A practical implementation of weighted kernel density estimation for handling shape constraints

Mark Anthony Wolters\textsuperscript{a}\* and Willard John Braun\textsuperscript{b}

Received 17 July 2018; Accepted 20 August 2018

The weighted kernel density estimator is an attractive option for shape-restricted density estimation, because it is simple, familiar, and potentially applicable to many different shape constraints. Despite this, no reliable software implementation has appeared since the method's original proposal in 2002. We found that serious numerical and practical difficulties arise when attempting to implement the method. We overcame these difficulties and in the process discovered that the weighted method and our own recently proposed method—controlling the shape of a kernel density using an adjustment curve—can be unified in a single computational framework. This article describes our findings and introduces the R package \texttt{scedensity}, which can be used to easily obtain density estimates that are unimodal, bimodal, symmetric, and more. © 2018 The Authors. \textit{Stat} Published by John Wiley & Sons Ltd

Keywords: shape restrictions; statistical computing; symmetric density estimation; unimodality; quadratic programming

\section{Introduction}

Shape-constrained probability density estimation has remained a topic of interest in the statistical literature since the work of Grenander (1956) on the estimation of unimodal densities. A variety of approaches to obtaining shape-restricted estimates have been attempted, and a variety of different constraints have been considered. Important progress has been made in recent years, particularly on the constraint of log-concavity (Dümbgen & Rufibach, 2009; Birke, 2009; Cule et al., 2010; Samworth, 2017), but also in other directions, including the use of exponential epi-splines to handle several different types of constraints (Royset & Wets, 2015), or Bernstein polynomials to handle unimodality (Turnbull & Ghosh, 2014; Papp & Alizadeh, 2014).

Much of the work to date has rightfully focused on statistical properties of shape-restricted estimators. In the present work, we take the statistical benefits of enforcing constraints as a given and instead focus on questions of implementation and real-world usability. We are concerned with the univariate setting, where exploratory analyses and visualization are major uses of density estimation. Shape-restricted non-parametric estimators have much to offer in this context, as they allow more flexibility than do parametric forms while reducing the visually distracting effects of sampling variability that can appear in ordinary non-parametric estimates.

\textsuperscript{a}Shanghai Center for Mathematical Sciences, Fudan University, 220 Handan Road, Shanghai 20433, China
\textsuperscript{b}Irving K. Barber School of Arts and Sciences, University of British Columbia, 3333 University Way, Kelowna, British Columbia, Canada, V1V 1V7
\textsuperscript{*Email: mark@mwolters.com

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
To make shape-restricted density estimation more accessible to practitioners, an ideal method would be (i) conceptually simple and familiar, (ii) numerically stable and robust, and (iii) able to accommodate a variety of different shape constraints. We believe that modifying a kernel density estimator (KDE) is a good approach to meeting these goals. Until now, however, a user-friendly software implementation for obtaining shape-constrained KDEs has been missing. Our recently released package scdensity (Wolters, 2018) for the R computing environment (R Core Team, 2018) provides such an implementation.

Our software package includes two approaches to shape-constrained kernel density estimation: the approach of Hall & Huang (2002), which we call the weighted KDE method, and our own approach (Wolters & Braun, 2018), called the adjusted KDE method. The first approach is the simpler of the two, but it has numerical and conceptual limitations that make it impractical for regular use. The second approach avoids the limitations of the first, but at the cost of some additional complexity in its construction.

We were able to overcome the limitations of the weighted KDE method, while eliminating the extra complexity of the adjusted KDE. As a result, the data analyst has a previously unavailable capability: to obtain a shape-constrained KDE, with a variety of possible constraints, in one line of code, just as easily as one can obtain a standard unconstrained estimate. The scdensity package can handle unimodality, bimodality, tail monotonicity, bounded support, constraints on the number of inflection points, and symmetry, as well as many of these in combination.

In this article, we describe the main results that form the basis of the package. In Section 2, the weighted and adjusted KDE methods are briefly reviewed, followed by an example. The example illustrates an important innovation of this work: an adaptive binning preprocessing step that makes the two methods’ optimization problems equivalent. Section 3 gives a detailed description of the implementation, including the formulation of the optimization problem, the binning procedure, and how the optimization was made numerically robust. Section 4 provides demonstrations of the package’s capabilities, and Section 5 contains additional technical notes.

## 2 Overview

In this section, we introduce our notation, describe the two estimation approaches, and illustrate the methods on the well-known chondrite data set (Ahrens, 1965; Good & Gaskins, 1980).

### 2.1 Notation

Let \( x \) be the observed data, a random sample of \( n \) scalar observations. The standard KDE is formed with \( n \) kernel functions, one centred at each data point. In developing our estimation procedure, we think of the KDE as just a convex combination of kernel functions, the number and locations of which are not directly related to \( x \). This slightly generalized version of the KDE can be written as

\[
  f_s(x | p) = \frac{1}{n} \sum_{j=1}^{r} \rho_j K \left( \frac{x - s_j}{h} \right),
\]

where \( K \) (the kernel function) is a probability density symmetric around zero, \( h \) is a positive scalar bandwidth, and \( p = [\rho_1, \ldots, \rho_r]^T \) is a vector of probability weights. The elements of \( s = [s_1, \ldots, s_r]^T \) are the kernel centres that determine the placement of the kernel functions. The standard (uniform-weighted) KDE is \( f_x(x | p_{\text{unif}}) \), where \( p_{\text{unif}} \equiv \frac{1}{n} 1 \), and \( 1 \) is a vector of ones.

