Vignette for QPdecon: An R Package for Density Deconvolution with Additive Measurement Errors using Quadratic Programming

Ran Yang, Daniel Apley*, Jeremy Staum, David Ruppert†

September 26, 2018

1 Introduction

The QPdecon package illustrated in this vignette estimates the probability density function (pdf) $f_X(\cdot)$ of a random variable (r.v.) X of interest when only a random sample of noisy observations $\{Y_1, \cdots, Y_n\}$ are available. The underlying model is $Y_i = X_i + Z_i, i \in \{1, 2, \cdots, n\}$, where the $Z_i's$ are mean-zero observation errors and are independent of the $X_i's$. As is typical in the extensive literature on density estimation with noisy observations, (e.g., Carroll and Hall [1988], Stefanski [1990], Fan [1991], Diggle and Hall [1993], Delaigle and Gijbels [2004], Hall and Meister [2007], Meister [2009]), the pdf f_Z of Z is assumed to be known.

I changed the references somewhat in the first two paragraphs. As far as I am aware, Silverman, Parzen, and Rosenblatt only discuss density estimation with noise-free observations. I added a few more references on kernel deconvolution. –DR

For the additive measurement error model, the pdf f_Y of Y is the convolution

$$f_Y(y) = (f_X * f_Z)(y) = \int_{-\infty}^{\infty} f_Z(y-x) f_X(x) \mathrm{d}x. \tag{1}$$

This convolution in the spatial domain corresponds to multiplication $\phi_Y(\omega) = \phi_X(\omega) \cdot \phi_Z(\omega)$ in the Fourier domain, where ϕ_Y denotes the Fourier transform of f_Y (likewise for ϕ_Z and ϕ_X), and ω denotes frequency. In light of this, one classic and popular method is the Fourier-based kernel deconvolution (KD) (e.g., Carroll and Hall [1988], Stefanski and Carroll [1990], Diggle and Hall [1993]). One estimates $f_X(\cdot)$ as the inverse Fourier transform of $\phi_K(h\omega)\widehat{\phi}_Y(\omega)/\phi_Z(\omega)$ (the overscore symbol $\hat{\cdot}$ denotes an estimate). The additional term $\phi_K(h\omega)$ is a frequency-domain

*Corresponding author. Department of Industrial Engineering & Management Sciences, Northwestern University, Evanston, Illinois,60208-3119

†Department of Statistical Science and School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853-3801

kernel weighting function that gives less weight to higher frequency values in the Fourier inversion integral to avoid numerical conditioning problems, and h here is the bandwidth parameter for kernel smoothing. This approach is referred to as KD (kernel deconvolution), because it is equivalent to kernel density estimation in the spatial domain, where the spatial domain kernel is the inverse Fourier transform of $\phi_K(h\omega)/\phi_Z(\omega)$, instead of some standard (e.g., Gaussian) kernel. Thus, KD is related to kernel density estimation for data observed without error [Rosenblatt, 1956, Parzen, 1962, Silverman, 1986].

Although KD methods have a sound theoretical foundation with well-understood asymptotic properties, their performance is sensitive to the choice of the bandwidth parameter which dictates the amount of smoothing [Fan, 1991, Barry and Diggle, 1995, Delaigle and Gijbels, 2004], and it may be difficult to achieve a desirable balance between over- and under-smoothing, as illustrated in the example below. Moreover, methods having desirable asymptotic results do not necessarily perform well in typical finite-sample situations. Other existing methodologies for density estimation include spline-based smoothing method by Green and Silverman [1993]. The smoothing splines methods are similar to KD in that they correspond approximately to smoothing by a kernel method with bandwidth depending locally rather than globally on the design points. Hence, such spline-based methods can suffer from similar issues with KD methods.

Fig. 1 illustrates the performance of KD methods with two types of kernels for a gamma example in which $X \sim Gamma(5, 1)$, $Z \sim N(0, \sigma_Z^2 = 3.2)$, and $n = 5000$. A histogram of the observed data $\{Y_1, \cdots, Y_n\}$, along with the true density $f_X(\cdot)$, are shown in each panel. Panel (a) also shows the KD estimate $\widehat{f_X}$ with rectangular frequency domain kernel $\phi_K(\omega) = I_{[-1,1]}(\omega)$ for bandwidth parameter $h \in \{0.87, 1.0, 1.16\}$. The salient characteristic here is the pronounced oscillation on the tails of $\widehat{f_X}$. This oscillation can be reduced by decreasing h, but the downside of this is oversmoothing of $\widehat{f_X}$. Even the largest $h = 1.16$ has not eliminated the tail oscillation, and yet the peak of $f_X(\cdot)$ is already being oversmoothed. Panel (b) shows similar results, but for triweight kernel $\phi_K(\omega) = (1 - \omega^2)^3 I_{[-1,1]}(\omega)$. The same problematic tradeoff regarding the choice of bandwidth parameter is evident. If one chooses a small enough bandwidth to avoid tail oscillation, this causes oversmoothing; if one chooses a large enough bandwidth to avoid oversmoothing, this causes tail oscillation. There may exist no value of bandwidth parameter that mitigates the tail oscillation without oversmoothing peaks.

Another undesirable characteristic of the KD method is that $\widehat{f_X}$ may be negative, as can be seen in Fig. 1. One can easily add a postprocessing adjustment of $\widehat{f_X}$ so that it is nonnegative and integrates to one, but this generally does not improve overall measures of quality of the estimator. As will be demonstrate later, it is much more effective to incorporate these constraints directly into the estimation process, as done in the **QPdecon** package. Moreover, it is even more difficult to incorporate more complex shape constraints (e.g., tail monotonicity or convexity, unimodality, etc.) into the KD method. In contrast, it is straightforward to incorporate known shape constraints into the **QPdecon** approach.

Motivated by the preceding, Yang et al. [2018] developed a quadratic programming (QP) optimization approach for density deconvolution and the **R** package **QPdecon** to implement it. This vignette gives an overview of the approach and describes how to use the **QPdecon** package. See Yang et al. [2018] for details of the QP approach and extensive performance comparisons demonstrating much better performance than existing methods. In **QPdecon**, the estimator $\widehat{f_X}$ is chosen to minimize a quadratic objective function that measures the difference between the convolution

$\widehat{f}_X * f_Z$ and an empirical density estimator $\widehat{f}_Y$. A variety of shape constraints are translated into linear constraints and can be easily incorporated into the QP formulation. The **QPdecon** objective function also include a quadratic regularization penalty for the purpose of ensuring the most appropriate level of smoothing. To select the regularization parameter (analogous to the KD's bandwidth), **QPdecon** includes a simple and computationally efficient method based on a concept similar to Stein's unbiased risk estimator (SURE) [Mallows, 1973, Stein, 1981, Efron, 1986]. **QPdecon** also includes a simple graphical method that serves as a check on the selected regularization parameter, and Yang et al. [2018] demonstrated that it is effective at preventing poor estimation results in the small proportion of cases where the SURE-like method selects too little regularization.

The examples in Yang et al. [2018] indicate that, even without shape constraints, the QP estimator performs substantially better than the KD method. With shape constraints (when applicable), the performance improvement is even larger. Even when the error density f_Z is Gaussian, which is notoriously difficult to deconvolve because of its smoothness [Carroll and Hall, 1988, Stefanski, 1990, Stefanski and Carroll, 1990, Fan, 1992, Wang and Wang, 2011], the QP estimator can achieve reasonable performance.

The remainder of this vignette is organized as follows. Section 2 describes the quadratic programming (QP) objective function for the density deconvolution problem (Section 2.1) and how to represent various shape constraints as linear constraints in the QP optimization (Section 2.2). Section 3 describes the SURE-like method for selecting the regularization parameter and method of regularization (Section 3.1) and then describes the simple, yet effective graphical check on the selected value (Section 3.2). Mixed throughout the vignette, we demonstrate how to use the **QPdecon** package to select the regularization parameter and obtain the density deconvolution estimator.

