Non-parametric methods in stochastic volatility density estimation

Author:
Mukul Tyagi (6163521)

Supervisors:
Dr. A.J. (Bert) van Es, Dr. P.J.C. (Peter) Spreij

August 26, 2011
Acknowledgements

I would like to express my sincere gratitude to my supervisors Dr. Bert van Es and Dr. Peter Spreij. Their guidance and constant encouragement were immensely helpful throughout the time I was writing this thesis. I would also like to thank them for lending me good books and literature. Every time I needed help, they were always willing to explain the theory and provided me with insightful comments.

I would also like to thank all my friends who made my stay in Amsterdam so much more fun and my family for their continuous support.
Contents

Acknowledgements i

1 Introduction 3

2 Non-parametric density estimation 5
  2.1 Univariate kernel estimator ........................................... 5
  2.2 Measures of efficiency .................................................. 6
  2.3 Examples of Kernel Estimation ........................................ 9
  2.4 Density estimation using wavelets ................................. 10
    2.4.1 Examples of wavelet estimation ............................... 14

3 Advanced methods in Kernel estimation 16
  3.1 Efficient computation of Kernel Density Estimators using fast
    Fourier Transform ....................................................... 16
    3.1.1 Algorithm ......................................................... 18
    3.1.2 Examples .......................................................... 19
  3.2 Deconvolution estimator .............................................. 19
    3.2.1 Examples .......................................................... 22

4 Stochastic volatility estimation 24
  4.1 Stochastic Volatility estimation ..................................... 24
  4.2 Examples of Stochastic Volatility estimation ...................... 26
    4.2.1 GARCH(1,1) ....................................................... 26
    4.2.2 Ornstein-Uhlenbeck process ................................... 28
    4.2.3 Multivariate case ................................................ 28

5 Wavelet Deconvolution 31
  5.1 Meyer-type wavelet ...................................................... 31
  5.2 Linear wavelet type Deconvolution estimator ..................... 32
  5.3 An Example ............................................................ 34
A Programs

A.1 Deconvolution ............................................. 37
A.2 Density estimation using fast Fourier tranformation ...... 38
A.3 GARCH(1,1) Stochastic volatility estimation .............. 39
A.4 GARCH(1,1) Deconvolution ............................... 40
A.5 Wavelet Deconvolution .................................. 41

References ......................................................... 43
Chapter 1

Introduction

The main focus of this thesis is to study density estimation, deconvolution and volatility density estimation and their computational methods. The problems are interrelated and we consider two tools to analyze these problems, the kernel density estimator and the wavelet estimator. We first study direct density estimation from an i.i.d. sample $X_1, X_2, \ldots, X_n$ from a continuous univariate density $f$. Density deconvolution is the method of estimating a function $f_X$ when the random variable $X$ is observed with an independent additive noise $Z$.

$$Y_i = X_i + Z_i, \quad i = 1, \ldots, n$$

Here we assume that $X_1, X_2, \ldots, X_n$ is a random sample having a common density $f_X$ that is of interest and the $Z_i$’s are random variables with common density $f_Z$ and independent of $X_i$. We assume that $f_Z$ is known and we want to estimate $f_X$. In this situation the density function $f_Y$ is convolution of $f_X$ and $f_Z$, $f_Y = f_X * f_Z$, so the problem of estimating $f_X$ is called deconvolution problem. To study this problem two types of non-parametric density estimators are reviewed, the deconvolution kernel density estimator and a linear wavelet deconvolution density estimator. For the latter, the idea is to represent the $f_X$ by a wavelet expansion and then to estimate the coefficients using a deconvolution algorithm. The estimator is based on Meyer-type wavelets.

Density estimation from a noisy sample is of fundamental importance in many practical situations and one of the area of application is volatility density estimation. Volatility is a measurement of variability of some asset’s return over a period of time. The estimation of volatility is important in financial applications for example in risk management as it is
required for derivative pricing e.g. options pricing. A lot of literature for
the study of finance is based on the Black-Scholes’ assumption of constant
volatility. However, there are contradictions when we observe real financial
data in markets e.g. the volatility smiles and volatility clustering cannot be
explained by constant volatility. Assuming stochastic volatility is one way
to circumvent this problem. We consider the following type of stochastic
volatility models,

\[ X_t = \sigma_t Z_t, \]

where \( Z_t \) is an i.i.d. Gaussian sequence and at each time \( t \) the random
variables \( \sigma_t \) and \( Z_t \) are independent. Simple transformations reduce the
equation to the following form

\[ \log(X_t^2) = \log(\sigma_t^2) + \log(Z_t^2), \]

and under the assumption the density of \( \log(Z_t^2) \) is known. So we return
to the original setting of deconvolution problem and deconvolution kernel
density estimator can be used to find the density of \( \log(\sigma_t^2) \).

The thesis is organized as follows. In chapter 2, we discuss the theory of
kernel density estimation and present the measures of efficiency such as mean
squared error(MSE) and mean integrated squared error(MISE) of kernel
density estimator. Two examples of estimating normal mixture densities
are also considered. Density estimation using wavelets is also explained and
same normal mixture densities are estimated by wavelets using the same
data sets as used in kernel density estimation.

In chapter 3, we discuss how fast Fourier transform(FFT) algorithm can
be used for efficient computation of kernel density estimator. Examples of
Gamma and Normal densities are given. We then consider the deconvolution
problem. The deconvolution estimator is derived and examples of Gamma
and Normal densities measured with Normal error are given.

In chapter 4, we derive the deconvolution stochastic volatility estimator.
Examples based on GARCH(1,1) processes in univariate and bivariate case
are given. Both use the FFT algorithm. The Ornstein-Uhlenbeck process
which is a mean reverting process is also considered.

In chapter 5, we discuss the linear wavelet type deconvolution estimator.
This estimator, which is based on Meyer-type wavelets is then applied to
the case when the error density is given by a double exponential distribution
and the density of interest is the standard normal density. The results are
also compared with the kernel estimator. The MATLAB codes can be found
in the appendix.
Chapter 2

Non-parametric density estimation

Density estimation is a very useful tool in data presentation and analysis. It can be said that as compared to cumulative distribution functions and the actual formula of a density, the plot of a density is more comprehensible and intuitive. Kernel estimation methods are a non-parametric way of finding structure in data. It is specially useful to use a non-parametric method when the underlying data is too complicated to fit a parametric model. In this thesis, we consider non-parametric methods of density estimation, in particular the kernel approach. Histograms are very well known non-parametric methods of data representation, however they depend very heavily on the initial origin and bin width. This can hide important characteristics and leads to inefficient use of data. They also lack mathematical rigor due to discontinuity since derivatives of estimates cannot be found, e.g. slope of density at a point. Kernel methods provide a mathematically more rigorous (possibly better) way to achieve the same goal. In the next section, we discuss kernel density estimation for univariate data, mostly based on [12].

