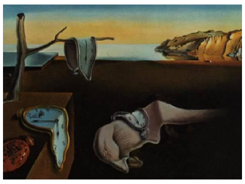

Journal of Statistical Software

October 2007, Volume 21, Issue 7. http://www.jstatsoft.org/

ks: Kernel Density Estimation and Kernel Discriminant Analysis for Multivariate Data in R

Tarn Duong
Institut Pasteur

Abstract

Kernel smoothing is one of the most widely used non-parametric data smoothing techniques. We introduce a new R package **ks** for multivariate kernel smoothing. Currently it contains functionality for kernel density estimation and kernel discriminant analysis. It is a comprehensive package for bandwidth matrix selection, implementing a wide range of data-driven diagonal and unconstrained bandwidth selectors.

Keywords: bandwidth selection, data-driven, non-parametric smoothing.

1. Introduction

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 packages **sm** (Bowman and Azzalini 2007) and **KernSmooth** (Wand and Ripley 2006) for R (R Development Core Team 2007). We introduce a new R package **ks**—available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/`—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.

In Section 2, we supply a brief review of kernel density estimation and indicate some reasons why unconstrained matrix selectors would be a useful generalisation over diagonal selectors. The optimal bandwidth selection problem for these unconstrained matrices is considered in Section 3. Also in this section are examples from **ks** for density estimation. We demonstrate the package's functionality for discriminant analysis in Section 4. With these unconstrained selectors now available, we conclude by offering some general recommendations.

2. Kernel density estimation

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

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

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top$ and $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{id})^\top, i = 1, 2, \ldots, n$. 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 $K(\boldsymbol{x}) = (2\pi)^{-d/2} \exp(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{x})$, the standard normal throughout. In contrast, the choice of $\mathbf{H}$ is crucial in determining the performance of $\hat{f}$.

According to Wand and Jones (1993), the most useful parameterisations of the bandwidth matrix are the diagonal $\mathbf{H} = \mathrm{diag}(h_1^2, h_2^2, \ldots, h_d^2)$ and the general or unconstrained which has no restrictions on $\mathbf{H}$, provided that $\mathbf{H}$ remains positive definite and symmetric.

To compare these two parameterisations, we examine the 'dumbbell' density, given by the normal mixture

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

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

From this data sample, we compute a diagonal and an unconstrained bandwidth matrix:

$$\begin{bmatrix} 0.4477 & 0 \\ 0 & 0.5612 \end{bmatrix} \text{ and } \begin{bmatrix} 0.5648 & -0.4045 \\ -0.4045 & 0.4935 \end{bmatrix}.$$

From these matrices, all the diagonal elements approximately lie in the range 0.45 to 0.56, so the smoothing in the co-ordinate directions are comparable. The main difference is that

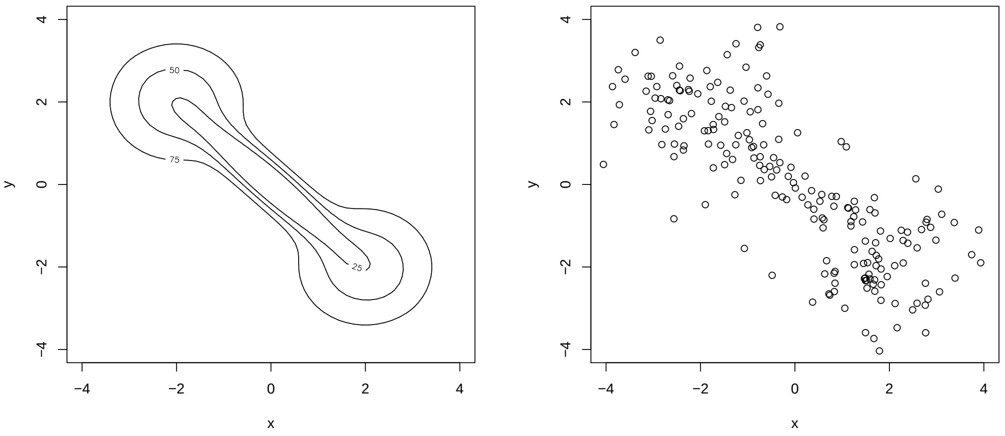

Figure 1: Target 'dumbbell' density: contour plot (left), scatter plot (right).

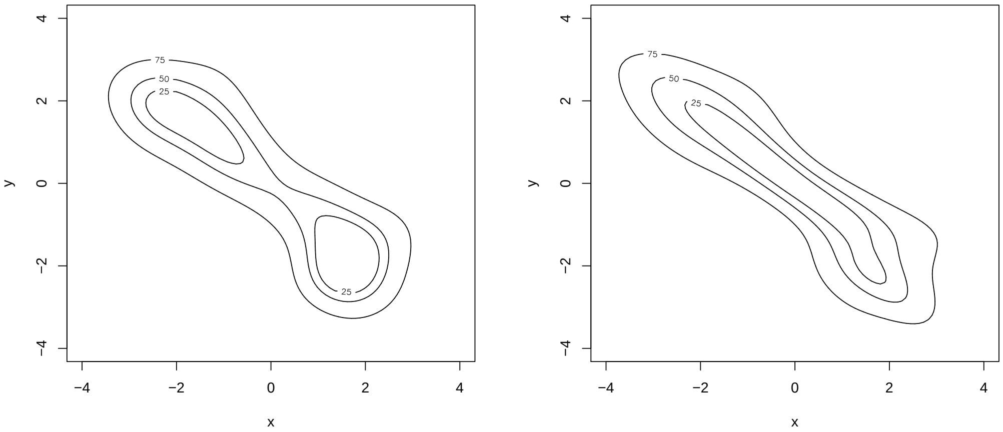

Figure 2: Kernel density estimates for dumbbell data: diagonal bandwidth matrix (left), unconstrained bandwidth matrix (right).

the off-diagonal elements in the latter matrix are almost the same size in magnitude as the diagonal elements, and so induce substantial smoothing in an oblique direction.

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 bimodal density estimate. The unconstrained bandwidth matrix correctly produces a unimodal density estimate.

Wand and Jones (1993) and Duong and Hazelton (2003, 2005b) contain more extensive simulation studies on a range of target densities concerning the relative gains in efficacy when using an unconstrained parametrisation as compared to a diagonal one. The general conclusion from these papers is that an unconstrained bandwidth matrix is most useful when there is large probability mass oriented away from the co-ordinate directions, such as the dumbbell density examined here.

