Fast Computation of Multivariate Kernel Estimators

M. P. Wand

Journal of Computational and Graphical Statistics, Vol. 3, No. 4. (Dec., 1994), pp. 433-445.

Stable URL:
http://links.jstor.org/sici?sici=1061-8600%28199412%293%3A4%3C433%3AFCOMKE%3E2.0.CO%3B2-5

Journal of Computational and Graphical Statistics is currently published by American Statistical Association.

Your use of the JSTOR archive indicates your acceptance of JSTOR's Terms and Conditions of Use, available at http://www.jstor.org/about/terms.html. JSTOR's Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at http://www.jstor.org/journals/astata.html.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

JSTOR is an independent not-for-profit organization dedicated to creating and preserving a digital archive of scholarly journals. For more information regarding JSTOR, please contact support@jstor.org.

http://www.jstor.org/
Tue Jul 25 03:12:36 2006

Fast Computation of Multivariate Kernel Estimators

M. P. WAND*

Multivariate extensions of binning techniques for fast computation of kernel estimators are described and examined. Several questions arising from this multivariate extension are addressed. The choice of binning rule is discussed, and it is demonstrated that linear binning leads to substantial accuracy improvements over simple binning. An investigation into the most appropriate means of computing the multivariate discrete convolutions required for binned kernel estimators is also given. The results of an empirical study indicate that, in multivariate settings, the fast Fourier transform offers considerable time savings compared to direct calculation of convolutions.

Key Words: Binning rules; Fast Fourier transform; Multivariate density estimation; Nonparametric regression.

1. INTRODUCTION

Multivariate kernel smoothing methods have tremendous practical potential as a simple means of recovering and highlighting structure in high-dimensional data sets, without the restrictions of parametric models; see, for example, Scott (1992). The applicability of kernel estimators is greatly enhanced by having their computational times kept to a minimum. This has motivated a considerable amount of recent research into the development of fast and efficient algorithms for their computation. Comprehensive expositions of this work are given by Fan and Marron (1994) and Seifert, Brockmann, Engel, and Gasser (1994) where it is demonstrated that, in the univariate case, fast computational methods can lead to savings of factors well into the hundreds. In this article we investigate a number of practical issues concerning the multivariate extension of one such fast kernel estimate – the binned or WARPed approximation. This is based on ideas first developed by Silverman (1982), Scott (1985), and Härdle and Scott (1992). Other early references are Härdle (1986) and Georgiev (1986).

Binned kernel estimates are usually computed over an equally-spaced mesh of *grid points*. The same ideas can be applied to obtain quickly computable approximations to kernel functional estimates, which arise in many common automatic bandwidth selection algorithms. Their calculation requires three distinct steps:

*Senior Lecturer, Australian Graduate School of Management, University of New South Wales, Kensington, NSW 2033, Australia

©1994 *American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America*

Journal of Computational and Graphical Statistics, Volume 3, Number 4, Pages 433–445

1. Bin the data by assigning the raw data to neighboring grid points to obtain *grid counts*. A grid count can be thought of as representing the amount of data in the neighborhood of its corresponding grid point.
2. Compute the required kernel weights. The fact that the grid points are equally spaced means that the number of distinct kernel weights is comparatively small.
3. Combine the grid counts and the kernel weights to obtain the approximation to the kernel estimate. This essentially involves a series of discrete convolutions.

Step 1 requires the choice of both the number of grid points and the binning rule. The first choice is quite crucial, because it represents a compromise between the computational speed of the algorithm and the approximation error due to binning. There are also several ways to bin the data (see Hall and Wand, 1993). We describe and compare the multivariate extensions of the two most popular binning rules in Section 4.

The multivariate extension of Step 2 can be accomplished quite easily, as explained in Section 2.

An important question concerns the most effective way of executing Step 3. The traditional way of computing a discrete convolution quickly is via the fast Fourier transform (FFT). However, direct computation of convolutions is a contender in the kernel estimation case because the arrays involved typically contain a high proportion of 0's. Scott (1992) presented algorithms that take advantage of this sparseness. In Section 5 we present a comparison of these approaches through an empirical study. It is demonstrated that, although there is little difference between the computational speeds of FFT and direct computation of convolution in univariate settings (Fan and Marron 1994), considerable gains can be realized by use of the FFT in multivariate settings.

Section 2 describes the essence of binned multivariate kernel estimation. Section 3 describes multivariate binning rules, and Section 4 is devoted to computation of multivariate convolutions. Section 5 contains the empirical study mentioned previously. Section 6 describes fast kernel functional estimation, and Section 7 gives discussions on implementation.

2. BINNED MULTIVARIATE KERNEL ESTIMATORS

Let $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ be a sample of pairs, where $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are $\mathbb{R}^d$-valued predictor variables having d-variate density f, and $Y_1, \ldots, Y_n$ are scalar response variables. The kernel density estimate of $f(\mathbf{x})$ and local polynomial kernel estimators of $m(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$ depend on quantities of the form

$$\hat{s}_{\mathbf{k}}(\mathbf{x}) = \sum_{i=1}^{n} (\mathbf{X}_i - \mathbf{x})^{\mathbf{k}} K_{\mathbf{h}}^{P}(\mathbf{X}_i - \mathbf{x})$$

and

$$\hat{t}_{\mathbf{k}}(\mathbf{x}) = \sum_{i=1}^{n} (\mathbf{X}_i - \mathbf{x})^{\mathbf{k}} K_{\mathbf{h}}^{P}(\mathbf{X}_i - \mathbf{x}) Y_i,$$

where $\mathbf{k} = (k_1, \ldots, k_d)$, and for a d-vector $\mathbf{u} = (u_1, \ldots, u_d)$ the convention $\mathbf{u}^{\mathbf{k}} = u_1^{k_1} \ldots u_d^{k_d}$ is used.

