Citation
Davies, T. and Baddeley, A. 2017. Fast computation of spatially adaptive kernel estimates. Statistics and Computing: pp. 1-20. http://doi.org/10.1007/s11222-017-9772-4

Statistics and Computing manuscript No.
(will be inserted by the editor)

Fast computation of spatially adaptive kernel estimates

Tilman M. Davies[1] · **Adrian Baddeley**[2,3]

Received: date / Accepted: date

Abstract Kernel smoothing of spatial point data can often be improved using an adaptive, spatially-varying bandwidth instead of a fixed bandwidth. However, computation with a varying bandwidth is much more demanding, especially when edge correction and bandwidth selection are involved. This paper proposes several new computational methods for adaptive kernel estimation from spatial point pattern data. A key idea is that a variable-bandwidth kernel estimator for d-dimensional spatial data can be represented as a slice of a *fixed*-bandwidth kernel estimator in $(d+1)$-dimensional scale space, enabling fast computation using Fourier transforms. Edge correction factors have a similar representation. Different values of global bandwidth correspond to different slices of the scale space, so that bandwidth selection is greatly accelerated. Potential applications include estimation of multivariate probability density and spatial or spatiotemporal point process intensity, relative risk, and regression functions. The new methods perform well in simulations and in two real applications concerning the spatial epidemiology of primary biliary cirrhosis and the alarm calls of capuchin monkeys.

Keywords Bandwidth selection · Edge correction · Fourier transform · Intensity · Scale space · Spatial point process

TMD was supported in part by the Royal Society of New Zealand, Marsden Fast-start grant 15-UOO-092: *Smoothing and inference for point process data with applications to epidemiology*. AB was supported in part by the Australian Research Council, Discovery Outstanding Researcher Award DP130104470.

✉ Tilman M. Davies
tdavies@maths.otago.ac.nz
· [1] Department of Mathematics & Statistics, University of Otago,
PO Box 56, Dunedin 9054, New Zealand
· [2] Department of Mathematics & Statistics, Curtin University,
GPO Box U1987, Perth WA 6845, Australia
· [3] Data61, CSIRO, Perth, Australia

1 Introduction

Kernel smoothing is a versatile tool for analysing spatial data: it is used to estimate probability density, point process intensity, relative risk, spatial regression functions and other quantities. Efficient algorithms for kernel smoothing are vital, and must continually be improved in order to handle larger and more complex datasets.

The classical fixed-bandwidth kernel estimator can be computed very rapidly, because it can be expressed as the convolution of the kernel function with the data points, and computed using the discrete Fourier transform.

However, employing a fixed bandwidth can have adverse consequences when the observation locations are highly heterogeneous. An elegant solution is to allow the bandwidth attached to each observation to adapt to local smoothing requirements, leading to an *"adaptive"* estimator, certain forms of which have been shown to have important theoretical and practical advantages over fixed-bandwidth estimation.

Unfortunately, adaptive smoothers are much more costly to compute, and their uptake in the applied literature for continuous spatial data remains quite limited. The goal of this paper is to develop efficient computational algorithms for such adaptive kernel estimators in an irregularly shaped region of space, including the non-trivial tasks of edge correction and bandwidth selection. Our attention is focussed on data observed in the two-dimensional plane $\mathbb{R}^2$, the most common setting for spatial analysis; the results can be extended to higher dimensions.

1.1 A Motivating Example

To demonstrate the need for adaptive smoothing, consider the data in Figure 1, which shows the geographic locations

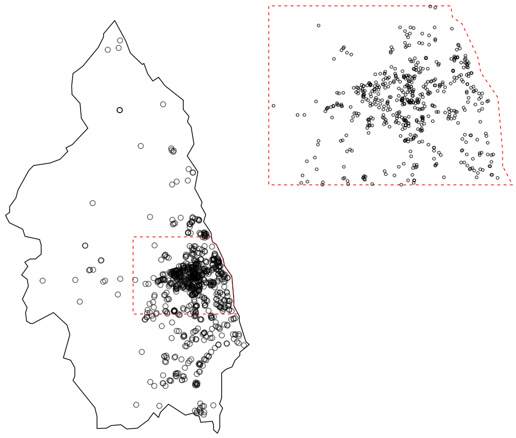


Fig. 1 Primary Biliary Cirrhosis (PBC) data. *Left:* full set of PBC cases (open circles) showing boundary of reporting region (solid lines). *Right:* enlarged view of the most densely populated sub-region.

of 761 cases of primary biliary cirrhosis (PBC) in a region of north-east England, from a study by Prince et al (2001). In the original analysis, these data were used in conjunction with a sample from the at-risk population, to obtain a fixed-bandwidth kernel-smoothed estimate of the spatially varying relative risk (cf. Kelsall and Diggle, 1995). For our purposes we consider only the PBC cases and estimate their *intensity*, the spatially-varying expected number of cases per unit area.

Figure 2 contrasts two intensity estimates for the PBC cases: one produced with a fixed bandwidth, the other with an adaptive smoothing technique described below. Several features can be seen immediately. The fixed bandwidth estimates have a far wider numerical range than the adaptive estimates, and show the hallmarks of simultaneous under- and over-smoothing. In the densely populated sub-region, the fixed-bandwidth smoothing is too generous, with relatively little fine-scale detail detectable. Conversely, toward the western border where observations are very scarce, there is not enough smoothing: the fixed-bandwidth estimate drops sharply, reaching values numerically close to zero (owing to the logarithmic scale, these are shown as missing pixels, coloured white). In the moderately-populated areas in the north, the fixed-bandwidth estimate has a ‘lumpy’ appearance caused by small clusters of cases. The use of a common, fixed bandwidth leads to an unhappy compromise between under- and over-smoothing in different parts of the study region.

In contrast, the adaptive estimate of intensity does not show any of these features. Areas of sparse data are smoothed over to avoid numerical problems and spurious bumps, while spatial detail is retained in the densely populated sub-region.

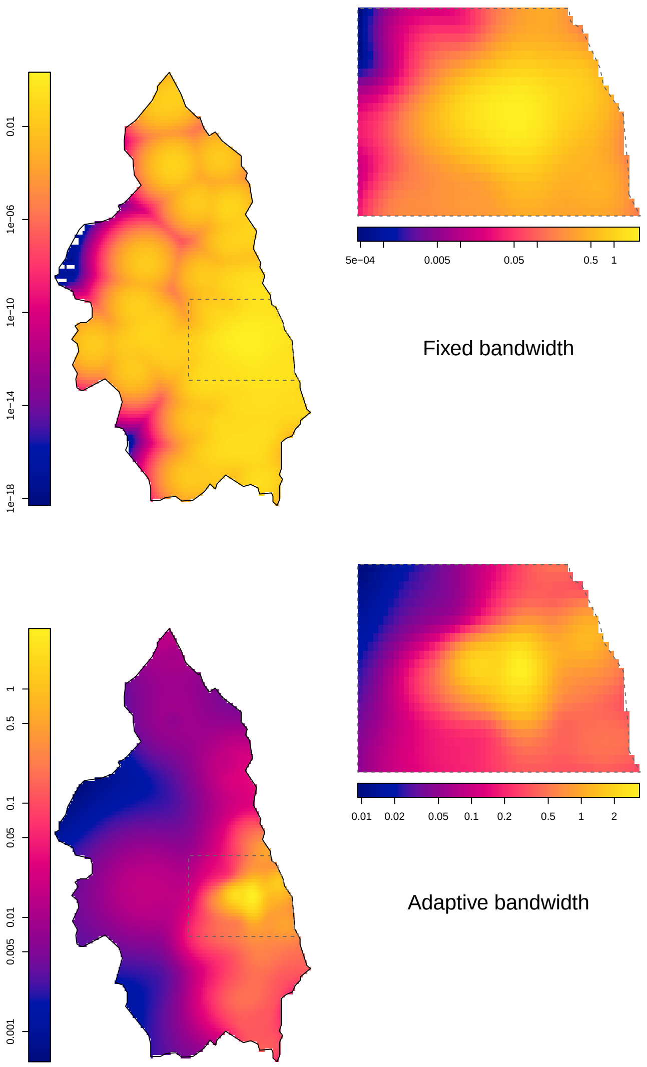


Fig. 2 Fixed bandwidth (*top*) and adaptive bandwidth (*bottom*) kernel estimates of intensity for the PBC data. Logarithmic colour scale; intensity expressed as number of cases per km^2.

1.2 A Brief Background on Adaptive Smoothing

Adaptive smoothing is typically discussed from two distinct perspectives. Adaptation may be performed either at each evaluation location, or alternatively, by rescaling the bandwidth of the individual kernels attached to the observations. The former techniques comprise methods referred to as “locally adaptive” or “balloon” estimators, the most well-known of which is the nearest-neighbour smoothing regimen of Loftsgaarden and Quesenberry (1965). The latter techniques are referred to as “point-adaptive”, “sample smoothing”, or “sample-point” methods, which began with the work of Breiman et al (1977). For a good general treatment of adaptive (or *variable bandwidth*) kernel estimators, see Terrell and Scott (1992) and references therein.

In this paper we focus on point-adaptive estimators, with particular attention paid to the 'square-root' methodology proposed by Abramson (1982). We do this for several reasons. First, unlike locally adaptive estimators, point-adaptive estimators result in densities that conserve mass and integrate appropriately to 1 over the study domain. This is an intrinsically desirable feature that leads to good performance in estimation of spatial intensity. Secondly, the Abramson smoother in particular has been shown to have theoretical benefits over fixed-bandwidth estimation, particularly so in the bivariate setting (see Hall and Marron, 1988; Davies and Hazelton, 2010, and our discussion in Section 2.2). Finally, the square-root rule of Abramson (1982) leads to an intuitively sensible smoothing regimen that we have found in practice to be particularly well-suited to spatial data, which often exhibit marked heterogeneity.

Hereinafter, we shall simply refer to point-adaptive estimators such as Abramson's approach without qualification as *adaptive* estimators.

1.3 Article Structure

In Section 2 we define the fixed bandwidth and adaptive bandwidth kernel estimators of intensity. We discuss "boundary bias" arising from observations in a bounded region of $\mathbb{R}^2$, and define edge-correction factors for reducing this bias. Section 3 reviews the standard algorithm for fixed-bandwidth kernel estimation using spatial discretisation and the fast Fourier transform. Sections 4 and 5 give two different algorithms for adaptive kernel estimation. The algorithm in Section 4 involves discretising the desired bandwidths into bins, and applying the fixed-bandwidth algorithm to each bin. In Section 5 we express the adaptive-bandwidth estimator as a slice of a fixed-bandwidth estimator in a higher-dimensional space, making it possible to calculate an edge-corrected adaptive kernel intensity estimate at multiple global bandwidth scales simultaneously using the fast Fourier transform. Performance of each algorithm is evaluated using two simulation studies. Section 6 describes an application to real data concerning the alarm calls of capuchin monkeys. Concluding remarks are in Section 7.

2 Kernel Intensity Estimation on $\mathbb{R}^2$

Let $Y = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$ be an observed set of points in a region of the plane $\mathbb{R}^2$, assumed to originate from a spatial point process with intensity function $\lambda(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^2$. Estimation of the intensity $\lambda(\cdot)$ is closely related to probability density estimation, since if the point process is assumed to be Poisson, then conditional on the number of points n, the locations $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$ are independent with common probability density $f(\boldsymbol{x})$ proportional to λ.

2.1 Fixed Bandwidth

For estimation at a point $\boldsymbol{x} = [x_1, x_2]^\top$ in $\mathbb{R}^2$, the fixed-bandwidth kernel estimate of $\lambda(\boldsymbol{x})$ is given by

$$\tilde{\lambda}_h(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in Y} \boldsymbol{K}_h(\boldsymbol{x} - \boldsymbol{y}), \tag{1}$$

where $h > 0$ is the scalar *bandwidth* or *smoothing parameter*, and $\boldsymbol{K}_h(\boldsymbol{u}) = h^{-2}\boldsymbol{K}(h^{-1}\boldsymbol{u})$ is the bandwidth-scaled *kernel function* (Silverman, 1986; Wand and Jones, 1995). Here $\boldsymbol{K}$ is a probability density on $\mathbb{R}^2$ with mean vector $[0,0]^\top$ and variance-covariance matrix equal to the identity matrix. The density is usually assumed to be symmetric in the sense that $\boldsymbol{K}(-\boldsymbol{u}) = \boldsymbol{K}(\boldsymbol{u})$ for all $\boldsymbol{u} \in \mathbb{R}^2$.

It is generally accepted in the literature that the specific functional form of the kernel $\boldsymbol{K}$ is of secondary importance to the amount of smoothing (see e.g. Wand and Jones, 1995). A popular practical choice is the Gaussian kernel—in particular, for spatial estimation, the infinite tails help ensure the final density/intensity is bounded away from zero. Other common choices include the biweight (quartic) and Epanechnikov kernels, both of which have compact support. In the theoretical exposition that follows, the results (including the necessary edge-correction techniques; see Section 2.3) are valid for all of these options and others that satisfy the assumption of being zero-centered symmetric densities. For the reason noted above, and given our objective of smoothing highly heterogeneous point patterns, we use the Gaussian kernel in our implementations.

In a more general form of (1), the scalar h is replaced by a 2×2 symmetric, positive-definite bandwidth matrix, which allows differing degrees of smoothing along each coordinate direction, and allows rotation of the kernel relative to the axes (see e.g. Wand and Jones, 1993). In what follows, we assume isotropic smoothing for simplicity.

2.2 Adaptive/Variable Bandwidth

As noted above, a major concern with the use of a constant bandwidth over all observations of Y is that it leads to poor estimator performance for heterogeneous point patterns. We often observe situations like the motivating example, with both undersmoothing and oversmoothing occurring simultaneously for sparsely and densely populated areas, respectively, even when h is chosen "optimally" in some sense.