Closely related to the bandwidth parameterisation is the pre-transformation of the data. Instead of basing our bandwidth selection on the original data $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$, we could use the transformed data $\boldsymbol{X}_1^*, \boldsymbol{X}_2^*, \ldots, \boldsymbol{X}_n^*$, where the transformation is either *sphering*

$$\boldsymbol{X}^* = \mathbf{S}^{-1/2}\boldsymbol{X}$$

where $\mathbf{S}$ is the sample covariance matrix of the untransformed data; or *scaling*

$$\boldsymbol{X}^* = \mathbf{S}_D^{-1/2}\boldsymbol{X}$$

where $\mathbf{S}_D = \text{diag}(s_1^2, s_2^2, \ldots, s_d^2)$ and $s_1^2, s_2^2, \ldots, s_d^2$ are the marginal sample variances. The bandwidth matrix $\mathbf{H}^*$ suitable for the sphered or scaled data can be back transformed to the original scale by $\mathbf{H} = \mathbf{S}^{1/2}\mathbf{H}^*\mathbf{S}^{1/2}$ or $\mathbf{H} = \mathbf{S}_D^{1/2}\mathbf{H}^*\mathbf{S}_D^{1/2}$, as appropriate.

The transformed data are more aligned to the co-ordinate axes which may lead us to believe that more restricted bandwidth parametrisations may be suitable for these transformed data.

Sphering or scaling with the most restricted parameterisation $\mathbf{H}^* = h^{*2}\mathbf{I}$, where $\mathbf{I}$ is the identity matrix, gives $\mathbf{H} = h^{*2}\mathbf{S}$ or $\mathbf{H} = h^{*2}\mathbf{S}_D$. However Wand and Jones (1993) strongly advise against these combinations of pre-transformation and bandwidth parametrisation.

In light of this, we need to use at least the diagonal bandwidth matrix with these transformations. Let $\mathbf{H}^* = \operatorname{diag}(h_1^{*2}, h_2^{*2}, \ldots, h_d^{*2})$. The bandwidth matrix obtained with pre-scaled data is $\mathbf{H} = \operatorname{diag}(s_1^2 h_1^{*2}, s_2^2 h_2^{*2}, \ldots, s_d^2 h_d^{*2})$ which is still a diagonal matrix, so it is not clear that this offers any advantage over a diagonal bandwidth matrix without pre-scaling. On the other hand, with pre-sphering, the resulting $\mathbf{H} = \mathbf{S}^{1/2}\mathbf{H}^*\mathbf{S}^{1/2}$ is non-diagonal. However, the off-diagonal elements are still constrained since they rely on the sample variance, whereas those in the unconstrained parametrisation do not have this restriction. So the latter will be better in situations where the sample variance is not an appropriate summary of the dispersion of the data. With this in mind, we focus on bandwidth selectors for the maximally general unconstrained parametrisation.

3. Optimal bandwidth selectors

We briefly present the main ideas behind optimal, data-based bandwidth selectors. The reader can consult Duong and Hazelton (2003, 2005b) and references therein for more details on the multivariate bandwidth selection problem.

We measure the performance of $\hat{f}$ (in common with the majority of researchers in this field) using the Mean Integrated Squared Error (MISE) criterion,

$$\operatorname{MISE}(\mathbf{H}) = \mathsf{E} \int_{\mathbb{R}^d} [\hat{f}(\boldsymbol{x}; \mathbf{H}) - f(\boldsymbol{x})]^2 \, d\boldsymbol{x}.$$

Our aim in bandwidth selection is to estimate

$$\mathbf{H}_{\text{MISE}} = \underset{\mathbf{H}}{\operatorname{argmin}} \operatorname{MISE}(\mathbf{H}),$$

over the space of all symmetric, positive definite $d \times d$ matrices. It is well known that the optimal bandwidth $\mathbf{H}_{\text{MISE}}$ does not have a closed form. To make progress it is usual to employ an asymptotic approximation, known as the AMISE (Asymptotic MISE)

$$\operatorname{AMISE}(\mathbf{H}) = n^{-1}(4\pi)^{-d/2}|\mathbf{H}|^{-1/2} + \frac{1}{4}(\operatorname{vech}^\top \mathbf{H})\boldsymbol{\Psi}_4(\operatorname{vech} \mathbf{H}) \tag{2}$$

where vech is the vector half operator e.g.

$$\operatorname{vech} \mathbf{H} = \operatorname{vech} \begin{bmatrix} h_1^2 & h_{12} \\ h_{12} & h_2^2 \end{bmatrix} = \begin{bmatrix} h_1^2 \\ h_{12} \\ h_2^2 \end{bmatrix}.$$

The subscript 4 on $\boldsymbol{\Psi}$ relates to the order of the derivatives involved. See Wand and Jones (1995, p. 98) for the general expression of the $\frac{1}{2}d(d+1) \times \frac{1}{2}d(d+1)$ matrix $\boldsymbol{\Psi}_4$. For the purposes of this paper, this expression is not required. All that we need is that the elements of $\boldsymbol{\Psi}_4$ are integrated density derivative functionals

$$\psi_{\boldsymbol{r}} = \int_{\mathbb{R}^d} f^{(\boldsymbol{r})}(\boldsymbol{x}) f(\boldsymbol{x}) \, d\boldsymbol{x}$$

where $\boldsymbol{r} = (r_1, r_2, \ldots, r_d)$, $|\boldsymbol{r}| = \sum_{i=1}^d r_i$, and the partial derivatives of f are

$$f^{(\boldsymbol{r})}(\boldsymbol{x}) = \frac{\partial^{|\boldsymbol{r}|}}{\partial_{x_1}^{r_1} \ldots \partial_{x_d}^{r_d}} f(\boldsymbol{x}).$$

We make use of the tractability of AMISE by seeking

$$\mathbf{H}_{\mathrm{AMISE}} = \underset{\mathbf{H}}{\operatorname{argmin}} \ \mathrm{AMISE}\,(\mathbf{H}).$$

For the next step we estimate the MISE or AMISE. A data-driven bandwidth selector is either

$$\hat{\mathbf{H}} = \underset{\mathbf{H}}{\operatorname{argmin}} \ \widehat{\mathrm{MISE}}\,(\mathbf{H}) \quad \text{or} \quad \hat{\mathbf{H}} = \underset{\mathbf{H}}{\operatorname{argmin}} \ \widehat{\mathrm{AMISE}}\,(\mathbf{H}). \tag{3}$$

Different selectors arise from the different methods used in the estimation step.

