lpdensity

Release 1.0.3

Rajita Chandak

May 27, 2022

CONTENTS:

`lpdensity` implements the local polynomial regression based density (and derivatives) estimator and bandwidth selection tools proposed in Cattaneo, Jansson and Ma (2020). Robust bias-corrected inference methods, both pointwise (confidence intervals) and uniform (confidence bands), are also implemented following the results in Cattaneo, Jansson and Ma (2020, 2021a). See Cattaneo, Jansson and Ma (2021b) for more implementation details and illustrations.

Install the `lpdensity` package by running `pip install lpdensity`.

Import density estimation and bandwidth selection functions by running the following lines

```
>>> from lpdensity import lpdensity
```

```
>>> from lpdensity import lpbwdensity
```

Related `Python`, `R` and `Stata` packages useful for nonparametric estimation and inference are available at https://nppackages.github.io/.

For source code and replication files, visit the lpdensity repository.

CHAPTER

ONE

REFERENCES

Calonico, S., M. D. Cattaneo, and M. H. Farrell. 2018. On the Effect of Bias Estimation on Coverage Accuracy in Nonparametric Inference *Journal of the American Statistical Association*, 113(522): 767-779.

Calonico, S., M. D. Cattaneo, and M. H. Farrell. 2020. Coverage Error Optimal Confidence Intervals for Local Polynomial Regression . Working paper.

Cattaneo, M. D., M. Jansson, and X. Ma. 2020. Simple Local Polynomial Density Estimators. *Journal of the American Statistical Association*, 115(531): 1449-1455.

Cattaneo, M. D., M. Jansson, and X. Ma. 2021a. Local Regression Distribution Estimators *Journal of Econometrics*, forthcoming.

Cattaneo, M. D., M. Jansson, and X. Ma. 2021b. lpdensity: Local Polynomial Density Estimation and Inference *Journal of Statistical Software*, forthcoming.

CHAPTER

TWO

AUTHORS

Matias D. Cattaneo, Princeton University. (cattaneo@princeton.edu).

Rajita Chandak (maintainer), Princeton University. (rchandak@princeton.edu).

Michael Jansson, University of California Berkeley. (mjansson@econ.berkeley.edu).

Xinwei Ma (maintainer), University of California San Diego. (x1ma@ucsd.edu).

2.1 lpdensity

2.1.1 lpdensity

Local Polynomial Density Estimation and Inference

Description

`lpdensity` implements the local polynomial regression based density (and derivatives) estimator proposed in Cattaneo, Jansson and Ma (2020). Robust bias-corrected inference methods, both pointwise (confidence intervals) and uniform (confidence bands), are also implemented following the results in Cattaneo, Jansson and Ma (2020, 2021a). See Cattaneo, Jansson and Ma (2021b) for more implementation details and illustrations.

Companion command: `lpbwdensity` for bandwidth selection.

Details

Bias correction is only used for the construction of confidence intervals/bands, but not for point estimation. The point estimates, denoted by *f_p*, are constructed using local polynomial estimates of order p, while the centering of the confidence intervals/bands, denoted by *f_q*, are constructed using local polynomial estimates of order q. The confidence intervals/bands take the form: *[f_q - cv * SE(f_q) , f_q + cv * SE(f_q)]*, where *cv* denotes the appropriate critical value and *SE(f_q)* denotes an standard error estimate for the centering of the confidence interval/band. As a result, the confidence intervals/bands may not be centered at the point estimates because they have been bias-corrected. Setting q and p to be equal results on centered at the point estimate confidence intervals/bands, but requires undersmoothing for valid inference (i.e., (I)MSE-optimal bandwdith for the density point estimator cannot be used). Hence the bandwidth would need to be specified manually when q=p, and the point estimates will not be (I)MSE optimal. See Cattaneo, Jansson and Ma (2020a, 2020b) for details, and also Calonico, Cattaneo, and Farrell (2018, 2020) for robust bias correction methods.

Sometimes the density point estimates may lie outside of the confidence intervals/bands, which can happen if the underlying distribution exhibits high curvature at some evaluation point(s). One possible solution in this case is to increase the polynomial order p or to employ a smaller bandwidth.