The notation $K_{\mathbf{h}}^P$ applies to the rescaling of the product kernel $K^P(\mathbf{x}) = K(x_1)\dots K(x_d)$ by the vector of bandwidths $\mathbf{h} = (h_1, \dots, h_d)$:

$$K_{\mathbf{h}}^P(\mathbf{x}) = K_{h_1}(x_1)\dots K_{h_d}(x_d), \tag{2.1}$$

where K is a symmetric probability density function and $K_h(x) = K(x/h)/h$ is a rescaling of K by the bandwidth $h > 0$. If K has compact support then we will let $[-\tau, \tau]$ denote the interval outside which K is 0. If K has infinite support then one could replace K by $K\mathrm{I}_{[-\tau,\tau]}$, where τ is chosen so that K is effectively 0 outside of $[-\tau, \tau]$. The choice of τ should be such that the truncation to $[-\tau, \tau]$ has negligible effect on the final estimation. For example, if K is the standard normal density, then $\tau \approx 4$ is a safe choice.

Note that $K_{\mathbf{h}}^P$ does not include all multivariate kernels of interest, such as those based on rotations of a univariate kernel (sometimes called *spherically symmetric* kernels). However, for computational purposes great savings in the number of kernel evaluations are possible if the product structure of (2.1) is available. Also not included is the possibility of smoothing in orientations different to those of the coordinate axes. This can be done by rescaling K^P by a *bandwidth matrix* and was demonstrated by Wand and Jones (1993) to be an important extension in certain circumstances. If smoothing in different orientations is desired, then it is recommended that the data be prerotated so that the product scaling (2.1) is adequate. The computations can be done on the rotated data, and then the result rotated back to correspond to the coordinates of the original data.

The simplest estimators of $f(\mathbf{x})$ and $m(\mathbf{x})$ are

$$\hat{f}(\mathbf{x}) = n^{-1}\hat{s}_{\mathbf{0}}(\mathbf{x}) \quad \text{and} \quad \hat{m}(\mathbf{x}; 0) = \hat{t}_{\mathbf{0}}(\mathbf{x})/\hat{s}_{\mathbf{0}}(\mathbf{x}).$$

The estimator $\hat{m}(\mathbf{x}; 0)$ corresponds to a local least squares constant fit, and is usually called the Nadaraya–Watson estimator. Higher-degree multivariate local polynomial estimators can be quite complicated (Ruppert and Wand 1994). Even local linear estimators require a $(d+1)\times(d+1)$ matrix inversion. For example, the local linear kernel estimator for bivariate predictor variables is

$$\hat{m}(\mathbf{x}; 1) = \mathbf{e}_1^T \begin{bmatrix} \hat{s}_{00}(\mathbf{x}) & \hat{s}_{10}(\mathbf{x}) & \hat{s}_{01}(\mathbf{x}) \\ \hat{s}_{10}(\mathbf{x}) & \hat{s}_{20}(\mathbf{x}) & \hat{s}_{11}(\mathbf{x}) \\ \hat{s}_{01}(\mathbf{x}) & \hat{s}_{11}(\mathbf{x}) & \hat{s}_{02}(\mathbf{x}) \end{bmatrix}^{-1} \begin{bmatrix} \hat{t}_{00}(\mathbf{x}) \\ \hat{t}_{10}(\mathbf{x}) \\ \hat{t}_{01}(\mathbf{x}) \end{bmatrix}.$$

Despite the increased complexity, significant gains can be realized by use of the local linear estimator compared to the Nadaraya–Watson estimator. In particular, local linear estimators are conditionally unbiased for linear m, adapt better to nonuniform designs, and exhibit superior boundary performance. For further discussion on the virtues of local linear fitting see Fan (1992), Hastie and Loader (1993), and Ruppert and Wand (1994).

For $i = 1, \dots, d$, let $g_{i1} < \dots < g_{i,M_i}$ be an equally spaced grid in the ith coordinate directions such that $[g_{i1}, g_{iM_i}]$ contains the ith coordinate values of the $\mathbf{X}$'s. Here M_i is a positive integer representing the *grid size* in direction i. Let

$$\mathbf{g}_{\mathbf{j}} = (g_{1j_1}, \dots, g_{dj_d}), \quad 1 \le j_i \le M_i, \quad i = 1, \dots, d$$

denote the grid point indexed by $\mathbf{j} = (j_1, \ldots, j_d)$ and the ith binwidth be denoted by $\delta_i = (g_{i,M_i} - g_{i1})/(M_i - 1)$. Fast binned approximations of kernel estimators involve binning the original data to obtain grid counts $(c_\mathbf{j}, d_\mathbf{j})$ that represent the amount of $(\mathbf{X}, Y)$ data near each grid point. Strategies for obtaining grid counts are described in Section 3. The binned approximation to $\hat{s}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$ is

$$\tilde{s}_\mathbf{k}(\boldsymbol{g}_\mathbf{j}) = \sum_{\ell_1=1}^{M_1} \cdots \sum_{\ell_d=1}^{M_d} (\boldsymbol{g}_\mathbf{j} - \boldsymbol{g}_\boldsymbol{\ell})^\mathbf{k} K_\mathbf{h}(\boldsymbol{g}_\mathbf{j} - \boldsymbol{g}_\boldsymbol{\ell}) c_\boldsymbol{\ell}.$$