3.1. Plug-in bandwidth selectors

The most well-known univariate plug-in selector is due to Sheather and Jones (1991). One multivariate extension of this is developed in Wand and Jones (1994). The plug-in estimate of the AMISE,

$$\mathrm{PI}(\mathbf{H}) = n^{-1}(4\pi)^{-d/2}|\mathbf{H}|^{-1/2} + \frac{1}{4}(\operatorname{vech}^\top \mathbf{H})\hat{\boldsymbol{\Psi}}_4(\operatorname{vech} \mathbf{H}), \tag{4}$$

can be numerically minimised to give the plug-in bandwidth matrix, $\hat{\mathbf{H}}_{\mathrm{PI}}$. If we note that $\psi_{\boldsymbol{r}} = \mathsf{E} f^{(\boldsymbol{r})}(\boldsymbol{X})$ where $\boldsymbol{X} \sim f$, then a natural estimator of $\psi_{\boldsymbol{r}}$ is

$$\hat{\psi}_{\boldsymbol{r}}(\mathbf{G}) = n^{-1} \sum_{i=1}^n \hat{f}^{(\boldsymbol{r})}(\boldsymbol{X}_i; \mathbf{G}) = n^{-2} \sum_{i=1}^n \sum_{j=1}^n K_{\mathbf{G}}^{(\boldsymbol{r})}(\boldsymbol{X}_i - \boldsymbol{X}_j), \tag{5}$$

where $\mathbf{G}$ is a pilot bandwidth matrix. These $\hat{\psi}_{\boldsymbol{r}}(\mathbf{G})$ are the elements of $\hat{\boldsymbol{\Psi}}_4$.

Like $\mathbf{H}$, we need to choose a sensible value for $\mathbf{G}$. We consider pilot bandwidth matrices of the form $\mathbf{G} = g^2\mathbf{I}$ along with the pre-transformations of Section 2. This combination of bandwidth parameterisation and pre-transformation is ill-advised for selecting the final bandwidth $\mathbf{H}$ but it is acceptable for selecting a pilot bandwidth $\mathbf{G}$. This is because it is not crucial to select $\mathbf{G}$ with the same accuracy as for $\mathbf{H}$. With this restricted parameterisation $g^2\mathbf{I}$, we are able to derive analytical expressions for optimal pilot selectors, allowing us to avoid computationally intensive numerical optimisation for pilot bandwidth selection.

The MSE (Mean Squared Error) for $\hat{\psi}_{\boldsymbol{r}}(g)$ is

$$\mathrm{MSE}\,(g) = \mathsf{E}[\hat{\psi}_{\boldsymbol{r}}(g) - \psi_{\boldsymbol{r}}]^2.$$

Wand and Jones (1994) select g to minimise the Asymptotic MSE (AMSE). However, attempting to minimise this criterion may lead to numerical instabilities, as documented in Duong and Hazelton (2003). These authors propose the Sum of AMSE (SAMSE) criterion which has better numerical and theoretical properties

$$\mathrm{SAMSE}\,(g) = \sum_{\boldsymbol{r}:|\boldsymbol{r}|=4} \mathrm{AMSE}\,(g).$$

Duong and Hazelton (2003) contains the explicit formula for the selector which minimises this SAMSE.

3.2. Cross validation bandwidth selectors

Cross-validation selectors are the main alternative to plug-in selectors. There are three main flavours of cross-validation selectors: least squares, biased and smoothed.

Least squares cross validation

The multivariate version of the least squares cross validation (LSCV) criterion of Rudemo (1982) and Bowman (1984) is

$$\mathrm{LSCV}(\mathbf{H}) = \int_{\mathbb{R}^d} \hat{f}(\boldsymbol{x};\mathbf{H})^2 \, d\boldsymbol{x} - 2n^{-1}\sum_{i=1}^{n} \hat{f}_{-i}(\boldsymbol{X}_i;\mathbf{H}),$$

where the leave-one-out estimator is

$$\hat{f}_{-i}(\boldsymbol{x};\mathbf{H}) = (n-1)^{-1}\sum_{\substack{j=1\\ j\neq i}}^{n} K_{\mathbf{H}}(\boldsymbol{x}-\boldsymbol{X}_j).$$

The LSCV selector $\hat{\mathbf{H}}_{\mathrm{LSCV}}$ is the minimiser of $\mathrm{LSCV}(\mathbf{H})$. We can rewrite LSCV as

$$\mathrm{LSCV}(\mathbf{H}) = n^{-1}(4\pi)^{-d/2}|\mathbf{H}|^{-1/2} + n^{-1}(n-1)^{-1}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}(K_{2\mathbf{H}} - 2K_{\mathbf{H}})(\boldsymbol{X}_i - \boldsymbol{X}_j). \tag{6}$$

We can show that $\mathsf{E}[\mathrm{LSCV}(\mathbf{H})] = \mathrm{MISE}(\mathbf{H}) - \int_{\mathbb{R}^d} f(\boldsymbol{x})^2 \, d\boldsymbol{x}$, indicating that LSCV estimates the MISE directly.

Biased cross validation

Plug-in methods use a pilot bandwidth matrix $\mathbf{G}$, which is independent of $\mathbf{H}$, to estimate $\boldsymbol{\Psi}_4$. For BCV, we set $\mathbf{G} = \mathbf{H}$ and use slightly different estimators. There are two versions of BCV, depending on the estimator of $\psi_{\boldsymbol{r}}$, see Sain, Baggerly, and Scott (1994). We can use

$$\check{\psi}_{\boldsymbol{r}}(\mathbf{H}) = n^{-2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} K_{2\mathbf{H}}^{(\boldsymbol{r})}(\boldsymbol{X}_i - \boldsymbol{X}_j)$$

or we could use

$$\tilde{\psi}_{\boldsymbol{r}}(\mathbf{H}) = n^{-1}\sum_{i=1}^{n}\hat{f}_{-i}^{(\boldsymbol{r})}(\boldsymbol{X}_i;\mathbf{H}) = n^{-1}(n-1)^{-1}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} K_{\mathbf{H}}^{(\boldsymbol{r})}(\boldsymbol{X}_i - \boldsymbol{X}_j).$$

The estimates $\check{\boldsymbol{\Psi}}_4$ and $\tilde{\boldsymbol{\Psi}}_4$ are obtained from $\boldsymbol{\Psi}_4$ by substituting $\check{\psi}_{\boldsymbol{r}}$ and $\tilde{\psi}_{\boldsymbol{r}}$ for $\psi_{\boldsymbol{r}}$. From this we obtain respectively

$$\begin{aligned}
\mathrm{BCV1}(\mathbf{H}) &= n^{-1}(4\pi)^{-d/2}|\mathbf{H}|^{-1/2} + \frac{1}{4}\mu_2(K)^2(\mathrm{vech}^\top \mathbf{H})\check{\boldsymbol{\Psi}}_4(\mathrm{vech}\,\mathbf{H})\\
\mathrm{BCV2}(\mathbf{H}) &= n^{-1}(4\pi)^{-d/2}|\mathbf{H}|^{-1/2} + \frac{1}{4}\mu_2(K)^2(\mathrm{vech}^\top \mathbf{H})\tilde{\boldsymbol{\Psi}}_4(\mathrm{vech}\,\mathbf{H}).
\end{aligned} \tag{7}$$

The BCV selectors $\hat{\mathbf{H}}_{\mathrm{BCV1}}$ and $\hat{\mathbf{H}}_{\mathrm{BCV2}}$ are the minimisers of the appropriate BCV function.