An interesting solution to this problem was suggested by Abramson (1982) and developed further by Hall and Marron (1988). This is a particular case of the *variable bandwidth* or *adaptive* kernel estimator, which allows h to vary from observation to observation in a specific way. Denote an adaptive kernel intensity estimate by $\hat{\lambda}_{h_0}(\boldsymbol{x})$. We have

$$\hat{\lambda}_{h_0}(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in Y} h(\boldsymbol{y})^{-2}\boldsymbol{K}\big(h(\boldsymbol{y})^{-1}\{\boldsymbol{x} - \boldsymbol{y}\}\big) = \sum_{\boldsymbol{y} \in Y} \boldsymbol{K}_{h(\boldsymbol{y})}(\boldsymbol{x} - \boldsymbol{y}),$$

(2)

where

$$h(\boldsymbol{y}) = h_0 n^{1/2} \lambda(\boldsymbol{y})^{-1/2}. \tag{3}$$

The above shows the variable bandwidths are inversely proportional to the square root of the underlying intensity function; scaled by a quantity $h_0 > 0$ hereinafter referred to as the *global bandwidth*. This means that by relative comparison, the variable bandwidth attached to each point will be larger in sparsely populated regions and smaller in densely populated areas.

Intuitively, (3) makes sense: essentially we wish to capture more detail where there are more observations, while smoothing more in regions where a lack of data heightens uncertainty. Furthermore, Abramson (1982) and Hall and Marron (1988) demonstrate that under reasonable practical conditions, the order of the asymptotic bias of the adaptive estimator $\hat{\lambda}$ is improved notably over that of the fixed estimator $\tilde{\lambda}$, namely $o(h_0^4)$ as opposed to $o(h^2)$. Beneficial properties also flow through to certain functionals of adaptive intensity estimates, such as the density-ratio or *relative risk* function (Davies and Hazelton, 2010; Davies et al, 2016).

Of course, λ is unknown in practice, so in implementation of (3) we replace λ with a pilot estimate; it is convenient to use a fixed-bandwidth kernel estimate, yielding estimated variable bandwidths $\hat{h}(\boldsymbol{y}) = h_0 n^{1/2} \tilde{\lambda}_h(\boldsymbol{y})^{-1/2}$.

2.3 Edge Correction

In real-world applications, it is usually never the case that we observe spatially unbounded data. Rather, we record Y on $W \subset \mathbb{R}^2$, referring to the (often irregular) subset of the plane, W, as the study *window*. This means probability weight from kernels assigned to observations near the window boundary is lost over the edge, and by the same token, points that could potentially be observed just outside the boundary (but of course are not) are not given the opportunity to contribute to the final density estimate on W.

One must therefore take care when performing kernel estimation of planar intensities using either (1) and (2) at the boundaries of the study region. *Boundary bias* can be severe—even leading to asymptotic inconsistency near W and hence dangerously unreliable density estimates, a problem exacerbated by a 'small' n and a 'large' bandwidth (Jones, 1993; Wand and Jones, 1995). Fortunately, it can be shown that an approximative correction exists by way of a relatively simple post-hoc rescaling of the raw kernel estimate.

One way to achieve this for the fixed-bandwidth estimator is to calculate $\tilde{\lambda}_h(\boldsymbol{x})/q_h(\boldsymbol{x})$, where

$$q_h(\boldsymbol{x}) = \int_W \boldsymbol{K}_h(\boldsymbol{z} - \boldsymbol{x})\, \mathrm{d}\boldsymbol{z}. \tag{4}$$

The corrective term in (4) can be recognised as the volume of the kernel (that has bandwidth h and is centered at the evaluation coordinate $\boldsymbol{x}$) that falls within W (Diggle, 1985; Jones, 1993; Diggle, 2010).

The analogous correction for the adaptive estimator was developed by Marshall and Hazelton (2010). In much the same way, we calculate $\hat{\lambda}_{h_0}(\boldsymbol{x})/q_{h(\boldsymbol{x})}(\boldsymbol{x})$, where

$$q_{h(\boldsymbol{x})}(\boldsymbol{x}) = \int_W \boldsymbol{K}_{h(\boldsymbol{x})}(\boldsymbol{z} - \boldsymbol{x})\, \mathrm{d}\boldsymbol{z}. \tag{5}$$

Here, $h(\boldsymbol{x})$ can be interpreted as the bandwidth assigned to a hypothetical observation were it to fall at location $\boldsymbol{x}$, calculated using the pilot density of Y as per (3).

The key difference between (4) and (5) is of course the varying bandwidth in the latter. Indeed, where (4) can be computed by translation, direct implementation of (5) is implicitly related to the fineness of the evaluation coordinates $\boldsymbol{x} \in W$ with, technically, a new bandwidth required for each different evaluation location.

2.4 Spatial Discretisation

In practice, space is discretised so a kernel estimate and edge-correction factors can be computed at a finite set of 2D coordinate locations. Assume $R \supseteq W$ is a rectangle that encloses W, with horizontal and vertical side lengths r_1 and r_2 respectively. We subsequently partition R into $M \times N$ rectangular pixels, each of area $A = a_1 a_2$, where $a_1 = r_1/M$ and $a_2 = r_2/N$.

Let $\boldsymbol{c}_{(i,j)} = [c_{i1}, c_{j2}]^\top$ denote the centroid of the pixel at the ith and jth horizontal- and vertical-axis position; define the grid $C = \{\boldsymbol{c}_{(i,j)} : i \in \{1, \ldots, M\}, j \in \{1, \ldots, N\}\}$. Direct implementation of either (1) or (2) therefore means that the equation is evaluated at all $\boldsymbol{x} \in C$.

3 Algorithms for Fixed Bandwidth

If estimates of $\lambda(\boldsymbol{x})$ are required only at a small set of prediction points $\boldsymbol{x}$, equation (1) can be evaluated directly for each $\boldsymbol{x}$ giving results with high precision. However, for estimation of the intensity at all pixels in the grid C, direct evaluation of (1) is prohibitive, as it would require a total of nMN kernel evaluations. While the evaluation grid resolution can be coarsened (reducing M and N), we cannot reduce the number of observations n, which is problematic in large samples.

3.1 Kernel Convolution

Consider the counting measure $\boldsymbol{\psi}$ which puts a mass of 1 unit on each of the observation points $\boldsymbol{y}_i = [y_{i1}, y_{i2}]^\top$. Then,

$$\begin{aligned}(\boldsymbol{K}_h * \boldsymbol{\psi})(x_1,x_2) &= \int_{\mathbb{R}^2} \boldsymbol{K}_h([x_1-t_1, x_2-t_2]^\top)\,\mathrm{d}\boldsymbol{\psi}(t_1,t_2) \quad (6)\\ &= \sum_{i=1}^{n} \boldsymbol{K}_h([x_1-y_{i1}, x_2-y_{i2}]^\top)\\ &= \tilde{\lambda}_h(\boldsymbol{x}).\end{aligned}$$

That is, (1) can be represented as a convolution of the kernel $\boldsymbol{K}_h$ with $\boldsymbol{\psi}$. This is important because in practice convolutions can be computed rapidly using Fourier transforms; practical details follow in Section 3.2.

Relatively little attention has been given to the issue of calculating the edge correction surface (4) for point pattern data in the literature. It is useful to note that the corrective surface has an analogous representation in terms of a convolution. Explicitly considering the indicator function $I_W(\boldsymbol{x})$ which returns 1 if $\boldsymbol{x} \in W$ and 0 elsewhere, and taking into account the symmetry of $\boldsymbol{K}_h$, we have

$$\begin{aligned}(I_W * \boldsymbol{K}_h)(\boldsymbol{x}) &= \int_{\mathbb{R}^2} I_W(\boldsymbol{z})\boldsymbol{K}_h(\boldsymbol{x}-\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \quad (7)\\ &= \int_W \boldsymbol{K}_h(\boldsymbol{z}-\boldsymbol{x})\,\mathrm{d}\boldsymbol{z}\\ &= q_h(\boldsymbol{x})\end{aligned}$$

See Koch et al (2003).

3.2 Implementation by Binning

The so called *binned* kernel estimator (Silverman, 1982; Wand, 1994) arises due to the need to discretise space for computation of (6) and (7).

The counting measure $\boldsymbol{\psi}$ will be discretised as a collection of weights $w_{(i,j)}$ attached to the centroids $\boldsymbol{c}_{(i,j)}$ of pixels in C. These weights may be chosen in several ways: the simplest is *binning*, in which $w_{(i,j)}$ is the number of observations from Y that fall in pixel (i,j). Fractional weighting is also possible, such as *linear binning* (Wand, 1994). While estimation accuracy at a given location can be improved by using fractional weights, the overall computational burden is not greatly affected. For simplicity, we use simple binning in the current work.

After obtaining the weights, we may approximate the direct estimate $\tilde{\lambda}_h(\boldsymbol{x})$ with its binned version, $\tilde{\lambda}_h^*(\boldsymbol{x})$, where

$$\begin{aligned}\tilde{\lambda}_h^*(\boldsymbol{c}_{(i,j)}) &= \sum_{t=1}^{M}\sum_{u=1}^{N} \boldsymbol{K}_h\left([c_{i1}-c_{t1}, c_{j2}-c_{u2}]^\top\right) w_{(t,u)},\\ &= \sum_{\ell_1=1-M}^{M-1}\sum_{\ell_2=1-N}^{N-1} \boldsymbol{K}_h\left([a_1\ell_1, a_2\ell_2]^\top\right) w_{(i-\ell_1,j-\ell_2)}. \quad (8)\end{aligned}$$

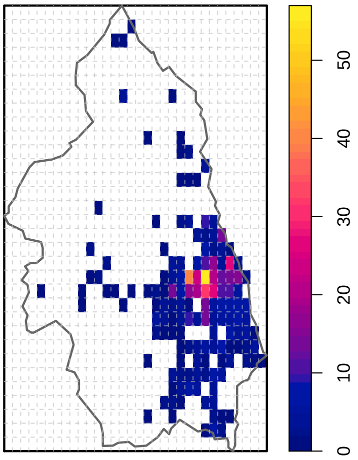


Fig. 3 Example evaluation grid and corresponding pixel weights based on a 32 × 32 resolution for the PBC data. Zero-count pixels are white.

Examination of (8) reveals it to be precisely the discretised version of the convolution in (6).

For the PBC data, Figure 3 illustrates the discretised pixel weights $w_{(i,j)}$ on a coarse 32×32 grid. The pixels themselves are delineated by grey dashed lines; W is visible as the irregular grey polygon, and the enclosing rectangle R is also shown.

Finally, the discretised version of the edge correction factor (7) is

$$\begin{aligned}q_h^*(\boldsymbol{c}_{(i,j)}) = a_1a_2 &\sum_{\ell_1=1-M}^{M-1}\sum_{\ell_2=1-N}^{N-1} \boldsymbol{K}_h\left([a_1\ell_1, a_2\ell_2]^\top\right)\\ &\times I_W\left(\boldsymbol{c}_{(i-\ell_1,j-\ell_2)}\right). \quad (9)\end{aligned}$$

3.3 Fast-Fourier Evaluation

The key property (Pinsky, 2002, p. 13, 92) is that the convolution $f * g$ of two integrable functions f and g satisfies $\mathscr{F}(f*g) = \mathscr{F}(f)\mathscr{F}(g)$ where $\mathscr{F}$ denotes the Fourier transform and the right-hand side is the pointwise product of the two functions. Consequently $f*g = \mathscr{F}^{-1}(\mathscr{F}(f)\mathscr{F}(g))$ where $\mathscr{F}^{-1}$ denotes the inverse Fourier transform. For computational purposes the functions will be discretised and we use the discrete Fourier transform (DFT). Fast algorithms for calculating the DFT and its inverse include the classical Cooley-Tukey method (Cooley and Tukey, 1965), which reduces the computational burden from $O(S^2)$ to $O(S\log S)$.

To apply the DFT we must ensure that the arrays of discretised kernel values $\boldsymbol{K}_h\left([a_1\ell_1, a_2\ell_2]^\top\right)$ and discretised weights $w_{(i,j)}$ can be treated as periodic functions. This is achieved by embedding them into larger arrays of size $P_M \times P_N$ as sketched in Figure 4. For simplicity we take $P_M = 2M$

$$\boldsymbol{w_0} = \begin{bmatrix} \boldsymbol{w} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{K_0} = \begin{bmatrix} \boldsymbol{K}([0,0]^\top) & \dots & \boldsymbol{K}([0,a_2(N-1)]^\top) & \boldsymbol{K}([0,-a_2N]^\top) & \dots & \boldsymbol{K}([0,-a_2]^\top) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{K}([a_1(M-1),0]^\top) & \dots & \boldsymbol{K}([a_1(M-1),a_2(N-1)]^\top) & \boldsymbol{K}([a_1(M-1),-a_2N]^\top) & \dots & \boldsymbol{K}([a_1(M-1),-a_2]^\top) \\ \boldsymbol{K}([-a_1M,0]^\top) & \dots & \boldsymbol{K}([-a_1M,a_2(N-1)]^\top) & \boldsymbol{K}([-a_1M,-a_2N]^\top) & \dots & \boldsymbol{K}([-a_1M,-a_2]^\top) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{K}([-a_1,0]^\top) & \dots & \boldsymbol{K}([-a_1,a_2(N-1)]^\top) & \boldsymbol{K}([-a_1,-a_2N]^\top) & \dots & \boldsymbol{K}([-a_1,-a_2]^\top) \end{bmatrix}$$

Fig. 4 Schematics of the extended $P_M \times P_N$ forms of the zero-padded pixel weight matrix ($\boldsymbol{w_0}$, within which $\boldsymbol{0}$ denotes a $M \times N$ matrix of zeros) and the reflected kernel evaluation matrix ($\boldsymbol{K_0}$) in preparation for Fourier transformation.

and $P_N = 2N$. The weights array is simply padded with additional rows and columns of zero values. Let $\boldsymbol{w} = [w_{(i,j)}]_{i=1,\dots,M;j=1,\dots,N}$ denote the $M \times N$ matrix of weights that approximates Y. We embed $\boldsymbol{w}$ in the upper leftmost corner of a larger $P_M \times P_N$ matrix of zeros, forming $\boldsymbol{w_0}$. Then we construct a $P_M \times P_N$ matrix $\boldsymbol{K_0}$ of kernel values in a permuted order, shown in Figure 4.