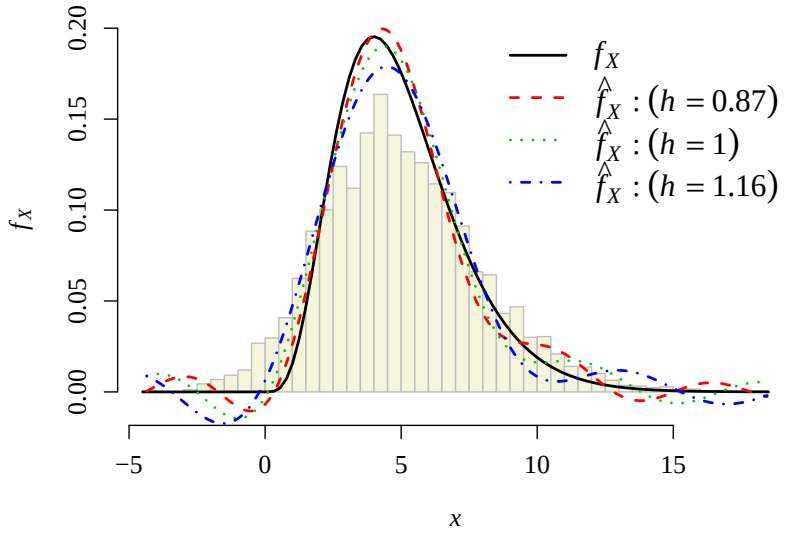
2 QP approach for density deconvolution

2.1 Basic QP Problem Formulation

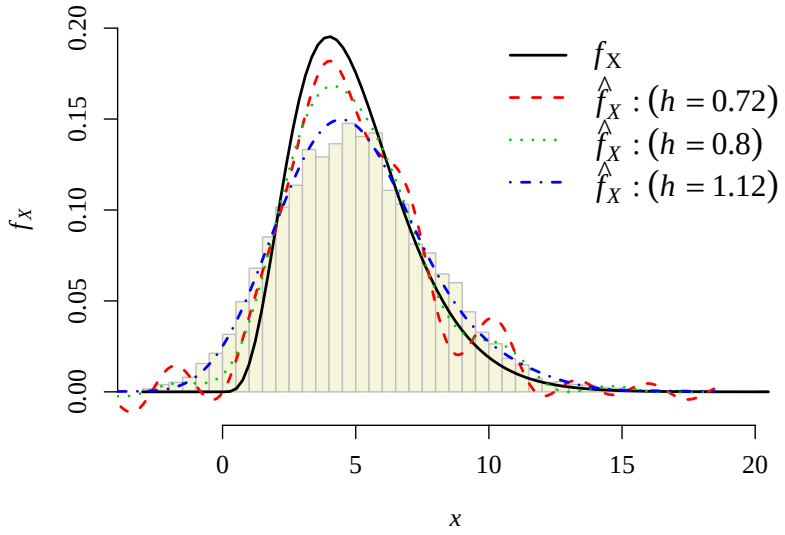
The QP approach works with a version of the continuous convolution in Eq. (1) discretized over a grid of equally spaced points $\mathbf{x} = \{x_j : 1 \le j \le K\}$ for the support of both $f_X(\cdot)$ and $f_Y(\cdot)$. Here $x_1 = \min\{Y_i : 1 \le i \le n\}$ and $x_K = \max\{Y_i : 1 \le i \le n\}$. More specifically, defining $\delta = (x_K - x_1)/(K-1)$, one can use the discrete approximation

$$f_X(x) \cong f_X(x_j) \equiv f_{X,j}, \text{if } x \in [x_j - \delta/2, x_j + \delta/2), \text{for } 1 \le j \le K,$$

and similarly for $f_Y(\cdot)$, as illustrated in Fig. 2. Let the vectors $\mathbf{f}_X = [f_{X,1}, f_{X,2}, \cdots, f_{X,K}]^T$ and $\mathbf{f}_Y = [f_{Y,1}, f_{Y,2}, \cdots, f_{Y,K}]^T$ represent the pdfs $f_X(\cdot)$ and $f_Y(\cdot)$, respectively. As an estimate of $\mathbf{f}_Y$, the histogram of $\{Y_1, \cdots, Y_n\}$ with bins centered at the same set of support points $\mathbf{x}$ is used. That is, the estimate $\widehat{f}_{Y,j}$ of $f_{Y,j}$ is the histogram bin height at x_j. The discretized estimator $\widehat{\mathbf{f}}_X$ of the pdf $f_X(\cdot)$ will also be represented as a K-length vector. It should be noted that the QP approach inherently produces a smoothed estimate $\widehat{\mathbf{f}}_X$, so that further smoothing is unnecessary. Guidelines for selecting K are discussed in Section 3.



(a)



(b)

Figure 1: The histogram is of $\{Y_1, \cdots, Y_n\}$ along with KD results for the $Gamma(5, 1)$ example for various levels of smoothing bandwidth h using (a) a rectangular kernel $\phi_K(\omega) = I_{[-1,1]}(\omega)$ and (b) a triweight kernel $\phi_K(\omega) = (1-\omega^2)^3 I_{[-1,1]}(\omega)$. Small h corresponds to undersmoothing, and large h corresponds to oversmoothing.

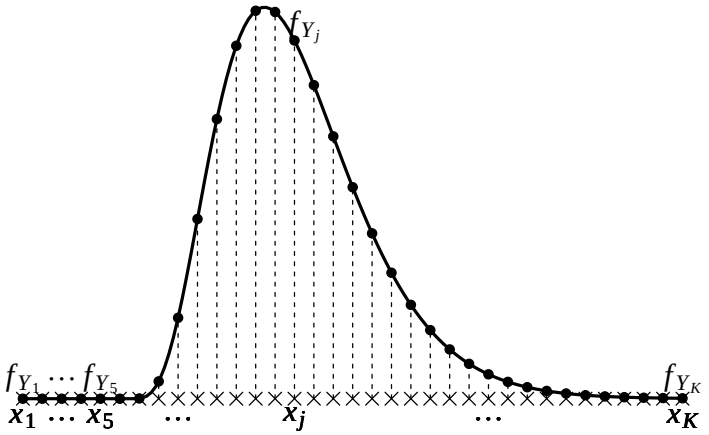


Figure 2: Illustration of the discrete approximation of f_X and the notation. The black solid curve is the density of f_X; the black dots are the discretized approximation of f_X; and the black crosses are the x-locations for the discretization.

The discretized version of Eq. (1) can be written as

$$\mathbf{f}_Y \cong \mathbf{C}\mathbf{f}_X \Longleftrightarrow \begin{bmatrix} f_{Y,1} \\ \vdots \\ f_{Y,K} \end{bmatrix} \cong \delta \begin{bmatrix} f_Z(x_1 - x_1) & \cdots & f_Z(x_1 - x_K) \\ \vdots & \ddots & \vdots \\ f_Z(x_K - x_1) & \cdots & f_Z(x_K - x_K) \end{bmatrix} \begin{bmatrix} f_{X,1} \\ \vdots \\ f_{X,K} \end{bmatrix}, \tag{2}$$

where the elements of the convolution matrix $\mathbf{C}$ are determined from the noise distribution, which is assumed known. At first glance, one may be tempted to use the estimate $\widehat{\mathbf{f}}_X = \mathbf{C}^{-1}\widehat{\mathbf{f}}_Y$, which is an exact solution to Eq. (2) with $\mathbf{f}_X$ and $\mathbf{f}_Y$ replaced by their estimates. However, as is well known in the deconvolution literature, $\mathbf{C}$ is typically (for typical noise distributions) so poorly conditioned that $\mathbf{C}^{-1}\widehat{\mathbf{f}}_Y$ is an unusable estimator subject to wild high-frequency oscillations.