We can write \( f_s(x | p) = k_s(x)^T p \), where \( k_s(x) = [k_s(x), \ldots, k_s(x)]^T \), and \( k_s(x) = \frac{1}{h} K \left( \frac{x - s_j}{h} \right) \). Let \( k'_s(x) \) and \( k''_s(x) \) be defined similarly, but using the first derivative of \( K \). Then \( f'_s(x | p) = k'_s(x)^T p \). Higher derivatives are defined in the same way.
2.2 The weighted KDE method

The weighted KDE method involves selecting $p$ to minimize the integrated squared error (ISE) between $f_x(x \mid p)$ and $f_x(x \mid p_{unif})$, subject to shape constraints on $f_x(x \mid p)$ (Hall & Huang, 2002, considered several alternatives before concluding that ISE was the best choice). The optimal weights are the solution to a quadratic program (QP).

Despite its promise as a modified KDE estimable using quadratic programming, the method has several significant problems:

- **Definiteness.** It is often the case that the matrix of the QP is ill conditioned, with many eigenvalues nearly zero, possibly negative. This makes the QP much more difficult to solve, as strict convexity cannot be assumed. We were not able to reliably solve the problem using any of the freely available QP alternatives available in R. In their work, Hall & Huang (2002) used a sequential quadratic programming (SQP) routine from the NAG optimization library (The Numerical Algorithms Group, 2018). This is a proprietary, industrial-grade library, and SQP is designed to solve non-convex problems.

- **Feasibility.** It is not guaranteed that feasible solutions will exist for a given combination of data set and shape constraint. Infeasible problems can easily arise owing to lack of data in a particular location. For example, requiring a unimodal estimate when the data are actually bimodal could lead to such a situation.

- **Efficiency.** As the sample size grows, the method becomes computationally inefficient. The dimension of the QP is $n$, because there is one weight per data point. Ill conditioning is also more likely to occur as $n$ grows and $x$ begins to contain more points that are very close together.

- **Realism.** The constraints may force the optimal estimate to directly contradict the observed data, by assigning a probability mass of zero to an interval that contains a data point. For example, this can happen in the tails of a unimodal estimate. The only way to avoid introducing a second mode may be to give a weight of zero to points in the tail.

The definiteness and feasibility problems are critical and must be addressed to provide a robust implementation. The efficiency and realism problems impact the usability of the method.

2.3 The adjusted KDE method

The adjustment curve approach involves modifying $f_x(x \mid p_{unif})$ not by changing its weights but rather by adding a function to it. The estimator is

$$
\hat{f}_x(x) = k_x(x)^T p_{unif} + \psi(x)^T a,
$$

where $\psi(x)$ is a vector of $m$ “adjustment densities” evaluated at $x$. The coefficients $a$ can be negative but are constrained to sum to zero to ensure the resulting estimate is a density. The estimator itself is also constrained to be non-negative. The optimal coefficients are again the solution to a QP.

The structure of $\psi(x)$ is specified independently of the data. For example, $m \neq n$ adjustment densities can be placed on a grid covering the support of the estimate. This strategy avoids the feasibility and efficiency problems experienced by the weighted KDE, by decoupling the adjustment curve from $n$. It also allows the shape of $f_x$ to change even in regions where data are scarce, which resolves the realism problem. The trade-off for these improvements is some additional complexity, because the structure of $\psi(x)$ needs to be specified.

When the adjustment densities are made equal to the kernel functions ($\psi(x) = k_x(x)$), the adjustment curve approach becomes equivalent to the weighted KDE method. The adaptive binning procedure, discussed later, automatically
sets up adjustment densities that can exploit this connection, thereby eliminating the complexity trade-off. In the scdensity package, the adjustment curve is the default estimation method.

### 2.4 Illustration: chondrite data

The overall approach is illustrated in Figure 1, using the chondrite data. The data are measurements of the silica percentage in 22 meteorites. Minnotte (1997) and Hall & Huang (2002) applied Gaussian kernel density estimation to these data, with a bandwidth of 0.7, producing an estimate with three modes (shown in Figure 1(a)). Hall & Ooi (2004) argued that the leftmost mode is spurious, and the true distribution should have only two modes. So we take bimodality as our constraint.

Instead of attempting to enforce constraints on \( f \) directly, we begin by finding \( s \) and \( \bar{w} \) such that \( f_s(x \mid \bar{w}) \) closely approximates \( f_s(x \mid p_{\text{unif}}) \). This is performed by first letting \( s \) be a sparse grid of kernel centres and then iteratively subdividing the grid in an adaptive way, until the best approximation to \( f_s(x \mid p_{\text{unif}}) \) is sufficiently close. At each stage of grid refinement, the best approximation is found by solving a QP without shape constraints. We refer to this as the binning step, because \( f_s(x \mid \bar{w}) \) is essentially a binned KDE with non-uniform bins.

