Journal of Statistical Software

March 2011, Volume 39, Issue 10. *http://www.jstatsoft.org/*

Deconvolution Estimation in Measurement Error Models: The R Package decon

Xiao-Feng Wang
Cleveland Clinic Foundation

Bin Wang
University of South Alabama

Abstract

Data from many scientific areas often come with measurement error. Density or distribution function estimation from contaminated data and nonparametric regression with errors-in-variables are two important topics in measurement error models. In this paper, we present a new software package **decon** for R, which contains a collection of functions that use the deconvolution kernel methods to deal with the measurement error problems. The functions allow the errors to be either homoscedastic or heteroscedastic. To make the deconvolution estimators computationally more efficient in R, we adapt the fast Fourier transform algorithm for density estimation with error-free data to the deconvolution kernel estimation. We discuss the practical selection of the smoothing parameter in deconvolution methods and illustrate the use of the package through both simulated and real examples.

Keywords: measurement error models, deconvolution, errors-in-variables problems, smoothing, kernel, faster Fourier transform, heteroscedastic errors, bandwidth selection.

1. Introduction

Data measured with errors occur frequently in many scientific fields. Ignoring measurement error can bring forth biased estimates and lead to erroneous conclusions to various degrees in a data analysis. One could think of several examples in which measurement error can be a concern:

- *In medicine:* The NHANES-I epidemiological study is a cohort study consisting of thousands of women who were investigated about their nutrition habits and then evaluated for evidence of cancer. The primary variable of interest in the study is the "long-term" saturated fat intake which was known to be imprecisely measured. Indeed, NHANES-I was one of the first studies where a measurement error model approach was used

(Carroll, Ruppert, Stefanski, and Crainiceanu 2006). Two more comprehensive studies, NHANES-II and NHANES-III, were published later.

- *In bioinformatics:* Gene microarray techniques have become very popular in recent years. A microarray consists of an arrayed series of thousands of microscopic spots of DNA molecules (genes). A gene present in the RNA sample finds its DNA counterpart on the microarray and binds to it. The spots then become fluorescent, and a microarray scanner is used to "read" the intensities in the microarray. The whole process to obtain the fluorescent intensities in a microarray study is subject to measurement error. Background correction is a critical step for microarray data analysis, which refers to correcting the effects of the measurement error for the observed intensities before performing further statistical analysis.

- *In chemistry:* The Massachusetts acid rain monitoring project was first described by Godfrey, Ruby, and Zajicek (1985), where water samples were collected from about 1800 water bodies, and chemical analyses were accomplished by 73 laboratories. Measuring chemistry values typically involves error, therefore external calibration/validation data were collected based on blind samples sent to the lab with "known" values. In the statistical analysis of the study, one faces the problem of measurement errors in the predictors. The essential insight underlying the solution of the measurement error problem is to recover the parameter of the latent variables by using extraneous information.

- *In astronomy:* Most astronomical data come with information on their measurement errors. Morrison, Mateo, Olszewski, Harding *et al.* (2000) studied galaxy formation with a large survey of stars in the Milky Way. The investigators were interested in the velocities of stars, which represent the "fossil record" of their early lives. The observed velocities involved heteroscedastic measurement errors. To verify the galaxy formation theories, one is to estimate the density function from contaminated data that are effective in unveiling the numbers of bumps or components.

- *In econometrics:* The stochastic volatility model has been fairly successful in modeling financial time series. As a basis for analyzing the risk of financial investments, it is an important technique used in finance to model asset price volatility over time. It can be shown that the stochastic volatility model can be rewritten as a regression model with errors-in-variables (Comte 2004). Therefore, the techniques in measurement error models can be used for solving the finance time series problems.

The consequences of ignoring measurement error include, for example, masking the important features of the data which further makes graphical model analysis confusing; losing the power to detect relationships among variables; and bringing forth bias in function/parameter estimation (Carroll *et al.* 2006). Here we use two simulated examples to illustrate the effects of ignoring errors. The first example is the density estimation of a variable X, where X is from $0.4N(-1.5, 1) + 0.6N(1.5, 1)$. However, instead of observing X, we observe $W = X + U$, where the error U is from $N(0, 1)$. We generate 1000 simulated observations from such a model. The left panel of Figure 1 presents the kernel density estimate from the uncontaminated sample (dashed line), the kernel density estimate from the contaminated sample (dotted line), and the true density function of X (solid line). We notice that even if the true density is bimodal, the kernel density estimate from the contaminated data may be unimodal. The

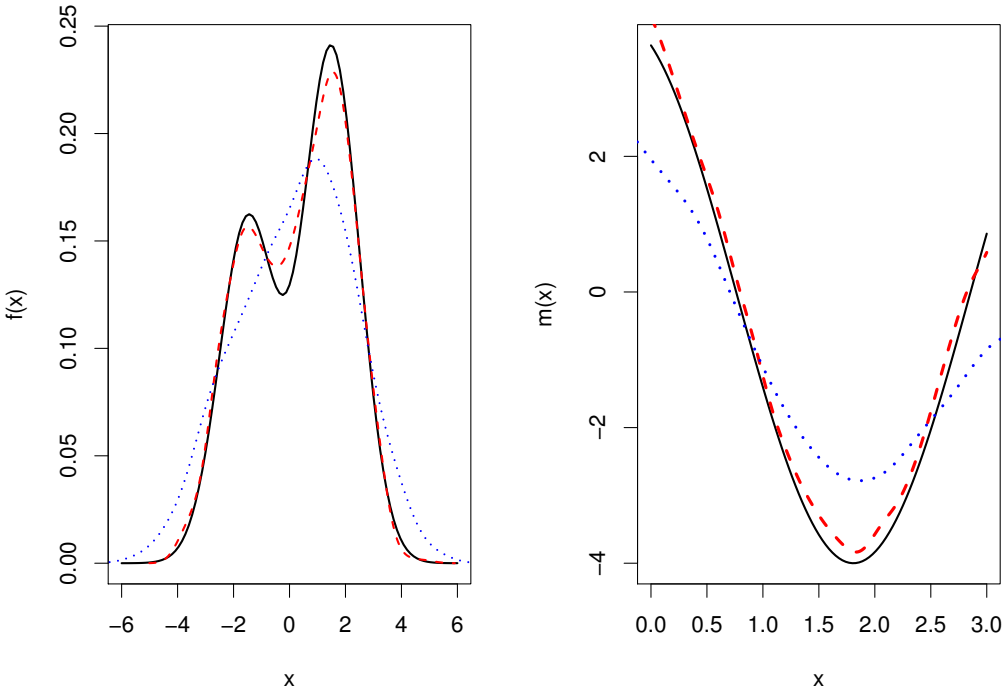


Figure 1: Simulation examples to illustrate the effects of measurement error: The solid lines denote the true curves; the dashed lines denote the kernel estimates from the uncontaminated sample; the dotted lines denote the kernel estimates from the uncontaminated sample.

second example is a regression of a response Y on a predictor X. We set the mean function as $4\sin(3X/2+2)$, where X is uniformly distributed on the interval $[0,3]$, and the residual variance $\sigma_\varepsilon^2 = 0.2$. Suppose that X is measured with error and we observe $W = X + U$, where the error U is from $N(0, 0.5^2)$. The simulated data were generated from the model with the size $n = 1000$. The right panel of Figure 1 displays the kernel regression estimates from the simulated case. The regression estimate from the uncontaminated sample (dashed line) gives an accurate estimate for the true curve (solid line), while the estimate from the contaminated sample (dashed line) is far from the target function. Thus, correcting the bias of naive estimators is critical in measurement error problems.

Statistical models for addressing measurement error problems can be classified as parametric or nonparametric (Carroll *et al.* 2006). Our interest here focuses on two nonparametric models and their estimation:

Model I: Assume that we observe the contaminated data $W_1, \cdots, W_n$ instead of the uncontaminated data $X_1, \cdots, X_n$, where W_j's are generated from an additive measurement error model

$$W_j = X_j + U_j, \qquad j = 1, \cdots, n. \tag{1}$$

We further assume that X_j's are independent and identically-distributed (i.i.d.) as X, the errors U_j's are i.i.d. as U, and X and U are mutually independent. The density function of U is denoted by f_U, assumed known. Under this additive error model, one is to recover the density function of X, f_X, or the distribution function of X, F_X, based on the data W_j's. Closely related to the density estimation is the problem of estimating the conditional density of X given W, $f_{X|W}(x|w)$.

Model II: Suppose that the observations are a sample of i.i.d. random vectors (W_1, Y_1), ..., (W_n, Y_n) generated by the model

$$\begin{cases} Y_j = m(X_j) + \varepsilon_j, \\ W_j = X_j + U_j, \qquad j = 1, \cdots, n, \end{cases} \tag{2}$$

where U_j's are the measurement error variables, independent of $(X_j, Y_j, \varepsilon_j)$, and ε_j's are the regression random errors assuming $E(\varepsilon_j | X_j) = 0$. The goal is to estimate the regression function $m(x)$ based on observations Y_j's with W_j's, where direct observation of X_j's is not possible.