Noting that $\mathbf{C}^{-1}\widehat{\mathbf{f}}_Y$ is the solution to $\widehat{\mathbf{f}}_X = \text{argmin}_{\mathbf{f}_X} \|\widehat{\mathbf{f}}_Y - \mathbf{C}\mathbf{f}_X\|^2$, this suggests using the estimator

$$\widehat{\mathbf{f}}_X = \text{argmin}_{\mathbf{f}_X} \|\widehat{\mathbf{f}}_Y - \mathbf{C}\mathbf{f}_X\|^2 + \lambda Q(\mathbf{f}_X), \tag{3}$$

where $Q(\mathbf{f}_X)$ is a regularization term that penalizes an $\mathbf{f}_X$ that is poorly behaved in some respect, and λ is a regularization parameter to be selected based on the data. For example, penalizing a large second derivative of $f_X(\cdot)$ can be achieved by using $Q(\mathbf{f}_X) = \|\mathbf{D}_2\mathbf{f}_X\|^2$, where $\mathbf{D}_2$ is an appropriately defined second-order difference matrix operator, which is referred as second derivative regularization. Another option is to use $Q(\mathbf{f}_X) = \|\mathbf{f}_X - \widehat{\mathbf{f}}_{\text{reg}}\|^2$ where $\widehat{\mathbf{f}}_{\text{reg}}$ is some easily-determined and well-behaved approximation to $\mathbf{f}_X$. For example, one can take $\widehat{\mathbf{f}}_{\text{reg}}$ to be a Gaussian distribution with mean $\widehat{\mu}_Y$ and variance $\widehat{\sigma}_Y^2 - \sigma_Z^2$, where $\widehat{\mu}_Y$ and $\widehat{\sigma}_Y^2$ are the sample mean and variance of $\{Y_1, \cdots, Y_n\}$. We refer to this as Gaussian regularization. In simulation studies, overall the two regularization approaches performed comparably, with one method working better for some examples, and vice-versa for other examples.

Because all pdfs integrate to one and are nonnegative, it makes sense to incorporate this

knowledge into the estimation of $\mathbf{f}_X$ by including constraints in the QP formulation:

$$\begin{array}{lcl} \widehat{\mathbf{f}}_X & = & \text{argmin}_{\mathbf{f}_X} \|\widehat{\mathbf{f}}_Y - \mathbf{C}\mathbf{f}_X\|^2 + \lambda Q(\mathbf{f}_X) \\ s.t. & & \delta \mathbf{1}^T \mathbf{f}_X = 1 \\ & & \mathbf{f}_X \geq \mathbf{0}, \end{array} \tag{4}$$

where $\mathbf{1}$ is a column vector of ones, and $\mathbf{f}_X \geq \mathbf{0}$ means that all elements of $\mathbf{f}_X$ are nonnegative..

2.2 Additional Shape Constraints

This section discusses a number of common features of density functions that can be translated into linear constraints on the solution, $\mathbf{f}_X$, to the QP. (Although linear on $\mathbf{f}_X$, these constraints are nonlinear on the estimated density f_X.) It is intuitively reasonable to suppose that including any such "prior" knowledge we may have regarding $\mathbf{f}_X$ should improve the estimation.

Tail monotonicity. Many pdfs have nonincreasing right tails and/or nondecreasing left tails. Suppose one knows that $f_X(x)$ is nonincreasing for $x \geq x^m$ for some specified $x^m \in \mathbf{x}$. This can be handled by incorporating additional inequality constraints into the QP formulation (4), as follows.

$$\begin{array}{lcl} \widehat{\mathbf{f}}_X & = & \text{argmin}_{\mathbf{f}_X} \|\widehat{\mathbf{f}}_Y - \mathbf{C}\mathbf{f}_X\|^2 + \lambda Q(\mathbf{f}_X) \\ s.t. & & \delta \mathbf{1}^T \mathbf{f}_X = 1 \\ & & \mathbf{f}_X \geq \mathbf{0} \\ & & \mathbf{A}_m \mathbf{f}_X \geq \mathbf{0}, \end{array}$$

where

$$\mathbf{A}_m = \begin{bmatrix} 0 & \cdots & 0 & 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & & \vdots & \ddots & & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & & \cdots & 0 & 1 & -1 & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 1 & -1 \end{bmatrix},$$

and the first non-zero column of $\mathbf{A}_m$ corresponds to x^m. A nondecreasing left tail can be handled in a similar manner, by augmenting $\mathbf{A}_m$ with additional rows.

Tail convexity. Many pdfs also have one or both tails that are convex. Suppose one knows that $f_X(x)$ is convex for $x \geq x^c$ for some specified $x^c \in \mathbf{x}$. This can be handled by adding the inequality constraints $\mathbf{A}_c \mathbf{f}_X \geq \mathbf{0}$, where

$$\mathbf{A}_c = \begin{bmatrix} 0 & \cdots & 0 & 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & & \vdots & & \ddots & & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & & 0 & 1 & -2 & 1 & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 1 & -2 & 1 \end{bmatrix},$$

and the first non-zero column of $\mathbf{A}_c$ corresponds to the location of $x^c \in \mathbf{x}$. A convex left tail can be handled similarly.

Unimodality. If we know the pdf is unimodal with mode at known location $x^u \in \mathbf{x}$, this is equivalent to a nonincreasing monotonicity constraint for $x \geq x^u$ and a nondecreasing monotonicity constraint for $x \leq x^u$. In analogy with the form of the monotonicity constraint given earlier, this can be handled by adding the inequality constraints $\mathbf{A}_u \mathbf{f}_X \geq \mathbf{0}$, where

$$\mathbf{A}_u = \begin{bmatrix} -1 & 1 & 0 & \cdots & & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \ddots & \vdots \\ \vdots & \ddots & -1 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & -1 & \ddots & 0 \\ \vdots & \ddots & & \ddots & & \ddots & 0 \\ 0 & & \cdots & & 0 & 1 & -1 \end{bmatrix},$$

and the row of $\mathbf{A}_u$ in which the order of the elements transitions from $\{-1, 1\}$ to $\{1, -1\}$ corresponds to the mode location x^u. The preceding is relevant when the mode location x^u is known in advance, which generally will not be the case. For an unknown mode location, one can add x^u as an additional decision variable and solve K separate QPs, each with a different unimodality constraint corresponding to each candidate $x^u \in \mathbf{x}$. The value of x^u resulting in the smallest QP objective function value estimates the mode location.

Support constraints. If there is information on the support of $f_X(x)$, e.g., that $f_X(x) = 0$ for $x < 0$, this can be easily taken into account. As an example, suppose that $X \geq 0$ is the concentration of a trace impurity in a chemical production process, and Y is a noisy measurement of X that can assume negative values, even though X is nonnegative. In situations like this, one can improve the estimate $\widehat{\mathbf{f}}_X$ by using the information that $f_X(x) = 0$ over certain regions, even though $f_Y(x) > 0$ over these regions. Supposing one knows that the support of $f_X(x)$ lies within the interval $[x_a, x_b]$ for some specified $x_1 \leq x_a < x_b \leq x_K$, one could solve (4) with the additional constraints that $f_{X,j} = 0$ for $j < a$ and $j > b$. In an equivalent but more computationally efficient formulation, one could simply replace the K-dimensional $\mathbf{f}_X$ in (4) by the reduced $(b - a + 1)$-dimensional counterpart $[f_{X,a}, f_{X,(a+1)}, \cdots, f_{X,b}]^T$ and also replace the $K \times K$ matrix $\mathbf{C}$ by its $K \times (b - a + 1)$ counterpart comprised of columns $\{a, a + 1, \cdots, b\}$ of $\mathbf{C}$.

