`lpcde`: Local Polynomial Conditional Density Estimation and Inference

Matias D. Cattaneo[*]   Rajita Chandak[†]   Michael Jansson[‡]   Xinwei Ma[§]

February 29, 2024

Abstract

This paper discusses the R package `lpcde`, which stands for local polynomial conditional density estimation. It implements the kernel-based local polynomial smoothing methods introduced in Cattaneo et al. (2024) for statistical estimation and inference of conditional distributions, densities, and derivatives thereof. The package offers mean square error optimal bandwidth selection and associated point estimators, as well as uncertainty quantification based on robust bias correction both pointwise (e.g., confidence intervals) and uniformly (e.g., confidence bands) over evaluation points. The methods implemented are boundary adaptive whenever the data is compactly supported. The package also implements regularized conditional density estimation methods, ensuring the resulting density estimate is non-negative and integrates to one. We contrast the functionalities of `lpcde` with existing R packages for conditional density estimation, and showcase its main features using simulated data.

1   Introduction

Conditional cumulative distribution functions (CDFs), probability density functions (PDFs), and derivatives thereof, are important parameters of interest in statistics, econometrics, and other data science disciplines. In this article, we discuss the main methodological features of the R package `lpcde` for estimation of and inference on conditional CDFs, PDFs, and derivatives thereof, employing the kernel-based local polynomial smoothing approach introduced in Cattaneo et al. (2024, CCJM hereafter).

Wand and Jones (1995), Fan and Gijbels (1996), Simonoff (2012), and Scott (2015) give textbook introductions to kernel-based density and local polynomial estimation and inference methods. The core idea underlying the estimator introduced in CCJM is to use kernel-based local polynomial smoothing methods to construct an automatically boundary adaptive estimator for CDFs, PDFs, and derivatives thereof. The estimation approach consists of two steps. The first step estimates the conditional distribution function using standard local polynomial regression methods, and the second step applies local polynomial smoothing to the (non-smooth) local polynomial conditional CDF estimate from the first step to obtain a smooth estimate of the CDF, PDF, and derivatives thereof.

For the case of PDF estimation, classical estimation approaches typically employ ratios of unconditional kernel density estimators, the derivative of kernel-based non-linear distribution function

---

[*]Department of Operations Research and Financial Engineering, Princeton University.

[†]Department of Operations Research and Financial Engineering, Princeton University.

[‡]Department of Economics, UC Berkeley.

[§]Department of Economics, UC San Diego.

regression estimators, or local polynomial estimators based on some preliminary density-like approximation. See, for example, Fan et al. (1996), Hall et al. (1999), De Gooijer and Zerom (2003), Hall et al. (2004), and references therein. These approaches are not boundary adaptive unless specific modifications (e.g., boundary corrected kernels) are introduced. CCJM's estimator is conceptually different and is boundary adaptive for a possibly unknown compact support of the data. Furthermore, the estimator has a simple closed form representation, which leads to easy and fast implementation. Unlike some other boundary adaptive procedures, it does not require pre-processing of data, and thus avoids the challenges of hyper-parameter tuning: only one bandwidth parameter needs to be selected for implementation.

Building on the theoretical and methodological work reported in CCJM, the package `lpcde` offers data-driven (pointwise and uniform) estimation and inference methods for conditional CDFs, PDFs, and derivatives thereof, which are automatically valid at interior, near-boundary, and boundary points on the support of both the variable of interest and the conditioning variables. For point estimation, the package offers mean squared error optimal bandwidth selection and associated point estimators. For inference, the package offers valid confidence intervals and confidence bands based on robust bias-correction techniques (Calonico et al., 2018, 2022). Finally, these statistical procedures can be easily used for visualization and graphical presentation of smooth empirical CDFs, PDFs, and derivative thereof. We give an overview of the main methods implemented in the package below, along with a discussion of more specific implementation issues. We also showcase the performance of the package with simulated data.

The package `lpcde` includes two main functions.

- `lpcde()`: This function implements the estimator of interest over a grid of evaluation points on the support of the variable of interest and at a pre-specified conditioning value. The function takes three main inputs: data, a bandwidth, and polynomial orders. When the bandwidth is not specified by the user, the function employs the companion function `lpbwcde()` for automatic, data-driven bandwidth selection. When the polynomial orders are not specified by the user, the function employs the next odd polynomial order relative to the parameter of interest. For example, for CDF estimation, the polynomial orders are set to $\mathfrak{p} = \mathfrak{q} = 1$, while for PDF estimation they are set to $\mathfrak{p} = 2$ and $\mathfrak{q} = 1$, where $\mathfrak{p}$ denotes the polynomial order for the variable of interest, and $\mathfrak{q}$ denotes the polynomial order for the conditioning variables. The Epanechnikov kernel is used by default. However, there are alternative kernel options that can be provided by the user, if desired. This function implements pointwise and uniform inference via robust bias-correction methods, employing the same grid of points used for point estimation.

- `lpbwcde()`: This function implements pointwise and integrated mean square error (IMSE) optimal bandwidth selection for the kernel-based local polynomial smoothing methods introduced in CCJM. The resulting bandwidth selection procedure leads to a IMSE-rate optimal point estimator whenever the difference of polynomial orders and derivatives order of interest is odd (see below for further details). This bandwidth choice is also valid, and in some cases optimal from a distributional approximation perspective, when coupled with robust bias-correction methods for statistical inference.

The methods `coef()`, `confint()`, `vcov()`, `print()`, `plot()` and `summary()` are supported for objects returned by the `lpcde` function, while the methods `coef()`, `print()` and `summary()` are supported for objects returned by the `lpbwcde` function. The `plot()` function builds on the `ggplot2` (Wickham et al., 2021) package in R and can be used for illustrations of conditional CDFs,

PDFs or higher order derivatives and their pointwise or uniform confidence bands for a given value of the conditioning variable(s).

The package `lpcde` contributes to a rather small set of `R` packages on CRAN for estimation and inference about conditional CDF, PDF, and derivatives thereof. More specifically, we identified two other packages that provide related methodology: `hdrcde` (Hyndman et al., 2021) and `np` (Hayfield and Racine, 2008). Table 1 summarizes some of the main differences between those packages and `lpcde`. As is noted in the table, `lpcde` is the only package available in `R` that currently provides both pointwise and uniform confidence interval construction that is asymptotically valid, in addition to producing boundary adaptive estimates, and proper conditional density estimates that are non-negative and integrate to one. These features are unique contributions of the package to the overall `R` toolkit.

Package Function	CDF / Derivative Estimation	Regularized Density	Valid at Boundary	Standard Error	Valid Inference	Confidence Bands	Bandwidth Selection
hdrcde							
cde	×	×	×	×	×	×	✓
np							
npcdens	×	×	×	✓	×	×	✓
lpcde							
lpcde	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of `R` packages for conditional PDF estimation

Notes: (i) all three packages provide conditional PDF point estimation; (ii) bandwidth selection is done via cross-validation in `hdrcde` and `np`, and using plug-in mean squared error approximations in `lpcde`.

The remainder of this article is organized as follows. Section 2 describes the derivation of our estimator along with details on how the bandwidth, covariance matrix and confidence intervals can be constructed. This section also highlights some key computational considerations when implementing the estimator. Section 3 discusses how the various functions and features of the package can be implemented in practice through examples of code snippets with a toy dataset. Section 4 illustrates the performance of the estimator through a simulation excercise. Finally, we conclude in Section 5. Additional information about the `R` package `lpcde`, including replication files and datasets, can be found at https://nppackages.github.io/lpcde/.