2.1 Univariate kernel estimator

It is assumed that we have an independent and identically distributed random sample\(^1\) \(X_1, X_2, \ldots, X_n\) from a continuous, univariate density \(f\). De-

\(^1\)i.i.d. assumption is not required for the definition of the estimator, however it is useful later when we consider MSE and MISE
note by $\hat{f}(x; h)$ the kernel density estimator of $f$, which is defined as

$$
\hat{f}(x; h) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right).
$$

(2.1)

The function $K$ is called the ‘kernel’ and it satisfies $\int K(x)dx = 1$, and $h$ is usually called the ‘bandwidth’ and is a positive number. The following rescaled version of the kernel is useful

$$
K_h(u) = \frac{1}{h} K\left(\frac{u}{h}\right).
$$

For example, if the kernel is $N(0, 1)$, then the scaled kernel is $N(0, h^2)$ as

$$
K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}
$$

and

$$
K_h(u) = \frac{1}{h} K\left(\frac{u}{h}\right) = \frac{1}{\sqrt{2\pi h^2}} e^{-\frac{u^2}{2h^2}}.
$$

This gives the following formula for kernel density estimator

$$
\hat{f}(x; h) = \frac{1}{n} \sum_{i=1}^{n} K_h\left(x - X_i\right).
$$

### 2.2 Measures of efficiency

A common way of measuring efficiency of an estimator is the mean square error (MSE). Suppose $\hat{f}(x; h)$ is an estimator of the density $f(x)$ at the point $x$ where $x \in \mathbb{R}$, then the MSE of $\hat{f}(x; h)$ is defined in the following way

$$
\text{MSE}\hat{f}(x; h) = \mathbb{E}(\hat{f}(x; h) - f(x))^2.
$$

We see that MSE is the second moment of $\hat{f}(x; h)$ about $f(x)$ and is a function of $f(x)$. The MSE can be worked out easily using the variance and bias of the estimator, which can be easily calculated (or are already known).

We have

$$
\text{MSE}\hat{f}(x; h) = \mathbb{E}(\hat{f}(x; h) - f(x))^2
$$

$$
= \mathbb{E}(\hat{f}(x; h) - \mathbb{E}\hat{f}(x; h) + \mathbb{E}\hat{f}(x; h) - f(x))^2
$$

$$
= \mathbb{E}(\hat{f}(x; h) - \mathbb{E}\hat{f}(x; h))^2 + 2(\mathbb{E}\hat{f}(x; h) - f(x))\mathbb{E}(\hat{f}(x; h)
$$

$$
- \mathbb{E}\hat{f}(x; h)) + (\mathbb{E}\hat{f}(x; h) - f(x))^2
$$

$$
= \text{Variance}(\hat{f}(x; h)) + [\text{Bias}(\hat{f}(x; h))]^2,
$$

6
Figure 2.1: This figure shows a kernel chosen to be $N(0,1)$ and two scaled kernels depicting that ‘$h$’ is the scaling factor, $h_1 = 0.9$ and $h_2 = 0.7$.

Figure 2.2: Three different kernels shown in this figure are Normal, Epanechnikov and Triangle kernel. For their formulae see (2.4)
since the second term in the second last inequality is zero and the first and the last terms are variance and squared bias of the estimator respectively. It is clear from the last expression that if the bias of the estimator is zero, then the MSE is equal to the variance of the estimator. To make the formula for the MSE explicit, we need to specify the formulae for the bias and variance of the estimator. Let $f$ be the density of the random variable $Y$. Then

$$\mathbb{E}\hat{f}(x; h) = \mathbb{E}K_h(x - Y)$$

$$= \int K_h(x - y)f(y)dy$$

$$= (K_h * f)(x)$$

Where $(K_h * f)(x) = \int K_h(x - y)f(y)dy$ is known as convolution. Using this notation the squared bias term for $f$ can be written as

$$[\text{Bias}(\hat{f}(x; h))]^2 = [(K_h * f)(x) - f(x)]^2$$

and the variance as

$$\text{Var}(\hat{f}(x; h)) = \text{Var}(\frac{1}{n} \sum_{i=1}^{n} K_h(x - Y))$$

$$= \frac{1}{n} \text{Var}(K_h(x - Y))$$

$$= \frac{1}{n} [\mathbb{E}(K_h(x - Y))^2 - (\mathbb{E}K_h(x - Y))^2]$$

$$= \frac{1}{n} [(K_h^2 * f)(x) - (K_h * f)^2(x)]$$

using the convolution notation. This gives the following expression for MSE

$$\text{MSE}\hat{f}(x; h) = \frac{1}{n}[(K_h^2 * f)(x) - (K_h * f)^2(x)] + [(K_h * f)(x) - f(x)]^2$$

It is worth noting that in parametric settings the rate of convergence of the MSE is typically $1/n$, however in non-parametric setting the rate of convergence is usually slower than $1/n$. The MSE gives the error related to a fixed point. To measure the error globally, criteria such as the ISE(integrated squared error) are used. The ISE is the square of the $L^2$ distance between $f(x)$ and $\hat{f}(x; h)$, given by

$$\text{ISE}\hat{f}(.; h) = \int [\hat{f}(x; h) - f(x)]^2 dx$$
It is easy to notice that the ISE has a drawback of depending on the particular data set in hand. It is, therefore, appropriate to consider the mean of the ISE known as the MISE, given by

$$\text{MISE} = E \int [\hat{f}(x; h) - f(x)]^2 dx$$

The change of order of integration is allowed (by Fubini’s theorem) because the integrand is non-negative, so it can be written as

$$\text{MISE} = E \int [\hat{f}(x; h) - f(x)]^2 dx = \int E[\hat{f}(x; h) - f(x)]^2 dx = \int \text{MSE} \hat{f}(x; h)$$

$$= \frac{1}{n} \int [(K_h^2 * f)(x) - (K_h * f)^2(x)]dx + \int [(K_h * f)(x) - f(x)]^2 dx$$

$$= \frac{1}{nh} \int K^2(x)dx + (1 - \frac{1}{n}) \int (K_h * f)^2(x)dx - 2 \int (K_h * f)(x)f(x)dx + \int f(x)^2 dx$$

where the second last step uses the expression for the MSE.

### 2.3 Examples of Kernel Estimation

Here we consider kernel density estimation of the following two known normal mixture densities.

$$f(x) = \frac{1}{2} N(0, .5) + \frac{1}{2} N(3, .5) \quad (2.2)$$

and

$$g(x) = \frac{1}{2} N(0, 1) + \frac{1}{10} N(-.1, .1) + \frac{1}{10} N(-.5, .1) + \frac{1}{10} N(0, .1)$$

$$+ \frac{1}{10} N(.5, .1) + \frac{1}{10} N(1, .1). \quad (2.3)$$

The kernels used in the example are:

- **Normal** $\frac{1}{\sqrt{2\pi}} \exp(-1/2t^2)$ \hfill (2.4)
- **Epanechnikov** $\frac{3}{4} (1 - 1/5t^2) / \sqrt{5}$ for $|t| < \sqrt{5}$, otherwise 0 \hfill (2.5)
- **Triangle** $1 - |t|$ for $|t| < 1$, otherwise 0 \hfill (2.6)
The figures 2.3 and 2.4 show that the choice of the kernel does not make a huge difference and estimates based on different kernels are almost indistinguishable with each other when the same bandwidth is used. The effect of the bandwidth is visible in the figures. A small value produces spurious peaks and troughs whereas a large value overshadows the individual modes. The effect of the bandwidth is specially more visible in the density (2.3). Also notable from figure 2.4 is that the middle panel gives a better estimate of the peaks and the bottom panel gives a better estimate of the tails of the density. Thus a universally acceptable bandwidth, i.e. a bandwidth that is good for all \( x \), in general cannot be found.