It is easy to show that

$$\tilde{s}_\mathbf{k}(\boldsymbol{g}_\mathbf{j}) = \sum_{\ell_1=-L_1}^{L_1} \cdots \sum_{\ell_d=-L_d}^{L_d} c_{\mathbf{j}-\boldsymbol{\ell}} \kappa_{\mathbf{k},\boldsymbol{\ell}} \tag{2.2}$$

where

$$\kappa_{\mathbf{b},\boldsymbol{\ell}} = \prod_{i=1}^{d} \{K_{h_i}(\ell_i \delta_i)(\ell_i \delta_i)^{k_i}\}, \quad L_i = \min(\lfloor \tau h_i/\delta_i \rfloor, M_i - 1),$$

and $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. The expression for the binned approximation to $\hat{t}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$, denoted by $\tilde{t}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$, is the same as $\tilde{s}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$ except that the $c_\boldsymbol{\ell}$ are replaced by the $d_\boldsymbol{\ell}$. If the $\tilde{s}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$ and $\tilde{t}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$ are substituted for $\hat{s}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$ and $\hat{t}_\mathbf{k}(\boldsymbol{g}_\mathbf{j})$ in the formulas for $\hat{f}(\boldsymbol{g}_\mathbf{j})$ and $\hat{m}(\boldsymbol{g}_\mathbf{j}; p)$ then their binned approximations $\tilde{f}(\boldsymbol{g}_\mathbf{j})$ and $\tilde{m}(\boldsymbol{g}_\mathbf{j}; p)$ result.

The binned approximations $\tilde{s}_\mathbf{k}$ and $\tilde{t}_\mathbf{k}$ represent enormous computational savings, because only $\sum_{i=1}^{d} L_i$ kernel evaluations are required to obtain the $\kappa_{\mathbf{k},\boldsymbol{\ell}}$ regardless of the value of n. Once the count and kernel vectors have been obtained, they need to be combined using (2.2) to obtain the $\tilde{s}_k$ and $\tilde{t}_k$. This is a discrete convolution problem and its solution will be discussed in Section 4.

3. MULTIVARIATE BINNING RULES

The two most common univariate binning rules are *simple binning* and *linear binning*. If a data point at y has surrounding grid points at x and z, then simple binning involves assigning a unit mass to the grid point closest to y. Linear binning assigns a mass of $(z-y)/(z-x)$ to the grid point at x, and $(y-x)/(z-x)$ to the grid point at z (see Jones and Lotwick 1983). Multivariate binning rules may be defined by taking the product of univariate rules. Figure 1 gives a graphical description of how a data point $\mathbf{X}$ distributes its weight to neighboring grid points for the bivariate extension of simple and linear binning. For simple binning, the point $\mathbf{X}$ gives all of its weight to its nearest grid point, in this case the grid point at the lower left vertex of the rectangle formed by joining the four grid points neighboring $\mathbf{X}$. In the case of linear binning, the contribution from $\mathbf{X}$ is distributed among each of the four surrounding grid points according to areas of the opposite subrectangles induced by the position of the data point. Higher-dimensional extensions of simple binning and linear binning, where areas are replaced by *volumes*,

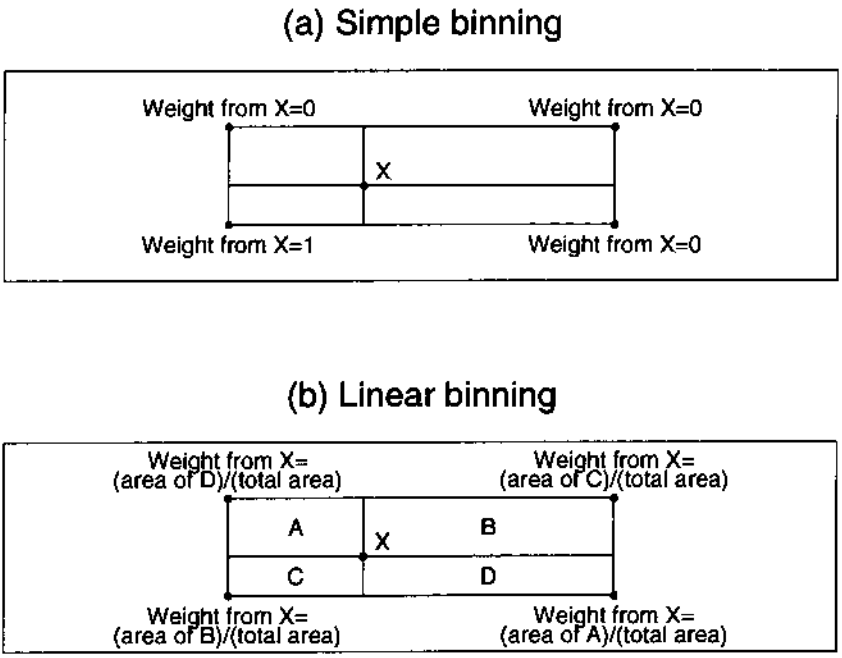

Figure 1. Graphical Representation of (a) Simple Binning and (b) Linear Binning When $d = 2$.

are obvious. Let $w_\ell(\mathbf{X})$ be the weight that $\mathbf{X}$ assigns to $\mathbf{g}_\ell$ by one of the described binning rules. Then the (c_ℓ, d_ℓ) are given by

$$c_\ell = \sum_{i=1}^{n} w_\ell(\mathbf{X}_i) \quad \text{and} \quad d_\ell = \sum_{i=1}^{n} w_\ell(\mathbf{X}_i) Y_i.$$

For both simple and linear binning the (c_ℓ, d_ℓ) can be computed using a fast $O(n)$ algorithm by extending the "integer division" idea of Fan and Marron (1994).