Each of the two models is the subject of ongoing research in statistics. The first model is an *additive measurement error model*. The problem of estimating f_X is also known as the *deconvolution problem*. It is often related to the application to imaging deblurring, microarray background correction, and bump hunting with measurement error. The second model is known as *regression with errors-in-variables*, which often occurs in bioscience, astronomy, and econometrics.

Methods for correcting the effects of measurement error based on the above two models have been widely investigated in the past two decades. In the additive measurement error model, Carroll and Hall (1988) and Stefanski and Carroll (1990) proposed the deconvolution kernel density estimator to recover the unknown density function from contaminated data, where the kernel idea and the Fourier inverse were employed in the construction of the estimator. Since then, the deconvolution kernel approach has been extensively studied. See for instance, Zhang (1990), Fan (1991, 1992), Efromovich (1997), Delaigle and Gijbels (2004a,b), Meister (2004) and van Es and Uh (2005), among others. The idea in deconvolution kernel density estimation was also generalized to nonparametric regression with errors-in-variables by Fan and Truong (1993). Recent contributions to the two measurement error problems include the consideration of heteroscedastic errors. Delaigle and Meister (2008) proposed a generalized deconvolution kernel estimator for density estimation with heteroscedastic errors. They also applied this idea to nonparametric regression estimation in the heteroscedastic errors-in-variables problem (Delaigle and Meister 2007). Hall and Lahiri (2008) studied estimation of distributions, moments and quantiles in the deconvolution problems. Wang, Fan, and Wang (2010) explored smooth distribution estimators with heteroscedastic error. Carroll, Delaigle, and Hall (2009) discussed the nonparametric prediction in measurement error models when the covariate is measured with heteroscedastic errors. A comprehensive discussion of nonparametric deconvolution techniques can be found in the recent monograph by Meister (2009). We thereby call all these kernel-type methods that require an inverse Fourier transform *deconvolution kernel methods* (DKM).

Despite the fact that DKM are shown to be the powerful tools in measurement error problems, there is no existing software to implement the methods systematically. In this paper, we present a newly-developed package **decon** for R (R Development Core Team 2010), which is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=decon` and provides a series of functions to recover the unknown density function, the distribution function, or the regression function using DKM in measurement error problems. We propose and apply an fast Fourier transform (FFT) algorithm in the deconvolution estimation, which adapts from the algorithm in kernel density estimation with error-free data by Silverman (1982). The resulting R functions become computationally very fast. Our R

functions allow both homoscedastic errors and heteroscedastic errors. Several bandwidth selection functions are also available in the package. The rest of the paper is organized as follows. Section 2 gives a summary of the DKM that are used in our package. Section 3 discusses the practical selection of the smoothing parameter in the measurement error problems. Section 4 addresses the FFT algorithm in the estimating procedures. Section 5 demonstrates our package through both simulated and real data examples. Finally, the paper ends with discussion.

2. Deconvolution methods in measurement error problems

In this section, we review DKM in the two measurement error models and discuss some computational technical details, which have been implemented in the software package.

2.1. Kernel methods in estimating density and distribution functions

Under Model I, let φ_X, φ_U, φ_W denote the characteristic function of X_j, U_j and W_j, respectively. Assume that $\varphi_U(t) \neq 0$ for $\forall t \in \mathbb{R}$. An inverse Fourier transform leads to,

$$f_X(x) = \frac{1}{2\pi} \int e^{-itx} \varphi_X(t) dt = \frac{1}{2\pi} \int e^{-itx} \frac{\varphi_W(t)}{\varphi_U(t)} dt. \tag{3}$$

A naive estimator of $f_X(x)$ can be obtained by substituting $\varphi_W(t)$ in (3) by its sample estimate

$$\tilde{\varphi}_W(t) = \frac{1}{n} \sum_{j=1}^{n} e^{itW_j},$$

and $\varphi_U(t)$ by its explicit expression (assumed known or estimated separately). However, in practice this naive estimate is unstable because the sample characteristic function has large fluctuations at its tails. To avoid this defect, one can replace $\varphi_W(t)$ with its kernel estimator,

$$\hat{\varphi}_W(t) = \int e^{itx} \hat{f}_W(w) dw,$$

where $\hat{f}_W(w) = (nh)^{-1} \sum_{j=1}^{n} K((w - W_j)/h)$ is the conventional kernel density estimator of f_W, and $K(\cdot)$ is a symmetric probability kernel with a finite variance. The resulting estimator of f_X based on $\hat{\varphi}_W(t)$ is the following deconvolution kernel density estimator (Stefanski and Carroll 1990),

$$\hat{f}_X(x) = \frac{1}{nh} \sum_{i=1}^{n} L\left(\frac{x - W_i}{h}\right), \tag{4}$$

where

$$L(z) = \frac{1}{2\pi} \int e^{-itz} \frac{\varphi_K(t)}{\varphi_U(t/h)} dt \tag{5}$$

is called the *deconvoluting kernel* such that φ_K is compactly supported and is the characteristic function of the kernel $K(\cdot)$, and $h = h(n) > 0$ is the bandwidth parameter depending on n. The distribution estimator $\hat{F}_X$ of F_X is thus defined as simply the integral of $\hat{f}_X$ over $(-\infty, x]$ (Hall and Lahiri 2008),

$$\hat{F}_X(x) = \frac{1}{2} + \frac{1}{2\pi n} \sum_{j=1}^{n} \int \frac{\sin(t(x - W_j)) \varphi_K(ht)}{t \varphi_U(t)} dt.$$

The difficulty of deconvolution depends heavily on the smoothness of the error density f_U: The smoother the error density the harder deconvolution is. In the classical deconvolution literature, the error distributions are classified into two classes: Ordinary smooth distribution and supersmooth distribution(Fan 1991). Examples of ordinary smooth distributions include Laplacian, gamma, and symmetric gamma; examples of supersmooth distributions are normal, mixture normal and Cauchy. Generally speaking, a supersmooth distribution is smoother than a ordinary smooth distribution, so f_X is more difficult to be deconvoluted when X is contaminated by supersmooth errors. It has been show that, for instance, the convergence rate is $O((\log n)^{-1/2})$ when errors belong to the normal family, and the convergence rate is $O(n^{-4/9})$ with Laplacian errors. In the **decon** package, two important cases of measurement error distributions are allowed: Normal (super-smooth) and Laplacian (ordinary-smooth).

In kernel density estimation for error-free data, the choice of the kernel function K does not have a big influence on the quality of the estimator. However, in deconvolution kernel estimation for contaminated data, the particular structure of the deconvolution estimators require the characteristic function of the kernel, φ_K, to have a compact and symmetric support. This requirement can be relaxed in the case of ordinary smooth errors or when the variance of measurement errors is small. We consider the following kernels in the package.

Kernels for normal errors

The normal distribution $N(0, \sigma^2)$ is the most commonly-used error distribution in practice. There are two typical choices of the kernel functions for normal errors. The first one is the following second-order kernel whose characteristic function has a compact and symmetric support (Fan 1992; Delaigle and Gijbels 2004a),

$$K(x) = \frac{48\cos x}{\pi x^4}\left(1 - \frac{15}{x^2}\right) - \frac{144\sin x}{\pi x^5}\left(2 - \frac{5}{x^2}\right). \tag{6}$$

Its characteristic function is

$$\varphi_K(t) = (1-t^2)^3 I_{[-1,1]}(t),$$

where $I_{[-1,1]}(t)$ is the indicator function. Hence, the resulting deconvoluting kernel with normal error is

$$L_1(x) = \frac{1}{\pi}\int_0^1 \cos(tx)(1-t^2)^3 e^{\frac{\sigma^2 t^2}{2h^2}}\, dt.$$

The requirement for this support kernel can be relaxed when the error variance is small in Gaussian deconvolution. Fan (1992) gave comprehensive discussions about the effects of error magnitude on the DKM. In the package, a user can select the standard normal density as the kernel function if the magnitude of error variance is small, where the corresponding deconvoluting kernel becomes

$$L_2(x) = \frac{1}{\sqrt{2\pi(1-\sigma^2/h^2)}} e^{-\frac{x^2}{2(1-\sigma^2/h^2)}}.$$

When could one use the normal kernel in a data analysis? Fan (1992) recommended to consider the case as $\sigma = O(n^{-1/5})$ and $\hat{h}_{opt} > \sigma$. If a user is not sure about error magnitude in a study, the support kernel is recommended.

Kernel for Laplacian errors

With Laplacian errors, U has density $f_U(x) = \frac{1}{2\sigma} exp(-|x|/\sigma)$. We consider the standard normal kernel function, so the resulting deconvoluting kernel for the case of Laplacian errors is

$$L_3(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \left(1 + \left(\frac{\sigma}{h}\right)^2 (1 - x^2)\right).$$

The R functions `DeconPdf` and `DeconCdf` in the **decon** package perform the deconvolution kernel density and distribution estimation from contaminated data, respectively. In deconvolution problems, it is common to assume an explicit form of the density function f_U of U, because f_X is not identifiable if f_U is unknown. There are two common ways to estimate the parameters of f_U in real data analysis. f_U is estimable from additional data $U'_1, ..., U'_m$ (i.i.d. as U), which are collected in a separate independent experiment. For example, additional "negative control" data are available in Lumina Bead microarray studies. One can also estimate f_U when replicated measurements of W are available. The Framingham study that we will present in Section 5 is such a case.