### 2.4 Density estimation using wavelets

In this section we discuss estimation of a univariate density \( f \) from independent and identically distributed data points using wavelets. This section is mostly based on [7] and [2]. Let \( X_1, X_2, \ldots, X_n \) be i.i.d. random variables with a common density \( f \). Suppose the density function \( f \) is square integrable i.e. \( f \in L^2(\mathbb{R}) \) and \( \psi(x) \) and \( \phi(x) \) are the wavelet and scaling function generated by an orthonormal multiresolution decomposition of \( L^2(\mathbb{R}) \). The formal wavelet series expansion of the density function \( f \) is then given by

\[
f(x) = \sum_{k \in \mathbb{Z}} c_{j_0 k} \phi_{j_0 k}(x) + \sum_{j \in \mathbb{J}} \sum_{k \in \mathbb{Z}} d_{j k} \psi_{j k}(x),
\]

where \( \mathbb{J} = \{ m \in \mathbb{Z} : m \geq j_0 \} \), where \( j_0 \) is called the primary resolution level. The coefficients \( c_{j_0 k} \) and \( d_{j k} \) are given by

\[
c_{j_0 k} = \int_{-\infty}^{\infty} \phi_{j_0 k}(x) f(x) dx = \mathbb{E}(\phi_{j_0 k}(X_i)),
\]

\[
d_{j k} = \int_{-\infty}^{\infty} \psi_{j k}(x) f(x) dx = \mathbb{E}(\psi_{j k}(X_i)).
\]

The wavelet basis functions at a resolution level \( j \) are given by

\[
\phi_{j k}(x) = 2^{j/2} \phi(2^j x - k)
\]

\[
\psi_{j k}(x) = 2^{j/2} \psi(2^j x - k),
\]

where \( \psi \) is the wavelet function and \( \phi \) is the associated scaling function.

To estimate the density function \( f \) we only need to estimate the coefficients
Figure 2.3: Kernel density estimates based on 1000 observations of (2.2). Normal, Epanechnikov and Triangle kernels are used for each Figure, actual density is shown by dotted line. Bandwidth is 0.1 for the first, 0.2 for the second and 0.3 for the third Figure.
Figure 2.4: Kernel density estimates based on 1000 observations of (2.3). Normal, Epanechnikov and Triangle kernels are used for each Figure, actual density is shown by dotted line. Bandwidth is 0.01 for the first, 0.05 for the second and 0.15 for the third Figure.
\( c_{j0k} \) and \( d_{jk} \). These can be estimated simply by

\[
\hat{c}_{j0k} = \frac{1}{n} \sum_{i=1}^{n} \phi_{j0k}(X_i),
\]

\[
\hat{d}_{jk} = \frac{1}{n} \sum_{i=1}^{n} \psi_{jk}(X_i).
\]

These estimates are calculated for \( \{j_0 \leq j < J\} \) for some large value of \( J \). The \( \hat{d}_{jk} \) are then ‘thresholded’. Thresholding can be described as the process of setting to zero the estimated whose absolute values are lower than the threshold. It helps in avoiding spurious oscillations and controlling the smoothness of the estimated density. The resulting estimate of the true density \( f \) is then given by

\[
\hat{f}(x) = \sum_{k \in \mathbb{Z}} \hat{c}_{j0k} \phi_{j0k}(x) + \sum_{j=j_0}^{J-1} \sum_{k \in \mathbb{Z}} \hat{d}_{jk} \psi_{jk}(x)
\]

The equation (2.7) needs some explanation. Let \( \{\phi_{j,s}(x)\} \) be the space of functions that we can derive from the different values of \( j \) and \( s \). It is easy to see that the subspace of functions \( \{\phi_{0,s}(x)\} \subset \{\phi_{1,s}(x)\} \) and so on. We need to chose a resolution level \( j_0 \) which will determine the smoothness of the estimator. The part of the function lying outside the subspace \( \{\phi_{j_0,s}(x)\} \) is then analyzed by the second term in the equation. Where the subspace formed by \( \{\psi_{j,s}(x)\} \) is the difference subspace of \( \{\phi_{j,s}(x)\} \) and \( \{\phi_{j+1,s}(x)\} \) i.e. for \( s \in \mathbb{Z} \), \( \{\psi_{j,s}(x)\} \oplus \{\phi_{j,s}(x)\} = \{\phi_{j+1,s}(x)\} \).

The following function is known as the Haar (mother) wavelet

\[
\psi(x) = \begin{cases} 
1 & \text{if } 0 \leq x < 0.5, \\
-1 & \text{if } 0.5 \leq x \leq 1, \\
0 & \text{if otherwise.}
\end{cases}
\]

The associated scaling function is

\[
\phi(x) = \begin{cases} 
1 & \text{if } 0 \leq x < 1, \\
0 & \text{if otherwise.}
\end{cases}
\]

Figure 2.6 shows shifted and scaled versions of the Haar wavelet. The shifted version just moves one step to the right and for the scaled version the base becomes halved and the height is multiplied by \( \sqrt{2} \).
2.4.1 Examples of wavelet estimation

The wavelet estimates based on the same sample of data points as used in kernel estimation (see Figures 2.3 and 2.4) are shown in the Figure 2.5. These estimates were calculated using the wavelet toolbox of MATLAB. The estimates were calculated using Haar wavelets and a global threshold utilizing a fast wavelet transform. It is important to notice that wavelet estimators adapt locally. The error in estimation is uniform throughout the estimators while in contrast kernel estimators are good at peaks or at tails.
Figure 2.6: The shifted and scaled versions of Haar Wavelet
Chapter 3

Advanced methods in Kernel estimation

3.1 Efficient computation of Kernel Density Estimators using fast Fourier Transform

Here we consider kernel estimation using the fast Fourier transform (FFT) as given in [10]. The direct computation of the kernel density estimate could be very time consuming. An approximation can be made realizing that a kernel estimate is a convolution of the data with the kernel. Indeed we have

\[ \hat{f}(x; h) = \int K_h(x - y)d\hat{F}_n(y), \]

where \( \hat{F}_n(y) \) denotes the empirical distribution function of the data. Utilizing the fast Fourier transform in calculating the Fourier transform to perform the convolution offers considerable efficiency. Let \( \mathbf{x} \) be a vector than its discrete Fourier transform is given by

\[ \mathbf{X}(k) = \sum_{j=0}^{N-1} \mathbf{x}(j)e^{-2\pi ijk/N}, \]

and inverse Fourier transform is given by

\[ \mathbf{x}(j) = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{X}(k)e^{2\pi ijk/N}. \]