2.3 Illustration of the QPdecon Package.

We now demonstrate how to use **QPdecon** to obtain the density estimator of X when it is corrupted with additive noise Z. **QPdecon** uses the **R** package **quadprog** [Turlach and Weingessel, 2015] as the solver for quadratic programming problems. The integrate-to-one and nonnegativity constraints are always recommended (as in (4)), while other constraints like monotonicity or convexity over specified tail regions can also be included if appropriate. We illustrate the package with the data generated in the following example.

```
### Generate observed data Y as a gamma X plus a normal Z
n <- 5000
X <- rgamma(n,5,1)
Z <- rnorm(n,0,sd=1.8)
Y <- X+Z # the observed sample of data
```

The `QPdecon(...)` function is the main function in the **QPdecon** package and is used to obtain the QP pdf estimator of $\mathbf{f}_X$. The usage is

```
QPdecon(Y,K,f_Z,lambda,reg="2Deriv",...)
```

The input arguments to the function are the observed data `Y`, number of histogram bins `K`, the noise density `f_Z`, the regularization parameter `lambda`, and the method of regularization (either `2Deriv` or `Gauss`). `K` is also the number of discretized points representing $\mathbf{f}_X$. For zero-mean Gaussian Z, `f_Z` should be set equal to the standard deviation of Z. For any other noise density, `f_Z` should be a user-defined function that takes a scalar input argument z and returns the noise density, $f_Z(z)$, at z. The default regularization method is `reg="2Deriv"`, which uses second derivative regularization. As mentioned previously, if `reg="Gauss"`, then the minimum L_2 distance to a Gaussian density is used for regularization; see Yang et al. [2018]. Section 3 discusses the choices of `K` and `lambda`.

In the following example, we set `lambda` below to a prespecified value, 0.011. Also, we could have simply set `f_Z` equal to 1.8, but instead we illustrate the more general case where `f_Z` is set equal to a function.

```
f_Z <- function(z){dnorm(z,mean=0,sd=1.8)} # specify the noise pdf
K <- 200 # desired number of discrete point in f_X
#### Obtain QP estimator ####
L <- QPdecon(Y=Y,K=K,f_Z=f_Z,lambda=0.011,reg="2Deriv",integr=TRUE,
    nonneg=TRUE)
#### Returned value of QPdecon()
names(L)
```

```
[1] "x"       "f_X"     "f_Y"     "lambda" "reg"     "mymode"
```

In the output object of the `QPdecon(...)`, `x` is a K-length vector of discretized values that is the support of the QP density estimator; `f_X` is the K-vector containing the QP-estimate of the density of X at the values in `x`; `f_Y` is the K-vector containing the empirical density of random variable Y (taken to be the histogram bin heights at the same values in `x`, using `K` bins corresponding to `x`); `lambda` and `reg` are the regularization parameter value and regularization method used in the QP method, respectively. The last object `mymode` returns the estimated mode location when the argument of `unimod` is set `TRUE` (`mymode="N/A"` when `unimod=FALSE`).

As discussed in Section 2.2, one can specify additional shape constraints to help improve the QP estimator. Constraints are added by specifying the corresponding arguments in the `QPdecon(...)` function, as in the example below.

The **R** code below generates Fig. 3, in which the red solid curve corresponds to the QP pdf estimator `f_X`, the black dashed curve corresponds to the true pdf f_X, and the histogram is `f_Y`. (Using `histo=TRUE` in `plot.QPdecon(...)` will plot this histogram, along with `f_X`). We decreased the sample size to 500 to better illustrate the potential improvements in the estimate due to constraints. We increased the bandwidth because of the smaller sample size. With the `plot.QPdecon(L, histo)` function in **QPdecon**, we can plot the QP estimator obtained from the `QPdecon(...)` function. Since we know the true pdf f_X for this illustrative example,

we can assess the performance of the QP pdf estimator by comparing with the $Gamma(5,1)$ density curve.

The four panels of Fig. 3 shows the pdf estimator with various combinations of constraints. The unimodality constraint is imposed by `unimod=TRUE`. In that case, the dashed vertical line indicates the mode location given by `mymode`, which is estimated automatically by the `QPdecon(...)` function. Specifying `support=c(0,20)` constraints the support to be (0,20). The upper bound on the support has no effect in this example, but constraining the support to be nonnegative improves the estimator; without that constraint $\widehat{f}(x) > 0$ for some $x < 0$. Specifying `monotone=c(2,8)` constrains the estimated density to be monotone below 2 and monotone above 8. In this case, the monotonicity constraints have the same effect as the unimodality constraint, but of course that will not happen in other examples.

```
library(QPdecon)
set.seed(8384)
n <-500
X <- rgamma(n,5,1)
Z <- rnorm(n,0,sd=1.8)
Y <- X+Z # the observed sample of data
#### Pdf estimator with only the nonnegativity
####    and integrate-to-one constraints   ####
L1 <- QPdecon(Y=Y,K=200,f_Z=1.8,lambda=0.02,reg="2Deriv",integr=TRUE,
          nonneg=TRUE)
#### Pdf estimator with additional unimodality constraint ####
L2 <- QPdecon(Y=Y,K=200,f_Z=1.8,lambda=0.02,reg="2Deriv",integr=TRUE,
          nonneg=TRUE,unimod=TRUE)
#### Pdf estimator with additional unimodality
####   and support constraints  ####
L3 <- QPdecon(Y=Y,K=200,f_Z=1.8,lambda=0.02,reg="2Deriv",integr=TRUE,
           nonneg=TRUE,unimod=TRUE,support=c(0,20))
#### Pdf estimator with additional monotonicity
####   and support constraints  ####
L4 <- QPdecon(Y=Y,K=200,f_Z=1.8,lambda=0.02,reg="2Deriv",integr=TRUE,
           nonneg=TRUE,support=c(0,20),monotone=c(2,8))

par(mfrow=c(2,2))
plot(L1,histo=TRUE,main="No additional constraints")
lines(L1$x,dgamma(L1$x,5,1),lwd=1.5,lty=2)
plot(L2,histo=TRUE,main="Unimodality constraint")
lines(L2$x,dgamma(L2$x,5,1),lwd=1.5,lty=2)
abline(v=L2$mymode,lty=2,lwd=1.5)
plot(L3,histo=TRUE,main="Unimodality and support constraints")
lines(L3$x,dgamma(L3$x,5,1),lwd=1.5,lty=2)
abline(v=L3$mymode,lty=2,lwd=1.5)
plot(L4,histo=TRUE,main="Monotonicity and support constraints")
lines(L4$x,dgamma(L4$x,5,1),lwd=1.5,lty=2)
```