We may then compute the intensity estimate using (8) on the extended grid:

$$\tilde{\lambda}_h^*(\boldsymbol{c}_{(i,j)}) = \Re\left[\mathscr{F}^{-1}\left(\mathscr{F}(\boldsymbol{w_0})\mathscr{F}(\boldsymbol{K_0})\right)\right]_{(i,j)}; \\ i \in \{1,\dots,M\}, j \in \{1,\dots,N\};$$

setting the result to zero for $\boldsymbol{c}_{(i,j)} \notin W$. Computational complexity is thus reduced from $O(M^2N^2)$ to $O(P_MP_N\log(P_MP_N)) = O(MN\log(MN))$. Here $\Re[\cdot]$ denotes the real part of a complex number; in theory this operation is redundant because the inverse Fourier transform is real-valued, but it is necessary in practice, because the imaginary component will contain small nonzero values because of numerical error.

The edge correction surface q_h can also be evaluated using the FFT after a similar modification to ensure periodicity. Let $\boldsymbol{I}_W$ be the $M \times N$ matrix with entries $I_W(\boldsymbol{c}_{(i,j)})$, and $\boldsymbol{I}_{W,\boldsymbol{0}}$ a zero-filled $P_M \times P_N$ matrix with $\boldsymbol{I}_W$ taking position in the upper-left quadrant analogously to $\boldsymbol{w_0}$. Then,

$$q_h^*(\boldsymbol{c}_{(i,j)}) = \Re\left[\mathscr{F}^{-1}\left(\mathscr{F}(\boldsymbol{I}_{W,\boldsymbol{0}})\mathscr{F}(\boldsymbol{K_0})\right)\right]_{(i,j)}; \\ i \in \{1,\dots,M\}, j \in \{1,\dots,N\}.$$

Our final, edge-corrected binned kernel estimate is therefore obtained on the grid C with $\tilde{\lambda}_h^*(\boldsymbol{c}_{(i,j)})/q_h^*(\boldsymbol{c}_{(i,j)})$.

The binned fixed bandwidth intensity estimator represents the most common standard for implementation in statistical software. Examples in the R language include the packages `spatstat` (Baddeley and Turner, 2005; Baddeley et al, 2015) and `ks` (Duong, 2007), with the former also invoking the FFT-based edge correction given by (9).

Alternatives to the traditional methods of kernel smoothing and bandwidth selection include the frequency-domain approach of Bernacchia and Pigolotti (2011) and O'Brien et al (2016).

4 Adaptive Bandwidth: Partitioning Algorithm

Whereas the fixed-bandwidth estimator (1) and edge correction (4) can be expressed as convolutions in $\mathbb{R}^2$, the variable-bandwidth counterparts (2) and (5) cannot, so that it is unclear how to compute them efficiently.

One strategy is to discretise the bandwidth so that a relatively small number of bandwidth values is involved, and to approximate the adaptive estimate by a sum of fixed-bandwidth estimates. This is similar in spirit to the work by Sain and Scott (1996) and Sain (2002), though we approach the problem differently by examining the empirical quantiles of the distribution of sample-point bandwidths.

4.1 Procedure

Denote our pre-computed set of variable bandwidths as $\hat{\boldsymbol{h}} = \{\hat{h}_1,\dots,\hat{h}_n\}$, where $\hat{h}_i$ is the bandwidth attached to the ith observation $\boldsymbol{y}_i$. Using the Abramson (1982) smoother, $\hat{\boldsymbol{h}}$ is obtained via (3), using a (possibly edge-corrected) fixed bandwidth FFT-based binned pilot kernel intensity estimate. Consider now the empirical ζth quantiles, $\hat{h}^{(\zeta)}$, of these n values—let $0 < \delta \leq 1$ be a 'quantile step' value such that $D = \delta^{-1}$ is an integer (naturally, we set $\delta \ll 0.5$). Define bandwidth bins using the sequence of values $\hat{h}^{(0)}$, $\hat{h}^{(\delta)}$, $\hat{h}^{(2\delta)}$, …, $\hat{h}^{(1)}$ and, based on the estimated variable bandwidths $\hat{h}_i$; $i \in \{1,\dots,n\}$, allocate each corresponding observation $\boldsymbol{y}_i$ to one of the D bandwidth bins

$$\left[\hat{h}^{(0)},\hat{h}^{(\delta)}\right], \quad \left(\hat{h}^{(\delta)},\hat{h}^{(2\delta)}\right], \quad \dots, \quad \left(\hat{h}^{(\{D-1\}\delta)},\hat{h}^{(1)}\right]. \tag{10}$$

Finally, let Y_d be the subset of the pattern Y that falls in the dth bin, for $d \in \{1,\dots,D\}$, where Y_d contains n_d points. This partitions the pattern Y into D disjoint subsets, $Y = Y_1 \cup Y_2 \cup \dots \cup Y_D$.

The adaptive smoother will be approximated by replacing each desired bandwidth $\hat{h}_i$ by the midpoint of the bin in which it falls. The approximation is the sum of D fixed-bandwidth estimates operation on the corresponding subsets of Y:

$$\hat{\lambda}_{h_0}(\boldsymbol{x}) \approx \tilde{\lambda}^*_{\bar{h}_1}(\boldsymbol{x}|Y_1) + \tilde{\lambda}^*_{\bar{h}_2}(\boldsymbol{x}|Y_2) + \ldots + \tilde{\lambda}^*_{\bar{h}_D}(\boldsymbol{x}|Y_D), \qquad (11)$$

where $\bar{h}_1, \ldots, \bar{h}_D$ are the midpoints of the D bins in (10), and recall h_0 is the pre-set global smoothing parameter (3). Here $\tilde{\lambda}^*_h(\boldsymbol{x}|Y)$ denotes the fixed bandwidth estimate (1) based on the dataset Y. Each of these D estimates is calculated with the fixed bandwidth FFT-based methods described above; $\boldsymbol{x}$ in (11) is taken at $\boldsymbol{c}_{(i,j)}$; $i \in \{1,\ldots,M\}, j \in \{1,\ldots,N\}$.

Figure 5 demonstrates this procedure with $D = 20$ on the illustrative 32×32 grid for the PBC data. Initial bandwidths $\hat{\boldsymbol{h}}$ were calculated from (3); bandwidths were divided into 20 bins; the three panels of Figure 5 show the calculations for bins 1, 10 and 20. On the left is the subset of points Y_j; on the right is the FFT-based fixed-bandwidth kernel intensity estimate $\tilde{\lambda}^*_{\bar{h}_j}(\boldsymbol{x}|Y_j)$, for $j = 1, 10, 20$. Note the zero-valued missing pixels in the intensities for the two smaller bandwidths, indicating effectively no contribution from those intensities at those coordinates to what will eventually become the final estimate via (11).

The question now arises as to how one might correct for edge effects in the result of (11) using $q_{h(\boldsymbol{x})}(\boldsymbol{x})$ as given by (5). Note that we generally expect the edge correction surface to be far less heterogeneous than the intensity itself (i.e. the edge correction surface will generally be close to 1 and vary quite slowly in the interior of W in comparison to $\hat{\lambda}_{h_0}$). This means that it is not unreasonable to consider evaluating $q_{h(\boldsymbol{x})}$ on a coarser $L_M \times L_N < M \times N$ grid, thereafter employing an efficient interpolation procedure such as bilinear methods to scale the resolution back up to the target $M \times N$, without a massive loss of accuracy. In addition, we could also apply much the same bandwidth-partitioning approach to the variable bandwidths at the $L_M \times L_N$ knots, further reducing the computational burden.

Assume L_M and L_N are divisors of M and N respectively. The grid $G = \{\boldsymbol{g}_{(k,l)} : k \in \{1,\ldots,L_M\}, l \in \{1,\ldots,L_N\}\}$ of pixel centroids $\boldsymbol{g}_{(k,l)} = [g_k, g_l]^\top$ is a subset of the full grid C. Each centroid $\boldsymbol{g}_{(k,l)}$ is assigned a bandwidth value, calculated in the same way as the observation-specific bandwidths $\hat{\boldsymbol{h}}$ using the pilot-estimated version of (3). Write $\hat{\boldsymbol{s}} = \{\hat{h}(\boldsymbol{g}) : \boldsymbol{g} \in G\}$. Then, we choose a value β such that $\beta^{-1} = B$ is an integer, and partition the $\boldsymbol{g} \in G$ into the B bins of β-quantiles of $\hat{\boldsymbol{s}}$ defined by

$$\left[\hat{s}^{(0)}, \hat{s}^{(\beta)}\right], \quad \left(\hat{s}^{(\beta)}, \hat{h}^{(2\beta)}\right], \quad \ldots, \quad \left(\hat{s}^{(\{B-1\}\beta)}, \hat{s}^{(1)}\right]. \quad (12)$$

Given the mid-points of these bandwidth intervals, labelled $\bar{s}_1, \ldots, \bar{s}_B$, the next step is to perform the kernel-window convolution as per (9) for each bandwidth value using the FFT and repopulate the $L_M \times L_N$ pixels with the appropriate edge correction value given the association of each $\boldsymbol{g} \in G$ with the B bins. If necessary, we then interpolate the results from

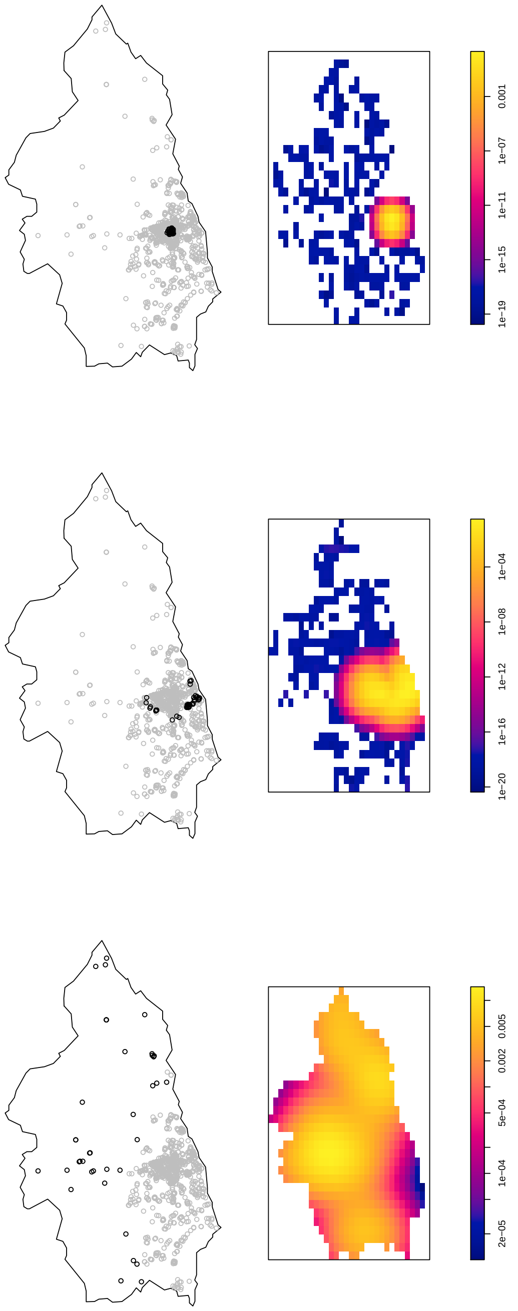


Fig. 5 Bandwidth-partitioned kernel estimation for the PBC data. Top, middle and bottom panels show the results for bandwidth bins $j = 1$, $j = 10$ and $j = 20$ respectively. Left panels show the full set Y of data points (grey) and the relevant subset Y_j (black). Right panels show the corresponding fixed-bandwidth kernel intensity estimate. Logarithmic colour map.

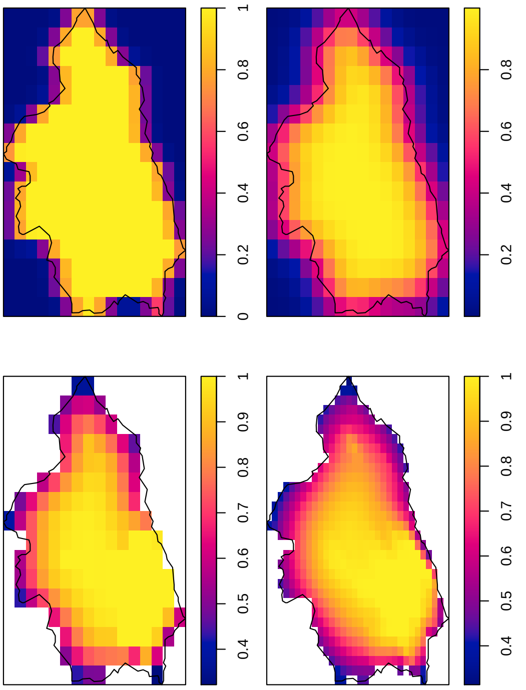


Fig. 6 Computing the bandwidth-partitioned, interpolated adaptive edge correction surface for the PBC data, using a coarsened 16×16 centroid grid, with the target being the original 32×32 grid. *Top left* and *right* respectively: Fixed bandwidth FFT-based surfaces using $\bar{s}_2$ and $\bar{s}_5$. *Bottom left* and *right* respectively: Repopulated adaptive corrective surface on the coarse grid, and following bilinear interpolation to the higher resolution.

the coarse $L_M \times L_N$ grid back to the full $M \times N$ grid using bilinear interpolation.

These strategies are illustrated for the PBC data in Figure 6. The top two images show the fixed bandwidth FFT-based corrective surfaces for the second and the fifth bandwidth bins, that is, $q^*_{\bar{s}_2}$ and $q^*_{\bar{s}_5}$, using a coarse 16×16 evaluation grid. The effect of a larger bandwidth resulting in corrective edge factors encroaching further into the interior of W is clear to see. The bottom two images show the appropriated adaptive corrective surface based on remapping the centroids $\boldsymbol{g} \in G$ to their corresponding values given the individual bandwidth bin to which they belong. This is done initially on the coarse grid; also shown is the bilinearly interpolated result to the target resolution of 32×32.

Noteworthy from these latter two plots is the fact that the corrective factors remain close to 1, even close up against the region boundary, in the vicinity of the densely populated subregion; this being owed to the relatively small bandwidths attached to observations in this area. This highlights another benefit of this kind of adaptive smoothing—if there is important detail in a populated area near the boundary it will be better preserved than in a fixed bandwidth estimate, due to lesser interference by correction factors.