Let us assume that we have a sample \( X_1, \ldots, X_n \) of independent and identically distributed observations from a continuous univariate distribution with
probability density \( f \) that we are trying to estimate. The Fourier transform (or characteristic function) of the function \( f \) is given by

\[
\phi_f(s) = \int e^{ist} f(t) \, dt.
\]

Define \( \phi_{emp}(s) \) to be the empirical characteristic function of the data,

\[
\phi_{emp}(s) = \int e^{ist} d\hat{F}_n(t) = \frac{1}{n} \sum_{j=1}^{n} \exp isX_j.
\]

Recall that, the kernel estimate is given by

\[
\hat{f}(x; h) = \frac{1}{nh} \sum_{j=1}^{n} K \left( \frac{x - X_j}{h} \right)
\]

where \( K \) is the kernel and \( h \) is called the bandwidth. The characteristic function of the scaled kernel \( K_h(x) = h^{-1}K(x/h) \) in terms of the characteristic function of the kernel is given by

\[
\phi_{K_h}(s) = \int e^{isx} K_h(x) \, dx = \int e^{isx} \frac{1}{h} K \left( \frac{x}{h} \right) \, dx = \int e^{isht} K(t) \, dt = \phi_K(sh).
\]

(3.1)

since

\[
\phi_{\hat{f}}(s) = \int e^{isx} \frac{1}{nh} \sum_{j=1}^{n} K \left( \frac{x - X_j}{h} \right) \, dx
\]

\[
= \frac{1}{n} \sum_{j=1}^{n} \int e^{isx} K_h(x - X_j) \, dx
\]

\[
= \frac{1}{n} \sum_{j=1}^{n} \int e^{is(x - X_i)} K_h(x - X_i) e^{isX_i} \, dx
\]

\[
= \phi_K(sh) \frac{1}{n} \sum_{j=1}^{n} e^{isX_i}
\]

\[
= \phi_K(sh) \phi_{emp}(s).
\]

Now we can invert \( \phi_{\hat{f}} \) using the inverse Fourier transform utilizing a fast Fourier transform algorithm to get the estimate of the unknown density \( f \).
Algorithm

We shall use the standard normal density as the kernel for the algorithm. Since the choice of a kernel does not make a huge impact on the estimate, the kernel can be suitably chosen for ease of computation. Since we will be using the fast Fourier transform algorithm which gives a discrete Fourier transform, we start with binning the data on $M = 2^r$ points. Take $a = \min(X_i) - 3h$ and $b = \max(X_i) + 3h$, where $h$ is the chosen bandwidth. This interval will be large enough in the case of the standard normal density as kernel to avoid any periodicity problems in fast Fourier transform (since $1 - P(-3 \leq Z \leq 3) = 0.0027$). Now, define $\delta$, the length between each of the $M$ bins at the $M$ points to be $(b - a)/M$ and $t_k = a + k\delta$ for $k = 0, \ldots, M - 1$. The binning can be done as follows. If a data point $X$ falls in the interval $[t_k, t_{k+1}]$, it is split into a weight $\frac{1}{n\delta^2}(t_{k+1} - X)$ at $t_k$ and $\frac{1}{n\delta^2}(X - t_k)$ at $t_{k+1}$. Now the sum of all the weights equals

$$\sum_i \left( \frac{1}{n\delta^2}(t_{k+1} - X_i) + \frac{1}{n\delta^2}(X_i - t_k) \right) = \frac{1}{n\delta^2} \sum_i (t_{k+1} - t_k) = \frac{n\delta}{n\delta^2} = \frac{1}{\delta}$$

Let us call the sequence of weights $w_k$. The next step is to find the empirical characteristic function, For $-\frac{M}{2} \leq l \leq \frac{M}{2}$, we define

$$Y_l = \frac{1}{M} \sum_{k=0}^{M-1} w_k \exp(i2\pi kl/M).$$

(3.2)

Define $s_l = \frac{2\pi l}{b-a}$. To make the algebra easy assume for a moment that $a = 0$. If $a \neq 0$ the algebra becomes a little complicated but the end result is the
same. So, if $a = 0$ then $t_k = k\delta$, $\delta = b/M$ and $s_l = 2\pi l/b$, then we have the rough approximation:

$$Y_l = \frac{1}{M} \sum_{k=0}^{M-1} w_k \exp(i2\pi kl/M)$$

$$= \frac{1}{M} \sum_{k=0}^{M-1} w_k \exp(i(2\pi l/b)(kb/M))$$

$$= \frac{1}{M} \sum_{k=0}^{M-1} w_k \exp(it_k s_l)$$

$$\approx \frac{1}{nM\delta} \sum_j \exp(is_l X_j)$$

$$= \sqrt{2\pi} \frac{u(s_l)}{(b-a)}.$$  

Here $u(s_l)$ denotes the empirical characteristic function. Define a sequence $\zeta_l^* = \exp(-.5h^2 s_l^2) Y_l$ and let $\zeta_k$ be the inverse discrete fourier transform of $\zeta_l^*$. Then the density to be estimated is given by $\zeta_k$ with

$$\zeta_k = \sum_{l=-M/2}^{M/2} \exp(-2\pi ikl/M)\zeta_l^*$$

$$\approx \sum_{l=-M/2}^{M/2} \exp(-is_l t_k)\sqrt{2\pi}(b-a) \exp(-.5h^2 s_l^2)u(s_l)$$

$$\approx \frac{1}{\sqrt{2\pi}} \int \exp(-ist_k) \exp(-.5h^2 s^2)u(s)ds$$

$$= \hat{f}(t_k).$$

### 3.1.2 Examples

Figure 3.2 show the Estimate of the univariate density of (a) Gamma(9,.5) and (b) Normal(0,1) random variable $X$ using FFT with $n = 1000$ samples and bandwidth $h = .4$. The estimator performs very well.

### 3.2 Deconvolution estimator

Let us now consider the deconvolution problem. Here we assume that $X_1, X_2, \ldots, X_n$ is a random sample having a common density $f_X$ that is
Figure 3.2: Estimate of univariate density of (a) Gamma(9,.5) and (b) Normal(0,1) random variable using FFT $n=1000$ samples, $h=.4$
of interest. However, the $X_i$’s are not directly observed. Instead $Y_i$’s with a common density $f_Y$ are observed where

$$Y_i = X_i + Z_i, \quad i = 1, \ldots, n,$$

and the $Z_i$’s are random variables with common density $f_Z$ and independent of $X_i$. We assume that $f_Z$ is known and we want to estimate $f_X$. Now the characteristic function or Fourier transform of the density $f_Y$ is given by

$$\phi_{f_Y}(t) = \mathbb{E}[\exp(itY)] = \int_{-\infty}^{\infty} \exp(ity) f_Y(y) dy$$

Thus we have that the characteristic function of $f_X, f_Y$ and $f_Z$ by independence satisfy

$$\phi_{f_Y}(t) = \mathbb{E}(\exp(itY)) = \mathbb{E}(\exp(it(X+Z))) = \mathbb{E}(\exp(itX))\mathbb{E}(\exp(itZ)) = \phi_{f_X}(t)\phi_{f_Z}(t).$$