Two obvious questions that arise are:

1. How do simple and linear binning compare?
2. How many bins should one use in each direction?

Question 1 is partially answered by asymptotic results of Hall and Wand (1993). A straightforward extension of their arguments leads to, for constants A_i and B_i,

$$\begin{aligned} E\{\tilde{s}_{\mathbf{k}}(\mathbf{x}) - \hat{s}_{\mathbf{k}}(\mathbf{x})\}^2 &= \textstyle\sum_{i=1}^{d} A_i \delta_i^2 + o\left(\sum_{i=1}^{d} \delta_i^2\right) \quad \text{for simple binning} \\ &= \textstyle\sum_{i=1}^{d} B_i \delta_i^4 + o\left(\sum_{i=1}^{d} \delta_i^4\right) \quad \text{for linear binning,} \end{aligned}$$

as $\delta_i \to 0$, $i = 1, \ldots, d$. Therefore, in terms of how well the $\tilde{s}_{\mathbf{k}}$ approximate the $\hat{s}_{\mathbf{k}}$, linear binning is an order of magnitude better than simple binning. Analogous results hold for $E\{\tilde{t}_{\mathbf{k}}(\mathbf{x}) - \hat{t}_{\mathbf{k}}(\mathbf{x})\}^2$.

It is impossible to give an absolute answer to Question 2 because functions with finer structure require more grid points to achieve a given level of accuracy. Insight into the effects of binning on accuracy can only be realized through examples. Table 1 shows

Table 1. Minimum Number of Bins in Each Direction to Achieve RMISE≈1%

	n = 100		n = 1,000		n = 10,000	
d	Simple	Linear	Simple	Linear	Simple	Linear
2	32	15	46	22	67	32
3	33	14	45	20	62	27
4	34	14	45	18	59	24

that minimum grid size $M = M_i$ in each direction is required to achieve an approximate relative mean integrated squared error (RMISE) of 1% for estimation of the $N(\mathbf{0}, \mathbf{I}_d)$ density over $[-3,3]^d$. Here RMISE is defined to be

$$E\int_{\mathbf{R}^d}\{\tilde{f}(\mathbf{x}) - \hat{f}(\mathbf{x})\}^2 d\mathbf{x} \Big/ E\int_{\mathbf{R}^d}\{\hat{f}(\mathbf{x}) - f(\mathbf{x})\}^2 d\mathbf{x} = \frac{\text{error due to binning}}{\text{total estimation error}}.$$

To produce Table 1 we replaced the numerator of RMISE by the leading term of the small delta approximation derived in Section 5 of Hall and Wand (1993). The denominator was replaced by the large sample approximation to the mean integrated squared error of a multivariate density estimator (see Wand and Jones, 1993) and the bandwidth minimizing this quantity, $h_i = h = [4/\{(d+2)n\}]^{1/(d+4)}$, $i = 1, \ldots, d$, was used in both numerator and denominator. See the Appendix for the derivation of results required for Table 1.

In this case the number of bins in each direction does not change very much between dimensions. However, remember that the total number of grid points is equal to M^d, so there is a big cost for higher dimensionality. Another point to note is that the normal density is a function with comparatively little structure, so the numbers in Table 1 represent an approximate lower bound on the number of grid points that one should use in each direction to achieve a 1% approximate RMISE. Higher numbers are necessary for more complex functions.

The results in Table 1 also provide a practical answer to Question 1, where it is seen that linear binning requires about half as many bins in each direction to achieve a given level of accuracy. The savings of linear binning can be enormous. For example, a 1% approximate RMISE is achieved for estimation of the three-dimensional normal density using linear binning with $27^3 = 19,683$ grid points, while $62^3 = 238,328$ grid points are required to achieve the same accuracy using simple binning.

Figure 2 shows contour plots of kernel density estimates based on 640 longitude/latitude pairs of the epicenters of earthquakes in Mount Saint Helens region. These data have been analyzed by O'Sullivan and Pawitan (1993). The values of $M_1 = M_2 = M$ are (a) 25, (b) 50, and (c) 100. For $M = 25$ the contours have a granular appearance and the binned estimator is slightly biased compared to the unbinned estimator. The difference between the $M = 50$ and $M = 100$ density estimates is marginal, but the $M = 100$ estimate is slightly smoother. For display purposes, about 50–70 grid points in each direction would be adequate for this density estimate.