2   Methodology

In this section, we give an overview of the methodology implemented in `lpcde`; technical details in full generality can be found in CCJM. We start by considering a random sample $(Y_1, \mathbf{X}_1^\mathrm{T}), \ldots, (Y_n, \mathbf{X}_n^\mathrm{T})$ from the continuously distributed random vector $(Y, \mathbf{X}^\mathrm{T}) \in \mathcal{Y} \times \mathcal{X}$. We assume $\mathcal{Y} \subseteq \mathbb{R}$ is a 1-dimensional and $\mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional possibly, but not necessarily, compactly supported set. The goal is to estimate and conduct inference on the conditional CDF, PDF, and derivatives thereof, of $Y|\mathbf{X}$. Prior to setting up our estimator, the following section establishes all necessary notation.

2.1 Notation

Notation introduced in this section will be used through the remainder of the text. Our parameter of interest is

$$F^{(\mu,\boldsymbol{\nu})}(y|\mathbf{x}) = \frac{\partial^{\mu+|\boldsymbol{\nu}|}}{\partial y^\mu \partial \mathbf{x}^{\boldsymbol{\nu}}} F(y|\mathbf{x}), \qquad F(y|\mathbf{x}) = \mathbb{P}[Y \leq y | \mathbf{X} = \mathbf{x}],$$

where $\mu \in \mathbb{N}_0$ denotes the derivative order with respect to the variable of interest Y and, employing multi-index notation, $\boldsymbol{\nu} \in \mathbb{N}_0^d$ denotes the multi-index for the corresponding derivatives of interest with respect to the conditioning variables $\mathbf{X}$. For example,

- $F(y|\mathbf{x}) = F^{(0,\mathbf{0})}(y|\mathbf{x})$: conditional CDF of $Y|\mathbf{X}$

- $f(y|\mathbf{x}) = F^{(1,\mathbf{0})}(y|\mathbf{x})$: conditional PDF of $Y|\mathbf{X}$,

- $f^{(1,0)}(y|\mathbf{x}) = F^{(2,\mathbf{0})}(y|\mathbf{x})$: derivative (with respect to y) of conditional PDF of $Y|\mathbf{X}$.

To simplify the exposition, we abstract from derivative estimation with respect to the conditioning variables in $\mathbf{X}$, and therefore set $\boldsymbol{\nu} = \mathbf{0}$ for the rest of this article. Consequently,

$$F^{(\mu)}(y|\mathbf{x}) = F^{(\mu,\mathbf{0})}(y|\mathbf{x}) \text{ for } \mu \in \mathbb{N}_0.$$

See CCJM for theoretical and methodological results concerning $|\boldsymbol{\nu}| > 0$, all of which are also implemented in the R package `lpcde`, thereby allowing for estimation of derivatives with respect to $\mathbf{X}$ of the conditional CDF of $Y|\mathbf{X}$.

The following notation is used in constructing and analysing our estimator:

- $\mathbf{e}_\ell$ is the conformable $(\ell+1)$-th unit vector

- $|\mathcal{A}|$ denotes the cardinality of a set $\mathcal{A}$.

- $\mathbf{q}(\mathbf{u})$: $\mathfrak{q}$-th order polynomial expansion for some $\mathfrak{q} \in \mathbb{N}$. It is a $(\mathfrak{q}_d + 1)$-dimensional vector collecting the ordered elements $\mathbf{u}^{\boldsymbol{\nu}}/\boldsymbol{\nu}!$ for $0 \leq |\boldsymbol{\nu}| \leq \mathfrak{q}$, where, employing multi-index notation, $\mathbf{u}^{\boldsymbol{\nu}} = u_1^{\nu_1} u_2^{\nu_2} \cdots u_d^{\nu_d}$, $\boldsymbol{\nu}! = \boldsymbol{\nu}_1! \boldsymbol{\nu}_2! \cdots \boldsymbol{\nu}_d!$, $|\boldsymbol{\nu}| = \nu_1 + \nu_2 + \cdots + \nu_d$, and $\mathfrak{q}_d = (d+\mathfrak{q})!/(\mathfrak{q}!d!) - 1$.

- $\mathbf{p}(u)$: $\mathfrak{p}$-th order polynomial expansion for some $\mathfrak{p} \in \mathbb{N}$. It is a $(\mathfrak{p}+1)$-dimensional vector collecting the ordered elements $u^\mu/\mu!$ for $0 \leq \mu \leq \mathfrak{p}$.

- $K_h(x;u) = K((x-u)/h)/h$, where $K(\cdot)$ is a kernel function and h is a bandwidth.

- $L_h(\mathbf{x};\mathbf{u}) = K_h(x_1 - u_1) K_h(x_2 - u_2) \cdots K_h(x_d - u_d)$.

- $\widehat{\mathbf{S}}_y = \frac{1}{n} \sum_{i=1}^n K_h(y_i;y) \mathbf{p}\left(\frac{y_i-y}{h}\right) \mathbf{p}\left(\frac{y_i-y}{h}\right)^{\mathrm{T}}$

- $\widehat{\mathbf{S}}_{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n L_h(\mathbf{x}_i;\mathbf{x}) \mathbf{q}\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right) \mathbf{q}\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)^{\mathrm{T}}$

- $\widehat{\mathbf{R}}_{y,\mathbf{x}} = \frac{1}{n^2 h} \sum_{j=1}^n \sum_{i=1}^n K_h(y_j;y) \mathbf{p}\left(\frac{y_j-y}{h}\right) L_h(\mathbf{x}_i;\mathbf{x}) \mathbf{q}\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)^{\mathrm{T}} \mathbb{1}(y_i \leq y_j)$,

4

2.2 General estimation idea

The construction of the conditional CDF, PDF and derivatives thereof involves two steps. First, the conditional distribution function $F(y|\mathbf{x})$ is estimated by standard local polynomial methods:

$$\widehat{F}_{\mathfrak{q}}(y|x) = \mathbf{e}_0^{\mathrm{T}}\widehat{\boldsymbol{\gamma}}_{\mathfrak{q}}(y|\mathbf{x}), \qquad \widehat{\boldsymbol{\gamma}}_{\mathfrak{q}}(y|\mathbf{x}) = \operatorname*{argmin}_{\mathbf{c}\in\mathbb{R}^{\mathfrak{q}d}} \sum_{i=1}^{n} \left(\mathbb{1}(y_i \leq y) - \mathbf{q}(\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}\mathbf{c}\right)^2 L_h(\mathbf{x}_i - \mathbf{x}), \quad (1)$$

Note here that the estimator $\widehat{F}_{\mathfrak{q}}(y|x)$ of $F(y|\mathbf{x})$ is not smooth as a function of y and therefore cannot be used to construct an estimator of the conditional PDF and higher-order derivatives with respect to y.