2.2. Heteroscedastic contamination

In many real applications, the distributions of measurement errors could vary with each subject or even with each observation, so the errors are heteroscedastic. Hence, consideration of heteroscedastic errors is very important. Recently, Delaigle and Meister (2008) altered Model I to allow that each U_j has its own density f_{U_j}, $j = 1, \cdots, n$. A typical case of heteroscedastic errors is that $f_{U_1}, \cdots, f_{U_n}$ are from the same distributional family, but the parameters of the measurement error distributions vary with the observation index. Through an inverse Fourier transform, Delaigle and Meister (2008)'s deconvolution estimator for the density with heteroscedastic errors can be also written as a form of kernel-type density estimator,

$$\hat{f}_{X,H}(x) = \frac{1}{nh} \sum_{j=1}^{n} L_j^H \left(\frac{x - W_j}{h}\right), \tag{7}$$

where

$$L_j^H(z) = \frac{1}{2\pi} \int e^{-itz} \frac{\varphi_K(t)}{\psi_{U_j}(t/h)} dt, \quad \psi_{U_j}(t) = \frac{\frac{1}{n}\sum_{k=1}^{n} |\varphi_{U_k}(t)|^2}{\varphi_{U_j}(-t)}. \tag{8}$$

Wang *et al.* (2010) discussed the deconvolution estimator of the smooth distribution function with heteroscedastic errors. The estimator is given by

$$\hat{F}_{X,H}(x) = \frac{1}{2} + \frac{1}{2\pi n} \sum_{j=1}^{n} \int \frac{\sin(t(x - W_j))\varphi_K(ht)}{t\psi_{U_j}(t)} dt.$$

The R functions `DeconPdf` and `DeconCdf` also allow us to estimate density and distribution functions with heteroscedastic errors. In the current version, only the case of heteroscedastic normal errors is considered.

2.3. Conditional density estimation

Under Model I, closely related to the density estimation is the problem of estimating the conditional density of X given W, $f_{X|W}(x|w)$. Conditional density estimation has an important

application to microarray background correction. Wang and Ye (2010) proposed a *re-weighted deconvolution kernel estimator*, which is defined by

$$\hat{f}_{X|W}(x|w) = \sum_{j=1}^{n} \tau(w|x) \frac{1}{h} L\left(\frac{x - W_j}{h}\right), \tag{9}$$

where the weight

$$\tau(w|x) = \frac{f_U(w-x)}{\frac{1}{b}\sum_{j=1}^{n} K_1\left(\frac{w-W_j}{b}\right)},$$

and $K_1(\cdot)$ is a conventional kernel function, $L(\cdot)$ is the deconvoluting kernel defined in (5), and b and h are the smoothing parameters. The function `DeconCPdf` allows us to estimate the conditional density function with homoscedastic errors.

2.4. Nonparametric regression with errors-in-variables

The ideas of the deconvolution kernel density estimators can be generalized to nonparametric regression with errors-in-variables. To describe the deconvolution kernel regression methods, let us start from the standard nonparametric regression case where the covariates X_j's are not contaminated. The goal here is to find the relationship between variables X_j's and Y_j's. One tries to estimate the conditional mean curve

$$m(x) = E(Y|X = x) = \frac{\int y f(x,y) dy}{f_X(x)} = \frac{r(x)}{f_X(x)}, \tag{10}$$

where $f(x, y)$ and $f_X(x)$ denote the joint density of (X, Y) and the marginal density of X, respectively. The denominator in (10) can be estimated by the standard kernel density estimator. Since the joint density $f(x, y)$ can be estimated using a multiplicative kernel, one could work out an estimator of the numerator in (10) by replacing the joint density $f(x, y)$ with its kernel estimate, which leads to

$$\tilde{r}(x) = (nh)^{-1} \sum_{j=1}^{n} Y_j K\left(\frac{x - X_j}{h}\right).$$

A natural estimate of $m(x)$ is now the combination of the estimates of the denominator and the numerator,

$$\tilde{m}(x) = \sum_{j=1}^{n} Y_j K\left(\frac{x - X_j}{h}\right) \Big/ \sum_{j=1}^{n} K\left(\frac{x - X_j}{h}\right).$$

This estimator is known as the *Nadaraya-Watson estimator*.

Back to Model II, extending the kernel idea becomes natural in the errors-in-variables setting. The denominator of the Nadaraya-Watson estimator may be replaced by the deconvolution kernel density estimator (4), which is an empirical version of $f_X(x)$ as in the error-free case. The estimator of the numerator of $m(x)$ must be constructed so that it does not require knowledge from the unobservable X_j's but only from the contaminated data W_j's. In the spirit of the deconvolution kernel density estimator, Fan and Truong (1993) suggest to estimate $r(x)$ with

$$\hat{r}(x) = \frac{1}{2\pi n} \sum_{j=1}^{n} Y_j \int e^{-itx} \psi_K(ht) e^{itW_j} / \psi_U(t) dt.$$

This leads to the final deconvolution kernel regression estimator,

$$\hat{m}(x) = \sum_{j=1}^{n} Y_j L\left(\frac{x - W_j}{h}\right) \Big/ \sum_{j=1}^{n} L\left(\frac{x - W_j}{h}\right), \tag{11}$$

where $L(\cdot)$ is the deconvoluting kernel defined in (5).

Delaigle and Meister (2007) further generalized the estimator (11) to the case of heteroscedastic errors, where each U_j has its own density f_{U_j}, $j = 1, \cdots, n$. The generalized regression estimator is defined by

$$\hat{m}_H(x) = \sum_{j=1}^{n} Y_j L_j^H\left(\frac{x - W_j}{h}\right) \Big/ \sum_{j=1}^{n} L_j^H\left(\frac{x - W_j}{h}\right), \tag{12}$$

where $L_j^H(\cdot)$ is defined in (8). In our package, the R function `DeconNpr` allows us to perform nonparametric regression analysis with either homoscedastic or heteroscedastic errors-in-variables.

3. Bandwidth selection

Bandwidth selection in deconvolution problems has been broadly discussed in many papers. Hesse (1999) carried out a theoretical study of the cross-validation (CV) bandwidth selection procedure. Delaigle and Gijbels (2004a) studied a bootstrap procedure to estimate the optimal bandwidth and showed its consistency. Delaigle and Gijbels (2004b) compared several plug-in bandwidth selectors with the CV bandwidth selector and the bootstrap bandwidth selector. Wang and Wang (2010) generalized the plug-in and the bootstrap bandwidth selection methods to the case of heteroscedastic errors. In the package, we provide a few bandwidth selection functions for practical use.

3.1. Rule of thumb

As in kernel density estimation with error-free data, the criterion of the bandwidth selection in deconvolution problems is the mean integrated squared error (MISE), defined by

$$MISE(h) = E \int (\hat{f}_X(x, h) - f_X(x))^2 dx.$$

The simplest bandwidth selection method available in the package is a rule-of-thumb, which is based on theorem 1 and theorem 2 of Fan (1991). In the case of the homoscedastic normal errors, by the definition of super-smooth distribution, the errors have a supersmooth distribution of order $\beta = 2$ with a positive constant $\gamma = 2/\sigma^2$. Working out with the error distribution, the kernel function, and the asymptotic MISE, we can obtain the rule-of-thumb bandwidth,

$$h_{ROT,N} = \left(\frac{4}{\gamma}\right)^{1/\beta} (\log n)^{-1/\beta} = \sqrt{2}\sigma(\log n)^{-1/2}. \tag{13}$$

In the case of the homoscedastic Laplacian errors (ordinary smooth), the rule-of-thumb bandwidth becomes,

$$h_{ROT,L} = \left(\frac{5\sigma^4}{n}\right)^{1/9}. \tag{14}$$

The R function `bw.dnrd` implements the above methods to choose the bandwidth depending on the type of errors.

3.2. Plug-in method

The plug-in bandwidth method is the normal reference approach to minimize the approximated MISE. Stefanski and Carroll (1990) showed that the asymptotic dominating term of the MISE of the deconvolution kernel density estimator (4) could be estimated by,

$$\widehat{MISE}(h) = \frac{1}{2\pi nh}\int \frac{|\varphi_K(t)|^2}{|\varphi_U(t/h)|^2}dt + \frac{h^4}{4}R(f_X'')\int x^2K(x)dx, \tag{15}$$

where $R(f_X'') = \int (f_X''(x))^2dx$. Evaluating the $\widehat{MISE}(h)$ involves estimating the unknown quantity $R(f_X'')$. If one assumes X to be normal, $R(f_X'') = 0.375\sigma_X^{-5}\pi^{-1/2}$. Hence, the estimator $R(\hat{f}_X'')$ is defined by $R(\hat{f}_X'') = 0.375\hat{\sigma}_X^{-5}\pi^{-1/2}$, where $\hat{\sigma}_X = \sqrt{\hat{\sigma}_W^2 - \sigma^2}$, $\hat{\sigma}_W^2$ is the sample variance of W, and σ^2 is the variance of the measurement error. Our package numerically evaluates $\widehat{MISE}(h)$ on a fine grid of h-values, then selects the optimal h that minimizes $\widehat{MISE}(h)$ on the grid. The R function `bw.dmise` implements the plug-in method to choose the bandwidth.