Approximating an adaptive kernel intensity estimate by partitioning the variable bandwidths and turning the problem into one of multiple fixed bandwidth estimates is a relatively ad-hoc solution, though one that is not entirely unlike the strategies behind the pixel-binning of the spatially continuous observations necessary for implementation of the FFT-based estimator itself. It can be shown that the approximation is uniformly convergent to the correct value as the discretisation becomes finer. Clearly the size of D, governed directly by the fineness of the bandwidth quantiles via δ, is of key interest with respect to computational load and numerical accuracy—since calculating $\hat{\lambda}_{h_0}$ will now rise to order $O(DMN\log(MN))$. Edge-correction introduces much the same decision in terms of setting β, as well as the option of reducing the target grid resolution, meaning a decision on setting L_M and L_N.

4.2 Simulation Study: Partition Accuracy

To investigate the consequences of these decisions, we now perform a series of simulations to compare—and time—explicit calculation of adaptive kernel intensities against variously dialled partitioned versions thereof.

We generate synthetic datasets according to a planar log-Gaussian Cox point process (LGCP) clipped to an irregular polygon W enclosed in the unit square. The LGCP class (Møller et al, 1998) is a flexible but analytically tractable model for a point process with spatially-varying random intensity. Let $Z(\boldsymbol{u})$, $\boldsymbol{u} \in \mathbb{R}^2$ be a stationary Gaussian random field (GRF). Define the "driving intensity" $\Lambda(\boldsymbol{u}) = \exp(Z(\boldsymbol{u}))$. Conditional on the realisation of the driving intensity $\Lambda(\boldsymbol{u}) = \lambda(\boldsymbol{u})$, $\boldsymbol{u} \in \mathbb{R}^2$, let $\boldsymbol{Y}$ be a Poisson point process with intensity function $\lambda(\boldsymbol{u})$. Then the unconditional distribution of $\boldsymbol{Y}$ is a log-Gaussian Cox process. The distribution is completely determined by the first and second moments of the Gaussian field Z, which can be chosen at will to produce random intensity surfaces that exhibit different styles of heterogeneity.

In this study we use an exponential covariance function r for all examples: $r(\boldsymbol{x}) = \sigma^2 \exp(-\|\boldsymbol{x}\|/\phi)$; $\sigma^2, \phi > 0$. Four problem cases are defined by fixing the parameters σ^2 and ϕ at different values: Case 1 has $\sigma^2 = 0.9, \phi = 0.03$; Case 2 has $\sigma^2 = 0.9, \phi = 0.09$; Case 3 has $\sigma^2 = 2, \phi = 0.03$; and Case 4 has $\sigma^2 = 2, \phi = 0.03$. Example realisations of zero-mean GRFs on W from each of these cases are shown in the top row of Figure 7. Where Case 3 is characterised by the tightest, spikiest peaks in observation intensity, point clusters in Case 4 are spatially larger though less extreme. The spatial functions generated in Cases 1 and 2 vary more slowly over W, with Case 2 exhibiting the least abrupt changes over space.

The simulations proceed as follows, repeated over 100 iterations. For each problem case, at each iteration we generate a zero-mean GRF with the defined values of σ^2 and

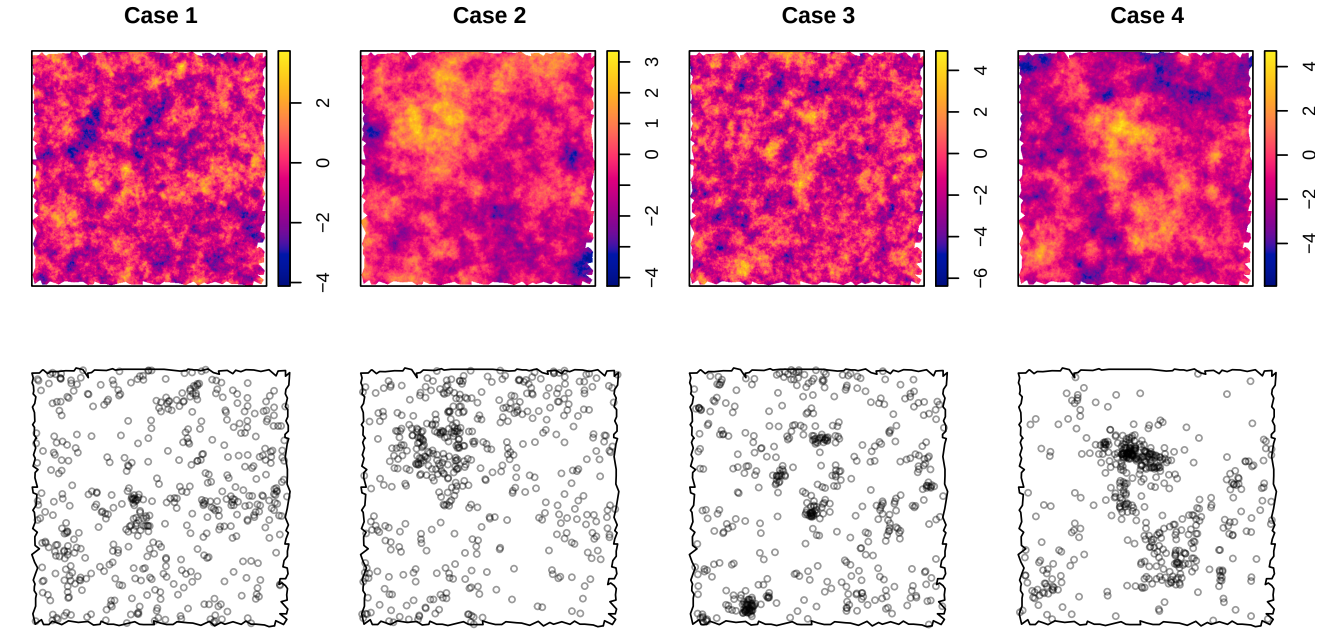


Fig. 7 Example log-intensity and corresponding point pattern realisations of the four LGCP problem cases for the partition accuracy simulation study. The irregular polygon is enclosed in the unit square.

ϕ and rescale its exponential to integrate to unity, giving a density function. This is followed by generation of a corresponding point pattern Y of size $n \sim \mathrm{POI}(\mu)$, where $\mu \in \{100, 1000\}$. A randomly generated point pattern for each of the four problem cases (each of size 500 for visual comparison) is given in the bottom row of Figure 7.

For a given data set, its intensity is estimated first via direct adaptive kernel estimation, followed by a battery of partitioned estimates using all combinations of the values $\delta, \beta \in \{0.01, 0.025, 0.05, 0.1\}$ and $L_M = L_N \in \{16, 32, 64\}$. The entire exercise is completed three times for differing target resolutions of $M = N \in \{64, 128, 256\}$. All computations were performed in R (R Core Team, 2017) on a modern desktop machine.

Accuracy of the approximation will be measured by the integrated squared error (ISE)

$$\mathrm{ISE}[\hat{f}] = \int \left(\hat{f} - f\right)^2 \tag{13}$$

where in this case f is the exact, directly computed estimate and $\hat{f}$ is the approximation using partitioned bandwidths.

Figure 8 shows boxplots of the ISE for each collection of partitioned estimates (on the density scale) for each generated data set in Case 3, based on $\mu = 1000$ with a target resolution of $M = N = 128$. Results for the other cases, target resolutions and mean sample size are provided as online supplementary material. All results tell the same story relatively speaking, simply with, as we would expect, larger overall error magnitudes and only slightly smaller computation times for the $\mu = 100$ case.

Immediately striking are the adverse consequences of excessive coarsening of the edge-correction surface via L_M and L_N. In the Figure 8 results, the attempts to interpolate the $L_M = L_N = 16$ surface back up to the desired $M = N = 128$ resolution introduce noticeable additional numeric error. That said, as soon as we consider either of the finer $L_M \times L_N$ resolutions, the overall discrepancy with respect to the direct estimates is reduced dramatically, with there being little discernable difference when we consider using either $L_M = L_N = 32$ or $L_M = L_N = 64$. Given an edge-correction resolution, a choice of bandwidth discretisation at either the intensity (δ) or edge-correction (β) stages has a minimal yet fairly consistent impact on the median ISEs. The drawback of the coarsest partitioning i.e. using $\delta = 0.1$ or $\beta = 0.1$ (for 10 bandwidth bins) seems to be a greater chance of outlying large error, though this tendency settles down as we decrease these to 0.05 (20 bins); 0.025 (40 bins); or 0.01 (100 bins).

In terms of total elapsed execution times relative to direct estimation, we note a marked improvement when the estimate is obtained by partitioning. We can see in Figure 8 that the most influential factor is the fineness of the bandwidth partitioning applied to the intensity itself. However, cross-referencing these boxplots against the ISE results, we can conclude that the additional computing time required for use of the finest $\delta = 0.01$ partitioning does not appear to yield a meaningful improvement with respect to proximity to the direct estimate. For example in this instance, provided we ensured an edge-correction resolution of at least $L_M = L_N = 32$, choosing $\delta = 0.05$ for 20 bandwidth bins would yield results little worse than direct estimation, with

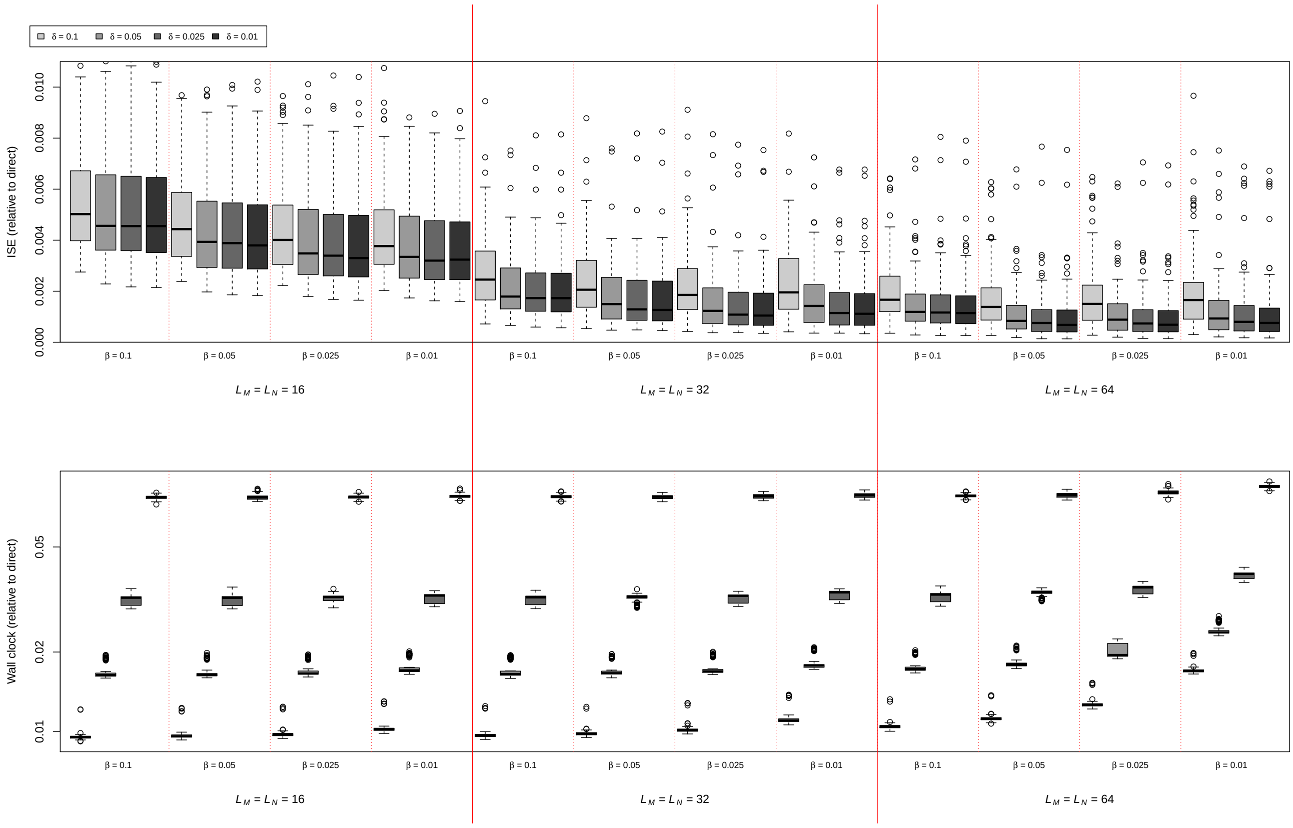


Fig. 8 ISEs relative to direct estimation (*top*) and wall clock timings relative to direct (*bottom*) for partitioned adaptive kernel intensity estimates using all combinations of δ, β, and $L_M = L_N$, for Case 3; set are $\mu = 1000$ and $M = N = 128$. For visual clarity we truncate the most extreme outliers in the ISE plots; and the relative timings are placed on a log scale.

only around 2% of the typical direct computation time. To place this into context, in our current implementation on a modern desktop machine, direct evaluation in this case averages roughly 20 seconds per estimate.

Choosing an ‘appropriate’ degree of partitioning in practice is largely application-dependent. That said, our collection of simulation results for all scenarios allow us to provide some general guidelines. First, it seems preferable to retain a sufficiently fine (i.e. 64^2 or greater) edge-correction surface; a decision that does not appear to have heavy consequences in computational terms. A similar comment applies to the edge-correction bandwidth partitioning parameter β, and we suggest using a value no coarser than $\beta = 0.05$. Most importantly, for the intensity/density partitioning parameter, a value of around $\delta = 0.025$ appears to provide universally satisfactory ISEs tempered with an acceptable computational burden—the additional time required to compute the finer partitions offering progressively smaller relative improvements to ISE.

5 Adaptive Bandwidth: Multi-Scale Estimation via 3D FFT

In the experiments reported above, the partitioning technique yielded good approximations to the adaptive intensity estimates with a substantial saving in computation. However, it is still costly to apply this technique at multiple values of the global bandwidth. The requirement to re-evaluate a kernel estimate using different bandwidths occurs both in interactive data analysis and in automatic data-driven bandwidth selection.