References

Calonico, S., M. D. Cattaneo, and M. H. Farrell. 2018. On the Effect of Bias Estimation on Coverage Accuracy in Nonparametric Inference *Journal of the American Statistical Association*, 113(522): 767-779.

Calonico, S., M. D. Cattaneo, and M. H. Farrell. 2020. Coverage Error Optimal Confidence Intervals for Local Polynomial Regression . Working paper.

Cattaneo, M. D., M. Jansson, and X. Ma. 2020. Simple Local Polynomial Density Estimators. *Journal of the American Statistical Association*, 115(531): 1449-1455.

Cattaneo, M. D., M. Jansson, and X. Ma. 2021a. Local Regression Distribution Estimators *Journal of Econometrics*, forthcoming.

Cattaneo, M. D., M. Jansson, and X. Ma. 2021b. lpdensity: Local Polynomial Density Estimation and Inference *Journal of Statistical Software*, forthcoming.

Authors

Matias D. Cattaneo, Princeton University. (cattaneo@princeton.edu).

Rajita Chandak (maintainer), Princeton University. (rchandak@princeton.edu).

Michael Jansson, University of California Berkeley. (mjansson@econ.berkeley.edu).

Xinwei Ma (maintainer), University of California San Diego. (x1ma@ucsd.edu).

lpdensity.lpdensity.**lpdensity**(*data*, *grid=None*, *bw=None*, *p=None*, *q=None*, *v=None*, *kernel='triangular'*, *scale=None*, *massPoints=True*, *bwselect='mse-dpi'*, *stdVar=True*, *regularize=True*, *nLocalMin=None*, *nUniqueMin=None*, *Cweights=None*, *Pweights=None*)

Parameters

- **data** (*vector*) – Numeric vector or one dimensional matrix/data frame, the raw data.
- **grid** (*number or vector*) – Numeric, specifies the grid of evaluation points. When set to default, grid points will be chosen as 0.05-0.95 percentiles of the data, with a step size of 0.05.
- **bw** (*number or vector*) – Numeric, specifies the bandwidth used for estimation. Can be (1) a positive scalar (common bandwidth for all grid points); or (2) a positive numeric vector specifying bandwidths for each grid point (should be the same length as *grid*).
- **p** (*int*) – Nonnegative integer, specifies the order of the local polynomial used to construct point estimates. (Default is *2*.)
- **q** (*int*) – Nonnegative integer, specifies the order of the local polynomial used to construct confidence intervals/bands (a.k.a. the bias correction order). Default is *p+1*. When set to be the same as *p*, no bias correction will be performed. Otherwise it should be strictly larger than *p*.
- **v** (*int*) – Nonnegative integer, specifies the derivative of the distribution function to be estimated. *0* for the distribution function, *1* (default) for the density funtion, etc.
- **kernel** (*string*) – Specifies the kernel function, should be one of *"triangular"*, *"uniform"*, and *"epanechnikov"*.
- **scale** (*number*) – Numeric, specifies how estimates are scaled. For example, setting this parameter to 0.5 will scale down both the point estimates and standard errors by half. Default

is *1*. This parameter is useful if only part of the sample is employed for estimation, and should not be confused with *Cweights* or *Pweights*.

- **massPoints** (*boolean*) – *True* (default) or *False*, specifies whether point estimates and standard errors should be adjusted if there are mass points in the data.
- **bwselect** (*string*) – String, specifies the method for data-driven bandwidth selection. This option will be ignored if *bw* is provided. Options are (1) *"mse-dpi"* (default, mean squared error-optimal bandwidth selected for each grid point); (2) *"imse-dpi"* (integrated MSE-optimal bandwidth, common for all grid points); (3) *"mse-rot"* (rule-of-thumb bandwidth with Gaussian reference model); and (4) *"imse-rot"* (integrated rule-of-thumb bandwidth with Gaussian reference model).
- **stdVar** (*boolean*) – *True* (default) or *False*, specifies whether the data should be standardized for bandwidth selection.
- **regularize** (*boolean*) – *True* (default) or *False*, specifies whether the bandwidth should be regularized. When set to *True*, the bandwidth is chosen such that the local region includes at least *nLocalMin* observations and at least *nUniqueMin* unique observations.
- **nLocalMin** (*int*) – Nonnegative integer, specifies the minimum number of observations in each local neighborhood. This option will be ignored if *regularize=False*. Default is *20+p+1*.
- **nUniqueMin** (*int*) – Nonnegative integer, specifies the minimum number of unique observations in each local neighborhood. This option will be ignored if *regularize=False*. Default is *20+p+1*.
- **Cweights** (*numeric vector*) – Numeric, specifies the weights used for counterfactual distribution construction. Should have the same length as the data.
- **Pweights** (*numeric vector*) – Numeric, specifies the weights used in sampling. Should have the same length as the data.

