Stata User Group - 9th UK meeting - 19/20 May 2003

Confidence intervals for kernel density estimation

Carlo Fiorio

c.fiorio@lse.ac.uk

London School of Economics and STICERD

Introduction

- Nonparametric density estimation have been widely applied for analyzing density of a given data set.
- Nonparametric density estimation can be seen as a development of histogram for density analysis.
- Probably the most frequently used nonparametric density estimation used is based on the kernel method.
- The most important parameter in kernel density estimation is the bandwidth: there exists a large literature on fixed and variable bandwidth (adaptive kernel).
- The kernel density estimation provides a point estimation. Considering several points along the data range and connecting them we can provide a picture of the estimated density.
- However, not large attention has been paid to **performing inference** with kernel density estimation in empirical works.

Outline of the presentation

- Recalling the main results from the literature
  - quickest rate of convergence for pointwise kernel estimation;
  - the issue of the asymptotic bias in non smooth functions of the sample moments;
  - coping with asymptotic bias;
  - difference between asymptotic and bootstrap tests or confidence intervals.
- Where are we with Stata?
- `kerden.ado`: a development of `kdensity.ado`;
- `bsciker.ado`: a new program for bootstrap confidence intervals for kernel density estimation.
- `asciker.ado`: a new program for asymptotic confidence intervals for kernel density estimation.

The kernel methodology for density estimation

- The kernel methodology aims to estimate the density f of a random variable, X, from a random sample $X_i, i = 1, 2, ..., n$ without assuming that f belongs to a known family of functions.
- The (fixed-width) kernel density estimation basically slides a window of given width along the data range counting and properly weighting the observation that fall into the window.
- Formally, the kernel estimator of f is:

$$f_n(x) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right) \tag{1}$$

- K is a kernel functions with given properties; $h_n, n = 1, 2, ..., n$ is a positive sequence of bandwidths; f is assumed to have $r \leq 2$ continuous derivatives in NBH of x (Silverman (1986)).

Performing inference on pointwise density estimation

- If nh^{2r+1} is bounded and $n \to \infty$:

$$Z_n(x) \equiv \frac{f_n - f(x) - b_n(x)}{\sigma_n(x)} = \frac{f_n(x) - E[f_n(x)]}{\sigma_n(x)} \xrightarrow{d} N(0,1) \qquad (2)$$

- We can compute a studentized statistic which is asymptotically pivotal for testing hypothesis or forming confidence interval for $f(x)$ with suitable estimator for $\sigma_n(x)$ and $b_n(x)$.
- `kerden.ado` provides and estimate of the variance of $f_n(x)$ computing:

$$s_n^2(x) = \frac{1}{(nh_n)^2} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right)^2 - \frac{f_n(x)^2}{n} \qquad (3)$$

The fastest rate of convergence of $f_n(x)$ to $f(x)$

- The fastest possible rate of convergence of $f_n(x)$ to $f(x)$ is obtained with $h_n \propto n^{-1/(2r+1)}$.
- With such a bandwidth: (a) $f_n - f(x) = O_p[n^{-r/(2r+1)}]$; (b) $b_n(x) \propto n^{-r/(2r+1)}$; (c) $\sigma_n(x) \propto n^{-r/(2r+1)}$.
- The studentized form of $Z_n(x)$ for asymptotic confidence interval is:

$$t_n(x) = \frac{f_n(x) - E[f_n(x)]}{s_n(x)} \xrightarrow{d} N(0,1) \tag{4}$$

- However, t_n is the asymptotic t statistic for testing hypothesis or forming CI for $E[f_n(x)]$ but cannot be used to test hypothesis and building CI for $f(x)$, unless $b_n(x)$ is negligibly small.
- The asymptotic bias causes the asymptotic distribution of t_n not to be centered at 0.

Methods for controlling the asymptotic bias

- Asymptotic bias is a characteristic of nonparametric estimators that is not shared by estimators that are not smooth functions of the sample moments (Horowitz, 1999).
- Asymptotic bias does affect the bootstrap as well.
- There are mainly two methods for dealing with asymptotic bias:
  - explicit bias removal;
  - undersmoothing.
- Hall (1992) explains that, nonparametric point estimation and nonparametric interval estimation (or testing) are different tasks that *require different degrees of smoothing*.
- Hall (1992) also shows that undersmoothing performs better in terms of errors in the coverage probability.