Smoothed cross validation

Smoothed cross validation (SCV) was introduced by Hall, Marron, and Park (1992). The SCV can be motivated by starting with a slightly modified version of LSCV in Equation (6), known as the leave-in-diagonals version, for data samples which have no repeated values

$$\mathrm{LSCV}(\mathbf{H}) = n^{-1}(4\pi)^{-d/2}|\mathbf{H}|^{-1/2} + n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n}(K_{2\mathbf{H}} - 2K_{\mathbf{H}} + K_{\mathbf{0}})(\boldsymbol{X}_i - \boldsymbol{X}_j)$$

where $K_{\mathbf{0}}$ is the Dirac delta function. To form SCV, we pre-smooth the data differences $\boldsymbol{X}_i - \boldsymbol{X}_j$ by $K_{2\mathbf{G}}$, i.e. replace $\boldsymbol{X}_i - \boldsymbol{X}_j$ by the convolution with $K_{2\mathbf{G}}(\boldsymbol{X}_i - \boldsymbol{X}_j)$:

$$\mathrm{SCV}(\mathbf{H}) = n^{-1}(4\pi)^{-d/2}|\mathbf{H}|^{-1/2} + n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n}(K_{2\mathbf{H}+2\mathbf{G}} - 2K_{\mathbf{H}+2\mathbf{G}} + K_{2\mathbf{G}})(\boldsymbol{X}_i - \boldsymbol{X}_j). \quad (8)$$

The SCV selector $\hat{\mathbf{H}}_{\mathrm{SCV}}$ is the minimiser of $\mathrm{SCV}(\mathbf{H})$. Again, we consider pilot bandwidth matrices of the form $\mathbf{G} = g^2\mathbf{I}$ with pre-transformations. A suitable selector for g is the minimiser of

$$Q(g) \equiv \operatorname{tr}\mathrm{MSE}(g) = \sum_{i=1}^{d}\sum_{j=i}^{d}\mathsf{E}[\hat{h}_{\mathrm{SCV},ij} - h_{\mathrm{AMISE},ij}]^2,$$

where $\hat{h}_{\mathrm{SCV},ij}$ is the (i,j)-th element of $\hat{\mathbf{H}}_{\mathrm{SCV}}$ and $h_{\mathrm{AMISE},ij}$ is the (i,j)-th element of $\mathbf{H}_{\mathrm{AMISE}}$. Its closed form expression is given in Duong and Hazelton (2005b).

Duong and Hazelton (2005a) show that for a selector $\hat{\mathbf{H}}$,

$$\sum_{i=1}^{d}\sum_{j=1}^{d}\mathsf{E}[\hat{h}_{ij} - h_{\mathrm{AMISE},ij}]^2 = O\left(\mathsf{E}\left[\frac{\partial}{\partial\operatorname{vech}\mathbf{H}}(\widehat{\mathrm{AMISE}} - \mathrm{AMISE})(\mathbf{H}_{\mathrm{AMISE}})\right]^2\right).$$

For the plug-in selector, they show that the right hand side is $O(\operatorname{tr}\mathsf{E}[\hat{\boldsymbol{\Psi}}_4(g) - \boldsymbol{\Psi}_4]^2)$, which is a weighted sum of the individual mean squared errors $\mathsf{E}[\hat{\psi}_{\boldsymbol{r}}(g) - \psi_{\boldsymbol{r}}]^2$, $|\boldsymbol{r}| = 4$. On the other hand, the SAMSE pilot selector minimises the unweighted sum of these terms, though we still have $\mathrm{SAMSE}(g) = O(\operatorname{tr}\mathsf{E}[\hat{\boldsymbol{\Psi}}_4(g) - \boldsymbol{\Psi}_4]^2)$. This establishes that the SAMSE pilot asymptotically minimises the distance between $\hat{\mathbf{H}}_{\mathrm{PI}}$ and $\mathbf{H}_{\mathrm{AMISE}}$. An alternative pilot selector based on the weighted MSEs may have better finite sample properties, but this is not pursued further here.

3.3. Code examples

Like in Section 2, we generate a sample of 200 points from the dumbbell density to compute bandwidth selectors and their corresponding density estimates.

```
R> library("ks")
R> samp <- 200
R> mus <- rbind(c(-2,2), c(0,0), c(2,-2))
R> Sigmas <- rbind(diag(2), matrix(c(0.8, -0.72, -0.72, 0.8), nrow = 2),
+  diag(2))
```

```
R> cwt <- 3/11
R> props <- c((1-cwt)/2, cwt, (1-cwt)/2)
R> x <- rmvnorm.mixt(n = samp, mu = mus, Sigma = Sigmas, props = props)
```

We use `Hpi` for unconstrained plug-in selectors and `Hpi.diag` for diagonal plug-in selectors. There are three other arguments which further specify the plug-in selector: `nstage` is the number of pilot estimation stages (1 or 2). We recommend using 2 stages of pilot estimation, see Wand and Jones (1995, pp. 73-74). The type of pilot estimation is set using `pilot` (`"amse"` or `"samse"`). The argument `pre` involves the pre-transformations (`"scale"` or `"sphere"`). We can use the pre-sphering or pre-scaling transformation with the unconstrained bandwidths. For the diagonal bandwidths, we should only use the pre-scaling, otherwise the back-transformation of pre-sphering results in a non-diagonal matrix.

```
R> Hpi1 <- Hpi(x = x, pilot = "amse", pre = "scale")
R> Hpi2 <- Hpi(x = x, pilot = "samse", pre = "scale")
R> Hpi3 <- Hpi(x = x, pilot = "amse", pre = "sphere")
R> Hpi4 <- Hpi(x = x, pilot = "samse", pre = "sphere")
R> Hpi5 <- Hpi.diag(x = x, pilot = "amse", pre = "scale")
R> Hpi6 <- Hpi.diag(x = x, pilot = "samse", pre = "scale")
```