In this section, we propose a certain reformulation of the adaptive smoothing problem for a point pattern in $\mathbb{R}^2$ into one of a convolution in $\mathbb{R}^3$ so that it can be calculated using Fourier techniques. It turns out that after evaluating a particular trivariate convolution *once*, we are essentially able to obtain the adaptive intensity estimate, with edge-correction, for *any* chosen global bandwidth scaling without any additional kernel evaluations.

5.1 Key Idea

The key idea is to extend the spatial domain into an extra dimension, by introducing a third axis representing the logarithm of the bandwidth. The result is "scale space" (Chaudhuri and Marron, 2000). The family of kernel functions $\boldsymbol{K}_h(\cdot)$ in two-dimensional space is replaced by a single kernel function in scale space

$$\mathscr{K}(v_1,v_2,v_3) = \boldsymbol{K}_{\exp(-v_3)}(v_1,v_2). \tag{14}$$

Given a set of data points $\boldsymbol{y}_1,\dots,\boldsymbol{y}_n$ and corresponding bandwidths $h_1,\dots,h_n$, consider the points $\boldsymbol{t}_i = [\boldsymbol{y}_i, \log h_i]^\top = [y_{i1}, y_{i2}, \log h_i]^\top$ in scale space. The convolution of the kernel $\mathscr{K}$ with the counting measure $\boldsymbol{\Psi}$ on $\boldsymbol{t}_1,\dots,\boldsymbol{t}_n$ is

$$\begin{aligned}
&(\mathscr{K} * \boldsymbol{\Psi})(v_1,v_2,v_3)\\
&\quad= \int_{\mathbb{R}^3} \mathscr{K}([v_1-t_1, v_2-t_2, v_3-t_3]^\top)\,\mathrm{d}\boldsymbol{\Psi}(t_1,t_2,t_3)\\
&\quad= \sum_{i=1}^n \mathscr{K}([v_1-y_{i1}, v_2-y_{i2}, v_3-\log h_i]^\top)\\
&\quad= \sum_{i=1}^n \boldsymbol{K}_{h_i\exp(-v_3)}([v_1-y_{i1}, v_2-y_{i2}]^\top).
\end{aligned}$$

Evaluating this convolution on the plane where $v_3 = 0$ yields

$$\begin{aligned}
(\mathscr{K} * \boldsymbol{\Psi})(v_1,v_2,0) &= \sum_{i=1}^n \boldsymbol{K}_{h_i}([v_1-y_{i1}, v_2-y_{i2}]^\top)\\
&= \hat{\lambda}_1(v_1,v_2),
\end{aligned}$$

the adaptive kernel estimate of intensity (without edge correction) using individual bandwidths h_i. Moreover, evaluating on the plane $v_3 = -\log s$ gives

$$(\mathscr{K} * \boldsymbol{\Psi})(v_1,v_2,-\log s) = \hat{\lambda}_s(v_1,v_2),$$

the adaptive estimate using individual bandwidths sh_i. Thus, a single convolution operation yields the adaptive kernel estimates using bandwidths sh_i for multiple values of s.

5.2 Technical Definition

The idea sketched above needs a careful restatement to ensure that the required quantities are well-defined and accessible. First, suppose $H = [h_{\min}, h_{\max}]$ represents the entire range of permissible bandwidths, taking into account all desired global scales via h_0. Nominate any single value $h^\# \in H$ to take the role of a 'reference bandwidth'. Then, for any bandwidth $h \in H$, let

$$z(h) = \log(h/h^\#) = \log h - \log h^\#.$$

This is a 1–1 map from H onto $Z = [z_{\min}, z_{\max}]$ where $z_{\min} = z(h_{\min})$ and $z_{\max} = z(h_{\max})$. Note that the role of $h^\#$ is simply to ensure $0 \in Z$, for practical reasons. The inverse map is

$$z^{-1}(z) = \exp(z)h^\# = \exp(\log h^\# + z).$$

Using our bandwidth-scaled planar kernel from the preceding sections, $\boldsymbol{K}_{h(\cdot)}(\cdot)$, define a 3D kernel as

$$\mathscr{K}(v_1,v_2,v_3) = \boldsymbol{K}_{z^{-1}(-v_3)}([v_1,v_2]^\top) = \boldsymbol{K}_{\exp(-v_3)h^\#}([v_1,v_2]^\top). \tag{15}$$

Now, assume we have our pre-computed set of observation-specific Abramson bandwidths $\hat{\boldsymbol{h}} = \{\hat{h}_1,\dots,\hat{h}_n\}$ in hand. One could calculate these initially via (3) simply setting $h_0 = 1$ for any data set, though for numerical reasons it can be prudent to use some scale-specific value, allowing our desired rescaling (once all computations are complete) to operate with respect to this initial choice of h_0.

The n observations of our data set are now considered as trivariate—the ith observation is represented by $[\boldsymbol{y}_i, z(\hat{h}_i)]^\top \equiv \boldsymbol{u}_i = [u_{i1}, u_{i2}, u_{i3}]^\top$. Then we may express the convolution of $\mathscr{K}$ with the counting measure $\boldsymbol{\Psi}$ on the points (u_{i1}, u_{i2}, u_{i3}) as

$$\begin{aligned}
&\mathscr{K} * \boldsymbol{\Psi}(v_1,v_2,v_3)\\
&\quad= \int_{\mathbb{R}^3} \mathscr{K}(v_1-t_1, v_2-t_2, v_3-t_3)\,\mathrm{d}\boldsymbol{\Psi}(t_1,t_2,t_3)\\
&\quad= \sum_{i=1}^n \mathscr{K}(v_1-u_{i1}, v_2-u_{i2}, v_3-u_{i3})\\
&\quad= \sum_{i=1}^n \boldsymbol{K}_{\exp\{-(v_3-z(\hat{h}_i))\}h^\#}([v_1,v_2]^\top - \boldsymbol{y}_i)\\
&\quad= \sum_{i=1}^n \boldsymbol{K}_{\exp(-v_3)h^\#}([v_1,v_2]^\top - \boldsymbol{y}_i).
\end{aligned} \tag{16}$$

Then, slicing (16) at $v_3 = -\log s$ yields

$$\begin{aligned}
&\mathscr{K} * \boldsymbol{\Psi}(v_1,v_2,-\log s)\\
&\quad= \sum_{i=1}^n \boldsymbol{K}_{z^{-1}(\log s + z(\hat{h}_i))}([v_1-u_{i1}, v_2-u_{i2}]^\top)\\
&\quad= \sum_{\boldsymbol{y}\in Y} \boldsymbol{K}_{\exp(\log s + \log \hat{h}_i - \log h^\#)h^\#}([v_1,v_2]^\top - \boldsymbol{y})\\
&\quad= \sum_{\boldsymbol{y}\in Y} \boldsymbol{K}_{sh(\boldsymbol{y})}([v_1,v_2]^\top - \boldsymbol{y})\\
&\quad= \hat{\lambda}_{sh_0}([v_1,v_2]^\top); \quad [v_1,v_2]^\top \in \mathbb{R}^2,
\end{aligned} \tag{17}$$

i.e. recovery of the adaptive kernel intensity estimate of the original planar observations $\boldsymbol{y}_i$ with corresponding bandwidths $s\hat{h}_i$; $i = 1,\dots,n$. This shows that theoretically, any value of $s > 0$, acting as the global bandwidth (or scaling of the initial choice thereof in pre-calculation of $\hat{\boldsymbol{h}}$) can be requested following a single execution of the convolution.

In practice we would start with the Abramson bandwidths $h(\boldsymbol{v}), \boldsymbol{v} \in W$ calculated from the pilot estimate as in (3) for all discretised locations $\boldsymbol{v}$. The range $[a_{\min}, a_{\max}]$ of these bandwidth values is calculated. Then we specify *a priori* the range of global bandwidths of interest, say $s \in [s_{\min}, s_{\max}]$ where $0 < s_{\min} < 1 < s_{\max}$. This determines the range of all permissible bandwidths $H = [h_{\min}, h_{\max}]$ used in intensity estimation and edge correction, by $h_{\min} = s_{\min} a_{\min}$ and $h_{\max} = s_{\max} a_{\max}$. Then we let $Z = [z_{\min}, z_{\max}]$ where $z_{\min} = z(h_{\min})$, $z_{\max} = z(h_{\max})$. Finally let

$$Z \oplus \check{Z} = \{z_1 - z_2 : z_1, z_2 \in Z\} = [z_{\min} - z_{\max}, z_{\max} - z_{\min}].$$

Calculations in the scale space are required to extend over the interval $Z \oplus \check{Z}$ on the third axis. (It does not suffice to compute results over Z, because the intensity estimate at any rescaling value $s \in [s_{\min}, s_{\max}]$ requires a slice to be taken at $-\log s = z(h) - z(sh)$ where $h \in H$ and $sh \in H$, implying $-\log s \in Z \oplus \check{Z}$.)

5.3 Edge Correction

The edge-correction surface can also be expressed as a convolution, analogous to (7). Identify the two-dimensional plane $\mathbb{R}^2$ with the $z = 0$ plane in three dimensions, $E = \{[t_1, t_2, t_3]^\top : t_3 = 0\}$. Identify the window W with the corresponding subset $W_E = \{[t_1, t_2, 0]^\top : [t_1, t_2] \in W\}$ of $\mathbb{R}^3$. Let $\mathscr{L}_W$ be the measure on $\mathbb{R}^3$ corresponding to Lebesgue measure on W, that is, $\mathscr{L}_W(B) = \int_W I_B(t_1, t_2, 0)\,\mathrm{d}t_1\,\mathrm{d}t_2$ for subsets $B \subseteq \mathbb{R}^3$. Then, noting $\mathscr{K}$ is symmetric in the first two coordinates, we have

$$\begin{aligned}
&(\mathscr{L}_W * \mathscr{K})(v_1, v_2, v_3) \\
&\quad = \int_{\mathbb{R}} \int_W \mathscr{K}(v_1 - t_1, v_2 - t_2, v_3 - t_3)\,\mathrm{d}t_1\,\mathrm{d}t_2\,\mathrm{d}t_3 \\
&\quad = \int_{\mathbb{R}} \int_W \boldsymbol{K}_{z^{-1}(-v_3+t_3)}([v_1 - t_1, v_2 - t_2]^\top)\,\mathrm{d}t_1\,\mathrm{d}t_2\,\mathrm{d}t_3 \\
&\quad = \int_{\mathbb{R}^3} \mathscr{I}_W(t_1, t_2, t_3) \\
&\qquad \times \boldsymbol{K}_{z^{-1}(t_3 - v_3)}([t_1 - v_1, t_2 - v_2]^\top)\,\mathrm{d}t_1\,\mathrm{d}t_2\,\mathrm{d}t_3 \qquad (18) \\
&\quad = \int_{\mathbb{R}^2} \mathscr{I}_W(t_1, t_2, 0) \boldsymbol{K}_{z^{-1}(0 - v_3)}([t_1 - v_1, t_2 - v_2]^\top)\,\mathrm{d}t_1\,\mathrm{d}t_2 \qquad (19)
\end{aligned}$$

with (19) resulting naturally from the trivariate convolution in (18) when we consider the fact that the original trivariate integral is non-zero only when $t_3 = 0$. Thus, for a planar coordinate $[v_1, v_2]^\top$, slicing (19) at the corresponding z-scaled adaptive bandwidth as $v_3 = -z\{h([v_1, v_2]^\top)\}$ gives

$$\begin{aligned}
&\mathscr{L}_W * \mathscr{K}(v_1, v_2, -z\{h([v_1, v_2]^\top)\}) \\
&\quad = \int_{\mathbb{R}^2} \mathscr{I}_W(t_1, t_2, 0) \\
&\qquad \times \boldsymbol{K}_{z^{-1}(z\{h([v_1, v_2]^\top)\})}([t_1 - v_1, t_2 - v_2]^\top)\,\mathrm{d}t_1\,\mathrm{d}t_2 \\
&\quad = \int_W \boldsymbol{K}_{h([v_1, v_2]^\top)}([t_1 - v_1, t_2 - v_2]^\top)\,\mathrm{d}t_1\,\mathrm{d}t_2, \qquad (20)
\end{aligned}$$

which is precisely the edge-correction factor for the adaptive estimator at a location $[v_1, v_2]^\top \in W$ as per (5). Edge-correction requires us to take a slice at $-z(h); h \in H$, which again demands access to values on the interval $Z \oplus \check{Z}$.

5.4 Implementing the 3D Convolution

For estimation in practice we must again discretise the observation space, which now entails construction of a 3D $M \times N \times Q$-cell evaluation lattice within an appropriate 3D rectangular box $\mathscr{R} \subset \mathbb{R}^3$. This enclosing box is defined by the original planar rectangle R in the first two coordinates, and by the interval $Z \oplus \check{Z}$ along the third axis.

Let $\boldsymbol{\gamma}_{(i,j,k)} = [\gamma_{i1}, \gamma_{j2}, \gamma_{k3}]^\top$ denote the centroid of the voxel at the ith, jth planar- and kth z-scaled bandwidth position, defining their set as $\mathscr{C} = \{\boldsymbol{\gamma}_{(i,j,k)} : i \in \{1, \ldots, M\}, j \in \{1, \ldots, N\}, k \in \{1, \ldots, Q\}\}$. Then, let $\omega_{(i,j,k)}$ be the discretised counting measure of $\boldsymbol{\Psi}$, on $\mathscr{C}$, defined as the number of observations falling into voxel (i, j, k); $\omega_{(i,j,\cdot)} = 0 \forall [\gamma_{i1}, \gamma_{j2}]^\top \notin W$. For binned estimation of the adaptive intensity expressed in (16), we therefore have a 3D discrete convolution on $\mathscr{C}$ as follows:

$$\begin{aligned}
&\mathscr{K} * \omega(\boldsymbol{\gamma}_{(i,j,k)}) \\
&= \sum_{\ell_1 = 1-M}^{M-1} \sum_{\ell_2 = 1-N}^{N-1} \sum_{\ell_3 = 1-Q}^{Q-1} \mathscr{K}(a_1 \ell_1, a_2 \ell_2, a_3 \ell_3) \omega_{(i-\ell_1, j-\ell_2, k-\ell_3)}. \qquad (21)
\end{aligned}$$

In (21), recall that a_1 and a_2 are defined as the side lengths of the voxels in the first and second coordinates respectively. Accordingly, allow a_3 to denote the cell length in the third dimension.