3.3. Bootstrap methods

Delaigle and Gijbels (2004a) studied the bootstrap bandwidth selection method by directly minimizing a bootstrap MISE. The method does not require the generation of any bootstrap sample in practice. The bootstrap-based method selects the bandwidth through minimization of the following quantity,

$$\begin{aligned}\widehat{MISE}^*_{boot}(h) =& \frac{1}{2\pi nh}\int |\varphi_K(t)|^2|\varphi_U(t/h)|^{-2}dt - \frac{1}{\pi}\int |\hat{\varphi}_{X,g}(t)|^2\varphi_K(ht)dt \\ &+ \frac{n-1}{2\pi n}\int |\hat{\varphi}_{X,g}(t)|^2|\varphi_K(ht)|^2dt,\end{aligned} \tag{16}$$

where g is a pilot bandwidth that can be determined from the rule-of-thumb or the plug-in bandwidth methods, and $\hat{\varphi}_{X,g}(t)$ is the Fourier transform of $\hat{f}_X(\cdot; g)$ given by $\hat{\varphi}_{X,g}(t) = \hat{\varphi}_W(t)\varphi_K(gt)/\varphi_U(t)$, with $\hat{\varphi}_W$ being the empirical characteristic function of W. The R function `bw.dboot1` implements the above bootstrap method to choose the bandwidth. We also provide another bootstrap bandwidth selection function `bw.dboot2`. It calculates the bootstrap MISE from real bootstrap samples and then finds the optimal bandwidth. As pointed out by Faraway and Jhun (1990), in bandwidth selection for error-free data, the additional computational cost of bootstrap bandwidth selection with real resampling often appears to result in better bandwidth selection, which provides another excellent candidate of bandwidth selectors.

The above bandwidth methods for the case of homoscedastic errors are also generalized to the case of heteroscedastic errors in the package except for the second bootstrap method with real resampling (the R function `bw.dboot2`). We did not provide the CV bandwidth selector in the package, since the bootstrap bandwidth outperforms the CV bandwidth according to the discussion by Delaigle and Gijbels (2004b). The deconvolution kernel regression estimators

have the same optimal rates as the deconvolution kernel density estimators, so one can easily apply the bandwidth selectors for density deconvolution to the regression estimators. Moreover, with the series of functions we provide in the package, it is not difficult to program the advanced method using simulation extrapolation (SIMEX) for bandwidth parameter choice in errors-in-variables problems proposed by Delaigle and Hall (2008).

It should be noticed by a user that the rule-of-thumb method may generate a "silly" selected bandwidth when sample size is small. Based on our extensive simulations, we recommend the two bootstrap bandwidth selectors as the data-driven selectors in practice.

4. Estimation using the fast Fourier transform

Deconvolution estimation involves n numerical integrations for each grid where the density is to be estimated, thus directly programming in R is quite slow. In the package, we program in R incorporating C and Fortran codes. Two options are provided for calculating the estimators in the **decon** package: The direct method based on the definitions discussed above, and the method with an FFT algorithm. We adapt the FFT algorithm for density estimation with error-free data proposed by Silverman (1982) to the DKM. Data are discretized to a very fine grid, then FFT is applied to convolve the data with a specific kernel to obtain the estimate. Specifically, we first take the Fourier transform of $\hat{f}_X(x)$ in (4) to obtain

$$\tilde{f}(t) = \sqrt{2\pi}\tilde{K}(ht)\tilde{u}(t), \tag{17}$$

where $\tilde{K}(\cdot)$ and $\tilde{u}(\cdot)$ are the Fourier transforms of the deconvoluting kernel and the data, respectively. For example, in the homoscedastic error cases, the Fourier transforms of $L_1(x)$, $L_2(x)$ and $L_3(x)$ are as follows,

$$\begin{aligned}
\tilde{L}_1(t) &= (1-t^2)^3 I_{[-1,1]}(t) e^{\frac{\sigma^2 t^2}{2h^2}}, \\
\tilde{L}_2(t) &= e^{-\frac{t^2}{2}\left(1-\frac{\sigma^2}{h^2}\right)}, \\
\tilde{L}_3(t) &= (1+\sigma^2 t^2) e^{-\frac{t^2}{2}}.
\end{aligned}$$

The Fourier transform of the data is given by $\tilde{u}(t) = (n\sqrt{2\pi})^{-1} \sum_{j=1}^{n} e^{itW_j}$, where a discrete approximation to $\tilde{u}(\cdot)$ is found by constructing a histogram on a grid of 2^d cells and then applying the FFT. Next the discrete Fourier transform of f_X is obtained from (17). Finally, the estimator of f_X is found by an inverse transform. With the adoption of FFT, the computational aspects of the deconvolution estimators become very efficient. Table 1 compares the system time spent by the FFT algorithm and the direct algorithm from the definitions in computing the deconvolution density estimators with different error types and different sample sizes. A MAC system with a 2.8 GHz Intel Core 2 Duo and 4GB memory was used for the simulation study.

In Table 1, n is the sample size and m is the number of points on the grid where the density functions were evaluated. The columns titled "FFT" list the system time used by the FFT algorithm, while the columns titled "Direct" list the system time used by the direct method. The deconvoluting kernels L_1, L_2, and L_3 are used in computing for the cases of normal error, small normal error and Laplacian error, respectively. We see that under the small normal errors and Laplacian errors, where the Gaussian kernel was used, the difference in computing

		Normal		Small normal		Laplacian	
n	m	FFT	Direct	FFT	Direct	FFT	Direct
40	64	0.001	0.020	0.000	0.001	0.000	0.001
	256	0.000	0.076	0.001	0.001	0.001	0.001
	512	0.000	0.149	0.000	0.002	0.000	0.002
	1024	0.001	0.304	0.001	0.004	0.001	0.004
200	64	0.000	0.094	0.000	0.002	0.000	0.001
	256	0.001	0.376	0.000	0.005	0.000	0.004
	512	0.001	0.749	0.000	0.009	0.001	0.007
	1024	0.000	1.499	0.001	0.017	0.001	0.014
2,000	64	0.001	0.934	0.001	0.011	0.000	0.009
	256	0.001	3.739	0.001	0.042	0.001	0.035
	512	0.000	7.466	0.001	0.082	0.001	0.072
	1024	0.001	14.923	0.001	0.168	0.001	0.142
4,000	64	0.001	1.867	0.000	0.022	0.000	0.018
	256	0.001	7.458	0.001	0.083	0.000	0.071
	512	0.001	14.926	0.001	0.166	0.000	0.142
	1024	0.001	29.849	0.001	0.332	0.001	0.283
20,000	64	0.001	9.329	0.001	0.104	0.001	0.089
	256	0.003	37.329	0.002	0.412	0.002	0.355
	512	0.002	74.619	0.002	0.826	0.001	0.701
	1024	0.002	149.396	0.002	1.651	0.002	1.407
200,000	512	0.047	742.521				

Table 1: Timings in seconds for calculations of deconvolution density estimators by the FFT algorithm and by the direct application of the definitions. n is the sample size and m is the number of points on the grid. The deconvoluting kernel L_1 is used for the case of normal error; L_2 is used for the case of small normal error; L_3 is used for the case of Laplacian error.

time is not very obvious. Even with $n = 20,000$ and $m = 1024$, the direct computing method with R interfacing with C/Fortran used only 1.651 seconds. However, when the support kernel was considered to deal with the normal errors, the method using Fourier transformations is far more efficient. We also adopted the 10 points Legendre-Gauss quadrature integration method to compute the support kernels where integral computations are needed. The difference in the precision of the "FFT" and "Direct" methods is very subtle and negligible. The current version of the package does not support the FFT algorithm in the case of heteroscedastic Laplacian error.

5. The decon package

The **decon** package contains four main deconvolution kernel estimation functions (`DeconPdf`, `DeconCdf`, `DeconCPdf`, `DeconNpr`), four bandwidth selection functions (`bw.dnrd`, `bw.dmise`, `bw.dboot1`, `bw.dboot2`) and other plot functions. In this section, we demonstrate the use of the **decon** package with several simulated examples and real data applications.