The binned approximation and its kernel centres are shown in Figure 1(b). There are 65 centres, 43 of which received a weight of zero. The 22 non-zero-weighted centres are the ones closest to the data points. In this case, the total number of centres exceeds the sample size. This does not occur for larger \( n \), where binning effects a significant data reduction (see Section 4).

Let \( \bar{w}_+ \) be the non-zero weights and \( s_+ \) the corresponding kernel centres. As \( f_s(x \mid \bar{w}) = f_{s+}(x \mid \bar{w}_+) \approx f_s(x \mid p_{\text{unif}}) \), we can use either \((s, \bar{w})\) or \((s_+, \bar{w}_+)\) as surrogates for \((x, p_{\text{unif}})\) when we enforce shape constraints.

![Figure 1](image-url)

**Figure 1.** A demonstration of the estimation approach, using the chondrite data. (a) The usual unconstrained KDE. (b) The binning procedure sets up a grid of kernel centres that allows a near-perfect approximation of the original KDE. Only some of the kernel centres receive non-zero weights. (c) The weighted KDE estimator is obtained by using only the non-zero-weighted centres. (d) The adjusted estimator is obtained using all of the centres. Operations (b), (c), and (d) all involve solving the quadratic program of Section 3.1, with different choices of kernel centres and/or constraints.
If we use \( (s_+, \hat{w}+) \), we obtain the weighted KDE, as shown in Figure 1(c). The method of Hall and Huang can be used directly, choosing new weights to minimize the ISE between \( f_{s+}(x | p) \) and \( f_{s+}(x | \hat{w}+) \).

If we use \( (s, \hat{w}) \) and set our adjustment densities to be \( k_s(x) \), we obtain the special case of the adjusted KDE method, where the kernel functions and the adjustment densities coincide. Therefore, we can again treat the estimation problem as a weighted KDE. The result is shown in Figure 1(d). The adaptive binning procedure ensures that \( k_s(x) \) is a suitable arrangement of adjustment densities.

In both cases, the optimal weights are found by quadratic programming. The QP to be solved is the same as that of the binning step, with the addition of shape constraints. This is the estimation step.

In the estimation step, the optimization problem is only a QP if certain constraint-dependent values, such as the locations of modes or inflection points, are prespecified. We use the term important points to refer to such fixed values. In the usual situation where the important points are unknown, the QP solver must be run inside an outer optimization loop to search for them. For the chondrite data, locations of the two modes and their intervening antimode were found in this way. Further comments are given in the next section.

The chondrite example also illustrates qualitative differences between the weighted and adjusted KDE approaches to constraint handling. The weighted KDE can only ensure bimodality by assigning near-zero probability mass to the four leftmost data points. The remainder of the estimate is shifted up to compensate. The adjusted KDE turns the left mode into a shoulder and retains fidelity to the original estimate in regions distant from the constraint violation. It can do this because of the extra kernel centres that were introduced in data-poor regions.

### 3 Our implementation

Most of the shape constraints we implement assume continuity of the density’s first, second, or third derivative. Of the commonly used kernel functions, only the Gaussian has well-defined derivatives up to the third order over the whole real line. In particular, none of the usual compactly supported kernels have three derivatives smoothly tending to zero at the boundaries of their support. Consequently, we employ the Gaussian kernel exclusively in our implementation.

The next section lays out the general QP that is used in both the binning and estimation steps. The following sections provide further remarks. Detailed pseudocode for the estimation procedure is given in Algorithm 1.

#### 3.1 A QP for ISE minimization

We now develop a QP for choosing the weights to minimize the ISE between a shape-constrained KDE and any other weighted KDE. Suppose we have two vectors of kernel centres, \( y \in \mathbb{R}^q \) and \( s \in \mathbb{R}^r \), and corresponding weighted KDEs, \( f_y(x | v) \) and \( f_s(x | w) \). It can be shown that the ISE between the two curves is

\[
\int_{-\infty}^{\infty} \left( k_y(x)^T v - k_s(x)^T w \right)^2 \, dx = h \left( v^T L_{yy} v - 2v^T L_{ys} w + w^T L_{ss} w \right),
\]

where \( L_{ys} \) is a \( q \times r \) matrix with \( (i,j)\)th element

\[
(L_{ys})_{ij} = \frac{1}{h} \int_{-\infty}^{\infty} k_{y_i}(x) k_{s_j}(x) \, dx = (K * K) \left( \frac{y_i - s_j}{h} \right).
\]

Matrices \( L_{yy} \) and \( L_{ss} \) are defined in the same way and are \( q \times q \) and \( r \times r \), respectively.
The rightmost equality in (4) follows from the symmetry of \( K \) and allows the ISE to be computed using \((K \ast K)(t)\); the convolution of the kernel with itself. The convolution is symmetric around zero, so \( L_{yy} \) and \( L_{ss} \) are symmetric, and \( L_{sy} = L_{ys}^T \). If \((K \ast K)(t)\) is known analytically, the ISE can be computed without any need for numerical integration. Note that the ISE is quadratic in \( v \) and \( w \).