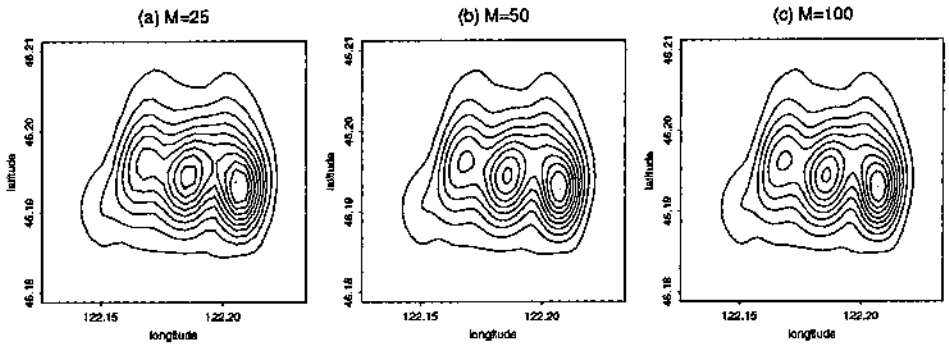

Figure 2. Binning Kernel Density Estimates of Mount Saint Helens Data for (a) M_i = 25, (b) M_i = 50, (c) M_i = 100.

4. COMPUTATION OF MULTIVARIATE CONVOLUTIONS

We now investigate the problem of efficient computation of multivariate discrete convolutions of the type given by (2.2). Scott (1992, p. 121) gives an efficient algorithm for computation of such quantities in the bivariate context. An alternative approach is to use the fast Fourier transform (FFT). It has the advantage of requiring only $O(M_1 \log M_1 \ldots M_d \log M_d)$ operations compared to the $O(M_1^2 \ldots M_d^2)$ operations required for direct computation of (2.2). We will describe how the FFT can be used to compute (2.2), starting with the case $d = 1$.

The *discrete Fourier transform* of a complex vector $\mathbf{z} = (z_0, \ldots, z_{N-1})$ is the vector $\mathbf{Z} = (Z_0, \ldots, Z_{N-1})$, where

$$Z_j = \sum_{\ell=0}^{N-1} z_\ell e^{2\pi i \ell j/N}, \quad j = 0, \ldots, N-1,$$

and i is the square root of -1. The vector $\mathbf{z}$ can be recovered from its Fourier transform $\mathbf{Z}$ by applying the *inverse discrete Fourier transform* formula

$$z_\ell = N^{-1} \sum_{j=0}^{N-1} Z_j e^{-2\pi i \ell j/N}, \quad \ell = 0, \ldots, N-1.$$

If N is a highly composite number, such as a power of 2, then discrete Fourier transforms and their inverses can be computed in $O(N \log N)$ operations using the FFT algorithm (Cooley and Tukey 1965). The discrete convolution of two vectors can be computed quickly using the FFT by appealing to the *Discrete Convolution Theorem*: multiply the Fourier transforms of the two vectors element-by-element and then invert the result to obtain the convolution vector (see Press, Flannery, Teukolsky, and Vetterling 1988, pp. 408–411). However, this theorem requires certain periodicity assumptions, so when these assumptions are violated appropriate *zero-padding* is required to avoid *wrap-around* effects. We will now give a description of what this entails for the FFT computation of the univariate convolution

$$\tilde{s}_k(g_j) = \sum_{\ell=-L_1}^{L_1} c_{j-\ell} \kappa_{k,\ell}, \quad j = 1, \ldots, M_1.$$

Let P be a highly composite number such that $P \geq M_1 + L_1$ and let $\mathbf{0}_p$ denote a vector of p zeroes. Define the "zero-padded" vectors

$$\mathbf{c}^Z = (c_1, \ldots, c_{M_1}, \mathbf{0}_{P-M_1})$$

and

$$\boldsymbol{\kappa}_k^Z = (\kappa_{k,0}, \kappa_{k,1}, \ldots, \kappa_{k,L_1}, \mathbf{0}_{P-2L_1-1}, (-1)^k \kappa_{k,L_1}, (-1)^k \kappa_{k,L_1-1}, \ldots, (-1)^k \kappa_{k,1}),$$

which are each vectors of length P. The zero-padding on the right end of the c_ℓ's is to account for wrap-around effects and the $\boldsymbol{\kappa}_k$ vector lists the $\kappa_{k,\ell}$'s in wrap-around order with appropriate zero-padding in the interior. Let $\mathbf{C}^Z$ and $\mathcal{K}_k^Z$ be the discrete Fourier transforms of $\mathbf{c}^Z$ and $\boldsymbol{\kappa}_k^Z$ respectively (computed using the FFT) and let $\tilde{\mathbf{S}}_k$ be the element-wise product of $\mathbf{C}$ and $\mathcal{K}_k$. Then the first M entries of the inverse FFT of $\tilde{\mathbf{S}}_k$ are equal to $\tilde{s}(g_j)$, $j = 1, \ldots, M_1$.