It is possible to estimate $f_X$, using the Fourier inverse theorem, by

$$f_X(x) = \frac{1}{2\pi} \int \exp(-itx) \phi_{f_X}(t) dt = \frac{1}{2\pi} \int \exp(-itx) \frac{\phi_{f_Y}(t)}{\phi_{f_Z}(t)} dt$$

If we use an estimate $\hat{f}_Y(y) = 1/n \sum K_h(y - Y_i)$ of $f_Y(y)$ to obtain an estimate $\phi_{f_Y}(t)$ of $\phi_{f_Y}(t)$, we obtain the estimate $\hat{f}_X$ of $f_X$ as

$$\hat{f}_X(x : h) = \frac{1}{2\pi} \int \exp(-itx) \frac{\phi_{f_Y}(t)}{\phi_{f_Z}(t)} dt, \quad (3.3)$$

where

$$\phi_{f_Y}(t) = \int_{-\infty}^{-\infty} \exp(ity) \hat{f}_Y dy$$

$$= \int_{-\infty}^{-\infty} \exp(ity) \frac{1}{n} \sum_{i=1}^{n} K_h(y - Y_i) \exp(-itY_i) \exp(itY_i) dy$$

$$= \frac{1}{n} \sum_{i=1}^{n} \phi_{K_h}(t) \exp(-itY_i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \int \exp(ity) \frac{1}{R} K\left(\frac{y}{R}\right) dy \exp(-itY_i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \phi_{K}(th) \exp(-itY_i) dy.$$
Thus (3.3) gives the following estimator

\[ \hat{f}_X(x : h) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(-itx) \frac{\hat{\phi}_Y(t)}{\phi_z(t)} \]

\[ = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(-itx) \frac{1}{n} \sum_{i=1}^{n} \frac{\phi_K(th) \exp(-itY_i)}{\phi_z(t)} dt \]

\[ = \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp[-is(x - Y_i)/h] \frac{\phi_K(s)}{\phi_z(s/h)} ds \]

\[ = \frac{1}{nh} \sum_{i=1}^{n} v_h \left( \frac{x - Y_i}{h} \right), \]

where

\[ v_h(x) = \int_{-\infty}^{\infty} \exp(-isx) \frac{\phi_K(s)}{\phi_z(s/h)} ds. \]

### 3.2.1 Examples

Figure 3.3 shows the deconvolution estimation of (a) Normal(0,1) density (b) Gamma(9,.5) both measured with Normal(0,0.1) error using a bandwidth of \( h = .4 \) with 100 observations. Again the estimator performs very well. The estimate were not calculated using FFT algorithm. The FFT algorithm can be used to calculate the estimates faster.
Figure 3.3: Deconvolution estimation of (a) Normal(0,1) density (b) Gamma(9,.5) both measured with Normal(0,.1) error , $h=.4$, $n=100$
Chapter 4

Stochastic volatility estimation

We consider stochastic volatility density estimation using a deconvolution estimator based on [5].

4.1 Stochastic Volatility estimation

Let $X$ denote the log price of an asset traded in the market. Assume that it follows an Itô diffusion process

$$dX_t = \mu_t dt + \sigma_t dW_t \quad X_0 = 0.$$  

Here $\mu$ is the drift process, $\sigma$ is the volatility process and $W$ is the standard Brownian motion. We consider the equation without the drift term i.e. $\mu = 0$. Therefore, we have

$$dX_t = \sigma_t dW_t.$$  

It is assumed that we observe the process at discrete time instants $0, \Delta, 2\Delta, \ldots, n\Delta$ such that $\Delta \to 0$ and $n\Delta \to \infty$. By $X_t^\Delta$ we denote the normalized increments

$$X_t^\Delta = \frac{1}{\sqrt{\Delta}}(X_{i\Delta} - X_{(i-1)\Delta}).$$  

24
Since \( \Delta \to 0 \), we have that

\[
X_i^\Delta = \frac{1}{\sqrt{\Delta}} \int_{(i-1)\Delta}^{i\Delta} \sigma_t dW_t \\
\approx \sigma_{(i-1)\Delta} \frac{1}{\sqrt{\Delta}} (W_i^\Delta - W_{(i-1)\Delta}) \\
= \sigma_{(i-1)\Delta} Z_i^\Delta,
\]

where

\[
Z_i^\Delta = \frac{1}{\sqrt{\Delta}} (W_i^\Delta - W_{(i-1)\Delta}).
\]

\( W_t \) is Brownian motion, since increments of BM are independent, \( Z_1^\Delta, Z_2^\Delta, \ldots, Z_n^\Delta \) are independent and identically distributed standard normal random variables with \( Z_i^\Delta \) standard normal. Since we have

\[
X_i^\Delta \approx \sigma_{(i-1)\Delta} Z_i^\Delta.
\]

Taking square of the equation and then logarithms, we return to our setting of data measured with error, from where we can evaluate the density of \( \sigma_{(i-1)\Delta} \) using the deconvolution method. We have

\[
\log (X_i^\Delta)^2 \approx \log (\sigma_{(i-1)\Delta})^2 + \log (Z_i^\Delta)^2.
\]

We drop the \( \Delta \) in the notation for the sake of simplicity. The density of \( Z_i \) is Gaussian, so \( x \mapsto \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \). First we compute the distribution function,

\[
P[\log(Z_i^2) \leq x] = P[Z_i^2 \leq e^x] \\
= P[-e^{x/2} \leq Z_i \leq e^{x/2}] \\
= 2P[Z_i \leq e^{x/2}] - 1 \\
= \frac{2}{\sqrt{2\pi}} \int_{-\infty}^{e^{x/2}} e^{-s^2/2} ds - 1.
\]
Let \( f_Z \) denote the density of \( \log Z_i^2 \). We obtain,

\[
    f_Z(x) = \frac{d}{dx} \left( \frac{2}{\sqrt{2\pi}} \int_{-\infty}^{x/2} e^{-s^2/2} ds - 1 \right)
\]

\[
    = \frac{2}{\sqrt{2\pi}} \left[ \left( \frac{d}{dx} e^{x/2} \right) e^{-e^x/2} \right]
\]

\[
    = \frac{2}{\sqrt{2\pi}} \left[ \frac{1}{2} e^{x/2} e^{-e^x/2} \right]
\]

\[
    = \frac{1}{\sqrt{2\pi}} e^{x/2} e^{-e^x/2}.
\]

The characteristic function of \( f_Z \) is given by (see [4])

\[
    \phi_{f_Z}(t) = \frac{2it}{\sqrt{\pi}} \Gamma \left( \frac{1}{2} + it \right)
\]

Since the choice of the kernel does not make a huge difference, the kernel chosen to estimate the density is chosen by relative ease of computation. The kernel used to compute estimates is given by

\[
    K(x) = \frac{48x(x^2 - 15) \cos x - 144(2x^2 - 5) \sin x}{\pi x^7}.
\]

(4.1)

It has a characteristic function given by

\[
    \phi_K(t) = (1 - t^2)^3, \quad |t| \leq 1
\]

which has a compact support as required for the deconvolution estimator to be well defined.