5.1. Simulated examples of deconvolution with homoscedastic errors

The first simple example is to recover the density function from data contaminated with Laplacian errors. We simulate the true random variables from $N(0, 1)$ and then add them with simulated measurement errors from the Laplacian distribution with the location parameter $\mu = 0$ and the scale parameter $\sigma = 0.5$:

```
R> n1 <- 500
R> x1 <- rnorm(n1, sd = 1)
R> sig1 <- 0.5
R> u1 <- ifelse(runif(n1) > 0.5, 1, -1) * rexp(n1, rate = 1/sig1)
R> w1 <- x1 + u1
```

To recover the density of the true variables, we simply use the rule-of-thumb bandwidth and compute the deconvolution density with `DeconPdf`:

```
R> bw1 <- bw.dnrd(w1, sig = sig1, error = "laplacian")
R> (f1 <- DeconPdf(w1, sig1, error = "laplacian", bw = bw1, fft = TRUE))

Call:
    DeconPdf(y = w1, sig = sig1, error = "laplacian", bw = bw1, fft = TRUE)

Data: y (500 obs.);        Bandwidth 'bw' = 0.4405

       x                  y
 Min.   :-6.04978   Min.   :0.0000000
 1st Qu.:-2.98680   1st Qu.:0.0006015
 Median : 0.07618   Median :0.0144636
 Mean   : 0.07618   Mean   :0.0818716
 3rd Qu.: 3.13915   3rd Qu.:0.1269764
 Max.   : 6.20213   Max.   :0.3533401
```

The output from the function `DeconPdf` is an object with class "`Decon`" whose underlying structure is a list containing the same components as in the function `density` in R.

We then consider a more complex case and estimate both density and distribution functions. Our simulated true model is a mixed normal with $0.5N(-3, 1) + 0.5N(3, 1)$ and the measurement errors are from $N(0, 0.8^2)$. We use the bootstrap bandwidth selector with resampling here.

```
R> n2 <- 1000
R> x2 <- c(rnorm(n2/2, -3, 1), rnorm(n2/2, 3, 1))
R> sig2 <- 0.8
R> u2 <- rnorm(n2, sd = sig2)
R> w2 <- x2 + u2
R> bw2 <- bw.dboot2(w2, sig = sig2, error = "normal")
R> f2 <- DeconPdf(w2, sig2, error = "normal", bw = bw2, fft = TRUE)
R> F2 <- DeconCdf(w2, sig2, error = "normal", bw = bw2)
```

We can further plot the deconvolution estimates from the two simulated cases. The R functions `plot.DeconPdf` and `plot.DeconCdf` in the package dispatch according to S3 rules, thus a user can simply use the generic R function, `plot`, to generate figures.

To evaluate the performance of DKM, we shall compare the deconvolution estimators with the kernel estimators from the uncontaminated sample and the uncontaminated sample. The following R function `SDF` is to estimate the kernel smooth distribution function from error-free data (Azzalini 1981).

```
R> SDF <- function(x, bw = bw.nrd0(x), n = 512, lim = 1) {
+    dx <- lim * sd(x)/20
+    xgrid <- seq(min(x) - dx, max(x) + dx, length = n)
+    Fhat <- sapply(x, function(x) pnorm((xgrid - x)/bw))
+    return(list(x = xgrid, y = rowMeans(Fhat)))
+  }
```

Figure 2 is generated from the R codes below.

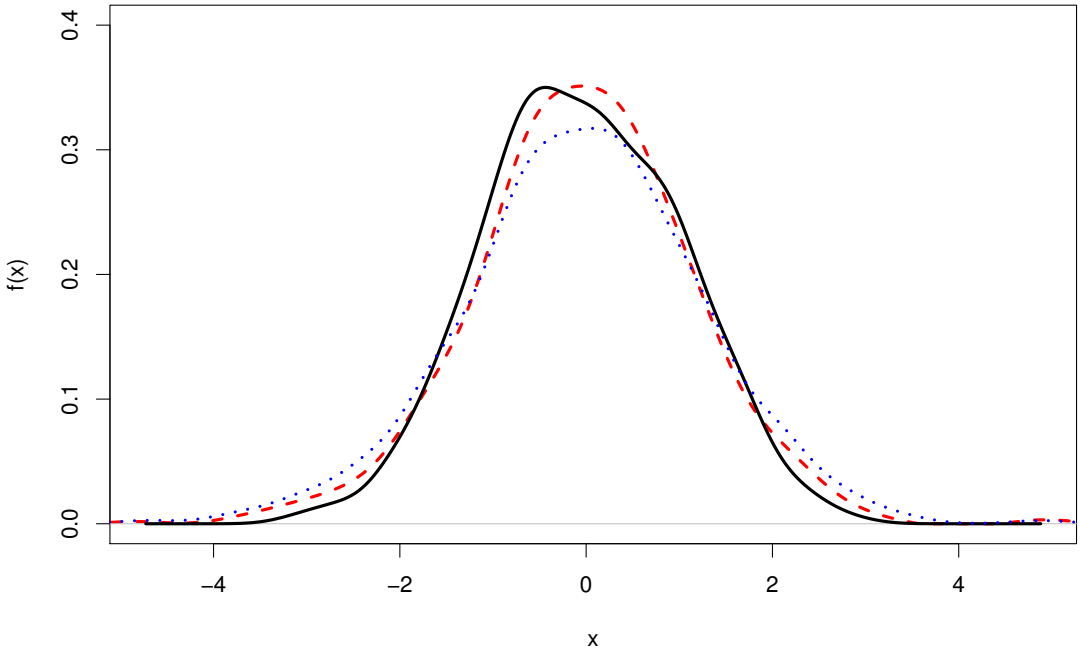
```
R> plot(f1, col = "red", lwd = 3, lty = 2, xlab = "x", ylab = "f(x)",
+    main = "")
R> lines(density(x1), lwd = 3, lty = 1)
R> lines(density(w1), col = "blue", lwd = 3, lty = 3)
R> par(mfrow=c(1,2))
R> plot(f2, col = "red", lwd = 3, lty = 2, xlab = "x", ylab = "f(x)",
+    main = "")
R> lines(density(x2), lwd = 3, lty = 1)
R> lines(density(w2), col = "blue", lwd = 3, lty = 3)
R> plot(F2, col = "red", lwd = 3, lty = 2, xlab = "x", ylab = "f(x)",
+    main = "")
R> lines(SDF(x2), lwd = 3, lty = 1)
R> lines(SDF(w2), col = "blue", lwd = 3, lty = 3)
```

In Figure 2, the solid lines denote the kernel estimates from the uncontaminated samples; the dashed lines denote the deconvolution estimates; and the dotted lines denote the the kernel estimates from the contaminated samples. We see that the deconvolution estimators work quite well to recover the functions in both simulated cases. Ignoring measurement error leads to biased estimates and can further lead to erroneous conclusions.

5.2. Simulated example of deconvolution with heteroscedastic errors

Deconvolution estimation with heteroscedastic errors involves more complex computation. Our R functions will automatically check whether the error type is homoscedastic or heteroscedastic. In the following example, we consider a case where the true model is a skewed distribution, with $X_j \sim \chi^2(1.5)$, $j = 1, ..., n$. The measurement errors are heteroscedastic, from $U_j \sim N(0, \sigma_j^2)$, where the error standard deviation σ_j depends on X_j, through $\sigma_j(X_j) = 0.7 + X_j / \max_{1 \le k \le n}\{X_k\}$. The R codes are displayed as follows.

a: *Estimation of density function with Laplacian errors.*



b: *Estimation of density and distribution functions with normal errors.*

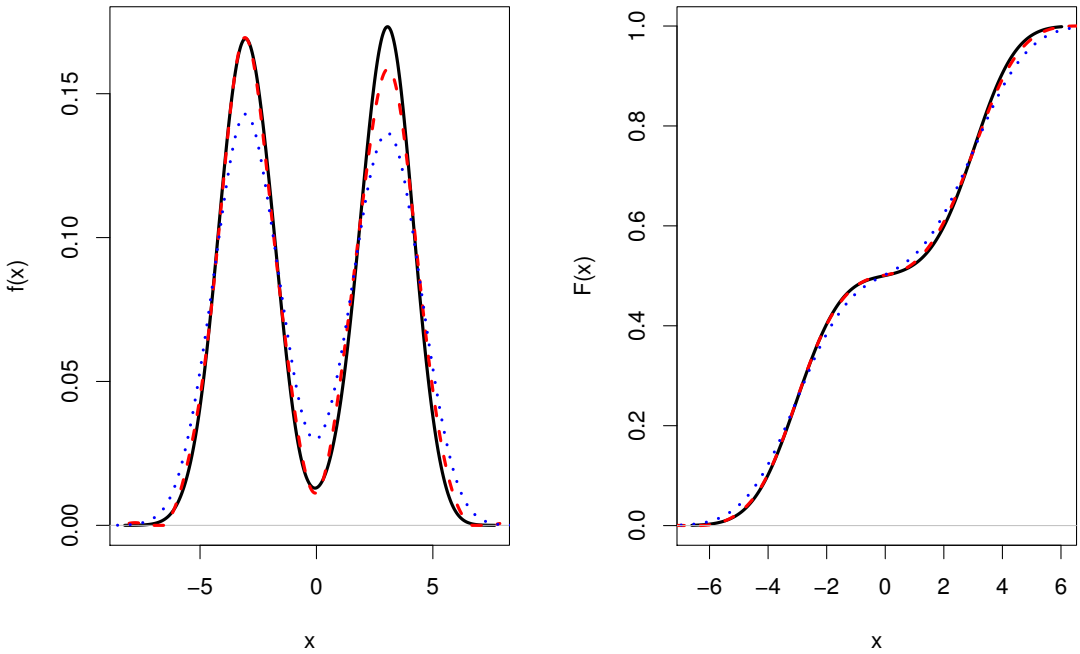


Figure 2: Two simulated examples with homoscedastic errors: (a) Laplacian errors (b) normal errors. The solid lines denote the kernel estimates from the uncontaminated sample; the dashed lines denote the estimate by DKM; the dotted lines denote the kernel estimates ignoring measurement errors.