Returns

- *Estimate* – A matrix containing (1) *grid* (grid points), (2) *bw* (bandwidths), (3) *nh* (number of observations in each local neighborhood), (4) *nhu* (number of unique observations in each local neighborhood), (5) *f_p* (point estimates with p-th order local polynomial), (6) *f_q* (point estimates with q-th order local polynomial, only if option *q* is nonzero), (7) *se_p* (standard error corresponding to *f_p*), and (8) *se_q* (standard error corresponding to *f_q*).
- *CovMat_p* – The variance-covariance matrix corresponding to *f_p*.
- *CovMat_q* – The variance-covariance matrix corresponding to *f_q*.

class lpdensity.lpdensity.**lpdensity_output**(*Estimate*, *CovMat_p*, *CovMat_q*, *p*, *q*, *v*, *kernel*, *scale*, *massPoints*, *n*, *ng*, *bwselect*, *stdVar*, *regularize*, *nLocalMin*, *nUniqueMin*, *data_min*, *data_max*, *grid_min*, *grid_max*)

Class of lpdensity function outputs.

Object type returned by *lpdensity()*.

coef()

Returns estimate coefficients.

confint(*alpha=0.05*, *CIuniform=False*, *CIsimul=2000*)

Returns confindence intervals/bands for prespecified confidence level.

alpha [number] Confindence level, must be between 0 and 1.

CIuniform [boolean] Boolean on wehther to construct uniform confidence bands, *True* or *False (default)*.

CIsimul [int] Number of simulations used to generate confidence intervals.

plot(*alpha=0.05*, *type='line'*, *CItype='region'*, *CIuniform=False*, *CIsimul=2000*, *hist=False*, *histData=None*, *histBins=None*, *histFillCol=3*, *histFillShade=0.2*, *histLineCol='white'*, *title=None*, *xlabel=None*, *ylabel=None*, *CIshade=0.2*)

Method to plot estimate and confidence bands. Requires ggplot.

alpha [number] Confindence level, must be between 0 and 1.

CIuniform [boolean] Boolean on wehther to construct uniform confidence bands, *True* or *False (default)*.

CIsimul [int] Number of simulations used to generate confidence intervals.

type [string] type of estimate plot, *line (default)*, or *points*, or *all*.

CItype [string] type of confidence interval plot, *region (default)*, *lines*, *ebar*, or *all*.

vcov()

Returns estimate standard error and covariance matrices.

Example

```
>>> import numpy as np
>>> from lpdensity import lpdensity
>>> data = np.random.normal(0,1,500)
>>> grid = np.linspace(min(data), max(data), 10)
>>> est = lpdensity(data=data, grid=grid)
>>> print(repr(est))
>>> est.plot()
```

2.1.2 lpbwdensity

Data-driven Bandwidth Selection for Local Polynomial Density Estimators

Description

`lpbwdensity` implements the bandwidth selection methods for local polynomial based density (and derivatives) estimation proposed and studied in Cattaneo, Jansson and Ma (2020, 2021a). See Cattaneo, Jansson and Ma (2021b) for more implementation details and illustrations.

Companion command: `lpdensity` for estimation and robust bias-corrected inference.

References

Cattaneo, M. D., M. Jansson, and X. Ma. 2020. Simple Local Polynomial Density Estimators. *Journal of the American Statistical Association*, 115(531): 1449-1455.

Cattaneo, M. D., M. Jansson, and X. Ma. 2021a. Local Regression Distribution Estimators *Journal of Econometrics*, forthcoming.

Cattaneo, M. D., M. Jansson, and X. Ma. 2021b. lpdensity: Local Polynomial Density Estimation and Inference *Journal of Statistical Software*, forthcoming.

Authors

Matias D. Cattaneo, Princeton University. (cattaneo@princeton.edu).