Therefore, in a second step, a smoothed (with respect to y) estimator of the CDF and its derivatives is constructed also using local polynomial methods: for any $0 \leq \mu \leq \mathfrak{p}$,

$$\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x}) = \mathbf{e}_\mu^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{\mathfrak{p},\mathfrak{q}}(y|\mathbf{x}), \quad \widehat{\boldsymbol{\beta}}_{\mathfrak{p},\mathfrak{q}}(y|\mathbf{x}) = \operatorname*{argmin}_{\mathbf{b}\in\mathcal{B}} \sum_{i=1}^{n} \left(\widehat{F}_{\mathfrak{q}}(y_i|\mathbf{x}) - \mathbf{p}(y_i - y)^{\mathrm{T}}\mathbf{b}\right)^2 K_h(y_i - y), \quad (2)$$

where $\mathcal{B}$ is some general constraint set. There are different forms of search space that may be of interest to researchers based on the application for which the estimator is being used. For example, it may be necessary that the first element of $\mathbf{b}$, corresponding to the conditional PDF estimator, be nonnegative. In this setting, the set over which $\mathbf{b}$ is minimized can be defined as $\mathcal{B} = \{\mathbf{b} \in \mathbb{R}^{\mathfrak{p}+1} : \mathbf{e}_1^{\mathrm{T}}\mathbf{b} \geq 0\}$. This case is studied further in Section 2.5. On the other hand, if $\mathcal{B} = \mathbb{R}^{\mathfrak{p}+1}$, no constraints are imposed on $\mathbf{b}$. This is the case that we focus on in the following sections.

2.3 Point estimation

Solving (1) and (2) (with $\mathcal{B} = \mathbb{R}^{\mathfrak{p}+1}$) gives a simple closed form for the general estimator:

$$\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x}) = \mathbf{e}_\mu^{\mathrm{T}}\widehat{\mathbf{S}}_y^{-1}\widehat{\mathbf{R}}_{y,\mathbf{x}}\widehat{\mathbf{S}}_{\mathbf{x}}^{-1}\mathbf{e}_0, \tag{3}$$

A complete derivation is provided in the supplemental material of CCJM.

In the R package, for a choice of derivative μ with respect to y (and a choice of of derivative $\boldsymbol{\nu}$ with respect of $\mathbf{x}$, a choice of polynomial orders $(\mathfrak{p}, \mathfrak{q})$, a choice of bandwidth h and kernel function $K(\cdot)$, the function `lpcde()` implements the estimator $\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})$ over a grid of points on $\mathcal{Y}$ for a given conditioning evaluation point $\mathbf{x}$. By default, the function sets $(\mu, \boldsymbol{\nu}) = (1, 0)$ (conditional PDF), $\mathfrak{q} = 1$ (local linear nonsmooth conditional CDF estimation), $\mathfrak{p} = 2$ (local quadratic smooth conditional CDF estimation), and $K(\cdot)$ is to chosen to be the Epanechnikov kernel. Generally speaking, it is recommended to choose the local polynomial order such that $\mathfrak{p} - \mu$ and $\mathfrak{q} - |\boldsymbol{\nu}|$ are both odd. Although the second-order Epanechnikov kernel is implemented by default, the function `lpcde()` can also be implemented with second-order uniform and triangular kernels by setting the variable `kernel_type` appropriately. The choice of the kernel does not affect the orders of the bias and the variance. Last but not least, the choice of bandwidth h is important: by default, whenever h is not supplied by the user, the function `lpcde()` relies on the companion function `lpbwcde()`, which implements data-driven bandwidth selection based on the minimization of the (approximate) mean squared error of the estimator $\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})$.

In the remainder of this section we review some of the main statistical properties and inference techniques developed in CCJM and the computational considerations in implementing these methods in the package `lpcde`.

5

2.4 Bandwidth selection

Once we have the closed form of the point estimator, we can derive the leading bias and variance of the estimator. The leading bias and variance for odd values of $\mathfrak{p} - \mu$ and $\mathfrak{q} - |\boldsymbol{\nu}|$ take the following form:

$$\text{Bias}\left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})\right] = h^{\mathfrak{q}+1} \sum_{|\mathbf{m}|=\mathfrak{q}+1} F_{p,\mathfrak{q}+1}^{(\mu,\mathbf{m})}(y|\mathbf{x}) B_{\mathbf{m}}^{(i)}(\mathbf{x}) + h^{\mathfrak{p}+1-\mu} F_{\mathfrak{p}+1,\mathfrak{q}}^{(\mathfrak{p}+1)} B_{\mathfrak{p}+1}^{(ii)}(y), \tag{4}$$

$$\text{Var}\left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})\right] = \frac{1}{nh^{d+2\mu+1}} F_{\mathfrak{p},\mathfrak{q}}^{(1)}(y|\mathbf{x}) V_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y,\mathbf{x}). \tag{5}$$

The quantities on the right hand side above implicitly depend on the kernel function. It is straightforward to show that both the bias and variance terms converge in probability to non-random, well-defined limits. Exact expressions and technical details for other cases can be found in the supplemental appendix of CCJM.

Equations (4) and (5) are valid for all evaluation points on the support of the data. As a result, the pointwise mean squared error (MSE) optimal bandwidth can be approximated as

$$h_{\mathfrak{p},\mathfrak{q}}^{\text{MSE}}(y,\mathbf{x}) = \underset{h>0}{\text{argmin}} \left[\text{Var}\left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})\right] + \text{Bias}\left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})\right]^2\right].$$

Under standard regularity conditions, $h^{\text{MSE}}(y,\mathbf{x})$ is MSE-optimal if $\mathfrak{p} - \mu$ and $\mathfrak{q} - |\boldsymbol{\nu}|$ are odd. Precise closed-form expressions for the MSE-optimal bandwidth can be found in the supplemental appendix of CCJM. In practice, the MSE-optimal bandwidth is estimated by plugging-in estimates of the unknown quantities in Equations (4) and (5), given some initial bandwidth choice and then directely solving for the optimal bandwidth.

The IMSE-optimal bandwidth is estimated similarly, with the main difference being that a set of grid points on the support of $\mathcal{Y}$ is used to approximate the integral. Detailed expressions are given in the supplemental material of CCJM. Bandwidth selection is implemented through the `lpbwcde()` function.

The number of grid points or specific locations of grid points (default is 19 equally-spaced points over the implied support) can be specified by the user as an input to both the `lpbwcde()` and `lpcde()` functions. For generating quantile-spaced grid points, the flag `grid_spacing` should be set to 'quantile'. Users should be aware of possible issues with using equally-spaced grid points at low-density regions or near boundary points. If a small banwidth is coupled with grid points that have few data points that can be used for estimation, the resulting point estimates as well as standard error approximations may have numerical inaccuracies that cause instability in the output. We recommend either prior checking of effective sample sizes for the choice of bandwidth or choosing quantile-spaced grid points.

2.5 Constrained density estimation

As mentioned in Section 2.2, some applications may require that the conditional density estimate satisfy additional constraints. For example, it may be desirable to ensure that the PDF estimate be non-negative on the support and integrates to one. Fortunately, our two-step formulation of the estimator allows for the non-negativity constraint to be incorporated directly into the second step given in (2):

$$\widehat{f}_{\texttt{N}}(y|\mathbf{x}) = \mathbf{e}_1^{\text{T}} \widehat{\boldsymbol{\beta}}_{\texttt{N}}(y|\mathbf{x}), \qquad \widehat{\boldsymbol{\beta}}_{\texttt{N}}(y|\mathbf{x}) = \underset{\mathbf{b}\in\mathbb{R}^{\mathfrak{p}+1}:\ \mathbf{e}_1^{\text{T}}\mathbf{u}\geq 0}{\text{argmin}} \sum_{i=1}^{n} \left(\widehat{F}(y_i|\mathbf{x}) - \mathbf{p}(y_i - y)^{\text{T}}\mathbf{b}\right)^2 K_h(y_i; y),$$