The extension of this idea to general dimensions is relatively straightforward. To give the flavor of what is involved, we will give a brief description of the FFT computation of the bivariate convolution required for computation of the $\tilde{s}_{00}(g_{j_1 j_2})$. For $i = 1, 2$ let P_i be a highly composite integer exceeding $M_i + L_i$. Also, let $\mathbf{c} = [c_{\ell_1 \ell_2}]$ be the $M_1 \times M_2$ matrix of counts and let $\mathbf{0}$ denote a generic matrix of zeroes. To make the notation less cumbersome we will write $\kappa_{00,\ell_1\ell_2}$ as $\kappa_{\ell_1\ell_2}$. Then appropriate zero-padded matrices are

$$\mathbf{c}^Z = \begin{bmatrix} \mathbf{c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\kappa}_{00}^Z = \begin{bmatrix} \kappa_{00} & \cdots & \kappa_{0L_2} & & \kappa_{0L_2} & \cdots & \kappa_{01} \\ \vdots & \ddots & \vdots & \mathbf{0} & \vdots & \ddots & \vdots \\ \kappa_{L_1 0} & \cdots & \kappa_{L_1 L_2} & & \kappa_{L_1 L_2} & \cdots & \kappa_{L_1 1} \\ & \mathbf{0} & & \mathbf{0} & & \mathbf{0} & \\ \kappa_{L_1 0} & \cdots & \kappa_{L_1 L_2} & & \kappa_{L_1 L_2} & \cdots & \kappa_{L_1 1} \\ \vdots & \ddots & \vdots & \mathbf{0} & \vdots & \ddots & \vdots \\ \kappa_{10} & \cdots & \kappa_{1L_2} & & \kappa_{1L_2} & \cdots & \kappa_{11} \end{bmatrix},$$

where the dimensions of the $\mathbf{0}$ matrices are chosen to ensure that both $\mathbf{c}^Z$ and $\boldsymbol{\kappa}_{00}^Z$ are $P_1 \times P_2$ matrices. Notice the mirror imaging of the $\kappa_{\ell_1\ell_2}$ values in the construction of $\boldsymbol{\kappa}_{00}^Z$. One should apply the FFT to each of $\mathbf{c}^Z$ and $\boldsymbol{\kappa}^Z$ and take the element-wise product of the results. The $M_1 \times M_2$ submatrix in the upper left corner of the inverse FFT of this product contains values of $\tilde{s}_{00}(g_{j_1 j_2})$, $j_i = 1, \ldots, M_i$, $i = 1, 2$, as defined by (2.2) in the case $d = 2$, $k = (0, 0)$.

5. SPEED COMPARISONS

Fan and Marron (1994) argued that, for typical grid sizes such as $M_i = 400$, there is not much difference between the times required for direct and FFT-based computation of *univariate* convolutions. However, because the asymptotic dominance of the FFT is greater in higher dimensions we might expect it to allow greater savings for more practical grid sizes. To test this theory, speed comparisons were performed for computation of a d-variate kernel density estimate for $d = 2$ and $d = 3$. The computations were performed over $[-3, 3]^d$ based on samples of size n =100, 1,000, and 10,000 of standard d-variate

Table 2. Average Discrete Convolution Computation Times (standard deviations)

	Bivariate density estimation								
	n = 100			*n = 1,000*			*n = 10,000*		
M	*FFT*	*Direct*	*Ratio*	*FFT*	*Direct*	*Ratio*	*FFT*	*Direct*	*Ratio*
25	.80	.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	(.42)	(.42)		(.00)	(.00)		(.00)	(.00)	
50	1.40	1.90	1.40	1.50	2.50	1.70	1.30	2.30	1.80
	(.52)	(.32)		(.53)	(.53)		(.48)	(.67)	
100	4.00	6.10	1.50	3.40	17.20	5.10	3.50	19.40	5.50
	(.00)	(.32)		(.52)	(1.14)		(.53)	(.97)	
	Trivariate density estimation								
	n = 100			*n = 1,000*			*n = 10,000*		
M	*FFT*	*Direct*	*Ratio*	*FFT*	*Direct*	*Ratio*	*FFT*	*Direct*	*Ratio*
25	8.50	18.60	2.20	8.60	26.00	3.00	8.30	30.70	3.70
	(.52)	(1.26)		(.70)	(1.25)		(.48)	(.67)	

normal observations. The kernel was the d-variate standard normal and the bandwidth was $h_i = h = [4/\{(d+2)n\}]^{1/(d+4)}$, which minimize asymptotic mean integrated square error for this setting. Grids for which $M = M_i = 25, 50$, and 100 were considered for the bivariate case but, due to storage restrictions, only $M = M_i = 25$ was used for the trivariate case.

For each setting and grid, 10 replications were performed and the kernel evaluation and convolution stages were timed, where the convolutions were computed using

1. The FFT-based algorithm described in the previous section. The FFTs were computed using the `S-PLUS fft()` function, and zero padding to matrices having dimensions equal to powers of 2 was used. For the bandwidths used in this study, the next highest power of 2 was sufficient.
2. Direct computation of convolution using FORTRAN implementation of two- and three-dimensional versions of the algorithm given in Scott (1992, p. 121). This is a very efficient direct convolution algorithm because it takes advantage of the fact that a high proportion of the grid counts and kernel weights are 0.

In all cases linear binning was used to obtain the grid counts. The computations were performed on the author's RS6000/220 Powerstation and times were recorded according to the elapsed time component of the `unix.time` function of `S-PLUS` (see Statistical Sciences, Inc. 1991). For each setting the average times and their ratios are shown in Table 2. The ratios are a better way of comparing the two computational methods, because they are less dependent on changes in computing environments. Nevertheless, the average times themselves give a real life aspect to the problem in that they indicate how long a user would have to wait for a picture of the estimate to appear on the screen in 1994 using a typical computing environment.