Rajita Chandak (maintainer), Princeton University. (rchandak@princeton.edu).

Michael Jansson, University of California Berkeley. (mjansson@econ.berkeley.edu).

Xinwei Ma (maintainer), University of California San Diego. (x1ma@ucsd.edu).

class lpdensity.lpbwdensity.**bw_output**(*BW*, *bws*, *p*, *v*, *kernel*, *n*, *ng*, *bwselect*, *massPoints*, *stdVar*, *regularize*, *nLocalMin*, *nUniqueMin*, *data_min*, *data_max*, *grid_min*, *grid_max*)

Class of lpbwdensity function outputs.

Object type returned by *lpbwdensity()*.

coef()

Returns estimate of bandwidths.

lpdensity.lpbwdensity.**lpbwdensity**(*data*, *grid=None*, *p=None*, *v=None*, *kernel='triangular'*, *bwselect='mse-dpi'*, *massPoints=True*, *stdVar=True*, *regularize=True*, *nLocalMin=None*, *nUniqueMin=None*, *Cweights=None*, *Pweights=None*)

Parameters

- **data** (*vector*) – Numeric vector or one dimensional matrix/data frame, the raw data.
- **grid** (*vector*) – Numeric, specifies the grid of evaluation points. When set to default, grid points will be chosen as 0.05-0.95 percentiles of the data, with a step size of 0.05.
- **p** (*int*) – Nonnegative integer, specifies the order of the local polynomial used to construct point estimates. (Default is *2*.)
- **v** (*int*) – Nonnegative integer, specifies the derivative of the distribution function to be estimated. *0* for the distribution function, *1* (default) for the density funtion, etc.
- **kernel** (*string*) – Specifies the kernel function, should be one of *"triangular"*, *"uniform"* or *"epanechnikov"*.
- **bwselect** (*string*) – Specifies the method for data-driven bandwidth selection. This option will be ignored if *bw* is provided. Can be (1) *"mse-dpi"* (default, mean squared error-optimal bandwidth selected for each grid point); or (2) *"imse-dpi"* (integrated MSE-optimal bandwidth, common for all grid points); (3) *"mse-rot"* (rule-of-thumb bandwidth with Gaussian reference model); and (4) *"imse-rot"* (integrated rule-of-thumb bandwidth with Gaussian reference model).
- **massPoints** (*boolean*) – *True* (default) or *False*, specifies whether point estimates and standard errors should be adjusted if there are mass points in the data.
- **stdVar** (*boolean*) – *True* (default) or *False*, specifies whether the data should be standardized for bandwidth selection.
- **regularize** (*boolean*) – *True* (default) or *False*, specifies whether the bandwidth should be regularized. When set to *True*, the bandwidth is chosen such that the local region includes at least *nLocalMin* observations and at least *nUniqueMin* unique observations.
- **nLocalMin** (*int*) – Nonnegative integer, specifies the minimum number of observations in each local neighborhood. This option will be ignored if *regularize=False*. Default is *20+p+1*.

- **nUniqueMin** (*int*) – Nonnegative integer, specifies the minimum number of unique observations in each local neighborhood. This option will be ignored if *regularize=False*. Default is *20+p+1*.
- **Cweights** (*vector*) – Numeric vector, specifies the weights used for counterfactual distribution construction. Should have the same length as the data. This option will be ignored if *bwselect* is *"mse-rot"* or *"imse-rot"*.
- **Pweights** (*vector*) – Numeric vector, specifies the weights used in sampling. Should have the same length as the data. This option will be ignored if *bwselect* is *"mse-rot"* or *"imse-rot"*.

Returns **BW** – A *bw_output()* class object containing a matrix of (1) *grid* (grid point), (2) *bw* (bandwidth), (3) *nh* (number of observations in each local neighborhood), (4) *nhu* (number of unique observations in each local neighborhood), and (5) *opt* additional parameters.

Return type object

Example

```
>>> import numpy as np
>>> from lpdensity import lpbwdensity
>>> data = np.random.normal(0,1,500)
>>> grid = np.linspace(min(data), max(data), 10)
>>> est = lpbwdensity(data=data, grid=grid, bwselect="mse-dpi")
>>> print(repr(est))
```

PYTHON MODULE INDEX

INDEX