6

where we use the subscript "N" to denote the non-negative estimator. The solution to this modified optimization problem leads to a simple closed form solution that can be written in terms of the unconstrained estimator $\widehat{f}(y|\mathbf{x})$:

$$\widehat{f}_{\mathtt{N}}(y|\mathbf{x}) = \max\left\{\widehat{f}(y|\mathbf{x}),\ 0\right\}.$$

In order to incorporate the constraint the the estimator also integrate to one, a global constraint must be imposed on the estimator. In CCJM, we propose and study a modification of the $\widehat{f}_{\mathtt{N}}$ based on the Kullback-Leibler divergence,

$$\widehat{f}_{\mathtt{I}}(y|\mathbf{x}) = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \operatorname{KL}\left(g \parallel \widehat{f}_{\mathtt{N}}(\cdot|\mathbf{x})\right), \qquad \text{where } \operatorname{KL}\left(g \parallel f\right) = \int_{\mathcal{Y}} g(y) \log\left(\frac{g(y)}{f(y)}\right) \mathrm{d}y,$$

where the subscript "I" stands for "integrating to one" and $\mathcal{G} = \{g \geq 0 : \int_{\mathcal{Y}} g(y)\mathrm{d}y = 1,\ g(y) = 0 \text{ for } y \notin \mathcal{Y}\}$.

Fortunately, $\widehat{f}_{\mathtt{I}}$ can be written in closed form as

$$\widehat{f}_{\mathtt{I}}(y|\mathbf{x}) = \frac{\widehat{f}_{\mathtt{N}}(y|\mathbf{x})}{\int_{\mathcal{Y}} \widehat{f}_{\mathtt{N}}(u|\mathbf{x})\mathrm{d}u}$$

Uniform rates of convergence as well as distributional convergence of both constrained estimators can be established with slight modifications from the theory that was established for the unconstrained estimator. Crucially, this means we can construct robust bias-correct uniform confidence bands for the constrained estimators as well. Further details regarding the convergence guarantees of the constrained estimators are provided in Section 4 of CCJM.

2.6 Distribution theory and robust bias-corrected inference

In order to conduct inference, we first construct a Wald-type test statistic that has the following distributional convergence

$$T_{\mathfrak{p},\mathfrak{q}}(y, \mathbf{x}) = \frac{\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x}) - F_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})}{\sqrt{\operatorname{Var}\left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})\right]}} \rightsquigarrow \mathcal{N}(B, 1),$$

where $\rightsquigarrow$ denotes weak convergence as $h \to 0$ and $n \to \infty$, $\mathcal{N}$ denotes the Gaussian distribution, and B denotes the standardized asymptotic bias emerging whenever a too "large" bandwidth is employed (e.g., when the MSE-optimal or IMSE-optimal bandwidth is used). See CCJM for details.

As a result, standard confidence intervals with nominal $(1 - \alpha)$ coverage takes the form:

$$\operatorname{CI}(y, \mathbf{x}) = \left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x}) \pm z_{1-\alpha/2}\sqrt{\widehat{\operatorname{Var}}\left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})\right]}\right],$$

where z_{α} is the α-th quantile of the standard normal distribution. However, for "large" bandwidths, this confidence interval would be invalid due to the asymptotic bias, B. In practice, undersmoothing is often used to address the asymptotic bias present. However, Calonico et al. (2018, 2022) show that undersmoothing is sub-optimal under the standard assumptions of the model. Instead, they propose a robust bias-correction (RBC) technique that has better higher-order approximations and asymptotically correct coverage probabilities. RBC requires bias-correction of the point estimator

and then adjusting the variance estimate appropriately to construct a bias-corrected Wald-type statistic.

For our estimator, we first correct for the first-order bias by using a point estimator that is generated by increasing the polynomial order for both variables, y and $\mathbf{x}$. To be specific, we use $\widehat{F}_{\mathfrak{p}+1,\mathfrak{q}+1}^{(\mu)}(y|\mathbf{x}; h_{\mathfrak{p},\mathfrak{q}}^{\mathrm{MSE}})$ in place of $\widehat{F}_{p,q}^{(\mu)}(y|\mathbf{x}; h_{p,q}^{\mathrm{MSE}})$. The bandwidth used is optimal for the point estimate with the lower order polynomials. The asymptotically valid confidence intervals now take the form

$$\mathrm{CI}^{\mathrm{RBC}}(y, \mathbf{x}) = \left[\widehat{F}_{\mathrm{RBC}}^{(\mu)}(y|\mathbf{x}) \pm z_{1-\alpha/2}\sqrt{\widehat{\mathrm{Var}}\left[\widehat{F}_{\mathrm{RBC}}^{(\mu)}(y|\mathbf{x})\right]}\right],$$

where $\widehat{F}_{\mathrm{RBC}}^{(\mu)}(y|\mathbf{x}) \equiv \widehat{F}_{\mathfrak{p}+1,\mathfrak{q}+1}^{(\mu)}(y|\mathbf{x}) = \widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x}) - \widehat{\mathrm{Bias}}\left[\widehat{F}_{\mathfrak{p},\mathfrak{q}}^{(\mu)}(y|\mathbf{x})\right]$.

Additionally, uniform confidence bands can be constructed as

$$\mathrm{CB}^{\mathrm{RBC}}(\mathcal{M}) = \left\{\left[\widehat{F}_{\mathrm{RBC}}^{(\mu)}(y|\mathbf{x}) \pm z_{\mathcal{M},1-\alpha/2}\sqrt{\widehat{\mathrm{Var}}\left[\widehat{F}_{\mathrm{RBC}}^{(\mu)}(y|\mathbf{x})\right]}\right], \quad y \in \mathcal{M}\right\},$$

where $\mathcal{M}$ is a collection of evaluation points on the support $\mathcal{Y}$ and $z_{\mathcal{M},\alpha}$ is the α-quantile over the collection of evaluation points for a normal distribution centered at 0 and with the same variance-covariance matrix as the estimator. The critical value $z_{\mathcal{M},1-\alpha/2}$, is defined by the upper α quantile of the supremum of the simulated Gaussian process on the grid $\mathcal{M}$:

$$z_{\mathcal{M},\alpha} = \inf\left\{u \geq 0: \ \mathbb{P}\left[\sup_{y \in \mathcal{M}} |\widehat{\mathcal{Z}}^{(\mu)}(y|\mathbf{x})| \leq u \middle| \mathrm{Data}\right] \geq 1 - \alpha\right\},$$

where $\widehat{Z}^{(\mu)}(y|\mathbf{x}) \overset{a}{\sim} \mathcal{N}\left(0, \widehat{\mathrm{Cov}}\left[\widehat{F}_{\mathrm{RBC}}^{(\mu)}(y|\mathbf{x})\right]\right)$. The confidence band depends on the entire collection of evaluation points. In lpcde, $z_{\mathcal{M},1-\alpha/2}$, is estimated by using the maximum over the grid points as an approximation for the supremum over $\mathcal{M}$. See CCJM (and its supplemental) for technical details and regularity conditions.

The RBC method leads to confidence intervals/bands that are not centered at the density point estimates since different order polynomials are used for the point estimates and for inference. Thus, it may happen that the point estimates lies outside of the RBC confidence intervals/bands if the underlying distribution has high curvature at some evaluation point(s). One solution in this case is to increase the polynomial orders $\mathfrak{p}$ and $\mathfrak{q}$, or to use a smaller-than-optimal bandwidth.