Dealing with the asymptotic bias

- The fastest rate of convergence of $f_n(x)$, is obtained with $h_n \propto n^{-1/2r+1}$
- However, $f_n(x)$ is asymptotically biased unless the bias is negligibly small.
- With undersmooting, $(nh_n)^{1/2} b_n(x) = o_p(1)$ as $n \to \infty$, i.e. the bias is asymptotically negligible (such a bandwidth minimizes the bias maximizing the variance).
- Horowitz (1999) suggests setting $h_n \propto n^{-\kappa}$, with $\kappa > -1/(2r+1)$; Hall (1992) suggests setting $h_n \propto \gamma n^{1/(2r+1)}$, with $0.1 < \gamma < 0.3$.
- `kerden.ado` can perform both undersmoothing.

Asymptotic vs. bootstrap CI

- Horowitz (1999) demonstrates that the bootstrap provides asymptotic refinements for tests of hypothesis and CI in nonparametric density estimation.
- With asy. critical values, the difference between the true and nominal rejection probabilities of a symmetrical t test is $O[(nh_n)^{-1}]$. This results relies on $nh_n^{r+1} \to 0$. If this does not happen the $ERP > O[(nh_n)^{-1}]$.
- With the BS critical values, the difference b/w true and nominal rejection probabilities of the symmetrical t test is $o[(nh_n)^{-1}]$
- Hence, the bootstrap provides asymptotic refinements for hypothesis tests and confidence intervals based on a kernel nonparametric density estimator (when the bandwidth h_n converges to zero sufficiently rapidly to make the asymptotic bias of the density estimator negligibly small).

Where are we with Stata?

- To the best of my knowledge, with Stata we can perform kernel density estimation but we cannot perform inference on the point density estimation.
- The popular program `kdensity.ado` has lots of features but does not compute the variance and allow undersmoothing.

the `kerden.ado`: a development of `kdensity.ado`

- `kerden.ado` is built on `kdensity.ado`.
- On top of what `kdensity.ado` does, `kerden.ado` computes the sample variance of pointwise estimation and allows to save it as an additional variable.
- Why `kerden.ado` and not `kdensity2.ado`?
  - No particular reason, just matter of names.
- This program could be of use for hypothesis testing as well as for confidence interval estimation.

What about `bsciker.ado`?

- Given the random sample $X_1, i = 1, 2, ..., n$ `bsciker.ado`:
  - generates B bootstrap samples $X_i^*, i = 1, 2, ..., n$ sampling X_i with replacement;
  - computes, *with undersmoothing*: $f_n^* = (1/nh_n) \sum_{i=1}^n K(x - X_i^*/h_n)$
  - computes: $s_n^{2*}(x) = (1/nh_n^2) \sum_{i=1}^n K(x - X_i^*/h_n)^2 - f_n^*(x)^2/n$
  - defines the bootstrap analog of t_n:

$$t_n^* = \frac{f_n^*(x) - f_n(x)}{s_n^*(x)} \tag{5}$$

  - Computes the BS critical values (for any given significance level) and saves $f_n(x)$, low/up bound as new variables.