Performing the trivariate discretisation for the PBC data using an $32 \times 32 \times 32$ illustrative lattice, Figure 9 shows the dimensionally augmented observations using the observation-specific Abramson bandwidths rescaled and placed on the appropriate interval $Z \oplus \check{Z}$. Also shown is the enclosing box $\mathscr{R}$. The topmost image draws those voxels of the illustrative lattice within $\mathscr{R}$ that have $\omega_{(i,j,k)} > 0$. The 2D study region is superimposed at the z value closest to zero, as defined for our edge-correction lookups in (20). Note, as dictated by the Abramson smoother, that densely clustered points are associated with smaller bandwidths. The bottom image displays isosurfaces of the 3D kernels $\mathscr{K}$ for four selected observations in the scale space. Note the intersection of the shaded 2D window W with the kernels, yielding the adaptive intensity estimate at a specified level of global smoothing. Taking parallel slices of the scale space by moving W up and down the z-axis corresponds to decreasing and increasing the global smoothing parameter respectively as per (17).

Prior to implementation of (21) using the FFT, in the same way the functionals of the 2D convolution (8) must allow for the periodic behaviour of the transform, the structures of the 3D convolution must also be modified appropriately. Apart from the increase in dimension, these objects are formed in much the same way: Retaining $P_M = 2M$ and $P_N = 2N$ from Section 3.3, let $P_Q = 2Q$ and define $\boldsymbol{\omega_0}$ as a $P_M \times P_N \times P_Q$ array with the $M \times N \times Q$ sub-box start-

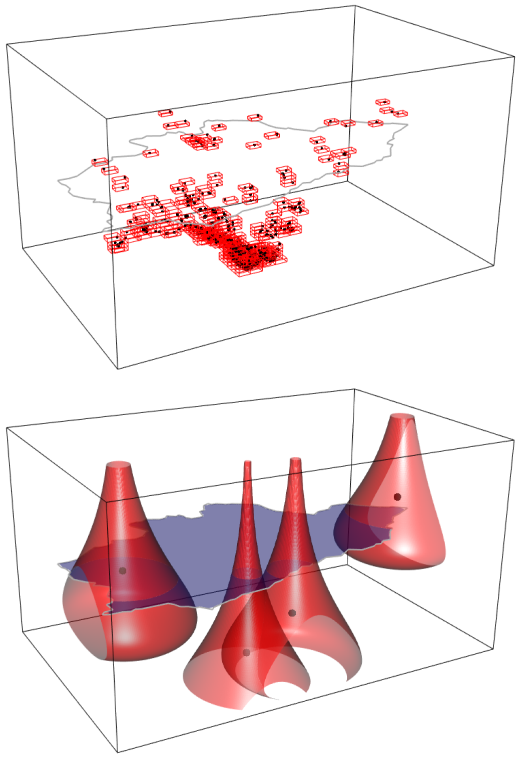


Fig. 9 Augmenting the dimension of our planar data to include the z-scaled bandwidth $z(\hat{h}_i)$ attached to the ith observation. *Top*: Voxels of the assumed $32 \times 32 \times 32$ evaluation lattice that contain at least one point in the 3D space. *Bottom*: Isosurfaces of the 3D kernels $\mathscr{K}$ for four selected observations; a given slice of the z-axis indicated by intersection of the shaded 2D study window with the 3D kernels in the scale space.

ing from the lowest vertex being occupied with the 3D array of counts $\boldsymbol{\omega} = \left[\omega_{(i,j,k)}\right]_{i=1,\ldots,M;j=1,\ldots,N;k=1,\ldots,Q}$; the remaining entries set to zero. Then, construct the 3D $P_M \times P_N \times P_Q$ reflected kernel evaluation array $\mathscr{K}_\mathbf{0}$ with 2D layers in the first two (planar, spatial) coordinates exactly as $\boldsymbol{K}_\mathbf{0}$ previously, with additional specification of the sequence $\xi = 0,\ldots,a_3(Q-1),-a_3Q,\ldots,-a_3$ along the third axis. Figure 10 provides schematics of $\boldsymbol{\omega}_\mathbf{0}$ along with one of the 2D layers of $\mathscr{K}_\mathbf{0}$ given a generic single value of ξ.

The trivariate convolution may then be computed on the extended array, and restricted to the original $M \times N \times Q$ array. Then for any global bandwidth sh_0 with $s \in [s_{\min}, s_{\max}]$ we may evaluate the adaptive kernel estimate by slicing at $z = -\log s$. The corresponding slice of the $M \times N \times Q$ array is at $k = \Omega(-\log s)$ where $\Omega(z) = \lceil Q(z - z_{\min})/(z_{\max} - z_{\min}) \rceil$. This provides the binned adaptive kernel intensity estimate on R:

$$\hat{\lambda}^*_{sh_0}(\boldsymbol{\gamma}_{(i,j,k)}) = \Re\left[\mathscr{F}^{-1}\left(\mathscr{F}(\boldsymbol{\omega}_\mathbf{0})\mathscr{F}(\mathscr{K}_\mathbf{0})\right)\right]_{(i,j,k)};$$
$$i \in \{1,\ldots,M\}, j \in \{1,\ldots,N\}, k = \Omega(-\log s).$$

Restriction to the study region W forms the final result for the adaptive intensity estimate with global bandwidth sh_0.

Lastly, the discretised version of the edge-correction convolution (18) we use in practice on the extended voxel lattice is defined as

$$\begin{aligned}
&\mathscr{L}_W * \mathscr{K}(\gamma_{i1}, \gamma_{j2}, \gamma_{k3}) \\
&= a_1 a_2 \sum_{\ell_1=1-M}^{M-1} \sum_{\ell_2=1-N}^{N-1} \sum_{\ell_3=1-Q}^{Q-1} \mathscr{K}(a_1\ell_1, a_2\ell_2, a_3\ell_3) \\
&\quad \times \mathscr{I}_W(\boldsymbol{\gamma}_{(i-\ell_1,j-\ell_2,k-\ell_3)}); \qquad (22) \\
&i \in \{1,\ldots,M\}, j \in \{1,\ldots,N\}, k \in \{1,\ldots,Q\}.
\end{aligned}$$

Consider once more the extended and reflected kernel evaluation array $\mathscr{K}_\mathbf{0}$ defined earlier, and the extended, zero-padded array $\mathscr{I}_{W,\mathbf{0}}$ defined in the same way as $\boldsymbol{\omega}_\mathbf{0}$ with the $M \times N \times Q$ lower sub-box now occupied by the appropriate entries of $\mathscr{I}_W$ at those coordinates. FFT operations subsequently provide, in the usual fashion, the $P_M \times P_N \times P_Q$ convoluted structure which may then be subsetted as appropriate to obtain the corresponding edge-correction factor for a given spatial location and associated bandwidth value:

$$q^*_{h([\gamma_{i1},\gamma_{j2}]^\top)}([\gamma_{i1},\gamma_{j2}]^\top) = \Re\left[\mathscr{F}^{-1}\left(\mathscr{F}(\mathscr{I}_{W,\mathbf{0}})\mathscr{F}(\mathscr{K}_\mathbf{0})\right)\right]_{(i,j,k)};$$
$$i \in \{1,\ldots,M\}, j \in \{1,\ldots,N\}, k = \Omega(-z\{h([\gamma_{i1},\gamma_{j2}]^\top)\})$$

Our edge-corrected, binned adaptive kernel intensity estimate computed via Fourier transformations is thus available as $\hat{\lambda}^*_{sh_0}(\boldsymbol{\gamma}_{(i,j,-\log s)})/q^*_{sh([\gamma_{i1},\gamma_{j2}]^\top)}([\gamma_{i1},\gamma_{j2}]^\top)\ \forall\ [\gamma_{i1},\gamma_{j2}]^\top \in W$.

One final technical issue warrants attention. In practice, we modify $\mathscr{K}$ slightly throughout and use

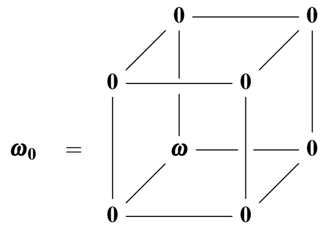
$$\mathscr{K}_\tau(v_1, v_2, v_3) = \tau(v_3)\mathscr{K}(v_1, v_2, v_3),$$

where $\tau(\cdot) : \mathbb{R} \to [0,1]$ is a *tapering function* on the relevant domain of the third axis, in place to ensure $\mathscr{K}$ is smooth and bounded. We stipulate that $\tau(z)$ will return 1 for any value $z \in Z \oplus \check{Z}$, tapering smoothly off to zero beyond these extremes. Our implementation uses the cosine taper (Harris, 1978), defined in the current context by

$$\tau(z) = \begin{cases} 0 & z \le d_1 \\ \left\{1 - \cos\left(\pi \frac{z-d_1}{d_2-d_1}\right)\right\}/2 & d_1 < z < d_2 \\ 1 & d_2 \le z \le d_3 \\ \left\{1 - \cos\left(\pi \frac{z-d_4}{d_3-d_4}\right)\right\}/2 & d_3 < z < d_4 \\ 0 & z \ge d_4 \end{cases}$$

where the change points d_1 through d_4 are calculated based on the span of $Z \oplus \check{Z}$ and the extension in this coordinate used in the construction of the discretised, reflected kernel evaluation array. We dictate that

$$d_1 = (z_{\max} - z_{\min} - a_3Q)/2;$$
$$d_2 = z_{\max} - z_{\min};$$
$$d_3 = z_{\min} - z_{\max}; \text{ and}$$
$$d_4 = (a_3(Q-1) + z_{\min} - z_{\max})/2.$$



$$[\mathscr{K}_{\mathbf{0}}]_{i=1,\dots,P_M;j=1,\dots,P_N;k\widehat{=}\xi} = \begin{bmatrix} \mathscr{K}(0,0,\xi) & \dots & \mathscr{K}(0,a_2(N-1),\xi) & \mathscr{K}(0,-a_2N,\xi) & \dots & \mathscr{K}(0,-a_2,\xi) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathscr{K}(a_1(M-1),0,\xi) & \dots & \mathscr{K}(a_1(M-1),a_2(N-1),\xi) & \mathscr{K}(a_1(M-1),-a_2N,\xi) & \dots & \mathscr{K}(a_1(M-1),-a_2,\xi) \\ \mathscr{K}(-a_1M,0,\xi) & \dots & \mathscr{K}(-a_1M,a_2(N-1),\xi) & \mathscr{K}(-a_1M,-a_2N,\xi) & \dots & \mathscr{K}(-a_1M,-a_2,\xi) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathscr{K}(-a_1,0,\xi) & \dots & \mathscr{K}(-a_1,a_2(N-1),\xi) & \mathscr{K}(-a_1,-a_2N,\xi) & \dots & \mathscr{K}(-a_1,-a_2,\xi) \end{bmatrix}$$

Fig. 10 Schematics of the extended $P_M \times P_N \times P_Q$ forms of the zero-padded pixel weight matrix ($\boldsymbol{\omega_0}$, within which **0** denotes a $M \times N \times Q$ array of zeros) and one layer of the $P_M \times P_N \times P_Q$ reflected kernel evaluation array $\mathscr{K}_{\mathbf{0}}$ at a third coordinate value of $\xi = 0,\dots,a_3(Q-1),-a_3Q,\dots,-a_3$ in preparation for 3D Fourier transformation.

If we rewrite equations (16)–(22) in terms of $\mathscr{K}_\tau$ instead of $\mathscr{K}$ *per se*, note that no results are affected when we are considering only those values $z \in Z \oplus \check{Z}$ that are relevant to our desired multi-scale intensities and corresponding edge-correction surfaces.

5.5 Simulation Study: Naïve Bootstrap

Choosing an appropriate level of smoothing in practice given an observed data set is an important and non-trivial exercise. Concerted research efforts over time have led to the development and appraisal of various data-driven bandwidth selectors for the fixed bandwidth kernel density estimator, and we benefit from the equivalence of bandwidth selection for densities/intensities (Diggle and Marron, 1988). Staple methods that now form the core of a researcher's univariate smoothing toolbox can be found in Wand and Jones (1995), Jones et al (1996), and references therein.

Several methods originally designed for univariate smoothing can be extended to multivariate kernel estimation, and Sain et al (1994) demonstrate this for various classical selectors. Sophisticated bandwidth selectors for non-scalar multivariate bandwidth parameterisations exist (see e.g. Wand and Jones, 1993; Duong and Hazelton, 2003, 2005), and Bayesian methods have also been discussed (Zhang et al, 2006).

To the best of our knowledge, however, no research has yet specifically targeted development of selectors for global bandwidth selection for the adaptive intensity estimator. Indeed, even an investigation of the performance of existing selectors designed for fixed-bandwidth smoothing has remained elusive, in part due to the heavy computational costs involved with repeated, direct evaluation of the target density. In this section, we demonstrate how the 3D FFT strategies can be used to facilitate these new pursuits, using a computationally demanding bootstrap approach.

In general, the idea of a bootstrapping procedure is to use resampling to assess properties of some random variable that depends upon the originally observed data (and any relevant additional parameters or functionals). In the context of a kernel estimate $\hat{f}$ of some density f, a common performance criterion is the integrated squared error (13). It has been suggested (Silverman, 1986; Taylor, 1989; Faraway and Jhun, 1990) that we may choose an approximately optimal bandwidth by using bootstrapping to minimise the mean ISE (MISE) with respect to the amount of smoothing.

Letting the ISE be our property of interest, a bootstrap version for our adaptive estimates can be defined as

$$\mathrm{MISE}[\hat{\lambda}_{h_0}] = \mathbb{E}^*\left[\int_W \left\{\hat{\lambda}_{h_0}(\boldsymbol{x}|Y^*) - \check{\lambda}_\eta(\boldsymbol{x}|Y)\right\}^2 \mathrm{d}\boldsymbol{x}\right]$$

where, given some (fixed or adaptive) reference kernel intensity estimate constructed of the original data Y, $\check{\lambda}_\eta(\boldsymbol{x}|Y)$, we have that $Y^*_{(i)} \sim \check{\lambda}_\eta(\boldsymbol{x}|Y)$; $\mathbb{E}^*$ denoting expectation with respect to the distribution of the bootstrapped data.