2.7 Implementation of the covariance estimator

Implementing the variance estimator for both estimating the MSE-optimal bandwidth and constructing confidence intervals, requires careful consideration. It is particularly crucial to consider the computational cost of estimating the variance when employing uniform confidence bands, which requires the construction of the full $|\mathcal{M}| \times |\mathcal{M}|$ covariance matrix in order to approximate the critical value $z_{\mathcal{M},1-\alpha/2}$. The discussion in this section focuses only on the covariance matrix estimation for the conditional PDF, $\widehat{f}(y|\mathbf{x})$, purely for simplicity of presentation.

The standard plug-in estimator (constructed by estimating the unknown quantities in (5)) that is proposed and studied in CCJM has a computational complexity of $O(|\mathcal{M}|^2 n^4)$. For large datasets and a fine grid of evaluation points, this is prohibitively slow to run in practice. As a result the

8

default covariance estimator used in `lpcde` implements a significantly faster jackknife covariance estimatior. The construction of the jackknife estimator relies on the fact that the closed-form of the estimator $\hat{f}$ can be written as a V-statistic to which the Hoeffding decomposition (van der Vaart, 2000, Chapter 11) can be applied. The covariance expression then decomposes to a sum of two independent functions that depend on the evaluation points and a small subset of the data in the neighborhood of the evaluation points, thus reducing the computational cost to only $O(|\mathcal{M}|^2(nh)^2)$.

Here we provide a sketch of how this estimator is constructed. The interested reader can find a complete derivation in Section 6 of the Supplementary Material of CCJM.

We start by first observing that the estimator $\hat{f}(y|\mathbf{x})$ is a V-statistic:

$$\hat{f}(y|\mathbf{x}) = \frac{1}{n^2 h} \sum_{i,j} \mathbb{1}(y_i \le y_j) \mathbf{e}_1^{\mathrm{T}} \hat{\mathbf{S}}_y^{-1} \mathbf{p}\left(\frac{y_j - y}{h}\right) K_h(y_j; y) \mathbf{q}^{\mathrm{T}}\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right) L_h(\mathbf{x}_i; \mathbf{x}) \hat{\mathbf{S}}_\mathbf{x}^{-1} \mathbf{e}_0$$

$$= \frac{1}{n^2} \sum_{i=1}^{n} a(y_i, y) b(\mathbf{x}_i, \mathbf{x}) + \frac{1}{n^2} \sum_{1 \le i \ne j \le n} \mathbb{1}(y_i \le y_j) a(y_j, y) b(\mathbf{x}_i, \mathbf{x}), \tag{6}$$

where

$$a(y_i, y) = h^{-2} \mathbf{e}_1^{\mathrm{T}} \hat{\mathbf{S}}_y^{-1} \mathbf{p}\left(\frac{y_i - y}{h}\right) K\left(\frac{y_i - y}{h}\right),$$

$$b(\mathbf{x}_i, \mathbf{x}) = h^{-d} \mathbf{e}_0^{\mathrm{T}} \hat{\mathbf{S}}_\mathbf{x}^{-1} \mathbf{Q}\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right) L\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right).$$

The scalar functions $a(\cdot)$ and $b(\cdot)$ are are non-zero only for data points that are within h distance of the evaluation point, a feature that the package `lpcde` leverages explicitly to improve numerical performance in applications. The second term in (6) can now be symmetrized and treated as a U-statistic. Then, the Hoeffding decomposition can be applied and plugged back into (6). This leads to a natural alternative jackknife covariance estimator, which is simple to write and computationally efficient:

$$\hat{\mathsf{C}}(y, \mathbf{x}, y', \mathbf{x}') = \frac{1}{n} \sum_{i=1}^{n} \hat{L}_{(i)}(y, \mathbf{x}) \hat{L}_{(i)}(y', \mathbf{x}').$$

where

$$\hat{L}_{(i)}(y, \mathbf{x}) = \frac{2}{n-1} \sum_{j \ne i} \left(u_{i,j} - \hat{f}(y|\mathbf{x}) \right).$$

and $u_{i,j} = \frac{1}{2}(\mathbb{1}(y_i \le y_j) a(y_j, y) b(\mathbf{x}_i, \mathbf{x}) + \mathbb{1}(y_j \le y_i) a(y_i, y) b(\mathbf{x}_i, \mathbf{x}))/2$. In particular, if the two evaluation points are equivalent, then we return the (approximately jackknife) variance estimator

$$\hat{\mathsf{V}}(y, \mathbf{x}) \equiv \hat{\mathsf{C}}(y, \mathbf{x}, y, \mathbf{x}) = \frac{1}{n-1} \sum_{i=1}^{n} \hat{L}_{(i)}^2(y, \mathbf{x}).$$

It can be easily verified that this jackknife covariance estimator is asymptotically equivalent to the theoretical variance expression in (5).

3 Implementation

In this section we illustrate some of the main features of the package with a simulated dataset. We consider a bi-variate jointly normal data generating process with mean 0 and variance 1.

Figures 1, 2 and 3 illustrate the performance of the package on a dataset generated from bi-variate jointly standard normal random variables for generating point estimates produced by `lpcde()` for the conditional (i) CDF (see Figures 1a and 1b), (ii) PDF (see Figures 2a and 2b), and (iii) first derivative (see Figures 3a and 3b) with both standard and robust bias-corrected confidence bands generated by the function `lpcde()`. The true functions are plotted in red for comparison.
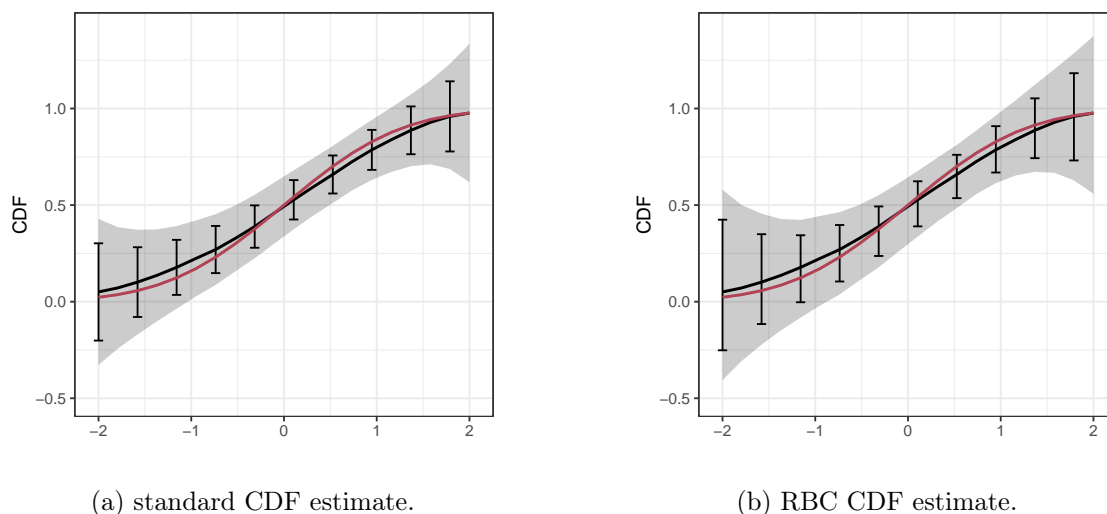

(a) standard CDF estimate.

(b) RBC CDF estimate.

Figure 1: Point estimates for the CDF Left: 95% standard confidence interval/bands. Right: 95% robust bias-corrected confidence intervals/bands.

(a) standard PDF estimate.

