ks: Kernel density estimation for bivariate data

Tarn Duong
Department of Statistics, University of New South Wales
Sydney Australia

4 October 2007

Kernel density estimation is a popular tool for visualising the distribution of data. See Simonoff (1996), for example, for an overview. When multivariate kernel density estimation is considered it is usually in the constrained context with diagonal bandwidth matrices, e.g. in the `R` packages `sm` (Bowman and Azzalini, 2005) and `KernSmooth` (Wand, 2006). We introduce a new `R` package `ks` which implements diagonal and unconstrained data-driven bandwidth matrices for kernel density estimation, which can also be used for multivariate kernel discriminant analysis. The `ks` package implements selectors for 2- to 6-dimensional data.

This vignette contains only a brief introduction to using `ks` for kernel density estimation for 2-dimensional data. See Duong (2007) for a more detailed account.

For a bivariate random sample $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ drawn from a density f, the kernel density estimate is defined by

$$\hat{f}(\boldsymbol{x}; \mathbf{H}) = n^{-1} \sum_{i=1}^{n} K_{\mathbf{H}}(\boldsymbol{x} - \boldsymbol{X}_i)$$

where $\boldsymbol{x} = (x_1, x_2)^T$ and $\boldsymbol{X}_i = (X_{i1}, X_{i2})^T, i = 1, 2, \ldots, n$. Here $K(\boldsymbol{x})$ is the kernel which is a symmetric probability density function, $\mathbf{H}$ is the bandwidth matrix which is symmetric and positive-definite, and $K_{\mathbf{H}}(\boldsymbol{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\boldsymbol{x})$. The choice of K is not crucial: we take $K(\boldsymbol{x}) = (2\pi)^{-1} \exp(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x})$ the standard normal throughout. In contrast, the choice of $\mathbf{H}$ is crucial in determining the performance of $\hat{f}$. The most common parameterisations of the bandwidth matrix are the diagonal and the general or unconstrained which has no restrictions on $\mathbf{H}$ provided that $\mathbf{H}$ remains positive definite and symmetric, that is

$$\mathbf{H} = \begin{bmatrix} h_1^2 & 0 \\ 0 & h_2^2 \end{bmatrix} \text{ or } \mathbf{H} = \begin{bmatrix} h_1^2 & h_{12} \\ h_{12} & h_2^2 \end{bmatrix}.$$

This latter parameterisation allows kernels to have an arbitrary orientation whereas the former only allows kernels which are oriented to the co-ordinate axes.

For our target density, we use the 'dumbbell' density, given by the normal mixture

$$\frac{4}{11} N\left(\begin{bmatrix} -2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) + \frac{3}{11} N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.8 & -0.72 \\ -0.72 & 0.8 \end{bmatrix} \right) + \frac{4}{11} N\left(\begin{bmatrix} 2 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right),$$

displayed on the left in Figure 1. This density is unimodal. On the right is a sample of 200 data points.

```
> library(ks)
> set.seed(8192)
> samp <- 200
> mus <- rbind(c(-2, 2), c(0, 0), c(2, -2))
> Sigmas <- rbind(diag(2), matrix(c(0.8, -0.72, -0.72, 0.8), nrow = 2),
+     diag(2))
> cwt <- 3/11
> props <- c((1 - cwt)/2, cwt, (1 - cwt)/2)
> x <- rmvnorm.mixt(n = samp, mu = mus, Sigma = Sigmas, props = props)
```