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

```
R> kde(x = x, H = Hpi1)
```

etc. We use the `plot` method for `kde` objects to display these kernel density estimates. The default is a contour plot with the upper 25%, 50% and 75% contours. These contours are the boundaries of the sample highest density regions, as defined in Bowman and Foster (1993) and Hyndman (1996). These regions are useful for displaying and summarising multivariate densities. For a random variable $\boldsymbol{X} \sim f$, the highest density region at level α, for a kernel density estimator $\hat{f}(\cdot;\mathbf{H})$, is $R_\alpha = \{\boldsymbol{x} : \hat{f}(\boldsymbol{x};\mathbf{H}) \geq Y_\alpha\}$ where Y_α is the α-quantile of $Y = \hat{f}(\boldsymbol{X};\mathbf{H})$. Then

$$\Pr(\boldsymbol{X} \in R_\alpha) = \Pr(\hat{f}(\boldsymbol{X};\mathbf{H}) \geq Y_\alpha) = \Pr(Y \geq Y_\alpha) = 1 - \alpha.$$

Hyndman (1996) shows that R_α is the region with the smallest hypervolume which contains $1-\alpha$ of the total probability mass. The sample highest density region is obtained by replacing Y_α with the α-quantile of $Y_i = \hat{f}(\boldsymbol{X}_i;\mathbf{H})$. These regions are also plotted by the **sm** library.

Recalling that the true density is unimodal, of the density estimates displayed in Figure 3, only the density estimates with unconstrained bandwidth matrices and pre-sphering reproduce this unimodality.

The commands `Hlscv` and `Hlscv.diag` are the unconstrained and diagonal LSCV selectors. The command `Hbcv` implements both BCV1 and BCV2. The default is BCV1; set `whichbcv = 2` to call BCV2. Their diagonal counterpart is `Hbcv.diag`. The unconstrained SCV selector is `Hscv` and its diagonal version is `Hscv.diag`. The argument `pre` is the same as before.

```
R> Hlscv1 <- Hlscv(x = x)
R> Hlscv2 <- Hlscv.diag(x = x)
```

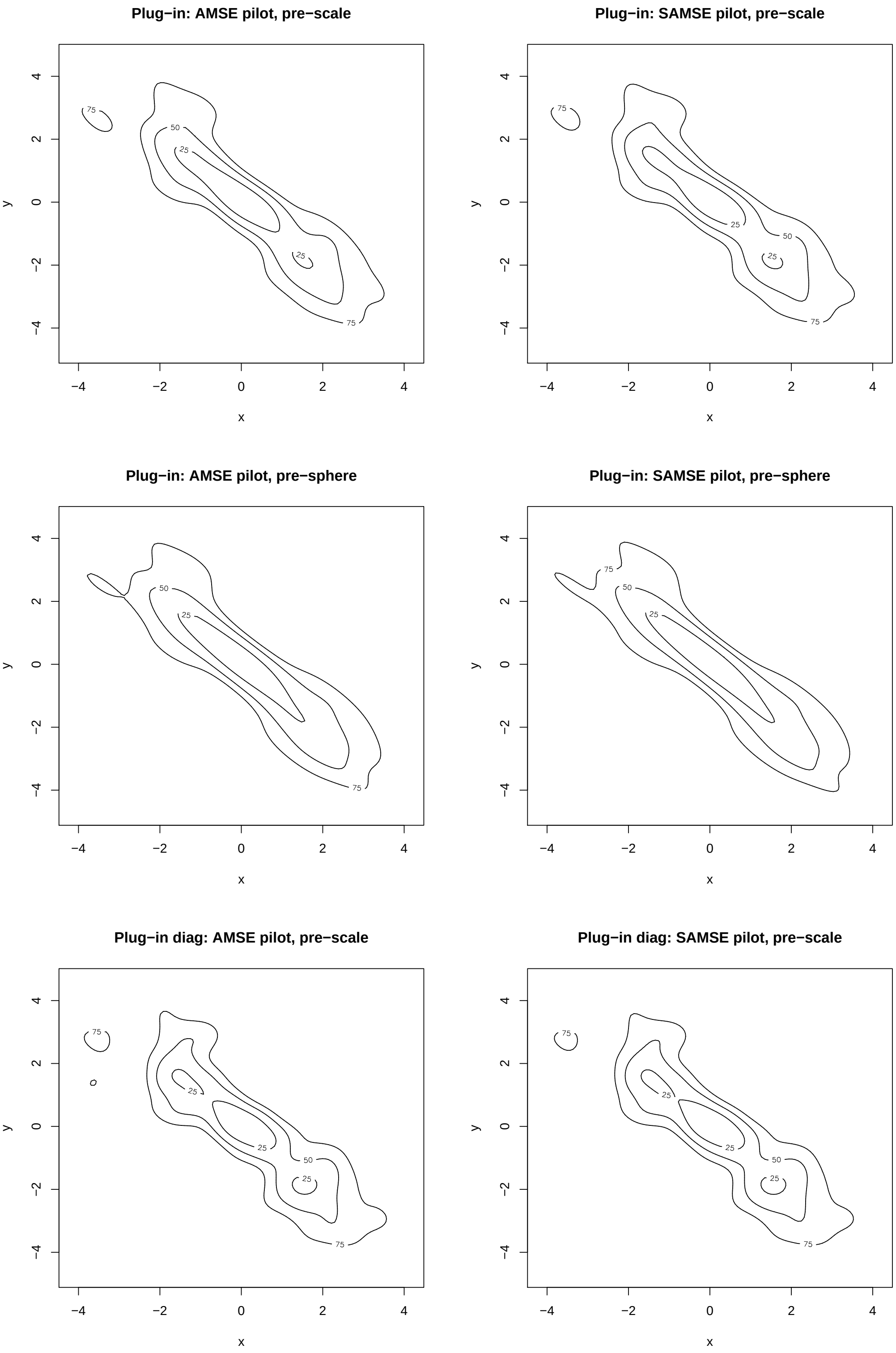

Figure 3: Kernel density estimates with plug-in selectors.

```
R> Hbcv1 <- Hbcv(x = x, whichbcv = 1)
R> Hbcv2 <- Hbcv.diag(x = x, whichbcv = 1)
R> Hscv1 <- Hscv(x = x, pre = "scale")
R> Hscv2 <- Hscv.diag(x = x, pre = "scale")
```

In Figure 4, we leave out the estimates with BCV2 since they are similar to BCV1.. Also omitted are the SCV selectors with pre-sphering, as they are similar to those with pre-scaling. The most reasonable density estimates are from the unconstrained SCV and LSCV selectors, though unlike the previous plug-in plots, the choice of pre-transformation has much less effect on the resulting density estimate.