Shape constraints can be enforced on \( f_y(x \mid v) \) by placing inequality restrictions on the estimate or its derivatives at a set of constraint-checking points. Let \( g = [g_1, g_2, \ldots, g_v]^T \) be a grid of such points, where \( l = g_1 < g_2 < \cdots < g_v = u \), and \([l, u]\) is chosen to cover the region where constraints are to be satisfied. With sufficiently many grid points spread over the interval, constraint violations can be made negligibly small.

Shape constraints expressed this way become linear restrictions on the weights. For example, suppose we wish our KDE to be unimodal, with mode at \( \theta \). To satisfy this constraint, restrict \( K_y'(g_i)^T v \geq 0 \) for \( g_i \leq \theta \), and \( K_y'(g_i)^T v \leq 0 \) for \( g_i \geq \theta \). If \( g_b < \theta < g_{b+1} \), these constraints can be combined in matrix form as

\[
Av \leq 0, \quad \text{where} \quad A = \begin{bmatrix}
-K_y'(g_1)^T \\
-K_y'(g_2)^T \\
\vdots \\
-K_y'(g_b)^T \\
K_y'(g_b+1)^T \\
\vdots \\
K_y'(g_v)^T
\end{bmatrix}. \tag{5}
\]

The constraints are satisfied by any \( v \) that produces a mode between \( g_b \) and \( g_{b+1} \). While this is sufficient for most purposes, the mode can be made to occur exactly at \( \theta \) by adding an additional equality constraint, \( K_y'(\theta)^T v = 0 \).

Other derivative-based shape constraints can be expressed using systems of inequalities similar to (5). In addition to unimodality with mode at \( \theta \), the `scdensity` package allows the following constraints: bimodality with stationary points at \((\theta_1, \theta_2, \theta_3)\); tail monotonicity to the left of \( \theta_1 \) and/or to the right of \( \theta_2 \); two inflection points of \( f' \), at \((\theta_1, \theta_2)\); or three inflection points of \( f'' \), at \((\theta_1, \theta_2, \theta_3)\). Any of these may be combined with constraints that the support of the estimate be bounded to the left or to the right, or that the estimate be symmetric around \( \xi \).

Values \( \theta_i \) or \( \xi \) are the important points, which are assumed known when constructing the QP. If they are actually unknown, the QP is wrapped inside a search routine, as described in Section 3.3. The symmetry constraint requires special handling, which is reviewed in Section 5. To use multiple shape restrictions in combination, the individual constraint matrices are constructed and then stacked to form a final system of inequalities.

With all of the aforementioned data put together, the optimal weight vector for minimizing the ISE between \( f_y \) and \( f_s(x \mid w) \), subject to constraints, is the solution in \( v \) to the QP

\[
\begin{align*}
\text{minimize} & \quad v^T L_{yy} v - 2v^T L_{ys} w \\
\text{subject to} & \quad Av \leq 0 \quad \text{(shape constraints)} \\
& \quad -L_y v \leq 0 \quad \text{(non-negativity)} \\
& \quad 1^T v = 1 \quad \text{(unit integral)},
\end{align*}
\tag{6}
\]

where the final two constraints are included to ensure that the KDE is a density.
3.2 The binning and estimation steps

During binning, a new set of kernel centres, $s$, is chosen in a data-adaptive manner. The centres must cover the entire range of the data (to accommodate the adjustment curve method), while not being too close together (for efficiency and numerical stability). At the same time, there must exist weights $w$ such that $f_s(x \mid w)$ closely approximates $f_x(x \mid p_{\text{unit}})$. The centres are established using a sequential partitioning approach. Initially, $s$ is set to have $M$ elements, covering the range $[l, u]$ with regular spacing. Optimal weights $w$ are those that minimize the ISE between $f_x(x \mid p_{\text{unit}})$ and $f_s(x \mid w)$, that is, the solution to

$$\begin{align*}
\text{minimize} & \quad w^T(nL_{ss})w - 2w^T L_{sx} 1 \\
\text{subject to} & \quad -Iw \leq 0 \quad \text{(non-negativity)} \\
& \quad 1^Tw = 1 \quad \text{(unit integral)}. 
\end{align*}$$

This QP is a special case of (6), with no shape constraints.

Centres $s$ divide the range $[l, u]$ into $M-1$ intervals. Each interval is marked for subdivision if the function values of $f_s$ and $f_x$ differ by more than $\epsilon$ at the midpoint of the interval, and the interval is wider than some minimum value $\delta$. In Algorithm 1, the test function $r_i(s, w)$ indicates whether the $i$th interval becomes marked. An extra kernel centre is inserted at the midpoint of every marked interval, increasing the length of $s$. Problem (7) is solved again, and the subdivision process continues until no intervals satisfy the test function. The lower bound $\delta$ on interval width guarantees that the process will terminate.

The value of $M$ is chosen so that the initial grid spacing is as large as possible without being larger than $h$, and we set $\delta = h/10$. These choices ensure that no data point is more than $h/2$ away from a kernel centre, and that no two kernel centres are closer than $h/10$. The tolerance on function agreement, $\epsilon$, serves mainly to discourage unnecessary bisection of intervals. If it is set too large, the initial grid with spacing approximately $h/2$ will be used. If it is set too small, the densest grid with spacing approximately $h/10$ will be used. By default, we set $\epsilon$ to $10^{-3}$ times the maximum height of the unconstrained estimate.