(b) RBC PDF estimate.
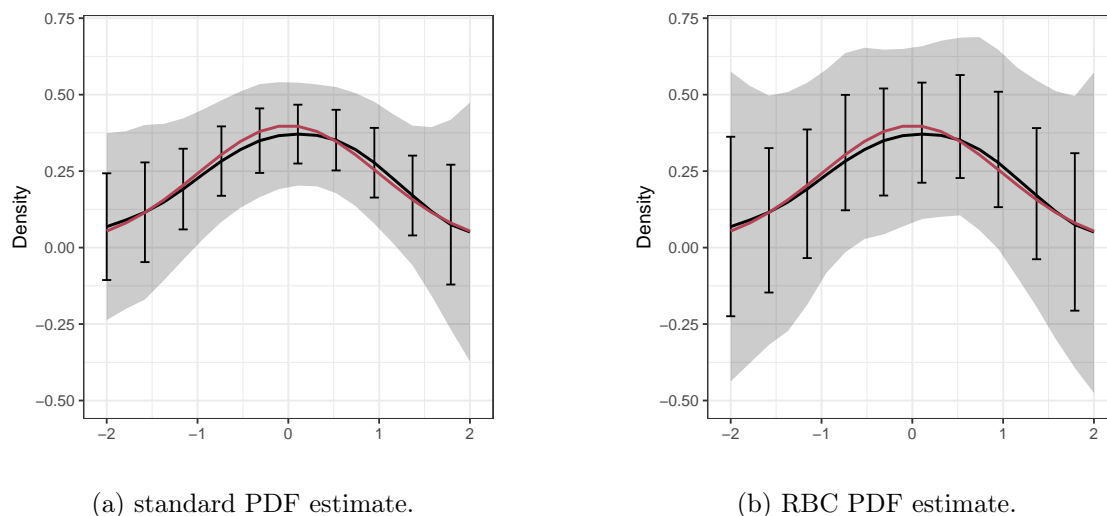
Figure 2: Point estimates for the PDF. Left: 95% standard confidence interval/bands. Right: 95% robust bias-corrected confidence intervals/bands.

(a) standard first derivative estimate.
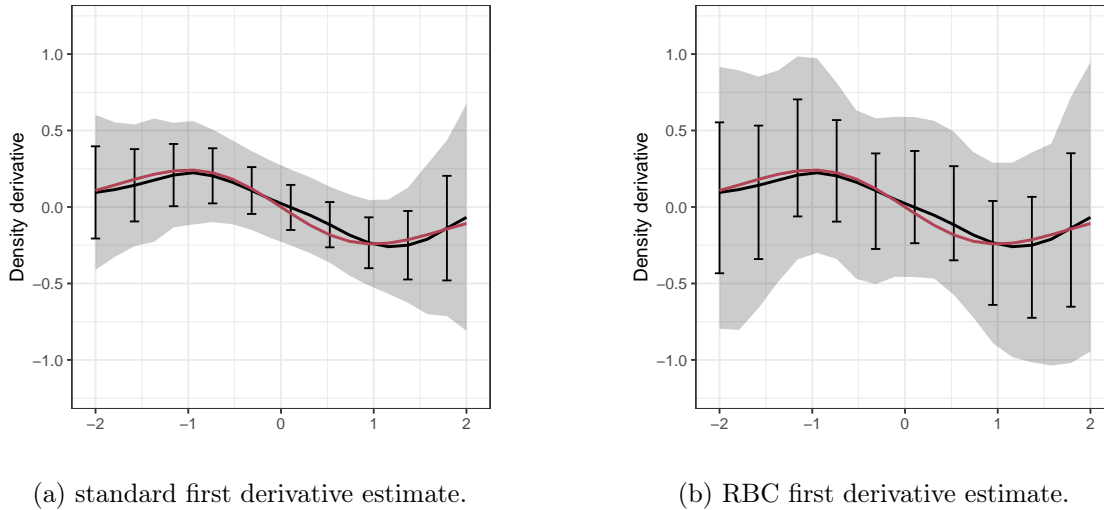
(b) RBC first derivative estimate.

Figure 3: Point estimates for the first derivative. Left: 95% standard confidence interval/bands. Right: 95% robust bias-corrected confidence intervals/bands.

In the remainder of this section we discuss how each of the functions provided in `lpcde` can be used with the aid of code snippets.

3.1 Density estimation

The function `lpcde()` provides information on point estimates, standard errors and confidence interval or bands for a given value of $\mathbf{x}$ over a range of grid points for y. If the grid points are not provided by the user, the function chooses nineteen equally-spaced grid points over the implied support of the data and, if no bandwidth is provided, computes the rule-of-thumb MSE bandwidth at each point.

The following example estimates the conditional density at $\mathbf{x} = 0$, with a fixed bandwidth of 1, using the default local polynomial approximation $\mathfrak{p} = 2$, $\mathfrak{q} = 1$. RBC confidence intervals over the grid are also computed, in this case using the default polynomial orders $\mathfrak{p} = 3$, $\mathfrak{q} = 2$.

```
R> set.seed(42)
R> n = 1000
R> x_data = as.matrix(stats::rnorm(n, mean = 0, sd = 1))
R> y_data = as.matrix(stats::rnorm(n, mean = x_data, sd = 1))
R> y_grid = seq(from=-2, to=2, length.out=10)
R> model1 = lpcde(y_data=y_data, x_data=x_data, y_grid=y_grid, x=0, bw = 1)
R> summary(model1)
```

The function returns an object of type `lpcde`. Standard R methods, `coef()`, `confint()`, `vcov()`, `print()`, `plot()` and `summary()`, can be used on objects of type `lpcde` to understand the output.

Below we reproduce the output of running the `summary` command on `model1`. The first part of the summary output provides basic information about some of the options specified to the function. The second part provides relevant information for each point estimate generated in a table with 7 columns, (i) grid evaluation points, (ii) bandwidth used at each point, (iii) effective number of data points used to generate the point estimate, (iv) point estimate, (v) standard error, (vi) lower $(1 - \alpha)$-confidence interval, and, (vii) upper $(1 - \alpha)$-confidence interval.

```
Call: lpcde

Sample size                                            1000
Polynomial order for Y point estimation       (p=)    2
Polynomial order for X point estimation       (q=)    1
Density function estimated                    (mu=)   1
Order of derivative estimated for covariates (nu=)    0
Kernel function                                        epanechnikov
Bandwidth method
```

				Point	Std.	Robust B.C.	
Index	Grid	B.W.	Eff.n	Est.	Error	[95% C.I.]	
1	-2.0000	1.0000	73	0.0073	0.0236	-0.0401 ,	0.0349
2	-1.5556	1.0000	159	0.0745	0.0086	-0.0103 ,	0.0605
3	-1.1111	1.0000	292	0.1873	0.0064	0.1454 ,	0.1961
4	-0.6667	1.0000	433	0.3473	0.0041	0.3760 ,	0.4178
5	-0.2222	1.0000	532	0.4584	0.0031	0.5046 ,	0.5374
6	0.2222	1.0000	528	0.4715	0.0027	0.5007 ,	0.5323
7	0.6667	1.0000	444	0.3624	0.0031	0.4037 ,	0.4386
8	1.1111	1.0000	304	0.1882	0.0056	0.1609 ,	0.2071
9	1.5556	1.0000	166	0.0752	0.0076	0.0060 ,	0.0735
10	2.0000	1.0000	67	0.0206	0.0171	-0.0468 ,	0.0300