More extensive simulation studies are reported in Duong and Hazelton (2003, 2005b). These confirm the general behaviour and performance of the bandwidth selectors demonstrated here for a wider range of target density shapes.

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 common technique for increasing computational speed for these large samples is binned kernel estimation, see Wand and Jones (1994, Appendix D), and is implemented in **KernSmooth** (Wand and Ripley 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. Suitable default binning grid sizes are $m = 401, 151^2, 51^3$ for $d = 1, 2, 3$. Computing density estimates on a grid becomes less feasible for $d > 3$.

Binned estimation is only defined with diagonal bandwidth matrices. Applicable cases include kernel density estimators with diagonal bandwidth matrices and the integrated density functional estimators with $g^2\mathbf{I}$ pilot bandwidth matrices for the plug-in and SCV selectors. In the `Hpi`, `Hpi.diag`, `Hscv`, `Hscv.diag` and `kde` commands, we set `binned = TRUE`, e.g.

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

4. Kernel discriminant analysis

We have ν populations, each associated with a density f_j and a prior probability π_j, $j = 1, 2, \ldots, \nu$. We wish to allocate a point $\boldsymbol{x}$ in the sample space to one (and only one) of these populations. A common allocation rule is the Bayes discriminant rule:

$$\text{Allocate } \boldsymbol{x} \text{ to group } j_0 \text{ where } j_0 = \underset{j \in \{1,2,\ldots,\nu\}}{\text{argmax}} \ \pi_j f_j(\boldsymbol{x}).$$

From each f_j, we have a random sample $\boldsymbol{X}_{j1}, \boldsymbol{X}_{j2}, \ldots, \boldsymbol{X}_{j,n_j}$. These $\boldsymbol{X}_{ji}$ are collectively known as the training data. In the cases where we wish only to classify a data sample, rather than the entire sample space, the random sample $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_m$ drawn from $\sum_{j=1}^{\nu} \pi_j f_j$ is known as the test data.

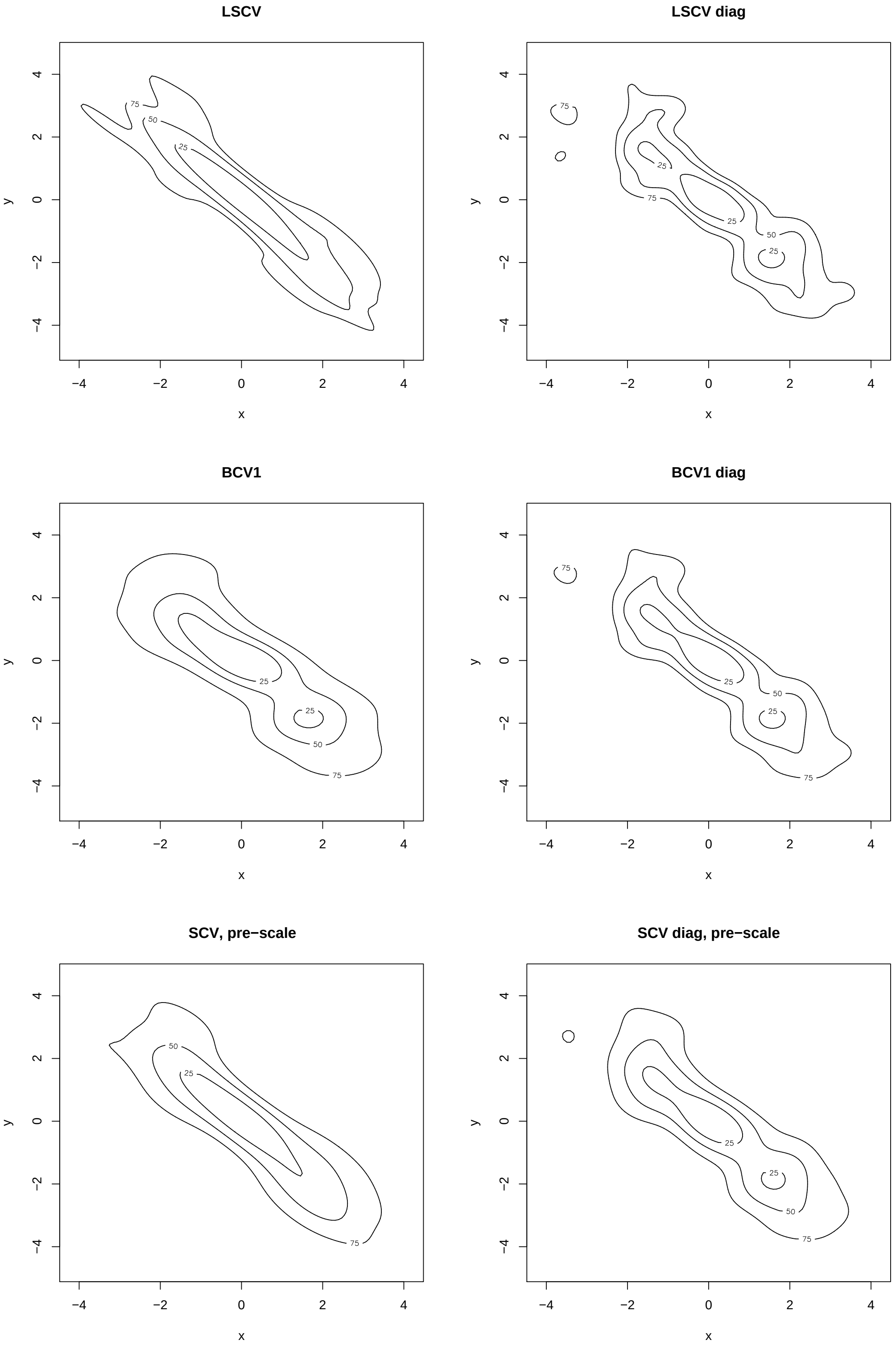

Figure 4: Kernel density estimates with cross validation selectors.