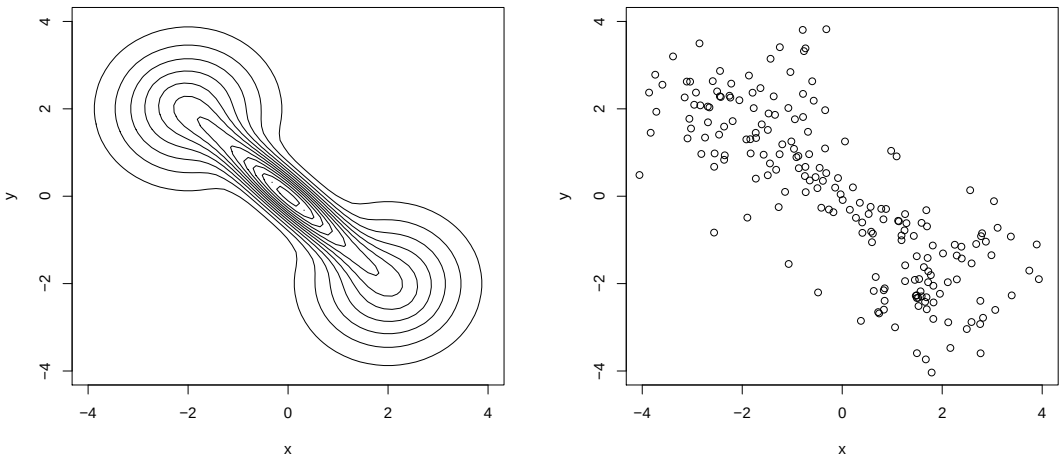

Figure 1: Target 'dumbbell' density. (Left) contour plot. (Right) Scatter plot.

We use `Hpi` for unconstrained plug-in selectors and `Hpi.diag` for diagonal plug-in selectors.

```
> Hpi1 <- Hpi(x = x)

           [,1]       [,2]
[1,]  0.4054542 -0.2778998
[2,] -0.2778998  0.3349442

> Hpi2 <- Hpi.diag(x = x)

          [,1]      [,2]
[1,] 0.1859013 0.0000000
[2,] 0.0000000 0.1312804
```

To compute a kernel density estimate, the command is `kde`, which creates a `kde` class object

```
> fhat.pi1 <- kde(x = x, H = Hpi1)
> fhat.pi2 <- kde(x = x, H = Hpi2)
```

We use the `plot` command to display these kernel density estimates. This generic `plot` command calls `plot.kde`. The default is a contour plot with the upper 25%, 50% and 75% contours of the (sample) highest density regions. These regions are also plotted by the `sm` library.

```
> plot(fhat.pi1)
> plot(fhat.pi2)
```

The respective kernel density estimates are produced in Figure 2. The diagonal bandwidth matrix constrains the smoothing to be performed in directions parallel to the co-ordinate axes, so it is not able to apply accurate levels of smoothing to the obliquely oriented central portion. The result is a multimodal density estimate. The unconstrained bandwidth matrix correctly produces a unimodal density estimate.

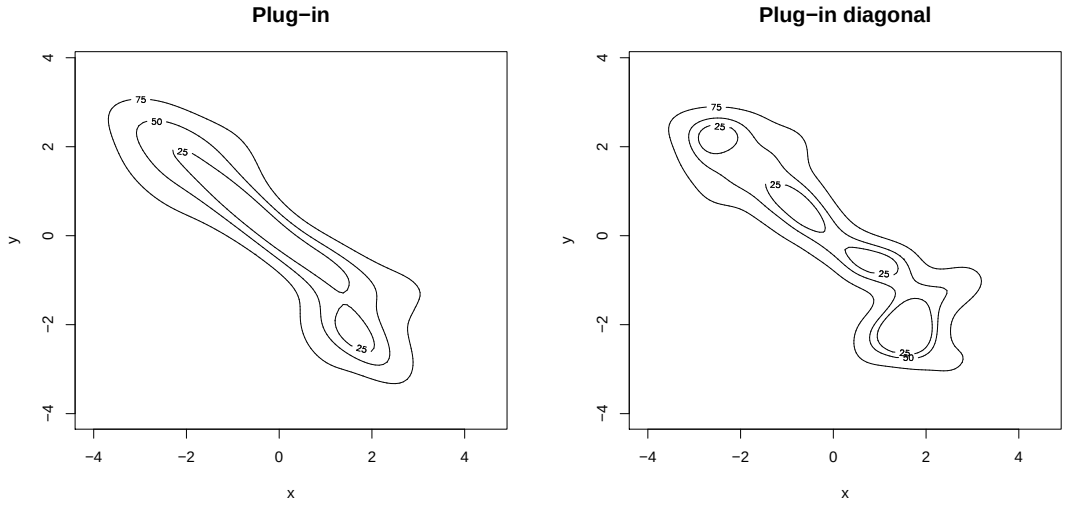

Figure 2: Kernel density estimates with plug-in selectors

The unconstrained SCV (Smoothed Cross Validation) selector is `Hscv` and its diagonal version is `Hscv.diag`. In Figure 3, the most reasonable density estimate is from the unconstrained SCV selector.

```
> Hscv1 <- Hscv(x = x)

           [,1]       [,2]
[1,]  0.5647800 -0.4044703
[2,] -0.4044703  0.4934641

> Hscv2 <- Hscv.diag(x = x)

         [,1]      [,2]
[1,] 0.284455 0.0000000
[2,] 0.000000 0.2460504
```

