Adaptive kernel density estimation in Stata

Philippe Van Kerm
CEPS/INSTEAD, G.-D. Luxembourg

9th London Stata User Group Meeting, 19-20 May 2003
Kernel density function estimation

- Official command: `kdensity`

\[
\hat{f}_f(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K \left( \frac{x - x_i}{h} \right)
\]  

- ‘Point mass’ of sample data diffused around \(x_i\)’s, and averaged at \(x\)
Kernel density function estimation

- Official command: \texttt{kdensity}

\[ \hat{f}_f(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K \left( \frac{x - x_i}{h} \right) \]  

- ‘Point mass’ of sample data diffused around \( x_i \)’s, and averaged at \( x \)
- Fixed/constant/global bandwidth \( h \): ‘degree of diffusion’ constant for all \( x_i \)’s
Adaptive kernel density function estimation

- akdensity

\[
\hat{f}_v(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i} K \left( \frac{x - x_i}{h_i} \right)
\]  

- Different bandwidths for different \( x_i \)'s
Adaptive kernel density function estimation

- \texttt{akdensity}

\begin{equation}
\hat{f}_v(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i} K \left( \frac{x - x_i}{h_i} \right)
\end{equation}

- Different bandwidths for different $x_i$’s
- Degree of diffusion varies inversely with $f(x_i)$
Adaptive kernel density function estimation

- akdensity

\[ \hat{f}_v(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i} K \left( \frac{x - x_i}{h_i} \right) \]  

(2)

- Different bandwidths for different \(x_i\)’s
- Degree of diffusion varies inversely with \(f(x_i)\)
- Greater precision where data are abundant and ...
- ... greater smoothness where data are sparse
Adaptive kernel density function estimation

- **akdensity**

\[
\hat{f}_v(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i} K \left( \frac{x - x_i}{h_i} \right)
\]  

(2)

- Different bandwidths for different \(x_i\)’s
- Degree of diffusion varies inversely with \(f(x_i)\)
- Greater precision where data are abundant and ...
- ... greater smoothness where data are sparse

- **Adaptive two-stage estimator (Abramson 1982):**

\[ h_i = h \times \lambda_i; \]
**Adaptive kernel density function estimation**

- akdensity

\[
\hat{f}_v(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i} K \left( \frac{x - x_i}{h_i} \right)
\]

- Different bandwidths for different \(x_i\)'s
- Degree of diffusion varies inversely with \(f(x_i)\)
- Greater precision where data are abundant and ...
- ... greater smoothness where data are sparse

- Adaptive two-stage estimator (Abramson 1982):

\[
h_i = h \times \lambda_i; \quad \lambda_i = \left( G / \tilde{f}(x_i) \right)^{0.5}
\]
Adaptive kernel density function estimation

- \texttt{akdensity}

\[
\hat{f}_v(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i} K \left( \frac{x - x_i}{h_i} \right)
\]  

- Different bandwidths for different \( x_i \)’s
- Degree of diffusion varies inversely with \( f(x_i) \)
- Greater precision where data are abundant and ...
- ... greater smoothness where data are sparse

- Adaptive two-stage estimator (Abramson 1982):

\[
h_i = h \times \lambda_i; \quad \lambda_i = \left( \frac{G/\hat{f}(x_i)}{\tilde{f}(x_i)} \right)^{0.5}
\]

- First step: compute pilot estimate (fixed bandwidth \( h \)) to generate \( \lambda_i \)
Adaptive kernel density function estimation

• **akdensity**

\[
\hat{f}_v(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i} K \left( \frac{x - x_i}{h_i} \right)
\]  

– Different bandwidths for different \(x_i\)'s
– Degree of diffusion varies inversely with \(f(x_i)\)
– Greater precision where data are abundant and ...
– ... greater smoothness where data are sparse

• **Adaptive two-stage estimator (Abramson 1982):**

\[
h_i = h \times \lambda_i; \quad \lambda_i = \left( \frac{G}{\tilde{f}(x_i)} \right)^{0.5}
\]