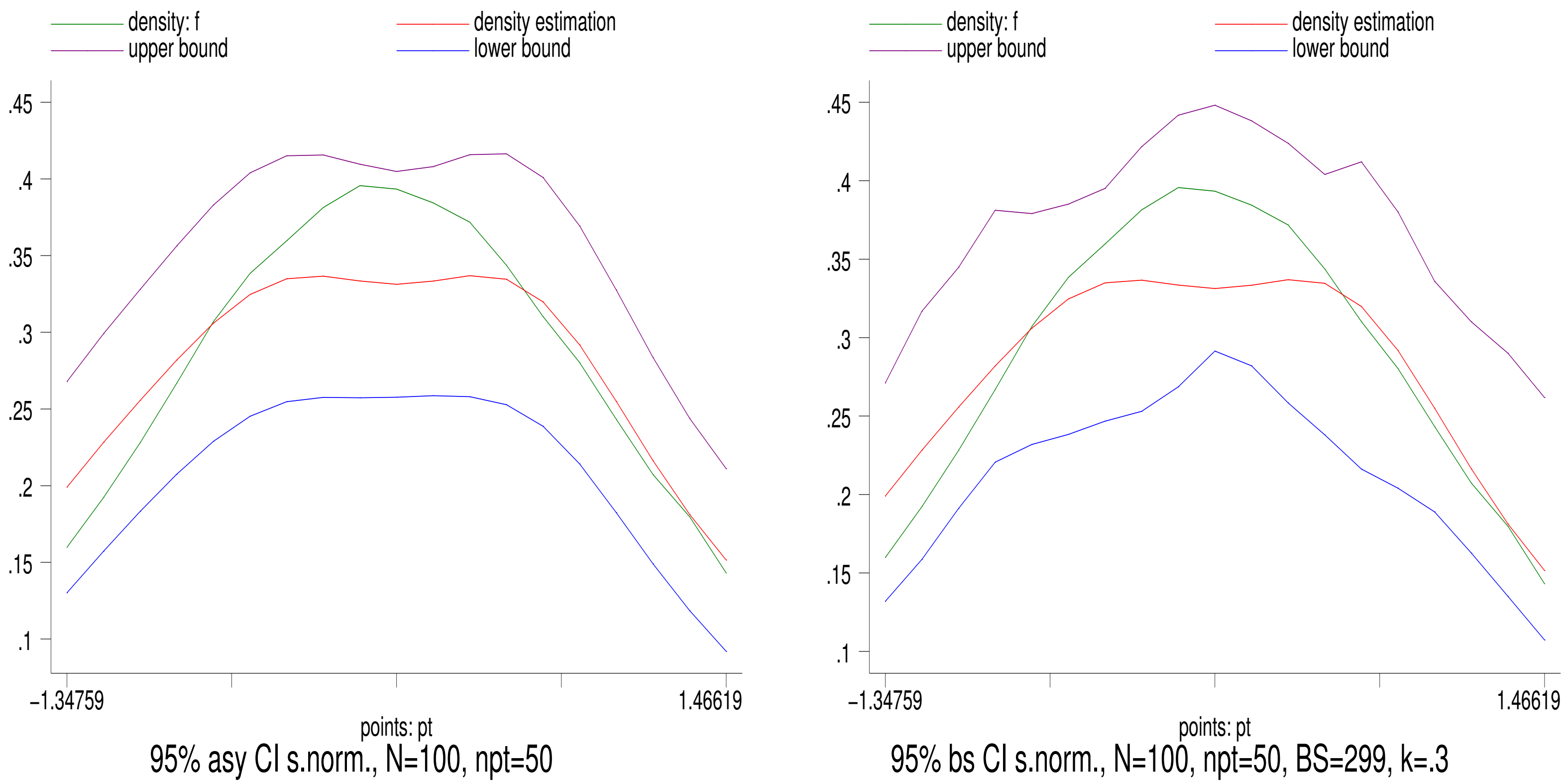
`bsciker.ado`: a program for BS CI of kernel density

- `bsciker.ado` develops in three steps:
  - it generates B bootstrap samples from the data set;
  - it computes the kernel density and its variance for each bootstrap data set using `kerden.ado` *with undersmoothing*;
  - it merges results from previous steps, compute the pivotal statistic, computes the relevant BS critical values, and finally saves the upper and lower bounds as additional variables to be plotted together with the kernel density estimation.

Simple illustration

- As simple illustration we generated a random sample of dimension $N = 100$ from a $N(0, 1)$.
- We than computed the asymptotic CI (for simplicity, $b_n(x) = 0$).
- We computed the boostrap CI with oversmoothing
- Plotted results together with "zero-bias" CI: $f(x) = f_n(x) \pm 1.96\sigma_n$.

A simple illustration



Brief discussion/conclusions

- `bsciker.ado` can be developed/accompanied by a program for testing hypothesis on $f(x)$.
- `bsciker.ado` can be quite time demanding but some improvement in programming could be helpful.

`bsciker.ado` and `kerden.ado` are useful program for performing inference on kernel density estimation.