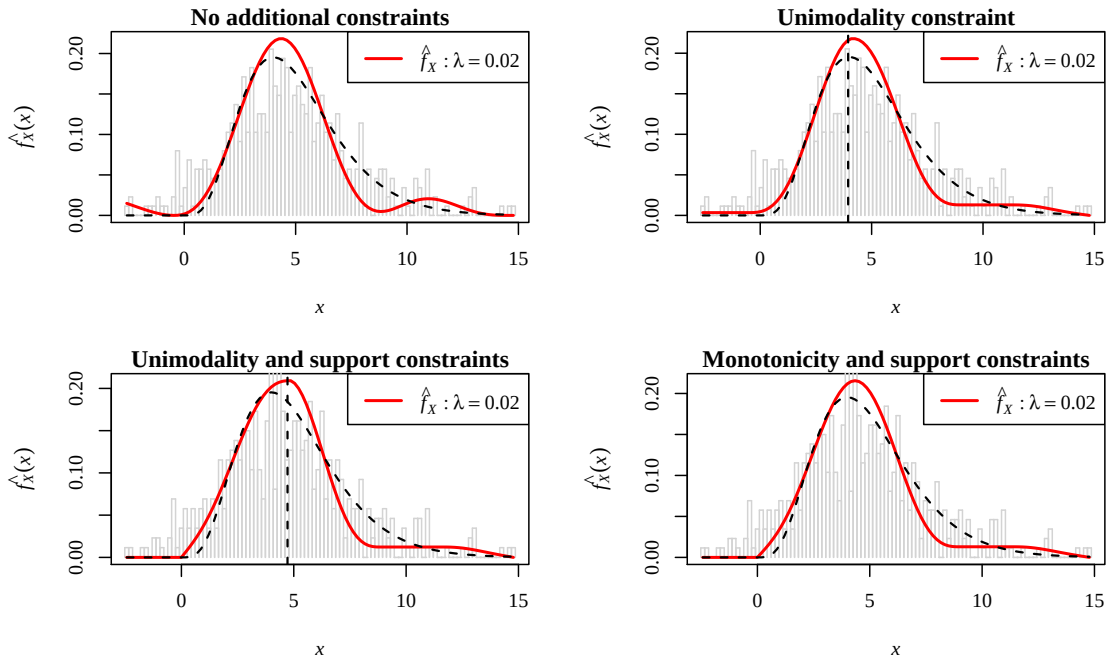


Figure 3: QP pdf estimators using various combinations of constraints added to the nonnegativity and integrate-to-one constraints. Each panel contains a histogram of the data (gray), the true pdf (black, dashed), and the QP estimator (red ,solid). The bandwidth λ is 0.02 in each case. When the unimodality constraint is imposed, a dashed vertical line is located at the estimated mode.

Top, left: No additional constraints.
Top, right: Unimodality constraint.
Bottom, left: Unimodality constraint and support constrained to be (0,20).
Bottom, right: Density constrained to be monotone on (0,2) and (8,∞), and support constrained to be (0,20).

3 Selection of the Regularization Parameter and Method

To use **QPdecon**, one must select K, λ, and $Q(\mathbf{f}_X)$ (either Gaussian or second derivative regularization).

We have found no adverse consequences to using a large K, other than an increase in computational expense. This is not surprising since the regularization parameter is λ, not K. Figure 4 shows that computation time increases rapidly with K. In that figure, the time to compute all four estimates in Figure 3 increases from 0.546 to 36.945 seconds as K increases from 100 to 400. Moreover, the estimated densities are virtually the same for K = 100, 200, 300, and 400. This can be seen in Figure 5 which contains estimated densities with the nonnegativity and integrate-to-one constraints for K = 100, 200, 300, and 400.

The rule-of-thumb that Yang et al. [2018] used is $K \approx 3\sqrt{n}$. That is, selecting K roughly three times the common $K \approx \sqrt{n}$ rule-of-thumb used with histograms. In Yang et al. [2018], n was large. If n is small, this choice of K might be too low, so we recommend $K = \min(100, 3\sqrt{n})$.

For both λ and the regularization method, our approach uses Stein's Unbiased Risk Estimate

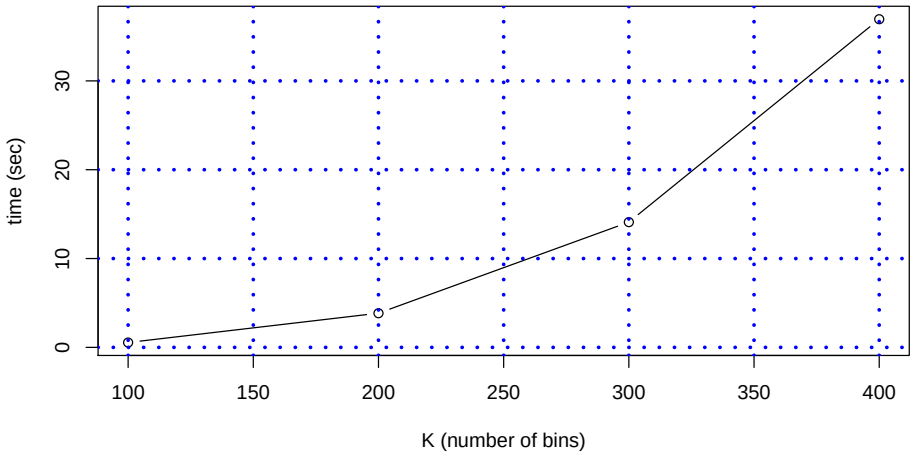


Figure 4: Computation times in seconds to compute all four estimates in Figure 3 with K = 100, 200, 300, and 400.

(SURE) method (Stein [1981]); see Section 3.1.

Occasionally, SURE selects λ so small that the density estimate is noticeably undersmoothed. In Section 3.2, we describe a graphical method, the "scree plot," that can select λ when the SURE method fails. The scree plot can also be used as a stand-alone method if one prefers not to use the SURE method.

3.1 A SURE Criterion for Selecting the Regularization Parameter and Method of Regularization

Let $\widehat{\mathbf{f}}_Y$ denote the histogram of Y for the "training" data $\{Y_1, \cdots, Y_n\}$, and let $\widehat{\mathbf{f}}^0_Y$ denote the same for hypothetical new "test" sample of n observations of Y drawn from the same distribution but independent of the training data. Let $\widehat{\mathbf{f}}_{X,\lambda}$ denote the estimate of $\mathbf{f}_X$ from the training data with regularization parameter λ.

The SURE method in Yang et al. [2018] minimizes an estimate $\mathbf{E}\{\|\widehat{\mathbf{f}}^0_Y - \mathbf{C}\widehat{\mathbf{f}}_{X,\lambda}\|^2\}$. To selection the regularization method, the estimate of $\mathbf{E}\{\|\widehat{\mathbf{f}}^0_Y - \mathbf{C}\widehat{\mathbf{f}}_{X,\lambda}\|^2\}$ is minimized over both λ and the regularization method.

3.2 A Graphical Scree-plot Approach for Selecting the Regularization Parameter

Yang et al. [2018] found in the numerous examples that the SURE choice of λ can be much too small on a relatively small percentage of Monte Carlo (MC) replicates. This problem is illustrated in Fig. 6 using 8,000 replicates of the $Gamma(5, 1)$ example in Fig. 1. For all replicates, $\widehat{\mathbf{f}}_X$ was