### 4.2 Examples of Stochastic Volatility estimation

#### 4.2.1 GARCH(1,1)

Let \( X_t \) be the detrended log return process which behaves as a GARCH(1,1) process. The GARCH(1,1) process, for given non-negative constants \( \alpha_0, \alpha_1 \) and \( \beta \) and iid sequence \( Z_t \), typically Gaussian noise, is described by the following equations.

\[
    X_t = \sigma_t Z_t,
\]

\[
    \sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \beta \sigma_{t-1}^2.
\]

Our aim is to estimate the univariate stationary density of \( \log \sigma_t^2 \). Its existence is guaranteed by [4] and [1].
Figure 4.1: Estimate of univariate density of $(\log \sigma^2_t)$ in GARCH(1,1) (a) Direct deconvolution (b) Using FFT; $h=.4$, $\alpha_0 = 2$, $\alpha_1 = 0.8$, $\beta = 0.1$
4.2.2 Ornstein-Uhlenbeck process

Let \( u_t = \log \sigma^2_t \). Then for given positive constants \( \alpha, b, c \), the Ornstein-Uhlenbeck process is defined as

\[
du(t) = \alpha(b - u(t))dt + cdW, \tag{4.2}
\]

where \( W_t \) is a standard Brownian motion, \( \alpha, b, c \) are positive constants. This process is mean reverting, since the drift is positive for \( b - u_t > 0 \) and negative for \( b - u_t < 0 \). Thus \( u_t \) is continually driven towards the long-term level \( b \) at the rate \( \alpha \). As mentioned in [6], \( u_t \) has a stationary distribution \( N(b, c^2/2\alpha) \). If \( u_0 = N(b, c^2/2\alpha) \) then \( u_t \) for all \( t \) has the same distribution. The solution of Equation (4.2) is given by

\[
u(t) = e^{-\alpha t}u(0) + \alpha b \int_0^t e^{-\alpha(t-s)}ds + c \int_0^t e^{-\alpha(t-s)}dW_s
\]

similarly for \( 0 < r < t \)

\[
u(t) = e^{-\alpha(t-r)}u(r) + \alpha b \int_r^t e^{-\alpha(t-s)}ds + c \int_r^t e^{-\alpha(t-s)}dW_s
\]

It follows that, given \( u(r) \), the value of \( u(t) \) is conditionally normally distributed with mean \( e^{-\alpha(t-r)}u(r) + b(1 - e^{-\alpha(t-r)}) \) and variance

\[
\sigma^2_u(r, t) = c^2 \int_r^t e^{-2\alpha(t-s)}ds = \frac{c^2}{2\alpha} (1 - \exp(-2\alpha(t - u))).
\]

Let \( \Delta_i = t_{i+1} - t_i \). To simulate the stationary version of \( u_t \), we need to draw \( u_0 \) from \( N(b, c^2/2\alpha) \) and simulate the process according to the following equation

\[
u_{t_{i+1}} = \exp(-\alpha \Delta_i)u_{t_i} + b(1 - \exp(-\alpha \Delta_i)) + c \sqrt{\frac{1}{2\alpha} (1 - \exp(-2\alpha \Delta_i)))} Z_{t_{i+1}}
\]

where \( Z_i \) are drawn from \( N(0,1) \).

4.2.3 Multivariate case

An estimate of bivariate density of \( \log \sigma^2_t, \log \sigma^2_{t-1} \) based on algorithm of [3], see also [8] can be calculated similarly. The FFT algorithm was used for deconvolution to speed up the process. The result can be seen in Figure 4.3.
Figure 4.2: Estimate of density of $(\log \sigma_t^2)$ which follows Ornstein-Uhlenbeck process with $\alpha = 1, c = .5, b = 1$ based on $n = 1000$ (a) Kernel estimate (b) FFT estimate
Figure 4.3: Estimate of bivariate density of (log $\sigma_t^2$, log $\sigma_{t-1}^2$) in GARCH(1,1) based on $n = 1000$ (a) $h = .4$ (b) $h = .5$ ; $\alpha_0 = 2$, $\alpha_1 = 0.8$, $\beta = 0.1$
Chapter 5

Wavelet Deconvolution

In this chapter, we construct a linear wavelet type deconvolution estimator using Meyer-type wavelets based on [9] and [11].

5.1 Meyer-type wavelet

In this chapter, we use a special class of band-limited wavelets whose Fourier transform have a bounded support. A particular of band limited wavelet-Meyer wavelet is used here. The Meyer wavelet and scaling function are defined in the Fourier domain by (see [2])(\( \hat{f} \) denotes Fourier transform of \( f \))

\[
\hat{\psi}(w) = \frac{1}{\sqrt{2\pi}} \exp \left( \frac{1}{2} iw \right) \sin \left( \frac{\pi}{2} \nu \left( \frac{3}{2\pi} |w| - 1 \right) \right) \quad \frac{2\pi}{3} \leq |w| \leq \frac{4\pi}{3}, \\
\hat{\psi}(w) = \frac{1}{\sqrt{2\pi}} \exp \left( \frac{1}{2} iw \right) \cos \left( \frac{\pi}{2} \nu \left( \frac{3}{2\pi} |w| - 1 \right) \right) \quad \frac{4\pi}{3} \leq |w| \leq \frac{8\pi}{3}, \\
\hat{\psi}(w) = 0 \quad \text{otherwise.}
\]

and

\[
\hat{\phi}(w) = \frac{1}{\sqrt{(2\pi)}} |w| \leq \frac{2\pi}{3}, \\
\hat{\phi}(w) = \frac{1}{\sqrt{(2\pi)}} \cos \left( \frac{\pi}{2} \nu \left( \frac{3}{2\pi} |w| - 1 \right) \right) \quad \frac{2\pi}{3} \leq |w| \leq \frac{4\pi}{3}, \\
\hat{\phi}(w) = 0 \quad \text{otherwise.}
\]

where

\[
\nu(x) = x^4(35 - 84x + 70x^2 - 20x^3), x \in [0, 1].
\]
5.2 Linear wavelet type Deconvolution estimator

Here we have our original setting where we assume that $X_1, X_2, \ldots, X_n$ be a random sample having a common density $f_X$ that is of interest. However, $X_i$’s are not directly observed. Instead $Y_i$’s with common density $f_Y$ are observed and

$$Y_i = X_i + Z_i \quad i = 1, \ldots, n$$

where $Z_i$’s are random variable with common density $f_Z$ and are independent of $X_i$. The density $f_Z$ is known. The basic idea is to present $f_X$ via a wavelet expansion and then to estimate the coefficients using a deconvolution algorithm. Let $\phi(x)$ and $\psi(x)$ be respectively the scaling function and a wavelet generated by an orthonormal multiresolution decomposition of $L^2(-\infty, \infty)$, and recall that

$$\phi_{jk}(x) = 2^{j/2} \phi(2^j x - k),$$
$$\psi_{jk}(x) = 2^{j/2} \psi(2^j x - k).$$

The formal wavelet expansion of the density function $f$ is then given by

$$f(x) = \sum_{j_0 \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} c_{j_0 k} \phi_{j_0 k}(x) + \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} d_{j k} \psi_{j k}(x),$$

(5.1)
where $\mathcal{J} = \{m \in \mathbb{Z} : m \geq j_0\}$, where $j_0$ is called the primary resolution level.