By default, the function provides estimates according to the original formulation of the estimator $\widehat{f}(y|\mathbf{x})$. If a constrained density estimate that is non-negative and integrates to one ($\widehat{f}_{\mathrm{I}}$ as defined in Section 2.5) is desired, the flags `nonneg` and `normalize` can be turned on.

```
R> model1 = lpcde(y_data=y_data, x_data=x_data, y_grid=y_grid, x=0,
+            bw = 1, nonneg=TRUE, normalize=TRUE)
```

Figure 4 shows how the estimates differ when the additional constraints are imposed:
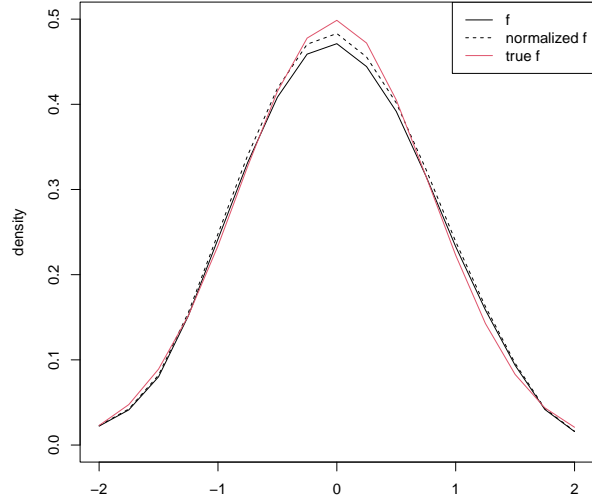
Figure 4: Comparing standard density estimate with normalized estimate.

3.2   Plotting

The `plot()` function uses the `ggplot2` package (Wickham et al., 2021) with objects of type `lpcde` to produce illustrations of point estimates and confidence intervals and/or bands. A simple plot of the conditional PDF with 95% confidence intervals can be generated by running the following code.

```
R> plot(model1) + theme(legend.position = "none")
```

This code snippet produces an image of the type shown in Figure 5.
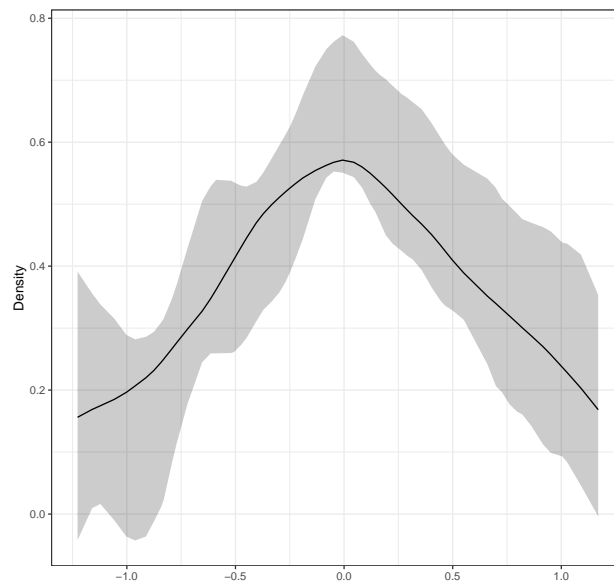

Figure 5: A simple density plot with robust 95% confidence intervals

By default the `plot()` function plots pointwise confidence intervals at 95% level with the point estimates. Additional options for confidence levels, bands and RBC inference are detailed in the package manual. Editing other visual aspects of the plots can be done by providing standard inputs to `ggplot2` functions.

3.3 Bandwidth selection

`lpbwcde()` implements the rule-of-thumb MSE- and IMSE- bandwidth selection by implementing the formulae provided in Section 2.3.

By default `lpbwcde()` computes the rule-of-thumb MSE optimal bandwidth for the conditional PDF with locally quadratic polynomial in y and locally linear polynomial in $\mathbf{x}$ and Epanechnikov kernel on nineteen equally-spaced grid points on the implied support of $\mathcal{Y}$ determined by the observed data. The output of this function is similar to that of `lpcde()` and provides basic information for the data and options specified. The summary of objects returned by this function additionally provides a table with three columns: (i) `y_grid`: values of the grid points for which the bandwidth is estimated, (ii) `B.W.`: the estimated bandwidth corresponding to each grid point, and (iii) `Eff.n.`: the number of effective data points at each evaluation point given the estimated bandwidth. An example of standard bandwidth selection is provided in the following output.

```
R> model2 = lpbwcde(y_data=y_data, x_data=x_data, x=0,
+           y_grid = y_grid, bw_type = "mse-rot")
R> summary(model2)


Call: lpbwcde

Sample size                                          1000
Polynomial order for Y point estimation        (p=)  2
Polynomial order for X point estimation        (q=)  1
Density function estimated                     (mu=) 1
Order of derivative estimated for covariates (nu=)   0
Kernel function                                      epanechnikov
Bandwidth method                                     mse-rot


==================================
Index    y_grid      B.W.    Eff.n
==================================
1       -1.8168    1.0651     117
2       -1.1740    1.7510     687
3       -0.7777    1.4587     681
4       -0.3617    1.1117     594
5       -0.0185    1.0589     577

----------------------------------
6        0.3476    1.1253     586
7        0.7088    1.4232     673
8        1.1133    1.8398     729
9        1.7728    1.0793     125
==================================
```

The estimated bandwidth from this function can be used as bandwidth input to `lpcde()` directly by using the option of `bwselect` to specify bandwidth selection type instead of running `lpbwcde()` first.

4   Simulations

In this section we illustrate the effectiveness of our estimator with a Monte Carlo study. For the sake of simplicity, we set $d = 1$ and assume that $\mathbf{x}$ and y are simulated by a joint normal distribution with variance 2 and covariance -0.1, truncated on $[-1, 1]^2$. We simulate 1000 data sets of 5000 independent samples each. The results of this study are presented in Table 2. The point estimates are generated on 20 evenly spaced grid points on $[0, 1]$ for y.

   We look at the performance of the estimator at three different derivative orders for the predictor variable, y, (i) the CDF, corresponding to $\mu = 0$, (ii) the PDF, corresponding to $\mu = 1$, and (iii) the first derivative of the conditional PDF with respect to y, corresponding to $\mu = 2$, and three different conditional values of the covariate, $\mathbf{x}$: (a) interior, (b) near-boundary and (c) at-boundary. For each conditional value, we present the average bandwidth, average bias, standard deviation, pointwise and uniform 95% coverage, and width of the confidence intervals/bands across the simulated datasets. We present these results for both the standard estimate (rows "WBC") which is generated with a quadratic polynomial ($\mathfrak{p} = 2$) with respect to the variable y, and linear polynomial ($\mathfrak{q} = 1$) with respect to the variable $\mathbf{x}$, as well as the robust bias-corrected estimates (rows "RBC") which uses cubic polynomial ($\mathfrak{p} = 3$) for y and quadratic polynomial ($\mathfrak{q} = 2$) for $\mathbf{x}$. Notice that robust bias-correction provides more accurate coverage, both pointwise and uniformly over the support across all derivative orders. We recommend, for accurate coverage probabilities, using uniform confidence bands with robust bias-corrected estimates.

   We can also analyze the pointwise-in-y performance of our estimator and inference methods. Table 3 presents the average pointwise results of 5000 simulated data sets. We consider three different evaluation points on the support of y for the conditional PDF and three different values of the conditioning variable $\mathbf{x}$. The first four columns of the table present average pointwise MSE-optimal bandwidth used in estimation, effective sample size at each evaluation point, bias, and standard error. The last four columns are the average pointwise confidence interval coverage and width of the confidence interval for the standard estimate and inference method ("WBC") and robust bias-corrected estimate and inference ("RBC"). Robust bias-corrected inference produces accurate empirical coverage across all pointwise combinations.