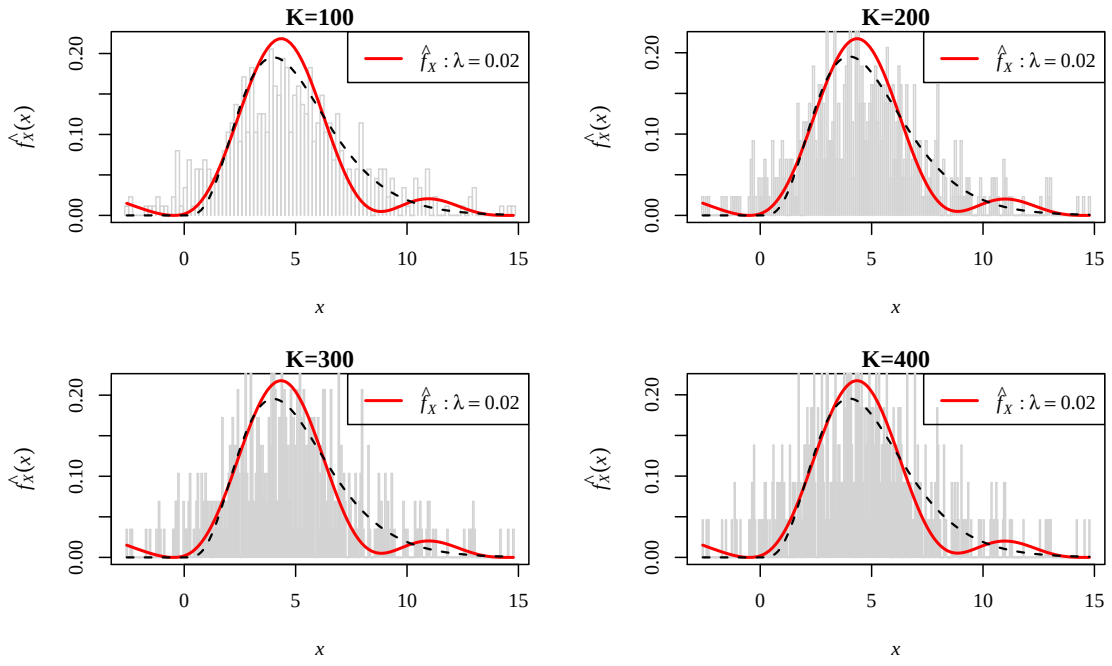


Figure 5: QP pdf estimators using the nonnegativity and integrate-to-one constraints. K = 100, 200, 300, and 400. Notice that, over this range of K, the choice of K has no noticeable effect on the estimated density.

estimated using the QP method with only the two universal shape constraints of integrate-to-one and nonnegativity.

Fig. 6a plots the estimation error measure

$$L_1(\widehat{f}_X, f_X) = \int |\widehat{f}_X(x) - f_X(x)| \mathrm{d}x$$

against $\log_{10}(\lambda_{\mathrm{SURE}})$. One can observe that 9.4% of replications have L_1 error more than twice the median L_1 error (the median is 0.089), and about 5.7% have error more than three times the median. The QP estimator in Fig. 6b corresponds to one of the occasional replicates for which λ_{SURE} is extremely underestimated, and its L_1 error is represented by the open red diamond in Fig. 6a. In comparison, Fig. 6c shows a much-improved estimation result using a corrected $\lambda = 0.011$ (corrected via the scree plot method, described below) for the data from the same replicate in Fig. 6b, and the L_1 error of the improved result is represented by the solid green diamond in Fig. 6a. Notice that the L_1 error is reduced from 1.05 to 0.19, a level that is far below the level for the extreme case and much more consistent with typical cases (twice the median L_1 error).

The corrected $\lambda = 0.011$ was obtained by inspection of the scree plot in Fig. 7. The screen plot is a simple, yet effective, graphical method for selecting an appropriate regularization parameter and avoiding the occasional poor pdf estimates when the SURE-like method selects λ too small. As illustrated in Fig. 7, the scree plot is a plot of $Q(\widehat{\mathbf{f}}_X)$ versus λ, and one looks for the elbow in the plot. The scree-plot choice for λ is the smallest value of λ that is comfortably to

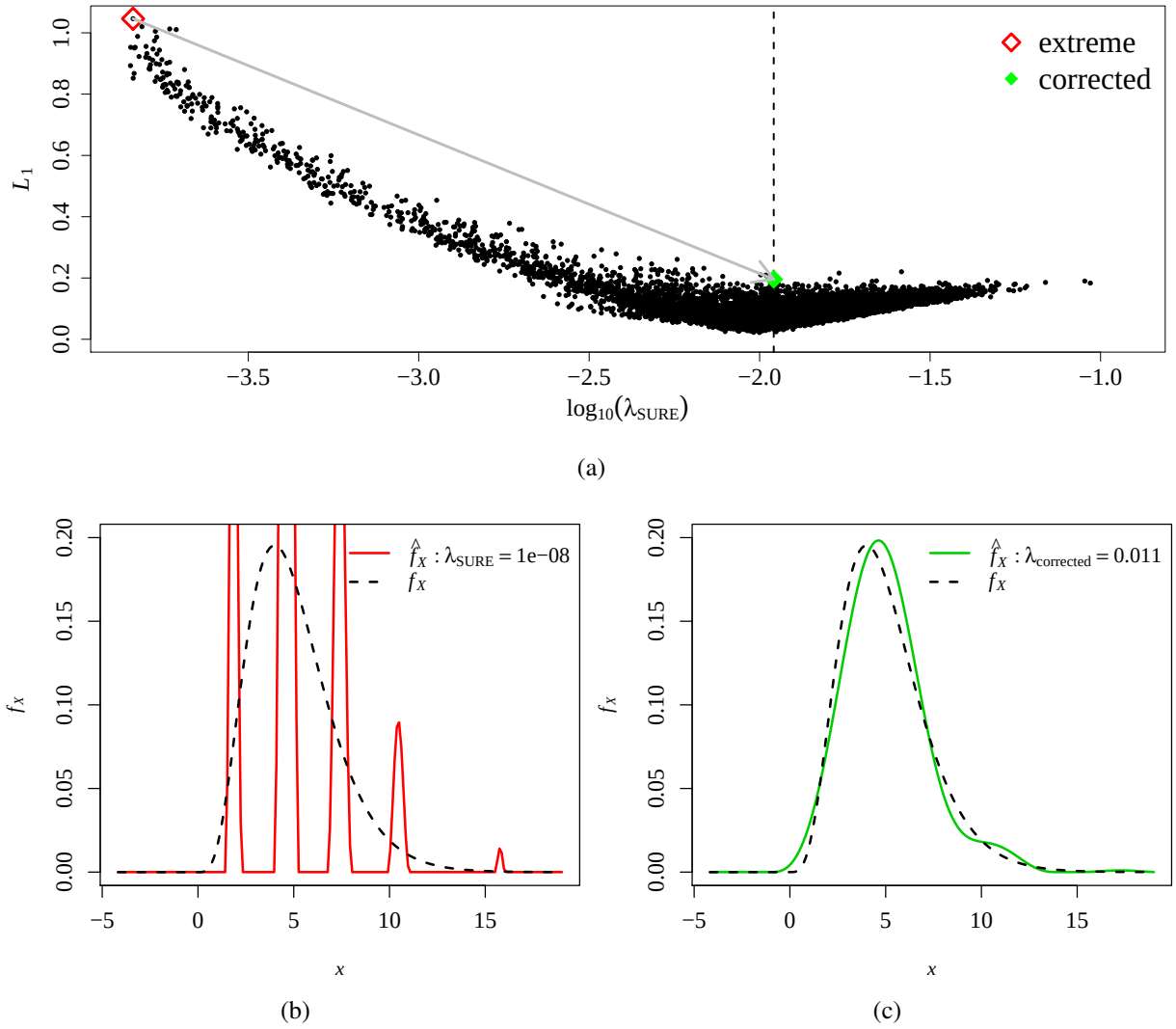


Figure 6: (a) Scatter plot of the L_1 error measure versus $\log_{10}(\lambda_{\text{SURE}})$ for 8,000 replicates of the $Gamma(5, 1)$ example; the dashed vertical line corresponds to $\lambda = 0.011$. (b) A poor pdf estimate for the replicate corresponding to the open red diamond in the upper-left corner of panel (a), for which λ_{SURE} is substantially underestimated. (c) A much-improved estimate $\widehat{\mathbf{f}}_X$ for the data for the same replicate featured in panel (b), but using a corrected $\lambda = 0.011$ obtained from the scree plot method. The L_1 error of this improved estimate is represented by the solid green diamond in panel (a)