As seen in the chondrite example, it is typical that many elements of $w$ are zero after binning completes. They correspond to kernel centres that are distant from any point in $x$. Continuing to denote the non-zero weights and their centres by $w_+$ and $s_+$, we can keep a unified notation in the estimation step by defining

$$(y, \bar{v}) = \begin{cases} 
(s, \bar{w}) & \text{if using the adjusted KDE method} \\
(s_+, \bar{w}_+) & \text{if using the weighted KDE method}
\end{cases} \quad (8)$$

In either case, $f_y(x \mid \bar{v})$ can act as a surrogate for $f_x(x \mid p_{\text{unit}})$.

The final, shape-restricted estimate is $f_y(x \mid \hat{v})$, where $\hat{v}$ is the solution to another QP. It minimizes the distance between $f_y(x \mid v)$ and $f_y(x \mid \bar{v})$:

$$\begin{align*}
\text{minimize} & \quad v^T L_{yy} v - 2\bar{v}^T L_{y\bar{y}} \bar{v} \\
\text{subject to} & \quad Av \leq 0 \quad \text{(shape constraints)} \\
& \quad A_{eq} v = 0 \quad \text{(important points)} \\
& \quad -Iv \leq 0 \quad \text{(non-negativity)} \\
& \quad 1^Tv = 1 \quad \text{(unit integral)}. 
\end{align*}$$

(9)
where $A$ is the appropriate constraint matrix as described previously and $A_{eq}$ is a matrix with one or more rows, used to fix the value of the estimate’s derivatives at the important points. This program is another special case of (6), where the two KDE’s centres coincide, and there are active shape constraints.

**Algorithm 1:** Shape Constrained Kernel Density Estimation.

**Input:** Data $x$; bandwidth $h$; a list of shape constraints, with corresponding important points; a method choice (either “weighted” or “adjusted”).

**Output:** Kernel centers $y$ and weights $\mathbf{v}$ such that $f_y(x|\mathbf{v})$ satisfies the constraints.

**Definitions**
- $\delta =$ minimum kernel spacing (default: $h/10$)
- $\epsilon =$ approximation tolerance (default: $0.001 \times \max_x f_x(x|\mathbf{p}_{\text{unif}})$)
- $v =$ number of constraint-checking points (default: 100)
- $m_i = (s_i + s_{i+1})/2$
- $\tau_i(s, \mathbf{w}) = \begin{cases} 1 & \text{if } |f_x(m_i|\mathbf{w}) - f_x(m_i|\mathbf{p}_{\text{unif}})| > \epsilon \text{ and } s_{i+1} - s_i > \delta \\ 0 & \text{otherwise} \end{cases}$
- $(s_+, \mathbf{w}+) =$ the centers and weights, excluding zero-weighted components.

**Binning step**

Set $l$ and $u$ such that $[l, u]$ covers $x$.

Set $M =$ smallest integer such that $u/M \leq h$.

Set $s =$ a regular grid of $M$ points on $[l, u]$.

*spacing is about $h$*

repeat

- Compute $\mathbf{L}_{ss}$ and set $\mathbf{L}_{ss} =$ SpectralShift($\mathbf{L}_{ss}$).
- Set $\mathbf{w} =$ solution to QP (7).
- for each $i$ such that $\tau_i(s, \mathbf{w}) = 1$
  - Insert the value $m_i$ into $s$, between $s_i$ and $s_{i+1}$.

until no intervals had $\tau_i(s, \mathbf{w}) = 1$

**Estimation step**

Set $(y, \mathbf{v}) =$

- $(s, \mathbf{w}) =$ if method is “adjusted”
- $(s_+, \mathbf{w}+) =$ if method is “weighted”

Set $\mathbf{g}$ to be a grid of $v$ constraint-checking points on $[l, u]$.

Construct constraint matrices $A$, $A_{eq}$.

if constraints are infeasible
  - return an error code.

Compute $\mathbf{L}_{yy}$ and set $\mathbf{L}_{yy} =$ SpectralShift($\mathbf{L}_{yy}$).

Set $\mathbf{v} =$ solution to QP (9).

*minimizes ISE($f_x(x|\mathbf{v})$, $f_x(x|\mathbf{p}_{\text{unif}})$), no constraints*

return $y$ and $\mathbf{v}$.

**3.3 Additional details**

It was previously observed that the matrix in the weighted KDE QP ($\mathbf{L}_{yy}$ in Equation (6)) can be ill conditioned, leaving the convexity of the optimization problem in doubt. When $K$ is the Gaussian kernel, the $(i,j)$th element of $\mathbf{L}_{yy}$ is

$$
(L_{yy})_{ij} = (K * K) \left( \frac{y_i - y_j}{h} \right) = \frac{1}{2\sqrt{\pi}} \exp \left\{ -\frac{(y_i - y_j)^2}{2h^2} \right\}. 
$$

(10)
From this, we see that the diagonal elements of \( L_{yy} \) are \( \frac{1}{2\sqrt{\pi}} \), and the elements of each row decrease toward zero away from the diagonal.