The kernel discriminant rule (KDR) is obtained from the Bayes discriminant rule by replacing f_j by its kernel density estimate

$$\hat{f}_j(\boldsymbol{x}; \mathbf{H}_j) = n_j^{-1} \sum_{i=1}^{n_j} K_{\mathbf{H}_j}(\boldsymbol{x} - \boldsymbol{X}_{ji}), \tag{9}$$

and π_j is usually replaced by the sample proportion n_j/n where $n = \sum_{j=1}^{\nu} n_j$; that is

$$\text{KDR : Allocate } \boldsymbol{x} \text{ to group } j_0 \text{ where } j_0 = \underset{j \in \{1,2,\dots,\nu\}}{\operatorname{argmax}} \ \hat{\pi}_j \hat{f}_j(\boldsymbol{x}; \mathbf{H}_j). \tag{10}$$

This now raises the question of the most appropriate bandwidth selectors for these density estimates. We take the simplest approach by using an optimal bandwidth selector for each of the training data sub-samples. These bandwidths are optimal for MISE for a single density function. An alternative is to select bandwidths which satisfy an error measure tailored to discriminant analysis, as is done by Hall and Kang (2005). Their data-driven bandwidth selectors are too computationally intensive for inclusion in **ks**. Fortunately they show that their discriminant analysis-optimal selectors are the same asymptotic order as density estimation-optimal selectors.

To illustrate, we use the well-known iris data which is part of the base software for R. There are 50 records each from 3 species of iris: *Iris setosa, I. versicolor* and *I. virginica.* We take the first three variables: sepal length, sepal width and petal length. Figure 5 is a scatter plot of these data.

```
R> library("MASS")
R> data("iris")
R> ir <- iris[,1:3]
R> ir.group <- iris[,5]
```

The commands for bandwidth selection in this case are similar to those for density estimation. The basic commands are `Hkda` and `Hkda.diag` for unconstrained and diagonal matrices. We

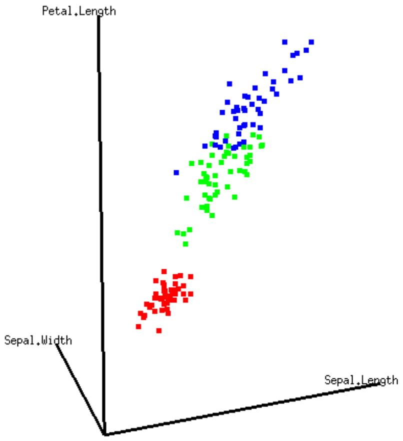

Figure 5: Scatter plot for iris data: *I. setosa* red, *I. versicolor* green and *I. virginica* blue.

set `bw = "plugin"`, `bw = "lscv"` or `bw = "scv"` for the plug-in, LSCV or SCV selectors. Due to the poor performance of BCV selectors for density estimation, they are not implemented for discriminant analysis. The other arguments to further specify the bandwidth selectors are the same as before.

```
R> Hpi1 <- Hkda(x = ir, x.group = ir.group, bw = "plugin",
+   pilot = "samse", pre = "sphere")
R> Hpi2 <- Hkda.diag(x = ir, x.group = ir.group, bw = "plugin",
+   pilot = "samse", pre = "scale")
R> Hscv1 <- Hkda(x = ir, x.group = ir.group, bw = "scv", pre = "sphere")
R> Hscv2 <- Hkda.diag(x = ir, x.group = ir.group, bw = "scv", pre = "scale")
```

To generate density estimates, the command is `kda.kde` which produces an object of class `kda.kde`, e.g.

```
R> kda.kde(x = ir, x.group = ir.group, Hs = Hpi1)
```

The `kda.kde` class has a `plot` method. For this example, it calls the **rgl** (Adler and Murdoch 2007) and **misc3d** (Feng and Tierney 2007) libraries to render the 3-dimensional display. In Figure 6, the density estimate for each group is plotted in a separate colour, using the same colours as in Figure 5. The default is to plot the contours of the upper 25% and 50% highest density regions, with the more opaque contour shell being 25% and the less opaque being 50%.

The computation of a misclassification rate (MR) for a kernel discriminant rule depends on whether the test data are independent of the training data. If the training data are independent, then

$$\widehat{\mathrm{MR}} = 1 - m^{-1} \sum_{j=1}^{m} \mathbf{1}\{\boldsymbol{Y}_j \text{ is correctly classified using KDR}\}$$

where KDR is the kernel discriminant rule in Equation (10). If the training and test data are the same, then it is appropriate to use a cross validated estimate of the misclassification rate:

$$\widehat{\mathrm{MR}}_{\mathrm{CV}} = 1 - n^{-1} \sum_{j=1}^{\nu} \sum_{i=1}^{n_j} \mathbf{1}\{\boldsymbol{X}_{ji} \text{ is correctly classified using KDR}_{-ji}\}$$

where KDR_{-ji} is a kernel discriminant rule in Equation (10) except that $\hat{\pi}_j$ and $\hat{f}_j(\boldsymbol{x}; \mathbf{H}_j)$ are replaced by $\hat{\pi}_{j,-i} = (n_j - 1)/(n-1)$ and

$$\hat{f}_{j,-i}(\boldsymbol{x}; \mathbf{H}_j) = (n_j - 1)^{-1} \sum_{\substack{i'=1 \\ i' \neq i}}^{n_j} (\boldsymbol{x} - \boldsymbol{X}_{ji'}).$$

For independent training data, the command is `compare`, which is not illustrated here. Since we have test data which is equal to the training data, we use