The coefficients $c_{j_0k}$ and $d_{jk}$ are given by

$$
c_{j_0k} = \int_{-\infty}^{\infty} \phi_{j_0k}(x)f_X(x)dx = \mathbb{E}(\phi_{j_0k}(X_i)),
$$

$$
d_{jk} = \int_{-\infty}^{\infty} \psi_{j_0k}(x)f_X(x)dx = \mathbb{E}(\psi_{j_0k}(X_i)).
$$

The basic idea is to approximate the function $f$ by the orthogonal projection given by the terms $\sum k \in \mathbb{Z} c_{j_0k}\phi_{j_0k}(x)$ (see (5.1)). For $j_0$ large enough and using the approximation properties of specific family of wavelets, here Meyer’s, the second term can be controlled [11].

Now, $c_{j_0k}$ and $d_{jk}$ can be viewed as mathematical expectations of the functions $u_{j_0k}$ and $v_{jk}$

$$
c_{j_0k} = \int_{-\infty}^{\infty} u_{j_0k}(y)f_Y(y)dy
$$

$$
d_{jk} = \int_{-\infty}^{\infty} v_{j_0k}(y)f_Y(y)dy,
$$

where $u_{j_0k}$ and $v_{jk}$ are the solutions to the following equations:

$$
\int_{-\infty}^{\infty} f_Z(y-x)u_{j_0k}(y)dy = \phi_{j_0k}(x), \quad (5.2)
$$

$$
\int_{-\infty}^{\infty} f_Z(y-x)v_{jk}(y)dy = \psi_{jk}(x). \quad (5.3)
$$

Indeed, noting that $f_Y(y) = \int_{-\infty}^{\infty} f_Z(y-x)f_X(x)dx$, we have

$$
c_{j_0k} = \int_{-\infty}^{\infty} u_{j_0k}(y)f_Y(y)dy
$$

$$
= \int_{-\infty}^{\infty} u_{j_0k}(y) \left( \int_{-\infty}^{\infty} f_Z(y-x)f_X(x)dx \right) dy
$$

$$
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u_{j_0k}(y)f_Z(y-x)f_X(x)dxdy
$$

$$
= \int_{-\infty}^{\infty} \phi_{j_0k}(x)f_X(x)dx.
$$

Now, if we take the Fourier transform of (5.2) (and the same steps can be done for (5.3)), we get

$$
u_{j_0,k}(y) = 2^{j_0/2}U_{j_0} \left( 2^{j_0/2}y - k \right), \quad (5.4)$$
where $U_{j_0}$ is the inverse Fourier transform of $\tilde{\phi}(w)/\tilde{f}_Z(-2^{j_0}w)$ where $\tilde{f}$ denotes the Fourier transform of a function $f$. We can now approximate $c_{j_0k}$ by

$$\hat{c}_{j_0k} = \frac{1}{n} \sum_{l=1}^{n} 2^{j_0/2} U_{j_0} \left(2^{j_0/2} Y_l - k\right).$$

Then truncating the series (5.1), we get a linear wavelet estimator $\tilde{f}_{X,n}$ of $f_X$

$$\tilde{f}_{X,n}(x) = \sum_{k \in \mathbb{Z}} \hat{c}_{j_0k} \phi_{j_0k}(x). \quad (5.5)$$

### 5.3 An Example

Now we consider a numerical example from [9]. We apply linear Meyer-type wavelet deconvolution to a situation where the error density is given by double exponential distribution and the density of interest is the standard normal density (see figure 5.3). We have

$$f_Z(x) = \frac{1}{2} \sigma \exp(-\sigma|x|), \quad (5.6)$$

$$f_X(x) = \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{1}{2}x^2\right).$$

If $f_Z$ is given by Equation (5.6) above, then its Fourier transform is given by $\tilde{f}_Z(w) = (1 + \sigma^2 w^2)^{-1}$. We also have $U_{j_0}$ is inverse Fourier transform of $\tilde{\phi}(w)/\tilde{f}_Z(-2^{j_0}w)$, Let $P$ be a probability measure with support in $[-\pi/3, \pi/3]$. As in [9] and [11], define the scaling function $\phi(x)$ and the wavelet function $\psi(x)$ as the functions whose Fourier transforms are given by the nonnegative square roots of the integrals

$$\tilde{\phi}(w) = \left[\int_{w-\pi}^{w+\pi} dP\right]^{1/2}, \quad (5.7)$$

$$\tilde{\psi}(w) = \exp(-iw/2) \left[\int_{|w|/2-\pi}^{-|w|+\pi} dP\right]^{1/2}.$$  

(5.8)
Figure 5.2: Linear Meyer-type wavelet deconvolution

Figure 5.3: Deconvolution using Normal kernel
Then we have,

\[ U_{j_0}(x) = \int e^{iwx} \tilde{\phi}(w) / f_Z(-2^{j_0}w) dw, \]

\[ = \int e^{iwx} \left( \int_{w-\pi}^{w+\pi} dP \right)^{1/2} / \left( 1 + \sigma^2 2^{j_0} w^2 \right)^{-1} dw, \]

\[ = \int e^{iwx} \left( \int_{w-\pi}^{w+\pi} dP \right)^{1/2} dw + \sigma^2 2^{j_0} \int w^2 e^{iwx} \left( \int_{w-\pi}^{w+\pi} dP \right)^{1/2} dw, \]

\[ =: \phi(x) + 2^{j_0} \sigma^2 \phi''(x). \]

Now we can evaluate \( u_{j_0,k} \) using equation (5.4). The estimate was calculated on \( n = 256 \) points with \( j_0 = -1 \), taking \( k < |10| \) for the finite series estimator of \( \hat{f}_{X,n}(x) \) (see equation (5.5)) taking \( \sigma = 0.1 \). The approximation of \( \phi(x) \) was done using Cubic spline interpolation. The cubic spline interpolation is a piecewise continuous curve passing through each value. There is a different curve for each interval having its own coefficients. For \( x \in [x_i, x_{i+1}] \), the part of the curve is given by

\[ CS_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i. \]

There will be \( n \) intervals and \( n \) curves for \( n+1 \) points. The \( 4n \) coefficients are calculated according to following equations: \( CS_i(x_i) = y_i, CS_i(x_{i+1}) = y_{i+1} \), \( CS'_{i-1}(x_i) = CS'_i(x_i), CS''_{i-1}(x_i) = CS''_i(x_i) \), and two more equation e.g. \( CS''_0(x_0) = 0, CS''_{n-1}(x_n) = 0 \). Here \( y_i \) denotes the value of the function at point \( x_i \). Spline interpolation has smaller interpolation error than polynomial interpolation even when using low degree polynomials for spline. Spline interpolation produces curves that appear to be seamless but they are only piecewise continuous.
Appendix A

Programs

Programs require Statistical, Financial and Wavelet Toolboxes of MATLAB wherever applicable.