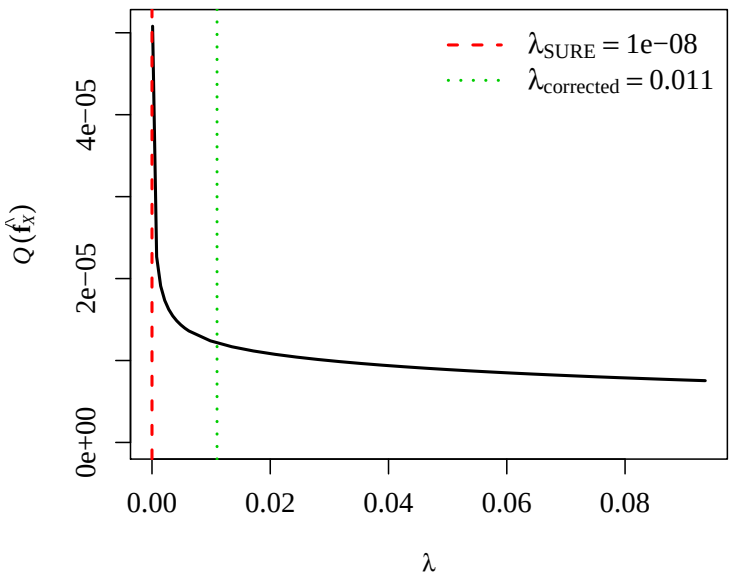


Figure 7: Scree plot for the replicate featured in Fig. 6b. The vertical red dashed line indicates the value for λ_{SURE}, which was much too small on this example and resulted in the poor pdf estimate in Fig. 6b. The vertical green dotted line indicates the corrected λ, chosen to the right of the elbow, which resulted in the substantially better pdf estimate shown in Fig. 6c.

the right of the elbow. This is analogous to plotting the estimated regression coefficients versus the regularization parameter in ridge regression to select the regularization parameter [Hoerl and Kennard, 1970a,b].

A number of conclusions can be drawn from Fig. 6a. First, we note that the best single value for λ in this $Gamma(5, 1)$ example was roughly $\lambda = 0.011$, which we found by comparing the MC average L_1 error values for a range of fixed λ values (the results of which are omitted, for brevity). We refer to this best single value of λ as the "oracle" value. The oracle value $\lambda = 0.011$ is also somewhat apparent from Fig. 6a, because if we smooth the scatter plot, the smoothed L_1 error would be smallest at approximately $\lambda = 0.011$. Also from Fig. 6a, the mode of the 8,000 λ_{SURE} values produced over the 8,000 MC replicates was also 0.011, the same as the oracle value, and in this respect the SURE-like method did an overall good job of selecting λ.

Another conclusion from Fig. 6a is that on replicates for which the SURE-like method did a poor job of selecting λ, resulting in large L_1 error, it was always because λ_{SURE} was *underestimated*. Moreover, and significantly, for all of the replicates with λ_{SURE} underestimated, the scree plots (not shown here, for brevity) always looked very much like the one shown in Fig. 7, and the corrected λ (selected to the right of the elbow) always substantially improved the pdf estimate, as in Figure 6c.

3.3 Illustration of the SURE method and of Scree Plot

We now illustrate how to have **QPdecon** find λ_{SURE} automatically and also how to select λ (or correct λ_{SURE}) using the scree plot. Users can have QPdecon find λ_{SURE} automatically by setting the argument of `lambda` in the `QPdecon(...)` function to `"SURE"`, which is the default, as

in the following code.

```
## Let QPdecon find lambda_SURE and use it in the QP estimator ##
L=QPdecon(Y=Y,Pdf=TRUE,K=K,f_Z=f_Z,lambda="SURE",reg="2Deriv",
  + integr=TRUE,nonneg=TRUE)
#### Plot the estimator ####
plot(L,histo=TRUE)
#### Compare with the true pdf ####
lines(L$x,dgamma(L$x,5,1),lwd=1)
```

Fig. 8 shows the histogram for the same sample of observations of Y depicted in Fig. 3, as well as the true and estimated pdf of f_X (black dashed curve and red solid curve, respectively). The automatically selected regularization parameter for this example is $\lambda_{\text{SURE}} = 0.007295$, and we can see this automated selection worked quite well for this example despite $\widehat{\mathbf{f}}_X$ having a slight oscillation on the right tail.

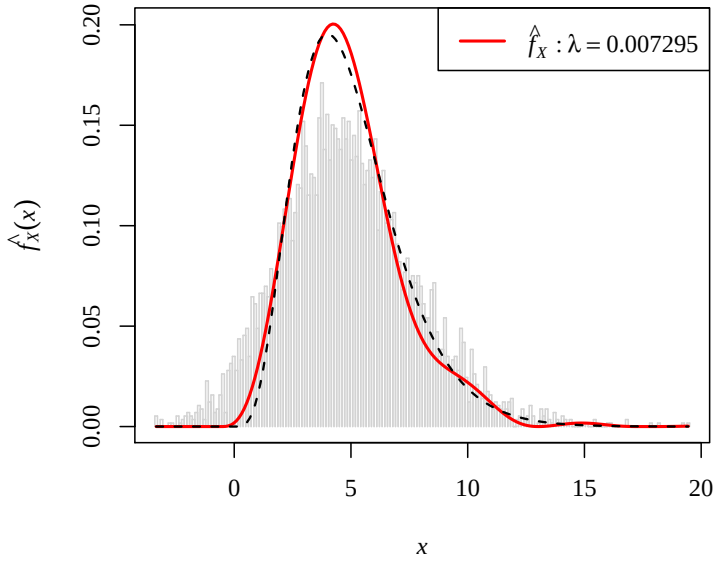


Figure 8: Histogram of the observed data Y together with the true pdf of X (black dashed curve) and the QP pdf estimator (red solid curve) using $\lambda_{\text{SURE}} = 0.0073$ obtained from the `QPdecon(...)` function.

As an alternative to using the SURE-like method to select λ, or as a check that λ_{SURE} is appropriate, we recommend also using the `ScreePlot(...)` function in **QPdecon**. The `ScreePlot(...)` function constructs the scree plot by repeatedly calculating the QP estimator $\widehat{\mathbf{f}}_X$ for a set of values of λ (which are determined automatically if the `lambda` argument is set to `"Scree"`, the default). The argument `lambda` can also be specified as a vector of fixed λ values for which the QP estimation problem will be solved for the scree plot. Other arguments to the `ScreePlot(...)` function are the same as the arguments to the `QPdecon(...)` function. In addition to creating the scree plot, the `ScreePlot(...)` function also returns two vectors, `lambda.seq` and `reg.penalty`, which corresponds to the horizontal and vertical

axes of the scree plot, respectively. The scree plot for the data depicted in Fig. 3 and Fig. 8 is shown in Fig. 9a (the arrows and dashed lines were added separately), and the corresponding **R** code is as follows.

```
#### Produce a scree plot for selecting or checking lambda ####
S=ScreePlot(Y=Y,K=K,f_Z=f_Z,lambda="Scree",reg="2Deriv",integr=TRUE,
  + nonneg=TRUE)
names(S)

[1] "lambda.seq"  "reg.penalty"
```

The long green arrow in Fig. 9a indicates $\lambda_{\mathrm{SURE}} = 0.073$, which was obtained from the automated SURE-like method. We have added the two dashed vertical lines to indicate roughly what may be viewed as the lower and upper bounds of the candidate λ values suggested by the scree-plot method, and we denote any λ falling in this range as λ_{Scree}. The arrow to the left of λ_{SURE} indicates a value ($\lambda = 0.001$) that falls substantially below the λ_{Scree} range and is clearly to the left of the elbow. The arrow to the right of λ_{SURE} indicates a value ($\lambda = 0.015$) that is within the λ_{Scree} range. The QP pdf estimators corresponding to these two λ values are shown in Fig. 9b, from which we can see that using a λ value that is too small results in tail oscillations and over-estimation in the middle quantiles, whereas using a moderate size of λ within the range of λ_{Scree} smooths the oscillations without deteriorating (oversmoothing) performance in the middle quantiles. For this particular replicate, the SURE method provides a regularization parameter λ_{SURE} that falls within the range of λ_{Scree} and results in good performance. However, as discussed earlier, there are replicates on which λ_{SURE} is chosen too small, and when this happens, the scree plot clearly indicates this (because λ_{SURE} falls to the left of the elbow, as in Fig. 7), so that a more appropriate λ can be selected to improve the performance of the QP method.