If \( y_i \) and \( y_j \) are nearly equal, then the entire \( i \)th and \( j \)th rows/columns of \( L_{yy} \) will be nearly equal. Thus, the ill-conditioning can be likened to a (multi)collinearity problem in regression. In the extreme case when \( y_i = y_j \), the two rows/columns will be exactly equal, and weights \( i \) and \( j \) will influence the ISE only through their sum. In this case, the optimal weights are clearly not unique.

Fortunately, it can be shown that matrices like \( L_{yy} \) are positive definite whenever \( K \) has finite variance and \( y \) contains no duplicate values (see Section 5 for a sketch of the proof). The ISE (3) is, then, a strictly convex function of either set of weights. This justifies a simple remedy, which Kung (2014) calls the “spectral shift approach.” If the matrix is not numerically positive definite, add a small constant to its diagonal elements. In our implementation, we let \( \lambda_{\min} \) be the smallest eigenvalue of \( L_{yy} \) and define the operation

\[
\text{SpectralShift}(L_{yy}) = \begin{cases} 
L_{yy} + (|\lambda_{\min}| + 10^{-10})I & \text{if } \lambda_{\min} < 10^{-10} \\
L_{yy} & \text{otherwise.}
\end{cases}
\]  

We have found this remedy to resolve the definiteness problem with no negative side effects. In Algorithm 1, it is used before solving the QP in both the binning and estimation stages.

When it is necessary to search for the best locations of the important points, we use the same approach we used with the adjustment curve method (Wolters & Braun, 2018, sec. 2.2). If there is only one important point, the golden-section search is used to find an optimum. When there are two or more points, the search is performed iteratively, one component at a time. This method has been found to provide good solutions and fast run times.

Whether or not the important points are known, the binning step only needs to be performed once (symmetry is an exception to this; see Section 5). After binning, the number of kernel centres is usually small, making it feasible to repeatedly solve QP (9) in searching for the optimal locations of the important points.

The optimal weights must satisfy a small number of equality constraints and a moderate number of inequality constraints. It is not possible to ensure that constraints will be numerically feasible for all possible choices of the important points. To obtain a robust implementation, it is necessary to detect and handle infeasible constraint specifications when they arise.

The problem of finding a feasible point of a given set of linear constraints is not trivial, but it can be expressed as a linear program (LP) and is routinely tackled as the first solution stage (“phase 1 LP”) in LP solvers (Antoniou & Lu, 2007). We make use of this by putting the constraints into an LP before running the QP. If the phase 1 LP does not return a solution, the QP in the estimation step is skipped and an error code is returned. This code can be handled by outer routines to give a robust user experience. During a search for the important points, detection of infeasibility allows the search to continue undisturbed.

It was mentioned previously that estimates can be constrained to have support that is bounded on the left or right. It is obviously not possible to achieve this exactly with the Gaussian kernel. Instead, our bounded-support constraint requires that \( f_y \) is less than some small number, such as 0.001 times the maximum height of the unconstrained estimate.

## 4 Demonstrations

This section demonstrates the capabilities of the \texttt{scdensity} package on two more real data examples (Figure 2) and summarizes the effect of sample size on the number of kernel centres and the run time (Figure 3).
4.1 Additional examples

Both of the following examples make use of the twoInflections+ shape constraint (which requires the first derivative of the density to have exactly three inflection points). This constraint produces density estimates with two inflection points and a high degree of qualitative smoothness.

The first data set is the 756 daily log-return values of the S&P 500 stock index, previously used for unimodal density estimation by Turnbull & Ghosh (2014). It has been argued (Fergusson & Platen, 2006) that such data are well modelled by a density symmetric around zero. Figure 2(a) shows a histogram of the data set, along with an unconstrained KDE and the corresponding KDE constrained to satisfy twoInflections+ while also being symmetric around zero. The constrained estimate was obtained with a single function call:

```r
scdensity(returns, bw="SJ", constraint = c("twoInflections+","symmetric"),
          opts=list(pointOfSymmetry=0, ncheck=200))
```

This example shows how constraints can be combined and how an `opts` list is used to provide optional extra information (in this case, the location of the point of symmetry, and `ncheck`, the number of constraint-checking points). The shape-constrained estimate eliminates spurious waves in the tails of the KDE and gives a smooth and symmetric estimate without making other assumptions, for example, about the shape of the tails.

The second data set consists of 289 axon diameter measurements from a single electron microscopy image in the collection analysed by Sepehrband et al. (2016). In the original analysis, the authors compared the fit of 16 different distribution families, searching for the most suitable parametric form. Clearly, the authors believed the true distribution to be qualitatively similar to common parametric forms but did not feel there was a priori justification to prefer one form over another. Situations like this are perfectly suited to shape-restricted estimation, particularly the twoInflections+ constraint, which can guarantee “parametric-like” smoothness while allowing the density’s shape to be data driven. The `scdensity` estimate for the axon data set was obtained with the code

```r
scdensity(diameters, bw="SJ", constraint=\texttt{\texttt{twoInflections+}})
```

and the results are shown in Figure 2(b). The constrained estimate does indeed have a shape suggestive of parametric forms, while still agreeing closely with the histogram.

