Realized Kernels: Recommended Implementation

Asger Lunde

Professor
Department of Economics and Business Economics
Aarhus University

September 28, 2016

Realised Kernels
Recommended Implementation

- Non-negative realised kernels of BNHLS (2011; Multivariate RK) is used:

$$K(X) = \sum_{h=-H}^{H} k\left(\frac{h}{H+1}\right)\gamma_h, \quad \gamma_h = \sum_{j=|h|+1}^{n} x_j x_{j-h}, \tag{1}$$

where $k(x)$ is a Parzen kernel

$$k(x) = \begin{cases} 1 - 6x^2 + 6x^3 & 0 \le x \le 1/2 \\ 2(1-x)^3 & 1/2 \le x \le 1 \\ 0 & x > 1. \end{cases}$$

Here x_j is the j-th high frequency return calculate over the interval τ_{j-1} to τ_j.

  - Notice no scaling by sample size. Related to, but different from, HAC and spectral density estimation.

Realised Kernels
Recommended Implementation

- The preferred choice of bandwidth is

$$H^* = c^* \xi^{4/5} n^{3/5},$$

$$\text{with } c^* = \left\{ \frac{k''(0)^2}{k_{\bullet}^{0,0}} \right\}^{1/5} \quad \text{and} \quad \xi^2 = \frac{\omega^2}{IQ},$$

where $c^* = ((12)^2/0.269)^{1/5} = 3.5134$ for the Parzen kernel.

- The bandwidth H^* depends on the unknown quantities ω^2 and IQ, where the latter is the integrated quarticity.

- We define an estimator of ξ, to get a bandwidth,

$$\hat{H}^* = c^* \hat{\xi}^{4/5} n^{3/5},$$

that can be implemented in practice.

Realised Kernels

Recommended Implementation

- To select H we need to estimate

$$\xi^2 = \frac{\omega^2}{\sqrt{IQ}} \simeq \frac{\omega^2}{IV}.$$

- To estimate ω^2 we compute the realised variance using every q-th trade or quote.
  - We obtain q distinct realised variances, $RV_{dense}^{(1)}, \ldots, RV_{dense}^{(q)}$ and compute

$$\hat{\omega}^2 = \frac{1}{q} \sum_{i=1}^{q} \frac{RV_{dense}^{(i)}}{2n_{(i)}},$$

    $n_{(i)}$: # non-zero returns used to compute $RV_{dense}^{(i)}$.
  - The reason that we choose $q > 1$ is robustness.
- IV is estimated by averaging 20 minute realised variances.

Realised Kernels

Recommended Implementation

- The actual implementation in Matlab is as function call that looks something like this:

  `MultivarRKernel(3/5,1,tim~obsPr,NtoS,'parzen',0)`

  or if `MultivarRKernel(A,B,C,D,E,F)` then

  `A(=3/5)`: the rate of n, that is α in $\hat{H}^* = c^* \hat{\xi}^{4/5} n^{\alpha}$

  `B(=1)`: the amount of jittering

  `C(=usePr)`: $n \times 2$ matrix with time stamps and prices

  `D(=NtoS)`: $\hat{\xi}^2$

  `E(='parzen')`: text string with the choice of kernel

  `F(=0)`: the amount of topflatness (=1 topflat kernels from BNHLS (2008; univariate RK))

Computing the Asympotic Variance
Recommended Implementation

- We have $\sqrt{n}\left(K(X)-[Y]\right) \stackrel{L}{\rightarrow} MN\left(0, \varpi(IV, IQ, \omega^2, n, H)\right)$, and a log-based version

$$\frac{\sqrt{n}\left[\log\left\{K(X)\right\}-\log\left\{[Y]\right\}\right]}{\sqrt{\varpi(IV, IQ, \omega^2, n, H)}/K(X)} \stackrel{L}{\rightarrow} N\left(0,1\right),$$

To compute $\hat{\varpi}(IV, IQ, \omega^2, n, H)$ we do as follows:

1. We have $\hat{K}(X)$, $\hat{H}$, $\bar{n}$ and n from the previous section.
2. Now we use our bias corrected estimator of ω^2 :

$$\breve{\omega}^2 = \exp\{\log\hat{\omega}^2 - \hat{K}(X)/(2\bar{n}\hat{\omega}^2)\}$$

Computing the Asympotic Variance
Recommended Implementation

3. To estimate IQ we use our subsampling bipower variation type estimator:

$$\left\{X_\delta, \breve{\omega}^2, \sqrt{n}\right\}^{[2,2]} = \sqrt{n}\sum_{j=1}^{\sqrt{n}} \left\{x_{j,\cdot}^2 - 2\breve{\omega}^2\right\}\left\{x_{j-2,\cdot}^2 - 2\breve{\omega}^2\right\}$$

with $\quad x_{j,\cdot}^2 = \frac{1}{\sqrt{n}+1}\sum_{s=0}^{\sqrt{n}} x_{j,s/(\sqrt{n}+1)}^2, \quad j = 1, \ldots, n.$

and the fact that $t\int_0^t \sigma_u^4 du \geq \int_0^t \sigma_u^2 du$, so

$$\widehat{IQ} = \max[\{\hat{K}(X)\}^2, \left\{X_\delta, \breve{\omega}^2, \sqrt{n}\right\}^{[2,2]}]$$

4. Now we estimate the asymptotic variance by

$$\hat{\varpi} \equiv \varpi(\hat{K}(X), \widehat{IQ}, \breve{\omega}^2, n, \hat{H})$$

Computing the Asympotic Variance

Recommended Implementation

- The actual implementation of $\varpi(\hat{K}(X), \widehat{IQ}, \breve{\omega}^2, n, \hat{H}) / \sqrt{n}$ in Matlab is a function call that looks something like this:

```
KernelVar(KerH,N,1,KdQ,MKern,EstOmega2,'parzen',0)
```

or if `KernelVar(A,B,C,D,E,F,G,H)` then

`A(=KerH)`: the bandwidth $\hat{H}$.

`B(=N)`: n number of observations used to compute $\hat{K}(X)$.

`C(=1)`: the amount of jittering

`D(=KdQ)`: $\widehat{IQ}$ the estimate of integrated quarticity.

`E(=MKern)`: $\hat{K}(X)$: the realised kernel.

`F(=EstOmega2)`: $\breve{\omega}^2$: bias corrected estimator of ω^2.

`G(='parzen')`: text string with the choice of kernel

`F(=0)`: the amount of topflatness (=1 topflat kernels from BNHLS (2008))