4 Summary

In this vignette, we have described the **QPdecon** package. This package uses quadratic programming (QP) with constraints for density deconvolution assuming an additive measurement error model. The **QPdecon** package implements the QP density deconvolution method via the `QPdecon(...)` function, and it also contains two schemes to select the regularization parameter. The first scheme is the automated SURE method, which is the default in **QPdecon**. The second scheme is a scree plot, implemented via the function `ScreePlot(...)`. It should be noted that one can obtain a cumulative distribution function (cdf) estimator simply by integrating the pdf estimator produced by **QPdecon**.

Yang et al. [2018] compared the **QPdecon** method to the KD method and also to the wavelet-like penalized contrast method (Comte et al. [2006]) implemented by the **R** package **deamer** (Stirnemann et al. [2012]). The latter was, aside from the **QPdecon** approach, the best performing **R** package that Yang et al. [2018] found. From Yang et al. [2018], the QP method appears to have a more favorable tradeoff between oversmoothing versus tail oscillation than existing methods. Additional advantages of the QP method are that a number of relevant constraints on the pdf can be easily incorporated into the QP formulation to further improve performance, and that the

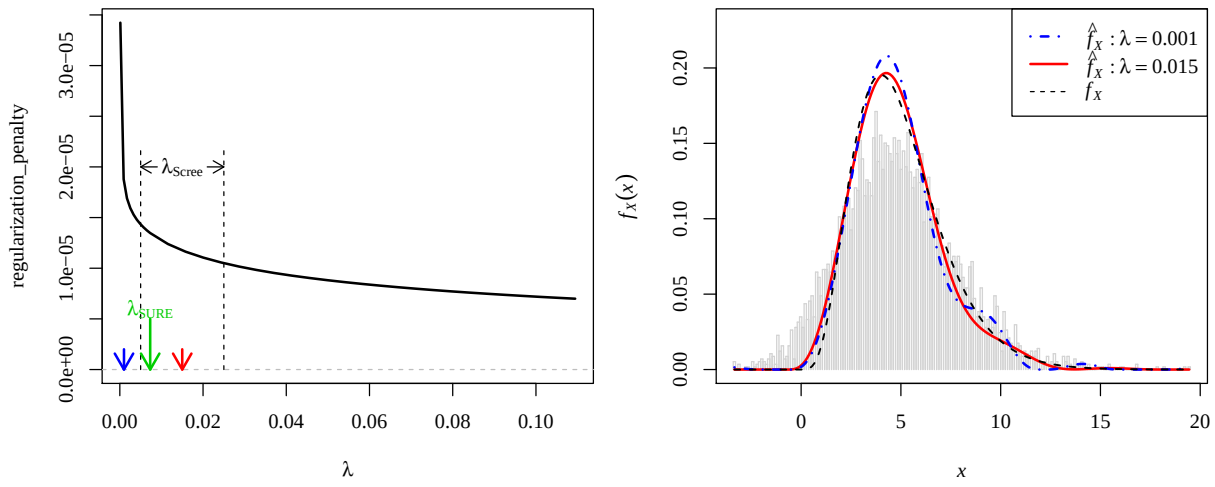


Figure 9: (a): Scree plot for the data depicted in Fig. 3 and Fig. 8 obtained using the `ScreePlot(...)` function in the **QPdecon** package. The long green arrow indicates the value of λ_{SURE} used in Fig. 8, and the two dashed vertical lines roughly indicate the range of λ values suggested by the scree-plot method. (b): The histogram of the observed data Y together with the QP estimators for the "inappropriately small" λ value (0.001) indicated by the blue arrow in panel (a) and for an appropriate λ value (0.015) indicated by the red arrow in panel (a), which falls within the λ_{Scree} range.

quadratic nature of the QP formulation is amenable to developing a computationally efficient SURE-like method of selecting the regularization parameter.

Yang et al. [2018] found, for a relatively small percentage of replicates, that the automatically selected λ_{SURE} was unreasonably small, which resulted in an erratic $\widehat{\mathrm{f}}_X$ like the one in Fig. 6b. This occurred on approximately 5% of the replicates for the gamma example and 1% of the replicates for the exponential example. This problem can be remedied using the scree-plot method as illustrated in Fig. 7.

References

J. Barry and P. Diggle. Choosing the smoothing parameter in a fourier approach to nonparametric deconvolution of a density estimate. *Journal of Nonparametric Statistics*, 4(3):223–232, 1995.

R. J. Carroll and P. Hall. Optimal rates of convergence for deconvolving a density. *Journal of the American Statistical Association*, 83(404):1184–1186, 1988.

F. Comte, Y. Rozenholc, and M.-L. Taupin. Penalized contrast estimator for adaptive density deconvolution. *Canadian Journal of Statistics*, 34(3):431–452, 2006.

A. Delaigle and I. Gijbels. Practical bandwidth selection in deconvolution kernel density estimation. *Computational statistics & data analysis*, 45(2):249–267, 2004.

P. J. Diggle and P. Hall. A fourier approach to nonparametric deconvolution of a density estimate. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 523–531, 1993.

B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American statistical Association*, 81(394):461–470, 1986.

J. Fan. On the optimal rates of convergence for nonparametric deconvolution problems. *The Annals of Statistics*, pages 1257–1272, 1991.

J. Fan. Deconvolution with supersmooth distributions. *Canadian Journal of Statistics*, 20(2): 155–169, 1992.

P. J. Green and B. W. Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press, 1993.

P. Hall and A. Meister. A ridge-parameter approach to deconvolution. *The Annals of Statistics*, 35(4):1535–1558, 2007.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970a.

A. E. Hoerl and R. W. Kennard. Ridge regression: applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, 1970b.

C. L. Mallows. Some comments on c p. *Technometrics*, 15(4):661–675, 1973.

A. Meister. *Deconvolution problems in nonparametric statistics*. New York, Springer, 2009.

E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

L. Stefanski and R. Carroll. Deconvoluting kernel density estimators. statistics 21 169–184. *Mathematical Reviews (MathSciNet): MR1054861 Digital Object Identifier: doi*, 10: 02331889008802238, 1990.

L. A. Stefanski. Rates of convergence of some estimators in a class of deconvolution problems. *Statistics & Probability Letters*, 9(3):229–235, 1990.

C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

J. Stirnemann, A. Samson, F. Comte, and C. Lacour. *Deconvolution density estimation with adaptive methods for a variable prone to measurement error*, July 2012. URL `https://cran.r-project.org/web/packages/deamer/index.html`. R package version 1.0.

B. A. Turlach and A. Weingessel. *Functions to solve Quadratic Programming Problems*, February 2015. URL `http://CRAN.R-project.org/package=quadprog`. R package version 1.5-1.

X.-F. Wang and B. Wang. Deconvolution estimation in measurement error models: The r package decon. *Journal of Statistical Software*, 39(10), 2011.

R. Yang, D. Apley, J. Staum, and D. Ruppert. Density deconvolution with additive measurement errors using quadratic programming. *submitted*, 2018.