Also shown on the plot are two other shape-restricted estimates obtained from other R packages that one might consider in this application: log-concave estimates (unsmoothed and smoothed) from the package `logcondens` (Dümbgen & Rufibach, 2011), and a unimodal and continuously differentiable estimate from the package `episplineDensity` (Buttrey et al., 2014). The estimation functions in these packages were used with their default settings. Of these alternative estimates, the unsmoothed log-concave estimate is closest to the `scdensity` estimate, but it has a non-differentiable peak and has difficulty fitting both the peak and the long right tail. The smoothed log-concave and epi-spline estimates are both inadequate, suggesting that manual adjustment of their default settings would be required to obtain a reasonable estimate.

4.2 Effect of sample size

Figure 3 shows the results of a small simulation designed to study the performance of the method as the sample size increases. At each of ten sample sizes ranging from $10$ to $10^5$, 250 samples were drawn from both a standard normal distribution and a $t$ distribution with five degrees of freedom. For each sample, `scdensity` was used to obtain a unimodal estimate using the adjustment curve approach.
Weighted KDE for handling shape constraints

Figure 2. Estimation results on two data sets. (a) Log returns of the S&P 500 index, showing the unconstrained KDE and the KDE with the \texttt{twoInflections+} and symmetric constraints. (b) Axon diameter measurements, comparing the KDE with \texttt{twoInflections+} constraint to the estimates provided by the \texttt{logcondens} and \texttt{episplineDensity} packages.

Figure 3. Effect of sample size on the estimation process, using the unimodality constraint and either $t_5$ data (black) or standard normal data (red). Each point is the median of 250 replications. (a) The number of kernel centres (total or zero weighted) after the binning step. (b) The run time required to obtain a constrained density estimate.

Figure 3(a) shows the median number of kernel centres after the binning step. For normally distributed data, even sample sizes of $10^5$ require only about 100 centres. Only a small proportion of the centres are zero weighted. For the $t$-distributed data, the number of centres required is much higher, and a much higher fraction of the centres are zero weighted. This can be explained by heavy tails of the $t_5$ distribution, which cause samples to cover a wider range, with larger gaps between data points.

The number of centres determines the dimension of the QP and is therefore one factor determining the run time. As shown in Figure 3(b), run times are very reasonable: less than 1 second for both distributions up to sample sizes of about $10^4$. At the largest sample sizes, run times increase faster for the $t_5$ data than for the normal data. This is due partly to the larger number of centres, but also due to the larger search range for determining the mode location.
5 Technical notes

We now provide a sketch of the proof that the matrix $L_{yy}$ is positive definite and then give more details about how the symmetry constraint is implemented.

5.1 Positive definiteness of the convolution matrix

The Gram matrix of a set of vectors $r_1, \ldots, r_t$ in an inner product space is the $t \times t$ symmetric matrix $G$, with $G_{ij} = \langle r_i, r_j \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. Such a matrix is positive semidefinite, and it is positive definite if and only if the vectors that generate the matrix are linearly independent (Horn & Johnson, 2013, thm. 7.2.10).

The set of continuous, real-valued functions with the usual addition and scalar multiplication operations is a real vector space. The set of kernel functions $\{k_v(x)\}_{v \in \mathbb{R}}$ is in this vector space, and the integral $\int_{-\infty}^{\infty} k_0(x)k_0(x)\,dx$, used to compute the elements of $L_{yy}$, is a valid inner product on the space. Hence, our matrices of the form $L_{yy}$ are Gram matrices and are at least positive semidefinite.

Positive definiteness follows if the kernel functions $k_{y_1}(x), \ldots, k_{y_r}(x)$ are linearly independent, that is, if $\sum_{i=1}^{q} a_i k_{y_i}(x) = 0$, $\forall x \Rightarrow a = 0$. This is trivially true when $q = 1$, and we may use mathematical induction to prove that it must be true for all $q$. Consider $j + 1$ kernel functions, of which the first $j$ are linearly independent, and for which $\sum_{i=1}^{j} a_i k_{y_i}(x) = 0$. If $a_{j+1} = 0$, then by the independence of $k_{y_1}(x), \ldots, k_{y_j}(x)$, $a = 0$ is the only solution. If $a_{j+1} \neq 0$, then $k_{j+1}(x) = \sum_{i=1}^{j} b_i k_{y_i}(x)$, where $b_i = -a_i/a_{j+1}$. For a compactly supported $k$, this produces an obvious contradiction as long as $y_1, \ldots, y_{j+1}$ are distinct. When the support is $\mathbb{R}$, we may add the requirement that $k$ has finite variance; then a contradiction is found by comparing the means and variances of $k_{j+1}(x)$ and $\sum_{i=1}^{j} b_i k_{y_i}(x)$.

5.2 Handling symmetry

Fundamentally, symmetry around a point $\xi$ is an equality constraint on points equidistant from $\xi$. It could be implemented as equality constraints on pairs of function values, or on pairs of weights. If the kernel centres are not positioned to be symmetric around $\xi$, this approach will lead to infeasible optimization problems. Even if the kernel centres are appropriately laid out, adding a large number of equality constraints needlessly increases the size and difficulty of the QP problem.

A much better approach is to make the KDE symmetric around $\xi$ by construction, so that the symmetry constraints do not need to be enforced at all in the QP. To do this, let $y = [y_L^T, y_R^T]^T$, where $y_L$ is a decreasing vector of $d$ kernel centres moving away from $\xi$ to the left and $y_R$ is an increasing vector of $d$ centres moving away from $\xi$ to the right. Their corresponding elements are equidistant from the point of symmetry: $(y_L)_i + (y_R)_i = 2 \xi$.