In practice, we generate N^* bootstrap samples $Y^*_{(i)} \sim \check{\lambda}_\eta(\boldsymbol{x}|Y)$ possessing corresponding sizes $n^*_{(i)} \sim \mathrm{POI}(n)$; $i = 1,\dots,N^*$. Taking the evaluation grid notation from Section 2.4, we minimise with respect to h_0 the quantity

$$\widehat{\mathrm{MISE}}[\hat{\lambda}_{h_0}] = \frac{a_1a_2}{N^*}\sum_{i=1}^{N}\sum_{\boldsymbol{c}\in C\cap W}\left\{\hat{\lambda}_{h_0}(\boldsymbol{c}|Y^*_{(i)}) - \check{\lambda}_\eta(\boldsymbol{c}|Y)\right\}^2. \tag{23}$$

Thus, in order to find

$$h_0^{(\mathrm{opt})} = \mathrm{argmin}_{h_0} \left\{ \widehat{\mathrm{MISE}}[\hat{\lambda}_{h_0}] \right\},$$

an inspection of (23) indicates the need to calculate N^* kernel intensity estimates, each one over a range of candidate global bandwidths. The 3D slicing technique allows the latter task to be completed quickly.

To time and test this bootstrap search for $h_0^{(\mathrm{opt})}$, we generated another 100 LGCP intensities and corresponding point patterns using the Case 1, $\mu = 1000$ specification from the simulation designs detailed in Section 4.2. In execution, for each of the 100 iterations, we used $N^* = 50$ bootstrapped data sets to trial each candidate h_0 via both the direct and 3D FFT approaches; and used an evaluation grid resolution of $M = N = 64$ (additionally specifying $Q = 64$ for the 3D FFT). For the fastest results in each case, when employing direct estimation, numerical optimisation routines were used to find $h_0^{(\mathrm{opt})}$; in using the 3D FFT, the individual MISEs were simply calculated for a fine increasing sequence of global bandwidth values; the value corresponding to the minimal estimated MISE being returned. In both cases, the fixed bandwidth pilot density used in calculation of the Abramson bandwidths utilises the classical leave-one-out least-squares cross-validation method to set the pilot bandwidth (see e.g. Silverman, 1986), and the reference intensity of the 'original' data $\breve{\lambda}_\eta(\cdot|Y)$ is also a fixed-bandwidth kernel estimate, with η set via Terrell (1990)'s oversmoothing factor (it is suggested in the univariate fixed bandwidth setting that a generous level of smoothing be used for the reference bandwidth; see Hall et al, 1992).

Figure 11 summarises the results, which take the form of a scatterplot of the chosen optimal global bandwidths at each iteration and a pair of boxplots comparing the ISEs between each 'true' intensity and the two respective estimates. Also displayed are the relative wall-clock timings.

Examining the selected bandwidths, we note overall agreement in the results. Some discrepancy in the individually chosen global bandwidths when comparing the directly obtained values to their 3D FFT counterparts is to be expected due to the discretisation of the z-axis in evaluting the trivariate convolution; however, any discrepancies appear minimal. Importantly, the chosen bandwidths are sufficiently similar such that the difference in ISE performance is also minimal. The third plot highlights the natural improvement in computational expense, with the 3D FFT method taking roughly 10% of the direct bootstrap time. (In our current environment, the direct bootstrap took nearly half an hour to complete per data set, with the 3D FFT implementation taking less than three minutes.)

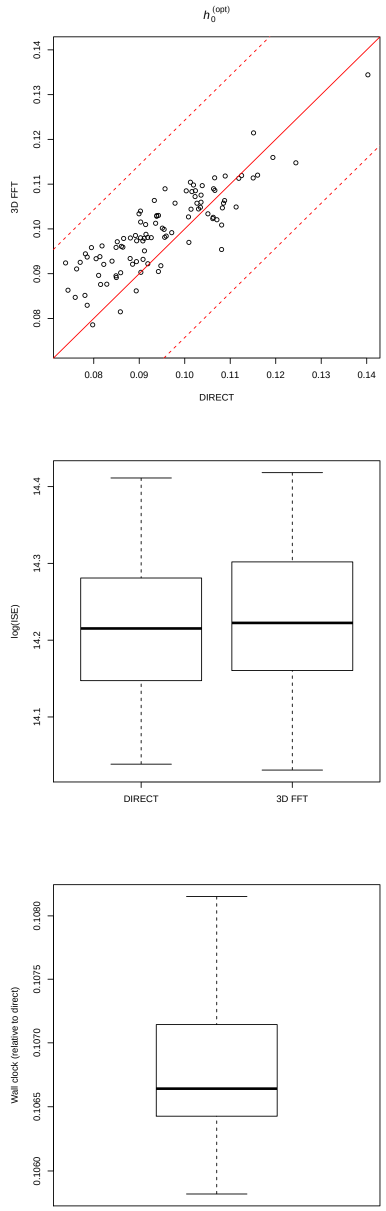


Fig. 11 Results of the naïve bootstrap simulation study. *Top:* Scatterplot of optimal global smoothing parameters selected using the 3D FFT approach against the direct approach, with a solid line denoting perfect equality; dashed lines denoting ±2 standard deviations of the direct approach. *Middle:* Boxplots of the log(ISE)s computed with respect to the true intensities generated at each iteration. *Bottom:* Relative wall-clock timings.

6 Application: Capuchin Predation Risk

For a real-world example, we turn to a novel analysis of the spatial variation in alarm calls attributed to capuchin monkeys spotting potential predatory threats in a region of Costa Rica. These data recently appeared in original work by Campos and Fedigan (2014), with one of the key objectives being the identification of possible areas where certain alarm calls occurred more than others.

6.1 Kernel Density Ratios

Campos and Fedigan (2014) used adaptive kernel density estimation to construct relative risk surfaces as per Davies and Hazelton (2010). For this example, we re-estimate one of these surfaces using an improved version of the adaptive relative risk function detailed in Davies et al (2016). In doing so, we employ the 3D FFT technique to choose a global smoothing bandwidth via bootstrapping as outlined in the previous section, followed by use of the bandwidth partitioning approach to estimate the required intensities.

The kernel log relative risk or *density ratio* function is an exploratory tool used to compare two estimated densities on $W \subset \mathbb{R}^2$. Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_X}\}$ and $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n_Y}\}$ be two point patterns observed on W. In the adaptive setting, we have

$$r(\boldsymbol{x}) = \log \frac{\hat{\lambda}_{h_0}(\boldsymbol{x}|X)/q_{h(\boldsymbol{x}|X)}(\boldsymbol{x})}{\hat{\lambda}_{h_0}(\boldsymbol{x}|Y)/q_{h(\boldsymbol{x}|Y)}(\boldsymbol{x})} + \log \frac{n_Y}{n_X}, \qquad (24)$$

where, in the same way as the intensity estimates themselves, the dependence of the edge-correction factors on the respective data sets is emphasised with the extended notation of the bandwidth function as $h(\boldsymbol{x}|\cdot)$ in the subscript.

We adopt the terminology of Bithell (1990, 1991); Kelsall and Diggle (1995) developed for applications in geographical epidemiology, and refer to X as the *case* data set, and Y as the *control* data set. A peak in resulting case/control log-ratio surface of kernel estimated densities thus represents a localised extreme over- or under-representation of cases relative to the density of the controls. Use of Abramson's adaptive smoothing regimen for the requisite densities was explored in Davies and Hazelton (2010); there are theoretical reasons for choosing the same global smoothing parameter for both numerator and denominator densities. In addition, some numerical instability can be avoided by also forcing the variable bandwidth factors for both density estimates to be identical (Davies et al, 2016). This so-called *symmetric* adaptive density ratio amounts to choosing the same pilot density estimate for both $\hat{\lambda}_{h_0}(\cdot|X)$ and $\hat{\lambda}_{h_0}(\cdot|Y)$; a situation where the need for edge correction is conveniently obviated due to cancellation of the subsequently identical corrective surfaces in the ratio.

For our purposes, we restrict attention to the subset of the alarm-calling data attributed to snakes—these will form the 'cases' X—shown in the leftmost panel of Figure 12. To adjust for the overall capuchin activity in the study region, habitat location data are also collected—the 'controls' Y—and can be seen in the middle panel of Figure 12. Noticeable is the shape formed by the at-risk habitats in comparison to the study window, which in our case we simply define as the rectangular region encapsulating the data. The region is merely clipped in a certain way in the south at approximate coordinates reflecting certain geographical impediments to habitation (Dr. F. A. Campos, personal communication). See the original work by Campos and Fedigan (2014) for a detailed account of the data.

6.2 Estimation via 3D FFT and Partitioning

The first step in implementation of the symmetric adaptive relative risk estimator is a decision regarding the common case/control pilot density which will be used to calculate the variable bandwidths. For the capuchin data, we pool the alarm and habitat observations $X \cup Y$, and use fixed bandwidth estimation with the smoothing parameter chosen via leave-one-out least-squares cross-validation. The chosen bandwidth is set at $h = 0.1516$ *km* (4 d.p.); the estimate is shown in the rightmost panel of Figure 12.

Next, we require specification of a global smoothing bandwidth h_0, common to both case and control densities. We execute the 3D FFT bootstrap approach detailed in the previous section using the pooled data, in which the reference density was a fixed bandwidth estimate with reference bandwidth $\eta = 0.2798$ *km* chosen according to Terrell (1990)'s oversmoothing factor. The bootstrap took approximately 170 seconds to complete, and an optimal bandwidth of $h_0^{(\mathrm{opt})} = 0.1613$ *km* was selected.

The case and control kernel-smoothed intensities were estimated with $h_0^{(\mathrm{opt})}$ and the aforementioned common pilot density, yielding a symmetric bandwidth function $h(\cdot|X \cup Y)$ for both. As noted, it was not necessary to employ edge-correction in the calculation of either of the two intensity functions due to cancellation of the equivalent $q_{h(\boldsymbol{x}|X \cup Y)}(\boldsymbol{x})$ factors in the ratio (24). Bandwidth quantile-partitioning was used for a target resolution of $M = N = 128$, setting $\delta = 0.01$ for 100 bandwidth bins. The wall-clock time taken to complete these two estimates totalled approximately 4 seconds.

Finally, to embellish sub-regions of potentially anomalous activity relative to the background habitat usage, asymptotically derived *tolerance contours* were computed at statistically significant 5% thresholds of both heightened and reduced risk. Theoretical details may be found in Davies and Hazelton (2010). Noteworthy is that calculation of the related p-value surfaces requires functionals that involve the edge-correction factors, regardless of symmetry of the bandwidths. Fortunately, we can make use of the partitioning approach exactly as described in Section 4.1 and demonstrated in Figure 6 to facilitate these calculations. In so doing, setting $\beta = 0.01$ for 100 bandwidth bins and without reducing the resolution of the edge-correction surfaces (i.e. keeping

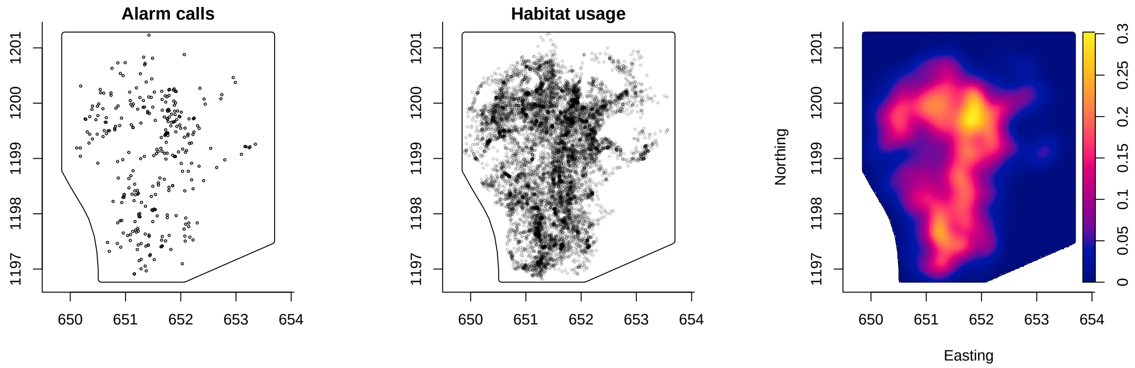


Fig. 12 Capuchin data; locations of alarm calls attributed to predatory snakes (*left*) and overall habitat usage (*middle*). On the *right* is a fixed-bandwidth kernel estimate of the pooled habitat/alarm data with smoothing parameter set via unbiased leave-one-out cross-validation, defining the function that serves as a common pilot density for calculation of the respective sets of adaptive bandwidths. Coordinates expressed in *km*.

L_M and L_N set at the target resolution $M = N = 128$), the tolerance contours took roughly 1.2 seconds to produce.

Thus, from start to finish, using the computational methods defined in the preceding sections we achieve all three of the bandwidth selection, intensity estimation and risk surface testing steps in around 175 seconds for this analysis. The topmost panel of Figure 13 shows the resulting log-risk function, with tolerance contours superimposed at the aforementioned 5% levels.

The observed risk function fluctuates over the study region, with pockets of low risk interspersed with peaks in the surface. According to the tolerance contours, there are several statistically significant low-risk areas where snake-related alarm calls appear minimal relative to the background habitat usage; with one larger area on the western border. Although there are several peaks in the surface, few appear to be flagged as significant by the contours, barring one small sub-area in the southeast, and a very small indication of another further west. Examining the top right image in Figure 3 of Campos and Fedigan (2014), we see general agreement in the appearance of the estimated risk function. One key difference with the original analysis, however, is that use of the symmetric adaptive risk estimator here has ‘dampened’ the enthusiasm of the tolerance contours when it comes to cordoning off significant fluctuations in risk, leading to a more conservative spatial test (a feature noted in Davies et al, 2016).