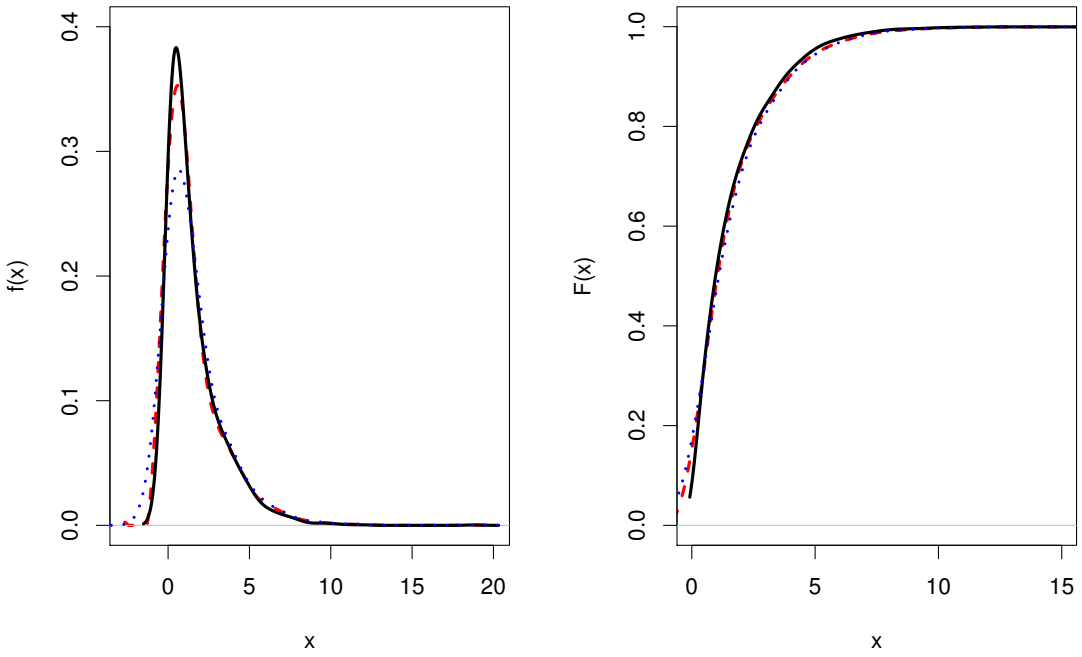


Figure 3: A simulated example with heteroscedastic errors: The solid lines denote the kernel estimates from the uncontaminated sample; the dashed lines denote the estimate by DKM; the dotted lines denote the kernel estimates ignoring measurement errors.

```
R> n3 <- 2000
R> x3 <- rchisq(n3, df = 1.5, ncp = 0)
R> sig3 <- 0.7 + x3/max(x3)
R> u3 <- sapply(sig3, function(x) rnorm(1, sd = x))
R> w3 <- x3 + u3
R> bw3 <- bw.dboot1(w3, sig = sig3, error = "normal")
R> f3 <-  DeconPdf(w3, sig3, error = "normal", bw = bw3, fft = TRUE)
R> F3 <-  DeconCdf(w3, sig3, error = "normal", bw = bw3)
R> par(mfrow = c(1,2))
R> plot(f3, col = "red", lwd = 3, lty = 2, xlab = "x", ylab = "f(x)",
+    main = "")
R> lines(density(x3, adjust = 2), lwd = 3, lty = 1)
R> lines(density(w3, adjust = 2), col = "blue", lwd = 3, lty = 3)
R> plot(F3, col = "red", lwd = 3, lty = 2, xlab = "x", ylab = "F(x)",
+    main = "")
R> lines(SDF(x3), lwd = 3, lty = 1)
R> lines(SDF(w3), col = "blue", lwd = 3, lty = 3)
```

Figure 3 displays the analysis results. We see that, even with the complex heteroscedastic model, our deconvolution density estimator works beautifully to recover the true density. We also notice that the effect of measurement errors on the distribution function $F(x)$ is relatively small with the pre-specified levels of error variances. However, our deconvolution distribution estimator still can correct the bias to a certain degree.

5.3. Simulated example of nonparametric regression with error in variables

Our last simulated example is to demonstrate the use of the function `DeconNpr` for estimating the regression function with errors-in-variables. We simulate the true covariates from the mixed normal $0.5N(2,1) + 0.5N(-2,1)$. The measurement errors are from $N(0, 0.8^2)$ and the regression random errors are from $N(0, 0.2^2)$. The true regression function is set to $m(x) = x^2 - 2x$. The R codes are displayed as follows.

```
R> n <- 2000
R> x <- c(rnorm(n/2, 2, 1), rnorm(n/2, -2, 1))
R> sig <- 0.8
R> u <- sig * rnorm(n)
R> w <- x + u
R> e <- rnorm(n, sd = 0.2)
R> y <- x^2 - 2 * x + e
R> m1 <- DeconNpr(w, sig, y , error = "normal")
R> plot(m1, col = "red", lwd = 3, lty = 2, xlab = "x", ylab = "m(x)",
+    main = "")
R> lines(ksmooth(x, y, kernel = "normal", 2, lwd = 3, lty = 1)
R> lines(ksmooth(w, y, kernel = "normal", 2, col = "blue", lwd = 3, lty = 3)
```

Figure 4 displays the results from the simulation study. We note that the dashed line (the deconvolution estimate) is very close to the solid line (the kernel estimate from the uncontaminated sample), while the dotted line (the the kernel estimates from the contaminated

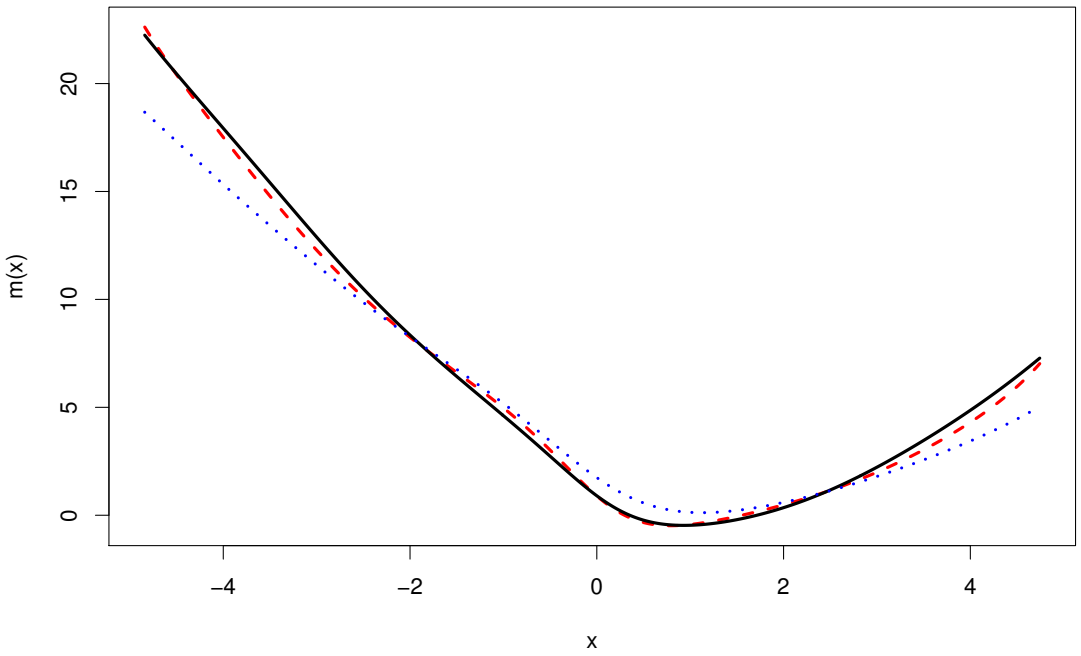


Figure 4: A simulated example for regression with errors-in-variables: The solid line denotes the kernel estimate from the uncontaminated sample; the dashed line denotes the estimate by DKM; the dotted line denotes the kernel estimate ignoring measurement errors.

sample) is away from the above two curves. The deconvolution method works well in errors-in-variables problems.