In each case the FFT exhibits equal or faster computation of convolutions than the direct approach, and can be as much as five times faster for the settings considered here. There is not much difference between the two methods for small grid sizes, but the FFT offers substantial improvements for finer grids. Use of the FFT comes at the cost of

careful zero-padding being required, but if speed is of major concern then it seems clear that FFT computation of convolutions is superior to direct computation for multivariate kernel estimation.

6. FUNCTIONAL ESTIMATION

The estimation of certain functionals of density and regression functions has been a topic of considerable research effort in recent years. For example, the problem of estimating the functional $\int f(\mathbf{x})^2\,d\mathbf{x}$, for a density f, is of importance to nonparametric rank statistics (see Sheather, Hettmansperger, and Donald in press). In the nonparametric multivariate density estimation context, data-driven rules for bandwidth selection rely on estimation of functionals of the form

$$\psi_{\mathbf{m}} = \int f^{(\mathbf{m})}(\mathbf{x}) f(\mathbf{x})\,d\mathbf{x}, \tag{6.1}$$

where

$$f^{(\mathbf{m})}(\mathbf{x}) = \frac{\partial^{m_1+\cdots+m_d}}{\partial x_1^{m_1}\ldots\partial x_d^{m_d}} f(\mathbf{x}).$$

This includes multivariate versions of least squares cross-validation (Stone 1984), biased cross-validation (Sain, Baggerly, and Scott in press) and plug-in bandwidth selection (Wand and Jones 1994). Similar bandwidth selection rules can be developed for multivariate kernel regression, but because this has not yet been done we will focus on the problem of estimating (6.1). The natural kernel estimate of $\psi_{\mathbf{m}}$ is

$$\hat{\psi}_{\mathbf{m}}(\mathbf{h}) = n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n} K_{\mathbf{h}}^{(\mathbf{m})}(\mathbf{X}_i - \mathbf{X}_j). \tag{6.2}$$

The first thing to note about (6.2) is that it involves a double summation over n. Therefore $O(n^2)$ operations are required for its direct computation, which can be prohibitively high for even moderate values of n. However, this can be overcome by using an approximation based on the bin counts $c_{\boldsymbol{\ell}}$ to obtain

$$\tilde{\psi}_{\mathbf{m}}(\mathbf{h}) = n^{-2}\sum_{j_1=1}^{M_1}\cdots\sum_{j_d=1}^{M_d} c_{\mathbf{j}}\left(\sum_{\ell_1=-L_1}^{L_1}\cdots\sum_{\ell_d=-L_d}^{L_d} c_{\mathbf{j}-\boldsymbol{\ell}}\kappa_{\boldsymbol{\ell}}^{(\mathbf{m})}\right),$$

where

$$\kappa_{\boldsymbol{\ell}}^{(\mathbf{m})} = \prod_{i=1}^{d}\{K_{h_i}^{(m_i)}(\ell_i\delta_i)\}.$$

The summation inside the brackets is a multivariate convolution, so it can be handled in the same way as for $\tilde{s}_{\mathbf{k}}(\boldsymbol{g}_{\mathbf{j}})$, described in the previous section. The resulting array only needs to be multiplied by the corresponding grid counts and summed to obtain the estimate.

7. COMPUTING PRACTICALITIES

The actual programming and execution of binned multivariate kernel estimates leads to several new questions concerning the utility of the programming language at hand. Most of the programming for the examples and comparisons in this article were done using S-SPLUS, which has the desirable features of multidimensional arrays and a built-in FFT routine called `fft`. The main exception is the direct convolution routine, which was programmed in FORTRAN because looping in this language is considerably faster.

Storage space is another practical concern, because multidimensional arrays can be quite large. If using the FFT to compute convolutions, then careful choice of the initial grid-sizes is recommended to ensure that zero-padding is not excessive. This is mainly due to the restriction of the zero-padded arrays requiring highly composite dimensions. For example, if powers of 2 are used for the zero-padded arrays, then having an initial grid size of 60×60 is unwise because 60 is very close to its next highest power of 2 and there is a good chance that 128×128 zero-padded arrays will be required, after accounting for wrap-around effects. In general, one should ensure that each M_i is chosen such that $M_i + \lfloor \tau h_i/\delta_i \rfloor$ does not exceed the next power of 2 above M_i.

S-PLUS/FORTRAN code for bivariate and trivariate kernel estimation is available by request from the author (e-mail: wand@agsm.unsw.edu.au).

APPENDIX: DERIVATION OF RESULTS REQUIRED FOR TABLE 1

Suppose that both f and K are both equal to the $N(\mathbf{0}, \mathbf{I}_d)$ density. Then, because f and K are spherically symmetric, we can take $\delta_i = \delta$ and $h_i = h$, $i = 1, \ldots, d$. Let ϕ be the univariate standard normal probability density function and unqualified integrals be taken over the whole space. Results given in Section 5 of Hall and Wand (1993) lead to (as $\delta \to 0$)

$$E \int \{\tilde{f}(\mathbf{x}) - \hat{f}(\mathbf{x})\}^2 \, d\mathbf{x} \approx \frac{1}{12n} d\delta^2 \int (\phi')^2 \left(\int \phi^2 \right)^{d-1} h^{-d-2}$$

for simple binning, and

$$E \int \{\tilde{f}(\mathbf{x}) - \hat{f}(\mathbf{x})\}^2 \, d\mathbf{x} \approx d\delta^4 \left\{ \frac{1}{120nh^{d+4}} + \frac{1 - n^{-1}}{144(1 + h^2)^{(d+4)/2}} \right\} \int (\phi'')^2 \left(\int \phi^2 \right)^{d-1}$$

