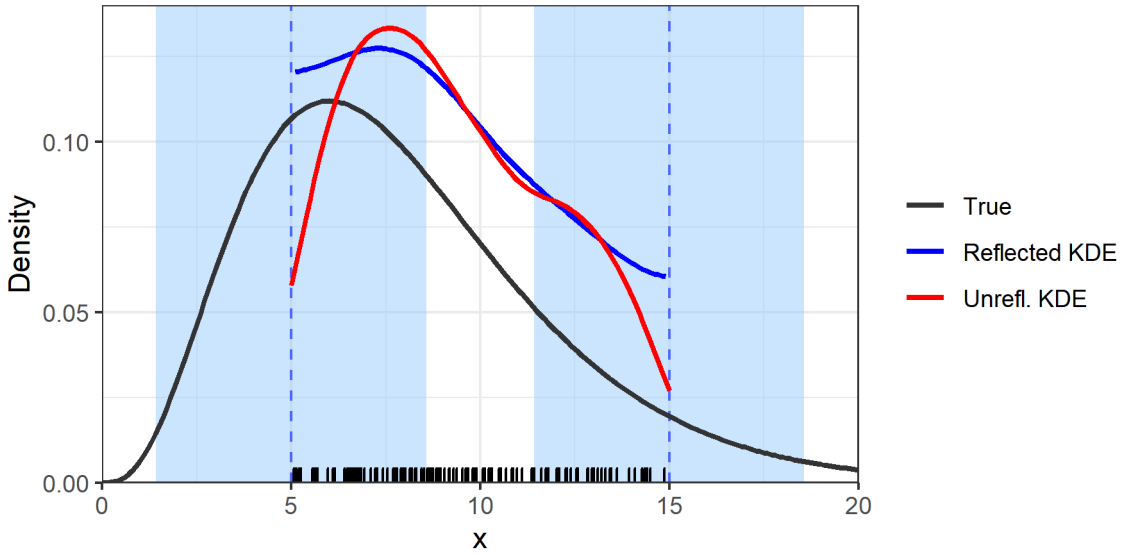

Figure 3: The true PDF is a chi-square distribution with 8 degrees of freedom (black line), from which random sample of 200 observations were drawn. Imagine a sampling bias that allows observations only on the interval $[5, 15]$; in this case, 145 observations remain (plotted as a rug at bottom) from which to estimate the density. The red curve plots the basic KDE `density(X, from = 5, to = 15)`. The blue curve is a KDE based on reflected data: all ovservations within the shaded blue region (4 times as wide as the bandwidth) were reflected across the boundaries (dashed lines).

Mean integrated squared error

The most common way to define the fitting error of an estimated density function is mean squared error, MSE. MSE is defined at a single point, while the error accumulated over the entire study interval is the mean integrated squared error, MISE. Most bandwidth selection methods attempt to estimate and then minimise the MISE as a function of the bandwidth, h. MISE can be written as the sum of systematic error (bias) and variance, and so the chosen bandwidth is a trade-off between these two terms. A large h will reduce variance but increase bias, and a small h will increase variance but reduce bias. This is a more

formal description of the intuitive idea that an overly-narrow bandwidth will undersmooth the data, and an overly-broad bandwidth will oversmooth. The following section is based on notes from Silverman (1986) and summarises the mathematical explanation of the variance-bandwidth trade-off and the criteria for an optimal bandwidth.

The sum of the squared bias and variance at point x is:

$$MSE_x(\hat{f}) = E\{\hat{f}(x) - f(x)\}^2$$

$$= \{E\hat{f}(x) - f(x)\}^2 + var\hat{f}(x)$$

For a global estimate, the mean integrated squared error is defined as

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

$$= \int MSE_x(\hat{f})dx$$

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

Exploring the bias term individually,

$$bias_h(x) = E\hat{f}(x) - f(x) = \int \frac{1}{h}K(\frac{x-y}{h})f(y)dy - f(x)$$

With some assumptions about the kernel function K,

$$bias_h(x) = \frac{1}{2}h^2 f''(x)k_2 + \text{ higher-order terms in } h$$

Therefore,

$$\int bias_h(x)^2 dx \approx \frac{1}{4}h^4 k_2 \int f''(x)^2 dx$$

where k_2 is a constant, for instance the variance of a Gaussian distribution K.

After some mathematical acrobatics, one can also derive an approximation of the variance term:

$$var\hat{f}(x) \approx \frac{1}{nh}f(x)\int K(t)^2 dt$$

$$\int var\hat{f}(x)dx \approx \frac{1}{nh}\int K(t)^2 dt$$

Combining the above expressions for MISE, bias, and variance, we have:

$$MISE \approx \frac{1}{4}h^4 k_2 \int f''(x)^2 dx + \frac{1}{nh}\int K(t)^2 dt$$

Note that the variance term depends on sample size n, while the bias term depends on the second derivative of the PDF. The bandwidth h affects both the variance and the bias. The value of h that minimises MISE is:

$$h_{opt} = k_2^{-2/5}\left\{\int K(t)^2 dt\right\}^{1/5}\left\{\int f''(x)^2 dx\right\}^{-1/5} n^{-1/5}$$

From inspection of this last result, we can tell that, unfortunately, h_{opt} depends on the (unknown) true density distribution f. Also, $h_{opt} \to 0$ as $n \to \infty$, but slowly. Lastly, smaller values of h are better for rougher (more 'wiggly') f. Roughness is indicated by the expression $\int f''^2$.

References

Broennimann, Olivier, Valeria Di Cola, and Antoine Guisan. 2020. *Ecospat: Spatial Ecology Miscellaneous Methods*. https://CRAN.R-project.org/package=ecospat.

Di Cola, Valeria, Olivier Broennimann, Blaise Petitpierre, Frank T Breiner, Manuela d'Amen, Christophe Randin, Robin Engler, Julien Pottier, Dorothea Pio, and Anne Dubuis. 2017. "Ecospat: An r Package to Support Spatial Analyses and Modeling of Species Niches and Distributions." Journal Article. *Ecography* 40 (6): 774–87.

Pavia, Jose M. 2015. *Testing Goodness-of-Fit with the Kernel Density Estimator: GoFKernel. Journal of Statistical Software, Code Snippets*. Vol. 66. http://www.jstatsoft.org/v66/c01/.

Silverman, Bernard W. 1986. *Density Estimation for Statistics and Data Analysis*. Book. Monographs on Statistics and Applied Probability. Bristol: Chapman; Hall Ltd.