The symmetric centres $y$ can be generated by performing the binning step exactly as described in Section 3, with the initial grid laid out such that it is symmetric around $\xi$. During binning, any interval subdivided on one side of $\xi$ is subdivided on the other side as well. The binned estimator $f_y(x | \tilde{v})$ uses the symmetric bins but $2d$ unique weights.

During estimation, symmetry is assured by letting $v = [p^T, p']^T$, where $p$ is a vector of $d$ non-negative weights that sum to $1/2$. This guarantees equality of the weights on pairs of points in $y_L$ and $y_R$. The objective function for optimizing the ISE between $f_y(x | v)$ and $f_y(x | \tilde{v})$ can be expressed in terms of $p$ as

$$p^T (L_{yL}y_L + L_{yR}y_R) - p^T (L_{yL}y_R + L_{yR}y_L) \tilde{v}.$$ 

The shape constraints can similarly be expressed in terms of $p$, and the resulting QP solved to obtain $\hat{p}$.
References

Ahrens, LH (1965), ‘Observations on the Fe—Si—Mg relationship in chondrites’, *Geochimica et Cosmochimica Acta*, 29(7), 801–806.

Antoniou, A & Lu, W (2007), *Practical Optimization: Algorithms and Engineering Applications*, Springer-Verlag, New York.

Birke, M (2009), ‘Shape constrained kernel density estimation’, *Journal of Statistical Planning and Inference*, 139, 2851–2862.

Buttrey, S, Royset, J & Wets, R (2014), *episplineDensity: density estimation with soft information by exponential epi-splines*. R package version 0.0-1.

Cule, M, Samworth, R & Stewart, M (2010), ‘Maximum likelihood estimation of a multi-dimensional log-concave density’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(5), 545–607.

Dümbgen, L & Rufibach, K (2009), ‘Maximum likelihood estimation of a log-concave density and its distribution function: basic properties and uniform consistency’, *Bernoulli*, 15(1), 40–68.

Dümbgen, L & Rufibach, K (2011), ‘logcondens: computations related to univariate log-concave density estimation’, *Journal of Statistical Software*, 39(6), 1–28.

Fergusson, K & Platen, E (2006), ‘On the distributional characterization of daily log-returns of a world stock index’, *Applied Mathematical Finance*, 13(1), 19–38. https://doi.org/10.1080/13504860500394052.

Good, I & Gaskins, R (1980), ‘Density estimation and bump-hunting by the penalized likelihood method exemplified by scattering and meteorite data’, *Journal of the American Statistical Association*, 75(369), 42–56.

Grenander, U (1956), ‘On the theory of mortality measurement, part II’, *Skandinavisk Aktuarietidskrift*, 39, 125–153.

Hall, P & Huang, L S (2002), ‘Unimodal density estimation using kernel methods’, *Statistica Sinica*, 12, 965–990.

Hall, P & Ooi, H (2004), ‘Attributing a probability to the shape of a probability density’, *The Annals of Statistics*, 32(5), 2098–2123.

Horn, RA & Johnson, CR (2013), *Matrix Analysis*, 2nd edn., *Cambridge University Press*, New York.

Kung, SY (2014), *Kernel Methods and Machine Learning*, *Cambridge University Press*, New York.

Minnotte, MC (1997), ‘Nonparametric testing of the existence of modes’, *The Annals of Statistics*, 25(4), 1646–1660.

Papp, D & Alizadeh, F (2014), ‘Shape-constrained estimation using nonnegative splines’, *Journal of Computational and Graphical Statistics*, 23(1), 211–231. https://doi.org/10.1080/10618600.2012.707343.

R Core Team (2018), *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Royset, JO & Wets, RJB (2015), ‘Fusion of hard and soft information in nonparametric density estimation’, *European Journal of Operations Research*, 247(2), 532–547.

Samworth, RJ (2017), Recent progress in log-concave density estimation. *ArXiv e-prints*, arXiv:1709.03154.

Sepehrband, F, Alexander, DC, Clark, KA, Kurniawan, ND, Yang, Z & Reutens, DC (2016), ‘Parametric probability distribution functions for axon diameters of corpus callosum’, *Frontiers in Neuroanatomy*, 10(59). 10.3389/fnana.2016.00059.
The Numerical Algorithms Group (2018), *The NAG Library*, Oxford, United Kingdom. https://www.nag.com/content/how-cite-nag.

Turnbull, BC & Ghosh, SK (2014), ‘Unimodal density estimation using Bernstein polynomials’, *Computational Statistics & Data Analysis*, 72, 13–29. https://doi.org/10.1016/j.csda.2013.10.021.

Wolters, MA (2018), *scdensity: shape-constrained kernel density estimation*. R package version 1.0.2.

Wolters, MA & Braun, WJ (2018), ‘Enforcing shape constraints on a probability density estimate using an additive adjustment curve’, *Communications in Statistics—Simulation and Computation*, 47(3), 672–691. https://doi.org/10.1080/03610918.2017.1288247.