5.4. Real data applications

Framingham data

The first real data example is from the Framingham Study on coronary heart disease described by Carroll *et al.* (2006). The data consist of measurements of systolic blood pressure (SBP) obtained at two different examinations in 1,615 males on an 8-year follow-up from the first examination. At each examination, the SBP was measured twice for each individual. The *framingham* data in the package contain four variables, "SBP11", "SBP12", "SBP21", "SBP22". We first take the average of the two measurements at each examination. Our goal here is to recover the density function of SBP measured at Exam 2. We use SBP at Exam 1 only to estimate the measurement error variance, but deconvolve SBP measured at Exam 2. Let us assume the measurement errors are normally distributed from $N(0, \sigma^2)$, and denote $SBP1$ and $SBP2$ to be the SBP measured at Exams 1 and 2, respectively. Thus, we have $SBP1|X, SBP2|X \sim N(X, \sigma^2)$, where X denotes the unobserved true blood pressure. It is easy to see $SBP1 - SBP2|X \sim N(0, 2\sigma^2)$. Therefore, the standard deviation of the measurement error can be estimated from difference between $SBP1$ and $SBP2$.

```
R> data(framingham)
R> SBP1 <- (framingham$SBP11 + framingham$SBP12)/2
R> SBP2 <- (framingham$SBP21 + framingham$SBP22)/2
R> sig <- sqrt(0.5 * var(SBP1-SBP2))
```

Graphically checking the distribution of $SBP1-SBP2$ supports the normal assumption of the measurement errors (the left panel of Figure 5). The observed SBP2 has mean 130.01, variance 395.65, and the estimated measurement error variance is 83.69. In this real data application, using the bootstrap bandwidth selection method without resampling cannot obtain an optimal bandwidth. The reason is that $\widehat{MISE}^*_{boot}(h)$ in (16) is not a concave function in the real study. Hence we consider an iterative bootstrap method with resampling to estimate the bandwidth.

```
R> bw0 <- bw.dnrd(SBP2, sig = sig, error = "normal")
R> temp <- bw0
R> ibw <- rep(0, 20)
R> for (i in 1:20) {
+    temp <- bw.dboot2(SBP2, sig = sig, h0 = temp, error = "normal",
+      B = 1000)
+    ibw[i] <- temp
R> }
R> ibw1 <- mean(ibw)
R> SBP2.dec <- DeconPdf(SBP2, sig = sig, error = "normal", bw = ibw1,
+    fft = TRUE)
R> plot(SBP2.dec, lwd = 3, main = "")
R> lines(density(SBP2, adjust = 1.6), lty = 3, lwd = 3, col = "blue")
```

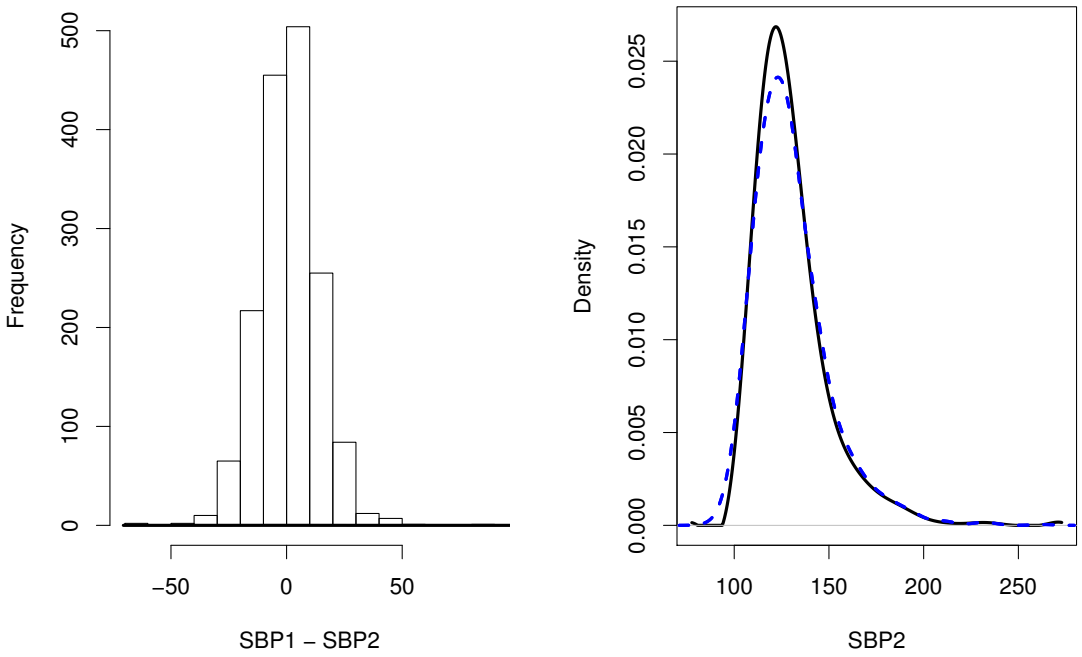


Figure 5: Density deconvolution of $SBP2$ in Framingham data. In the left panel, the histogram of $SBP1 - SBP2$ is displayed to examine graphically the distribution of measurement errors. In the right panel, the solid line denotes the deconvolution density estimate and the dotted line denotes the kernel estimate ignoring measurement error.

The right panel of Figure 5 displays the analysis result. It is noted that the kernel estimate ignoring measurement error underestimates the peak of the density function of the unobserved variables. The deconvolution method also corrects the bias on the left side of the naive density estimate.

Galaxy data

The astronomical position-velocity data set is partially from a sample of 26 low surfaces brightness (LSB) galaxies (De Blok, McGaugh, and Rubin 2001). The data contain 318 stars with their radiuses in kiloparsec (kpc), and observed velocities of stars in km/s (relative to center, corrected for inclination) from 26 LSB galaxies. It was known that the velocities were measured with errors. In the data set, each velocity includes its estimated standard deviation of measurement errors. Here we shall investigate the nonlinear relation between *Velocity* (V) and *Radius* (Rkpc). It is reasonable to assume that *Velocity* is the covariate measured with heteroscedastic normal error, hence we consider the deconvolution method using the estimator in (12). The bandwidth is chosen by eye to be as small as possible while retaining smoothness.

```
R> data(galaxy)
R> m1 <- DeconNpr(galaxy$V, galaxy$Err, galaxy$Rkpc, error = "normal",
+    bw = 9.3)
R> plot(m1, xlim = c(0,250), ylim = c(0,15), lwd = 3, main = "", xlab = "x")
R> lines(ksmooth(galaxy$V, galaxy$Rkpc, kernel="n", 61.8), lwd = 3,
+    lty = 2, col = 4)
```

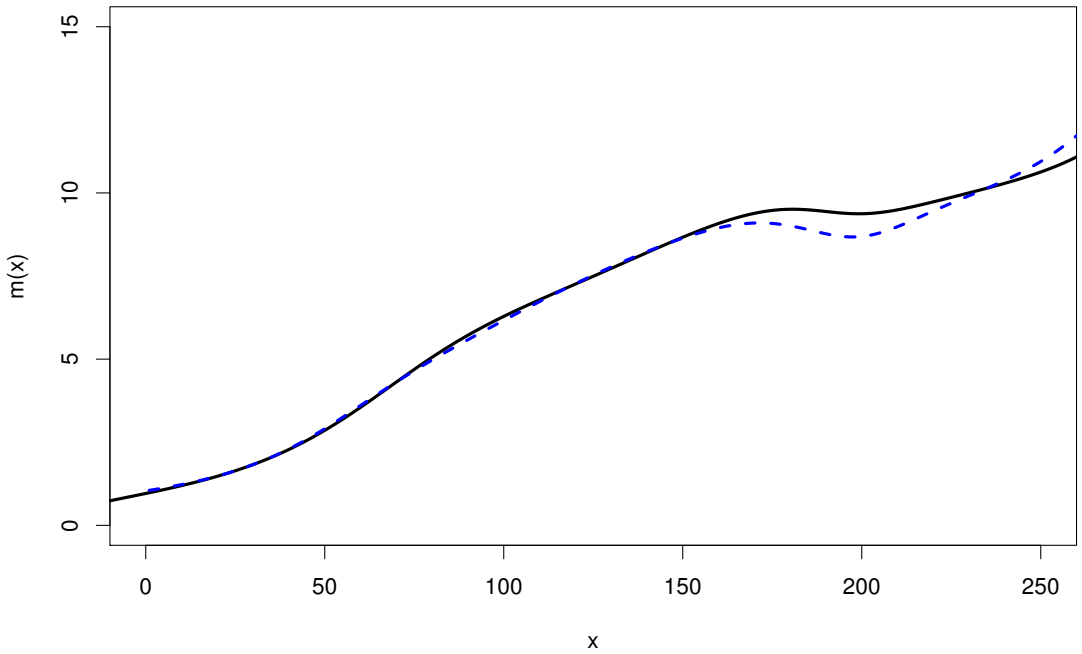


Figure 6: Regression with errors-in-variables in Galaxy data: The solid line denotes the deconvolution regression function estimate and the dashed line denotes the kernel estimate ignoring measurement error.