6.3 Direct Estimation

We repeated the analysis with direct estimation of all required functions. Bootstrap bandwidth selection was achieved in 4377 seconds and a slightly larger optimal global bandwidth of 0.1682 *km* was returned. The intensity estimation step took around 19 seconds, and tolerance contour calculation extended to approximately 48 seconds. Thus, in stark contrast to the grand total wall clock time of less than three minutes in Section 6.2, performing the entire analysis in a direct fashion requires a total time of almost one-and-a-quarter hours.

The result of direct estimation is shown in the middle panel of Figure 13. Visually, there is little difference between the direct estimate and previous one, with magnitude and spatial appearance consistent over the study region. The tolerance contours flag the same regions of significantly lower and higher risk, with some minor differences observable for the smallest subregions.

The bottom panel of Figure 13 shows the difference between the approximative and direct estimates of the log relative risk function. By and large, the differences are minor across the whole region. Although there are some small areas with larger discrepancies, these occur in areas far from any observations and against the boundary—understandably where we would expect the partition approximation to perform worse—and also where interpretation under any circumstances should be cautious at best.

7 Discussion

There are many disciplines in which observed planar data are complex in appearance. Reliable estimation of spatial structure for both exploration and modelling is typically a key goal of statistical analyses, and the flexibility of kernel smoothing has made it a popular and standard tool in any analyst’s toolbox for continuous spatial density/intensity function estimation.

The classical fixed bandwidth estimator can often be ill-equipped to handle this task, particularly when the data are highly heterogeneous (indeed, as we have seen, even moderate heterogeneity can be problematic). The data-driven adaptive estimator of Abramson (1982) can improve both theoretically and practically upon its fixed bandwidth counterpart. However, to date, the high computational demands of implementing the adaptive estimator have hampered both

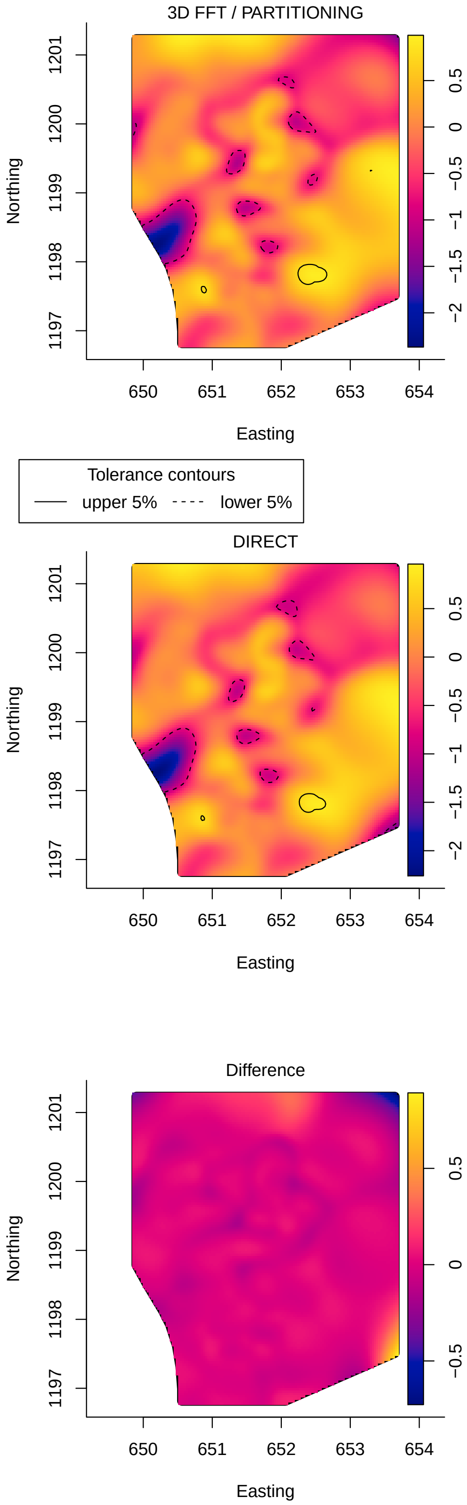


Fig. 13 Log relative risk functions for the capuchin predation data, including asymptotic tolerance contours, produced via symmetric adaptive kernel intensity estimation. *Top*: Result based on employing the novel 3D FFT and partitioning method. *Middle*: Result of direct estimation for all components. *Bottom*: Difference between the approximative and direct result.

development of additional methods of inference and the ability of applied researchers to access the existing techniques for their own analyses.

We have proposed and appraised the performance of computational alternatives to direct estimation of Abramson's adaptive kernel intensity estimator. Using the variable bandwidths assigned to each data point, we can form partitions of the data based on the bandwidth quantiles, leading to fast approximations as a sum of fixed bandwidth estimates. Our simulations reveal very satisfactory approximations to direct estimates, even using rather coarse bandwidth bins.

An alternative option involves augmenting the dimension of our planar data to include the variable bandwidths themselves along a third axis. This enables expression of a trivariate convolution for both intensity estimation and the associated adaptive edge-correction factors, in turn allowing 3D fast-Fourier evaluation of the required calculations. In fact, if we choose a particular scaling of the bandwidths, the convolution effectively yields the intensity at all global bandwidths in a prespecified interval simultaneously; individual 2D intensities subsequently available as virtually instantaneous array slices at the desired bandwidth level. The real benefit of the trivariate convolution approach therefore lies with problems where kernel estimates must be repeatedly calculated at different bandwidths, such as in the development of bandwidth selection algorithms. We implemented a computationally demanding bootstrap to explore this capability, and simulations showed excellent performance of the 3D FFT version at a dramatic reduction to computational expense when compared to direct estimation.

Finally, we brought together the 3D FFT-based bootstrap selector and partition estimation for producing an adaptive kernel spatial relative risk surface of capuchin alarm calls in a region of Costa Rica. We achieved this in around three minutes using the new computational methods, with direct estimation taking well over an hour. As the pair of simulation studies suggested, the faster estimate showed excellent agreement to the direct version. Importantly, the messages gained from an assessment of tolerance contours for statistically significant peaks and troughs in the surfaces were the same.

To motivate further research, testing, and applications utilising the adaptive estimator (some ideas follow below), we have implemented the 3D FFT technique (with edge correction) in the R language. The code is freely available in the latest versions of the `spatstat` and `sparr` packages on the Comprehensive R Archive Network (CRAN). The bandwidth partitioning method can be achieved in `sparr`.

The benefits of the computational methods discussed here are not limited to the specific techniques and applications we have described. As noted, the most immediate area of future research lies in the development of bandwidth selectors for which, thanks to the fast slice-lookup ability afforded by the 3D convolution, we are no longer required to reevaluate the planar estimate for each new global bandwidth. This opens up a range of new possibilities for data-driven bandwidth selectors; a scrutiny of a collection of classical selectors for the fixed bandwidth estimator (e.g. bootstrap, least-squares/likelihood cross-validation) is the next logical step. Further investigation from a Bayesian perspective is another option, see for example Zhang et al (2006) and Hu et al (2012).

Our methods can also be applied to a spatially anisotropic kernel, provided the global scaling parameter s is still a scalar. However, new challenges are presented when it comes to bandwidth selection for unconstrained bandwidth matrices. Recent work by Gramacki and Gramacki (2017a,b) focusses on these issues.

Finally, we can in theory easily extend the ideas to data sets of higher dimension. One particular area of interest is the estimation of *spatiotemporal* intensities and relative risk functions; see e.g. the recent work by Fernando and Hazelton (2014) and references therein. This estimator is currently defined only for a fixed bandwidth, and a natural area of future research is adaptive smoothing for the same types of problems. It is therefore imperative that computational expense is reined in, and the work here provides viable avenues for this pursuit.

Acknowledgements The authors gratefully acknowledge Fernando A. Campos and Linda M. Fedigan for providing the capuchin data and advice related to their original analysis. Two referees and an associate editor are thanked for their constructive feedback.

References

Abramson IS (1982) On bandwidth estimation in kernel estimates – a square root law. Annals of Statistics 10(4):1217–1223

Baddeley A, Rubak E, Turner R (2015) Spatial Point Patterns: Methodology and Applications with R. Chapman and Hall/CRC Press, London

Baddeley AJ, Turner R (2005) `spatstat`: An R package for analyzing spatial point patterns. Journal of Statistical Software 12(6):1–42

Bernacchia A, Pigolotti S (2011) Self-consistent method for density estimation. Journal of the Royal Statistical Society Series B 73:407–422

Bithell JF (1990) An application of density estimation to geographical epidemiology. Statistics in Medicine 9:691–701

Bithell JF (1991) Estimation of relative risk functions. Statistics in Medicine 10:1745–1751

Breiman L, Meisel W, Purcell E (1977) Variable kernel estimates of multivariate densities. Technometrics 19:135–144

Campos FA, Fedigan LM (2014) Spatial ecology of perceived predation risk and vigilance behavior in white-faced capuchins. Behavioral Ecology 25:477–486

Chaudhuri P, Marron J (2000) Scale space view of curve estimation. Annals of Statistics 28:408–428

Cooley JW, Tukey JW (1965) An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation 19:297–301

Davies TM, Hazelton ML (2010) Adaptive kernel estimation of spatial relative risk. Statistics in Medicine 29(23):2423–2437

Davies TM, Jones K, Hazelton ML (2016) Symmetric adaptive smoothing regimens for estimation of the spatial relative risk function. Computational Statistics & Data Analysis 101:12–28

Diggle P (2010) Nonparametric methods. In: Gelfand A, Diggle P, Fuentes M, Guttorp P (eds) Handbook of Spatial Statistics, CRC Press, Boca Raton, FL, chap 18, pp 299–316

Diggle PJ (1985) A kernel method for smoothing point process data. Applied Statistics 34(2):138–147

Diggle PJ, Marron JS (1988) Equivalence of smoothing parameter selectors in density and intensity estimation. Journal of the American Statistical Association 83:793–800

Duong T (2007) `ks`: Kernel density estimation and kernel discriminant analysis for multivariate data in R. Journal of Statistical Software 21(7):1–16

Duong T, Hazelton ML (2003) Plug-in bandwidth matrices for bivariate kernel density estimation. Journal of Nonparametric Statistics 15(1):17–30

Duong T, Hazelton ML (2005) Cross-validation bandwidth matrices for multivariate kernel density estimation. Scandinavian Journal of Statistics 32:485–506

Faraway JJ, Jhun M (1990) Bootstrap choice of bandwidth for density estimation. Journal of the American Statistical Association 85:1119–1122

Fernando WTPS, Hazelton ML (2014) Generalizing the spatial relative risk function. Spatial and Spatio-temporal Epidemiology 8:1–10

Gramacki A, Gramacki J (2017a) FFT-based fast bandwidth selector for multivariate kernel density estimation. Computational Statistics & Data Analysis 106:27–45

Gramacki A, Gramacki J (2017b) FFT-based fast computation of multivariate kernel density estimators with unconstrained bandwidth matrices. Journal of Computational and Graphical Statistics 26:459–462

Hall P, Marron JS (1988) Variable window width kernel density estimates of probability densities. Probability Theory and Related Fields 80:37–49

Hall P, Marron JS, Park BU (1992) Smoothed cross-validation. Probability Theory and Related Fields 92:1–20

Harris F (1978) On the use of windows for harmonic analysis with the Discrete Fourier Transform. Proceedings of the IEEE 66(1):51–83

Hu S, Poskitt DS, Zhang X (2012) Bayesian adaptive bandwidth kernel density estimation of irregular multivariate distributions. Computational Statistics & Data Analysis 56:732–740

Jones MC (1993) Simple boundary correction for kernel density estimation. Statistics and Computing 3:135–146

Jones MC, Marron JS, Sheather SJ (1996) A brief survey of bandwidth selection for density estimation. Journal of the American Statistical Association 91:401–407

Kelsall JE, Diggle PJ (1995) Kernel estimation of relative risk. Bernoulli 1:3–16

Koch K, Ohser J, Schladitz K (2003) Spectral theory for random closed sets and estimating the covariance via frequency space. Advances in Applied Probability 35:603–613

Loftsgaarden DO, Quesenberry CP (1965) A nonparametric estimate of a multivariate density function. Annals of Mathematical Statistics 36:1049–1051

Marshall JC, Hazelton ML (2010) Boundary kernels for adaptive density estimators on regions with irregular boundaries. Journal of Multivariate Analysis 101:949–963

Møller J, Syversveen AR, Waagepetersen RP (1998) Log Gaussian Cox Processes. Scandianvian Journal of Statistics 25:451–482

O'Brien TA, Kashinath K, Cavanaugh NR, Collins WD, O'Brien JP (2016) A fast and objective multidimensional kernel density estimation method: fastKDE. Computational Statistics & Data Analysis 101:148–160

Pinsky M (2002) Introduction to Fourier Analysis and Wavelets. No. 102 in Graduate Studies in Mathematics, American Mathematical Society, Providence, RI

Prince MI, Chetwynd A, Diggle PJ, Jarner M, Metcalf JV, James OFW (2001) The geographical distribution of primary biliary cirrhosis in a well-defined cohort. Hepatology 34:1083–1088

R Core Team (2017) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/

Sain SR (2002) Multivariate locally adaptive density estimation. Computational Statistics & Data Analysis 39:165–186

Sain SR, Scott DW (1996) On locally adaptive density estimation. Journal of the American Statistical Association 91:1525–1534

Sain SR, Baggerly KA, Scott DW (1994) Cross-validation of multivariate densities. Journal of the American Statistical Association 89:807–817

Silverman BW (1982) Kernel density estimation using the fast Fourier transform. Applied Statistics 31:93–99

Silverman BW (1986) Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York

Taylor CC (1989) Bootstrap choice of the smoothing parameter in kernel density estimation. Biometrika 76:705–712

Terrell GR (1990) The maximal smoothing principle in density estimation. Journal of the American Statistical Association 85:470–477

Terrell GR, Scott DW (1992) Variable kernel density estimation. Annals of Statistics 20:1236–1265

Wand MP (1994) Fast computation of multivariate kernel estimators. Journal of Computational and Graphical Statistics 3(4):433–445

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 (1995) Kernel Smoothing. Chapman and Hall, Boca Raton, USA

Zhang X, King ML, Hyndman RJ (2006) A Bayesian approach to bandwidth selection for multivariate kernel density estimation. Computational Statistics & Data Analysis 50:3009–3031