So far the calls to `kde` compute $\hat{f}$ exactly. This exact computation is $O(n^2)$ complexity which becomes infeasible for large sample sizes, say $n = 10\ 000$ on a current desktop PC. One

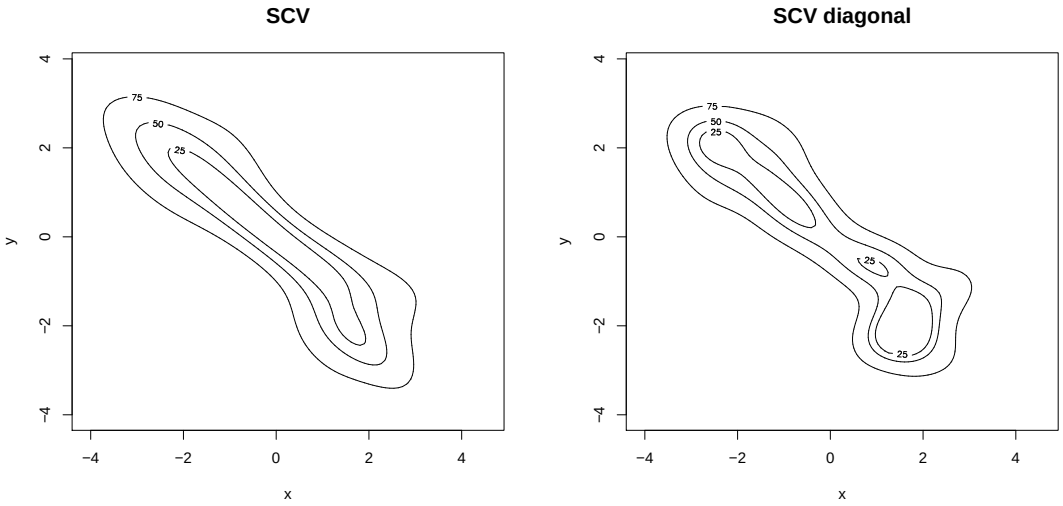

Figure 3: Kernel density estimates with cross validation selectors

common technique for increasing computational speed for these large samples is binned kernel estimation, as implemented in `KernSmooth` (Wand, 2006). Binning converts the data sample of size n to a grid of size m, so binned estimation remains $O(m)$ regardless of the sample size. A suitable binning grid size for bivariate data is $m = 151^2$.

Binned estimation is only defined with diagonal bandwidth matrices. Applicable cases include kernel density estimators with diagonal bandwidth matrices and the pilot estimators for the plug-in and SCV selectors. In the `Hpi`, `Hpi.diag`, `Hscv`, `Hscv.diag` and `kde` commands, we set `binned=TRUE`, e.g.

```
> x <- rmvnorm.mixt(10000, mus, Sigmas, props)
> H <- Hpi(x = x, binned = TRUE)
> Hdiag <- Hscv.diag(x = x, binned = TRUE)
> fhatdiag <- kde(x = x, H = Hdiag, binned = TRUE)
```

The different bandwidth selectors available in `ks` may now pose a problem of too much choice. The unconstrained bandwidth selectors will be better than their diagonal counterparts when the data have large mass oriented obliquely to the co-ordinate axes, like for the dumbbell data. The unconstrained plug-in and the SCV selectors can be viewed as generally recommended selectors.

References

Bowman, A. W. and Azzalini, A. (2005). *sm: kernel smoothing methods: Bowman and Azzalini (1997)*. R package version 2.0-14. Ported to R by B. D. Ripley.

Duong, T. (2007). ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R. Submitted.

Simonoff, J. S. (1996). *Smoothing Methods in Statistics*. Springer-Verlag, New York.

Wand, M. P. (2006). *KernSmooth: Functions for kernel smoothing for Wand & Jones (1995).* R package version 2.22-19. R port by B. D. Ripley.