Figure 6 displays the analysis results, where the solid line denotes the deconvolution regression function estimate and the dashed line denotes the kernel estimate ignoring measurement errors. Both estimated curves show that the radius increases nonlinearly as the velocity increases. Both curves are accordant when the velocity is small, nevertheless, they are different when the velocity becomes large. This is probably due to the fact that the variance of measurement errors is small with low velocity stars but becomes large with high velocity stars.

6. Discussion

In this paper we have illustrated the use of R package **decon** with simulation, visualization, and real data analysis for the measurement error problems. To our knowledge, the package is the first publicly-available software for the estimation in nonparametric measurement error models. We adapt the FFT algorithm for density estimation with error-free data to the deconvolution kernel density and regression with errors-in-variables. The deconvolution estimators thus become computationally efficient in R. By providing such specialist functionality within a standard software package as R we hope to make statistical analysis of data with measurement error a bit more routine.

Extensions towards the software package with more complex measurement error problems can be done in the future. It is of interest to implement estimation methods for deconvolution with repeated measurements (Delaigle, Hall, and Meister 2008), and nonparametric prediction in measurement error models (Carroll *et al.* 2009). In errors-in-variables problems, we only implement the deconvolution kernel regression estimator, which is a special case of the

local polynomial regression estimator with errors-in-variables, proposed by Delaigle, Fan, and Carroll (2009) recently. We are also interested in implementing the local polynomial method in the future.

The confidence band construction in nonparametric measurement error problems is very challenging. The literature on this topic is limited. Bissantz, Dümbgen, Holzmann, and Munk (2007) studied nonparametric confidence bands in density deconvolution with the homoscedastic ordinary smooth error. It has been shown that asymptotic confidence bands are of little use in practice because of the extremely low convergence rate. The confidence bands of the deconvolution density may be obtained by bootstrap algorithms using the series of functions in the package. The confidence band construction of regression curve with errors-in-variables remains as an open problem.

We focus on the DKM in this paper and in the package. There are existing non-Fourier type methods for the nonparametric measurement error models. (More discussion can be found in Sun and Wang (2009)). SIMEX is a popular simulation-based approach for measurement error problems. Staudenmayer, Ruppert, and Buonaccorsi (2008) presented a Bayesian method for density deconvolution, which involves a spline-based density estimation with a Monte Carlo Markov chain and a random-walk Metropolis-Hastings algorithm. Developing a statistical software package for these non-Fourier type methods will be of interest.

Acknowledgments

We are grateful to the reviewers for their valuable comments. We also thank Dr. Cynthia Schneider for her help in revising this paper. The research of Xiao-Feng Wang is supported in part by the NIH grant UL1 RR024989.

References

Azzalini A (1981). "A Note on the Estimation of a Distribution Function and Quantiles by a Kernel Method." *Biometrika*, **68**, 326–328.

Bissantz N, Dümbgen L, Holzmann H, Munk A (2007). "Nonparametric Confidence Bands in Deconvolution Density Estimation." *Journal of the Royal Statistical Society B*, **69**, 483–506.

Carroll RJ, Delaigle A, Hall P (2009). "Nonparametric Prediction in Measurement Error Models." *Journal of the American Statistical Association*, **104**(487), 993–1003.

Carroll RJ, Hall P (1988). "Optimal Rates of Convergence for Deconvolving a Density." *Journal of the American Statistical Associations*, **83**, 1184–1186.

Carroll RJ, Ruppert D, Stefanski LA, Crainiceanu C (2006). *Measurement Error in Nonlinear Models: A Modern Perspective*. 2nd edition. Chapman Hall, New York.

Comte F (2004). "Kernel Deconvolution of Stochastic Volatility Models." *Journal of Time Series Analysis*, **25**(4), 563–582.

De Blok WJG, McGaugh SS, Rubin VC (2001). "High-Resolution Rotation Curves of Low Surface Brightness Galaxies: Mass Models." *The Astronomical Journal*, **122**, 2396–2427.

Delaigle A, Fan J, Carroll RJ (2009). "A Design-Adaptive Local Polynomial Estimator for the Errors-in-Variables Problem." *Journal of the American Statistical Association*, **104**(485), 348–359.

Delaigle A, Gijbels I (2004a). "Bootstrap Bandwidth Selection in Kernel Density Estimation from a Contaminated Sample." *Annals of the Institute of Statistical Mathematics*, **56**(1), 19–47.

Delaigle A, Gijbels I (2004b). "Practical Bandwidth Selection in Deconvolution Kernel Density Estimation." *Computational Statistics & Data Analysis*, **45**, 249–267.

Delaigle A, Hall P (2008). "Using SIMEX for Smoothing-Parameter Choice in Errors-in-Variables Problems." *Journal of the American Statistical Association*, **103**(481), 280–287.

Delaigle A, Hall P, Meister A (2008). "On Deconvolution with Repeated Measurements." *The Annals of Statistics*, **36**(2), 665–685.

Delaigle A, Meister A (2007). "Nonparametric Regression Estimation in the Heteroscedastic Errors-in-Variables Problem." *Journal of the American Statistical Association*, **102**, 1416–1426.

Delaigle A, Meister A (2008). "Density Estimation with Heteroscedastic Error." *Bernoulli*, **14**, 562–579.

Efromovich S (1997). "Density Estimation for the Case of Supersmooth Measurement Error." *Journal of the American Statistical Association*, **92**(438), 526–535.

Fan J (1991). "On the Optimal Rates of Convergence for Nonparametric Deconvolution Problems." *The Annals of Statistics*, **19**, 1257–1272.

Fan J (1992). "Deconvolution with Supersmooth Distributions." *Canadian Journal of Statistics*, **20**, 155–169.

Fan J, Truong YK (1993). "Nonparametric Regression with Errors in Variables." *The Annals of Statistics*, **21**(4), 1900–1925.

Faraway JJ, Jhun MS (1990). "Bootstrap Choice of Bandwidth for Density-Estimation." *Journal of the American Statistical Association*, **85**(412), 1119–1122.

Godfrey PJ, Ruby A, Zajicek OT (1985). "The Massachusetts Acid Rain Monitoring Project: Phase 1." In *Water Resource Research Center*. University of Massachusetts.

Hall P, Lahiri SN (2008). "Estimation of Distributions, Moments and Quantiles in Deconvolution Problems." *The Annals of Statistics*, **36**(5), 2110–2134.

Hesse CH (1999). "Data-Driven Deconvolution." *Journal of Nonparametric Statistics*, **10**(4), 343–373.

Meister A (2004). "On the Effect of Misspecifying the Error Density in a Deconvolution Problem." *Canadian Journal of Statistics*, **32**(4), 439–449.

Meister A (2009). *Deconvolution Problems in Nonparametric Statistics*. Springer-Verlag, New York.

Morrison HL, Mateo M, Olszewski EW, Harding P, *et al.* (2000). "Mapping the Galactic Halo I: The "Spaghetti" Survey." *The Astronomical Journal*, **119**, 2254–2273.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Silverman BW (1982). "Kernel Density-Estimation Using the Fast Fourier-Transform." *Journal of the Royal Statistical Society C*, **31**(1), 93–99.

Staudenmayer J, Ruppert D, Buonaccorsi J (2008). "Density Estimation in the Presence of Heteroskedastic Measurement Error." *Journal of the American Statistical Association*, **103**, 726–736.

Stefanski LA, Carroll RJ (1990). "Deconvoluting Kernel Density Estimators." *Statistics*, **21**, 169–184.

Sun J, Wang XF (2009). "Comment on Nonparametric Prediction in Measurement Error Models by Carroll, R.J., Delaigle, A., and Hall, P." *Journal of the American Statistical Association*, **104**, 1012–1013.

van Es B, Uh HW (2005). "Asymptotic Normality of Kernel-Type Deconvolution Estimators." *Scandinavian Journal of Statistics*, **32**(3), 467–483.

Wang XF, Fan Z, Wang B (2010). "Estimating Smooth Distribution Function in the Presence of Heterogeneous Measurement Errors." *Computational Statistics & Data Analysis*, **54**, 25 – 36.

Wang XF, Wang B (2010). "Simultaneous Confidence Bands and Bootstrap Bandwidth Selection in Deconvolution with Heteroscedastic Error."

Wang XF, Ye D (2010). "Conditional Density Estimation with Measurement Error."

Zhang CH (1990). "Fourier Methods for Estimating Mixing Densities and Distributions." *The Annals of Statistics*, **18**, 806–830.

Affiliation:

Xiao-Feng Wang
Department of Quantitative Health Science/Biostatistics Section
Cleveland Clinic Foundation
9500 Euclid Ave
Cleveland OH 44195, United States of America
E-mail: wangx6@ccf.org

Bin Wang
Department of Mathematics and Statistics
University of South Alabama
Mobile, AL 36688, United States of America
E-mail: bwang@jaguar1.usouthal.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 39, Issue 10 *Submitted:* 2010-03-21
March 2011 *Accepted:* 2010-11-04