   Finally, we test the rule-of-thumb MSE bandwidth selection by simulating point estimation and coverage at varying bandwidth values. We choose the range of bandwidth values to be between 0.5 and 1.3 times the average ROT bandwidth chosen by the `lpbwcde()` function. Table 4 presents the average bias, standard error, root mean-squared error, pointwise coverage rate and average width of confidence intervals for 5000 simulations at the interior point $y = 0$, $\mathbf{x} = 0$.

5   Conclusion

This article introduced the software package `lpcde`, which implements local polynomial kernel based regression estimation and inference for conditional densities and higher-order derivatives. Additional information and replication files can be found at https://nppackages.github.io/lpcde/.

		$\widehat{h}_{\mathrm{ROT}}$	bias	se	Coverage		Average Width	
					Pointwise	Uniform	Pointwise	Uniform
CDF ($\mu = 0$)								
$\mathbf{x} = 0$	WBC	0.33	0.15	0.11	73.2	83.9	0.42	0.60
	RBC	0.33	0.15	0.13	85.8	93.5	0.51	0.73
$\mathbf{x} = 0.8$	WBC	0.18	0.18	0.05	21.7	34.7	0.19	0.28
	RBC	0.18	0.18	0.14	70.8	91.7	0.54	0.82
$\mathbf{x} = 1.0$	WBC	0.20	0.19	0.05	18.4	30.5	0.18	0.27
	RBC	0.20	0.19	0.14	68.1	90.8	0.53	0.80
PDF ($\mu = 1$)								
$\mathbf{x} = 0$	WBC	0.38	0.09	0.03	60.4	78.9	0.11	0.23
	RBC	0.38	0.09	0.09	87.9	93.3	0.37	0.68
$\mathbf{x} = 0.8$	WBC	0.35	0.10	0.04	73.0	84.6	0.22	0.40
	RBC	0.35	0.10	0.18	91.6	95.9	0.65	0.81
$\mathbf{x} = 1.0$	WBC	0.38	0.10	0.06	55.8	77.0	0.24	0.45
	RBC	0.38	0.10	0.20	81.2	91.1	0.67	0.81
PDF Derivative ($\mu = 2$)								
$\mathbf{x} = 0$	WBC	0.76	0.07	0.03	30.9	55.2	0.10	0.18
	RBC	0.76	0.07	0.12	67.1	92.2	0.48	0.87
$\mathbf{x} = 0.8$	WBC	0.75	0.09	0.04	35.3	62.0	0.14	0.24
	RBC	0.75	0.09	0.16	74.9	91.4	0.64	1.17
$\mathbf{x} = 1.0$	WBC	0.77	0.09	0.04	38.5	66.9	0.15	0.27
	RBC	0.77	0.09	0.18	78.8	92.9	0.69	1.25

Table 2: Table of average coverage for jointly normal data at three different conditional values. $\mathbf{x} = 0$ is an interior point, $\mathbf{x} = 0.8$ is a near-boundary point, $\mathbf{x} = 1$ is at-boundary. WBC: without bias-correction, RBC: robust bias-corrected.

Eval. point	$\widehat{h}_{\mathrm{ROT}}$	eff.n	bias	se	Coverage		AW	
					WBC	RBC	WBC	RBC
$\mathbf{x} = 0$								
$y = 0$	0.29	249	0.07	0.04	56.4	96.0	0.14	0.49
$y = 0.8$	0.33	374	0.00	0.02	71.4	91.0	0.07	0.22
$y = 1.0$	0.34	284	0.12	0.05	24.8	96.3	0.18	0.40
$\mathbf{x} = 0.8$								
$y = 0$	0.24	140	0.07	0.07	77.4	98.6	0.20	0.68
$y = 0.8$	0.30	297	0.00	0.02	78.0	94.1	0.07	0.24
$y = 1.0$	0.36	256	0.13	0.05	54.0	91.6	0.18	0.37
$\mathbf{x} = 1.0$								
$y = 0$	0.27	131	0.07	0.08	72.4	96.8	0.31	1.04
$y = 0.8$	0.41	345	0.00	0.02	73.2	91.3	0.07	0.19
$y = 1.0$	0.40	221	0.14	0.08	61.1	95.4	0.31	0.49

Table 3: Pointwise results
WBC: without bias-correction, RBC: robust bias-corrected.

$\times \widehat{h}_{\mathrm{MSE}}$	$\widehat{h}$	bias	se	rmse	WBC CR	RBC CR	WBC AW	RBC AW
0.5	0.15	0.07	0.31	0.34	100.0	100.0	1.23	4.14
0.6	0.18	0.07	0.18	0.20	99.7	100.0	0.69	2.37
0.7	0.20	0.07	0.11	0.14	97.7	100.0	0.43	1.46
0.8	0.23	0.07	0.07	0.11	87.8	99.5	0.28	0.96
0.9	0.26	0.07	0.05	0.09	74.8	98.6	0.20	0.67
1	0.29	0.07	0.04	0.08	56.4	96.0	0.14	0.49
1.1	0.32	0.06	0.03	0.07	39.6	89.8	0.11	0.37
1.2	0.35	0.07	0.02	0.07	25.5	81.4	0.08	0.28
1.3	0.38	0.06	0.02	0.07	16.0	72.6	0.06	0.22

Table 4: Bandwidth selection at interior point ($y = 0, \mathbf{x} = 0$).
WBC: without bias-correction, RBC: robust bias-corrected.

6 Acknowledgments

References

Sebastian Calonico, Matias D. Cattaneo, and Max H. Farrell. On the effect of bias estimation on coverage accuracy in nonparametric inference. *Journal of the American Statistical Association*, 113(522):767–779, 2018.

Sebastian Calonico, Matias D. Cattaneo, and Max H. Farrell. Coverage error optimal confidence intervals for local polynomial regression. *Bernoulli*, 28(4):2998–3022, 2022.

Matias D. Cattaneo, Rajita Chandak, Michael Jansson, and Xinwei Ma. Local polynomial conditional density estimators. *Bernoulli*, 2024.

Jan G De Gooijer and Dawit Zerom. On conditional density estimation. *Statistica Neerlandica*, 57(2):159–176, 2003.

Jianqing Fan and Irene Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman & Hall/CRC, 1996.

Jianqing Fan, Qiwei Yao, and Howell Tong. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206, 1996.

Peter Hall, Rodney CL Wolff, and Qiwei Yao. Methods for estimating a conditional distribution function. *Journal of the American Statistical Association*, 94(445):154–163, 1999.

Peter Hall, Jeff Racine, and Qi Li. Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99(468):1015–1026, 2004.

Tristen Hayfield and Jeffrey S. Racine. Nonparametric econometrics: The np package. *Journal of Statistical Software*, 27(5), 2008. URL https://www.jstatsoft.org/v27/i05/.

Rob J Hyndman, Jochen Einbeck, and Matthew P Wand. *hdrcde: Highest Density Regions and Conditional Density Estimation*, 2021. URL https://pkg.robjhyndman.com/hdrcde/. R package version 3.4.

David W Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2015.

Jeffrey S Simonoff. *Smoothing Methods in Statistics*. Springer Science & Business Media, 2012.

Aad W van der Vaart. *Asymptotic Atatistics*. Cambridge University Press, 2000.

M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall/CRC, 1995.

Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2021. URL https://CRAN.R-project.org/package=ggplot2. R package version 3.3.5.