for linear binning. Standard large sample results from density estimation (see Wand and Jones 1993) show that, as $h \to 0$, $nh \to \infty$ and $n \to \infty$,

$$\inf_{h>0} E \int \{\hat{f}(\mathbf{x}) - f(\mathbf{x})\}^2 \, d\mathbf{x} \approx (4\pi)^{-d/2} \{(d+4)/4\} \{(d+2)/(4n)\}^{4/(d+4)}$$

with the asymptotically optimal h equal to $[4/\{(d+2)n\}]^{1/(d+4)}$. These results can be now substituted into the RMISE formula. Direct algebra then shows that the minimum value of M required to achieve an approximate $100\alpha\%$ RMISE for estimation of the

normal density over $[-3,3]^d$ is

$$M = \left\lceil 1 + 6\left\{\frac{dn^{2/(d+4)}4^{6/(d+4)}(d+2)^{(d-2)/(d+4)}}{24(d+4)\alpha}\right\}^{1/2}\right\rceil$$

(where $\lceil x \rceil$ is the smallest integer greater than or equal to x) for simple binning, and

$$M = \left\lceil 1 + 6\left\{\frac{d(4n)^{4/(d+4)}\left(3(d+2) + \frac{10n(d+2)}{[4^{2/(d+4)} + \{n(d+2)\}^{2/(d+4)}]^{(d+4)/2}}\right)}{480(d+4)(d+2)^{4/(d+4)}\alpha}\right\}^{1/4}\right\rceil$$

for linear binning. The values of M in Table 1 are obtained from these formulas by setting $\alpha = .01$, $n = 100$, 1,000, and 10,000 and $d = 2, 3$, and 4.

[*Received March 1993. Revised April 1994.*]

ACKNOWLEDGMENTS

The author is grateful to Jianqing Fan, Steve Marron, David Scott and Berwin Turlach for their very helpful comments during the course of this research. Programming assistance from Eugene Dubossarsky is also gratefully acknowledged.

REFERENCES

Cooley, J. W., and Tukey, J.W. (1965), "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, 19, 297–301.

Fan, J. (1992), "Design-Adaptive Nonparametric Regression," *Journal of the American Statistical Association*, 87, 998–1004.

Fan, J., and Marron, J. S. (1994), "Fast Implementations of Nonparametric Curve Estimators," *Journal of Computational and Graphical Statistics*, 3, 35–56.

Georgiev, A. A. (1986), "A Fast Algorithm for Curve Fitting," in *COMPSTAT: Proceedings in Computational Statistics*, eds. F. de Antoni, N. Lauro and A. Rizzi, Vienna: Physica-Verlag, pp. 97–101.

Hall, P., and Wand, M.P. (1993), "On the Accuracy of Binning Approximations to Kernel Density Estimators," *Working Paper Series 93–003*, University of New South Wales, Australian Graduate School of Management.

Härdle, W., and Scott, D. W. (1992), "Smoothing by Weighted Averaging of Rounded Points," *Computational Statistics*, 7, 97–128.

Hastie, T. J., and Loader, C. (1993), "Local Regression: Automatic Kernel Carpentry" (with discussion), *Statistical Science*, 8, 120–143.

Jones, M. C., and Lotwick, H. (1983), "On the Errors Involved in Computing the Empirical Characteristic Function," *Journal of Statistical Computing and Simulation*, 17, 133–149.

O'Sullivan, F., and Pawitan, Y. (1993), "Multivariate Density Estimation by Tomography," *Journal of the Royal Statistical Society*, Ser. B, 55, 509–521.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W.T. (1988), *Numerical Recipes: The Art of Scientific Computing*, Cambridge, U.K.: Cambridge University Press.

Ruppert, D., and Wand, M.P. (1994), "Multivariate Locally Weighted Least Squares Regression," *Annals of Statistics*, 22, in press.

Sain, S. R., Baggerly, K. A., and Scott, D. W. (in press), "Cross-Validation of Multivariate Densities," *Journal of the American Statistical Association*, 89.

Scott, D. W. (1985), "Average Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions," *The Annals of Statistics*, 13, 1024–1040.

——— (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley.

Seifert, B., Brockmann, M., Engel, J., and Gasser, Th. (1994), "Fast Algorithms for Nonparametric Curve Estimation," *Journal of Computational and Graphical Statistics*, 3, 192–213.

Sheather, S. J., Hettmansperger, T. P., and Donald, M. R. (in press), "Data-Based Bandwidth Selection for Kernel Estimators of the Integral of $f^2(x)$," *Scandinavian Journal of Statistics*, 21.

Silverman, B. W. (1982), "Kernel Density Estimation Using the Fast Fourier Transform," *Applied Statistics*, 31, 93–97.

Statistical Sciences, Inc. (1991), *S-PLUS Reference Manual*, Seattle, WA: author.

Stone, C. J. (1984), "An Asymptotically Optimal Window Selection Rule for Kernel Density Estimates," *Annals of Statistics*, 12, 1285–1297.

Wand, M. P., and Jones, M. C. (1993), "Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation," *Journal of the American Statistical Association*, 188, 520–528.

——— (1994), "Multivariate Plug-in Bandwidth Selection," *Computational Statistics.*, 9, 97–116.