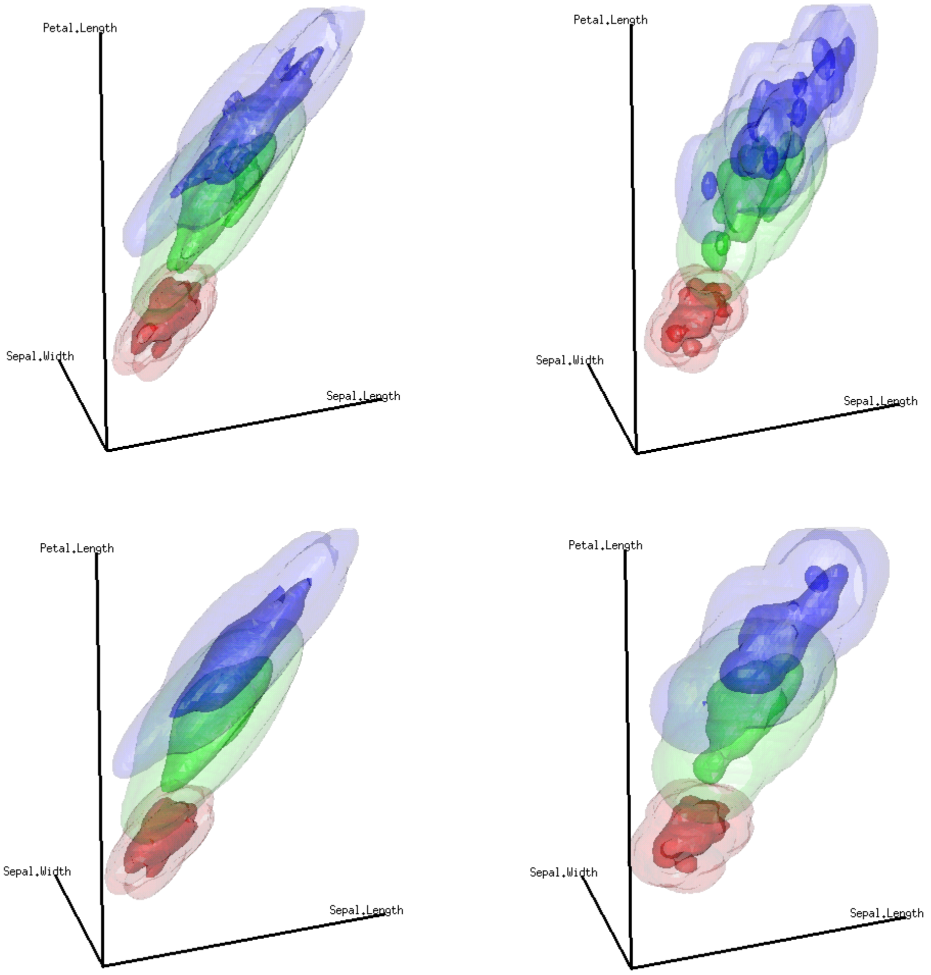

Figure 6: Kernel density estimates for kernel discriminant analysis: *I. setosa* red, *I. versicolor* green and *I. virginica* blue: plug-in (upper left), plug-in diagonal (upper right), SCV (lower right), SCV diagonal (lower right).

```
R> compare.kda.cv(x = ir, x.group = ir.group, bw = "plugin",
+  pilot = "samse", pre = "sphere")
R> compare.kda.diag.cv(x = ir, x.group = ir.group, bw = "plugin",
+  pilot = "samse", pre = "scale")
R> compare.kda.cv(x = ir, x.group = ir.group, bw = "scv", pre = "sphere")
R> compare.kda.diag.cv(x = ir, x.group = ir.group, bw = "scv",pre = "scale")
```

to give misclassification rates: plug-in 0.0533, plug-in diagonal 0.0667, SCV 0.0400 and SCV diagonal 0.0533. The unconstrained SCV selector performs the best in this case.

5. General recommendations

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. Amongst the unconstrained selectors, we advise against using the BCV selector. The LSCV selector is useful in some cases though its performance is known to be highly variable. The 2-stage plug-in and the SCV selectors can be viewed as generally recommended selectors.

Acknowledgments

The author would like to acknowledge the support of Martin Hazelton as supervisor during the author's Ph.D. candidature at the University of Western Australia, during which much of this research was carried out. The candidature was supported financially by an Australian Postgraduate Award.

References

Adler D, Murdoch D (2007). **rgl***: 3D Visualization Device System (OpenGL).* R package version 0.74, URL http://rgl.neoscientists.org/.

Bowman AW (1984). "An Alternative Method of Cross-Validation for the Smoothing of Density Estimates." *Biometrika*, **71**, 353–360.

Bowman AW, Azzalini A (2007). **sm***: Smoothing Methods for Nonparametric Regression and Density Estimation.* University of Glasgow, UK and Universita di Padova, Italia. R package version 2.2, URL http://www.stats.gla.ac.uk/~adrian/sm/.

Bowman AW, Foster P (1993). "Density Based Exploration of Bivariate Data." *Statistics and Computing*, **3**, 171–177.

Duong T, Hazelton ML (2003). "Plug-in Bandwidth Matrices for Bivariate Kernel Density Estimation." *Journal of Nonparametric Statistics*, **15**, 17–30.

Duong T, Hazelton ML (2005a). "Convergence Rates for Unconstrained Bandwidth Matrix Selectors in Multivariate Kernel Density Estimation." *Journal of Multivariate Analysis*, **93**, 417–433.

Duong T, Hazelton ML (2005b). "Cross-Validation Bandwidth Matrices for Multivariate Kernel Density Estimation." *Scandinavian Journal of Statistics*, **32**, 485–506.

Feng D, Tierney L (2007). **misc3d***: Miscellaneous 3D Plots.* R package version 0.4-0, URL http://CRAN.R-project.org/.

Hall P, Kang KH (2005). "Bandwidth Choice for Nonparametric Classification." *The Annals of Statistics*, **33**, 284–306.

Hall P, Marron JS, Park BU (1992). "Smoothed Cross-Validation." *Probability Theory and Related Fields*, **92**, 1–20.

Hyndman RJ (1996). "Computing and Graphing Highest Density Regions." *The American Statistician*, **50**, 120–126.

R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rudemo M (1982). "Empirical Choice of Histograms and Kernel Density Estimators." *Scandinavian Journal of Statistics*, **9**, 65–78.

Sain SR, Baggerly KA, Scott DW (1994). "Cross-Validation of Multivariate Densities." *Journal of the American Statistical Association*, **89**, 807–817.

Sheather SJ, Jones MC (1991). "A Reliable Data-based Bandwidth Selection Method for Kernel Density Estimation." *Journal of the Royal Statistical Society B*, **53**, 683–690.

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

Wand MP, Jones MC (1993). "Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation." *Journal of the American Statistical Association*, **88**, 520–528.

Wand MP, Jones MC (1994). "Multivariate Plug-in Bandwidth Selection." *Computational Statistics*, **9**, 97–116.

Wand MP, Jones MC (1995). *Kernel Smoothing*. Chapman and Hall Ltd., London.

Wand MP, Ripley BD (2006). **KernSmooth***: Functions for Kernel Smoothing for Wand & Jones (1995)*. R package version 2.22-19, URL http://CRAN.R-project.org/.

Affiliation:

Tarn Duong
Unité Analyse d'Images Quantitative
Institut Pasteur
25 rue du Docteur Roux
75015 Paris, France
E-mail: tduong@pasteur.fr

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 21, Issue 7 *Submitted:* 2007-03-04
October 2007 *Accepted:* 2007-09-12