– First step: compute pilot estimate (fixed bandwidth \(h\)) to generate \(\lambda_i\)
– Second step: compute density estimate with \(h_i\) local bandwidths
Variability bands as a bonus

- \texttt{akdensity} estimates variability bands:
  
  \[ \hat{f}_v(x) \pm b \times SE(x) \]
Variability bands as a bonus

- \texttt{akdensity} estimates variability bands:
  - \( \hat{f}_v(x) \pm b \times SE(x) \)
  - Pointwise variability bands
Variability bands as a bonus

- `akdensity` estimates variability bands:
  - $\hat{f}_v(x) \pm b \times SE(x)$
  - Pointwise variability bands
  - Not confidence intervals (accounts for sample variability but not bias!)
Variability bands as a bonus

- \texttt{akdensity} estimates variability bands:
  - \( \hat{f}_v(x) \pm b \times SE(x) \)
  - Pointwise variability bands
  - Not confidence intervals (accounts for sample variability but not bias!)

- Also for fixed bandwidth kernel estimates!
Syntax extract

- ‘High-level’ command (mimicks `kdensity`):

  ```plaintext
  akdensity varname ... [ , noadaptive width(\#) \\
  [ epan | gauss ] stdbands(\#) ... ]
  ```
Syntax extract

- ‘High-level’ command (mimicks `kdensity`):

  \[ \text{akdensity } varname \ldots [ , \text{ noadaptive width(#)} \]
  \[ [ \text{ epan | gauss } ] \text{ stdbands(#)} \ldots ] \]

- ‘Low-level’ command (rarely used, but full control):

  \[ \text{akdensity0 } varname \ldots , \text{ width(# | varname)} \ldots [ \]
  \[ \text{ lambda(string) } \ldots ] \]
A simulated example

True underlying density

Adaptive kernel density estimation May 17, 2003
Adaptive kernel density estimation
FIXED bandwidth estimates

h = 0.83

Adaptive kernel density estimation May 17, 2003
FIXED bandwidth estimates

h = 0.55

Adaptive kernel density estimation May 17, 2003
VARIABLE bandwidth estimates

h = 1.11 (default)

Adaptive kernel density estimation May 17, 2003
VARIABLE bandwidth estimates

Adaptive kernel density estimation May 17, 2003
VARIABLE bandwidth estimates

Adaptive kernel density estimation May 17, 2003
FIXED bandwidth estimates

VARIABLE bandwidth estimates

Adaptive kernel density estimation May 17, 2003
A real example

Real GDP per capita – FIXED and VARIABLE bandwidths

Adaptive kernel density estimation

May 17, 2003
Final remarks

- Fast implementation for large datasets (use of linear interpolation for the pilot estimate)
Final remarks

- Fast implementation for large datasets (use of linear interpolation for the pilot estimate)
- Works under both version 7 and version 8:
Final remarks

• Fast implementation for large datasets (use of linear interpolation for the pilot estimate)

• Works under both version 7 and version 8:
  – use new graphics engine if called from Stata 8, and former engine if called from Stata 7
Final remarks

• Fast implementation for large datasets (use of linear interpolation for the pilot estimate)

• Works under both version 7 and version 8:
  – use new graphics engine if called from Stata 8, and former engine if called from Stata 7
  – graphics options therefore vary (in Stata 8, option `plot(...)` especially useful!)
Final remarks

- Fast implementation for large datasets (use of linear interpolation for the pilot estimate)

- Works under both version 7 and version 8:
  - use new graphics engine if called from Stata 8, and former engine if called from Stata 7
  - graphics options therefore vary (in Stata 8, option \texttt{plot(...)} especially useful!)
  - coded using \texttt{if _caller()<8} statements
Final remarks

• Fast implementation for large datasets (use of linear interpolation for the pilot estimate)

• Works under both version 7 and version 8:
  – use new graphics engine if called from Stata 8, and former engine if called from Stata 7
  – graphics options therefore vary (in Stata 8, option `plot(...)` especially useful!)
  – coded using `if _caller()<8` statements

• Available in next *Stata Journal* issue (vol.3(2))