A.1 Deconvolution

function []=meas_w_err()
%simulating the process as a sum of normal distribution with standard normal error
Num=50;points=200;
xt=normrnd(0,1,points,1);
zt=normrnd(0,.1,points,1);
yt=xt+zt;
% trying to recover xt
h=.4;
x=linspace(-3, 3,Num);
x1=linspace(-3,+3,Num);
f=zeros(Num,1);
for j=1:Num
    for i =1:points
        u=(x(j)-yt(i))/h;
        f(j)= f(j)+ker(u,h);
    end
    f(j)=f(j)/(points*h);
end
set(gcf,'Color','w');
plot(x,f,x1,normpdf(x1,0,1),'--');
legend('Deconvolution Estimate','Actual density')
title('Deconvolution estimation of Normal density measured with Normal error')
%effective kernel
function [integl]=ker(u,h)
    function fval= vh(t,u,h)
        fval = real(1./(pi) .*exp(-1i.*t.*u) .* (1-t.^2).^3 .*exp(t.^2./2./h^2));
    end
    vhh=@(t) vh(t,u,h);
    integl=quad(vhh,0,1);
end
end

A.2 Density estimation using fast Fourier tranformation

function []=silvermanfftkernel()
l=1000;
x=gamrnd(9,.5,l,1);
M=2^7;
h=.4;
a=min(x)-3*h;
b=max(x)+3*h;
delta=(b-a)/M;
x1=linspace(a,b,M+1);
%data discretization
weight=zeros(M+1,1);
for k=1:M
    t_k=a+(k-1)*delta;
    t_k1=a+(k)*delta;
    for i=1:l
        if (x(i)>=t_k && x(i)<t_k1)
            weight(k)=weight(k)+(t_k1-x(i))/(l*delta^2);
            weight(k+1)=weight(k+1)+(-t_k+x(i))/(l*delta^2);
        end
    end
end
% % check
% sum(weight)
% (1/delta)
lll=((0:M)-M/2)';
ll=[lll(M/2+1:end); lll(1:M/2)]; % some functions like the fft require the
  % -ve frequency components to be mirrored around the highest frequency
y_l=ifft(weight);
s_l=2*pi*ll/(b-a);
zeta=exp(-.5*h^2.*(s_l.^2)).*y_l;
f=real(fft(zeta));
set(gcf,'Color','w')
plot(x1,f,x1,gampdf(x1,9,.5),'--');
legend('FFT','Actual')
title('Efficient Density estimation using FFT');
end

A.3 GARCH(1,1) Stochastic volatility estimation

function Final_garch1dfft()
  % Set the random number generator seed for reproducibility.
  randn('seed', 100)
  Alpha0 = 2;
  Alpha = 0.1;
  Beta = 0.8;
  Num = 1000;

  [U , H] = ugarchsim(Alpha0, Alpha, Beta, Num);
  u1=U(2:end);
  u2=U(1:end-1);
  u3=u1-u2;
  logX=log(u3.^2);
  logsigma=log(H);
  h=.4;
  [fsigma,xis]=ksdensity(logsigma,'width',h);
  l=length(logX);
  a=min(logX)+1;
  b=max(logX)+1;
  M=2^7;
  delta=(b-a)/(M);
  xs=linspace(a,b,M+1);
weight=zeros(M+1,1);
for k=1:M
    t_k=a+(k-1)*delta;
    t_k1=a+(k)*delta;
    for i=1:l
        if (logX(i)>=t_k && logX(i)<t_k1)
            weight(k)=weight(k)+(t_k1-logX(i))/(l*delta^2);
            weight(k+1)=weight(k+1)+(-t_k+logX(i))/(l*delta^2);
        end
    end
end
% check for weightsum= deltainverse
% weightsum=sum(weight)
% deltainverse=1/delta
%-------------------------------------------
Yl=ifft(weight);
lll=((0:M)-M/2)';
ll=[lll(M/2+1:end); lll(1:M/2)];
sl=ll.*(2*pi)./(b-a);
phi_w=zeros(M+1,1);
for i=1:M+1
    if (h*sl(i)>=-1 && h*sl(i)<=1)
        phi_w(i,1)=(1-(h*sl(i))^2)^3;
    end
end
phi_k=1/sqrt(pi)*2.^(1i.*sl).*cgamma(.5+1i*sl);
zetastar=phi_w.*Yl./phi_k;
estf=real(fft(zetastar));
set(gca,'Color','w')
plot(xs(1:M+1),estf,xis,fsigma);
legend('FFT','KDE')
title('Final estimated density using FFT in GARCH(1,1)');
end

A.4 GARCH(1,1) Deconvolution

function Final_kdedecov()
    randn('seed', 100)
Alpha0 = 2;
Alpha = 0.1;
Beta = 0.8;
Num = 200;

[U, H] = ugarchsim(Alpha0, Alpha, Beta, Num+1);
u1=U(2:end);
u2=U(1:end-1);
u3=u1-u2;
logX=log(u3.^2);
logh=log(H);
h=.4;
[fsigma,xis]=ksdensity(logh,'width',h);
x=linspace(min(logX)+1, max(logX)+1,50);
f=zeros(length(x),1);
for j=1:length(x)
    for i =1:Num
        u=(x(j)-logX(i))/h;
        f(j)= f(j)+Nker(u,h);
    end
    f(j)=f(j)/(Num*h);
end
set(gcf,'Color','w');
plot(x,f,xis,fsigma);
legend('Deconvolution Estimate','Direct Kernel Estimate')
title('Final estimated density using deconvolution in GARCH(1,1)');
end

function [integl]=Nker(u,h)
    function fval= vh(t,u,h)
        fval = real(1./(pi) .*exp(-1i.*t.*u) .* (1-t.^2).^3 .*sqrt(pi)...
            ./2.^((i.*t./h)./cgamma(.5+i.*t./h)));
    end
    vhh=@(t) vh(t,u,h);
    integl=quad(vhh,0,1);
end

A.5 Wavelet Deconvolution

function []=meas_w_err_wave()
%simulating the process as a sum of normal distribution with
double exponential error
randn('seed',1)
points=256;
r = -10 + (10+10).*rand(points,1);
xt=normrnd(0,1,points,1);  %standard normal dist
zt=normrnd(0,.1,points,1);
zt=.5 * .1 .*exp(-.1 * abs(r));  % double exponential distribution
yt=xt+zt;

%Set effective support and grid parameters.
lb = -8; ub = 8; n =128;m=-1;

% Compute and plot Meyer wavelet and scaling functions.
[phi,psi,x] = meyer(lb,ub,n);
[phi2d,psi2d,x2d]=meyer2diff(lb,ub,n);

Um =@(t) interp1(x,phi,t,'spline') - 2^(2*m)*(.1)^2* interp1(x,phi2d,t,'spline');
theta=linspace(-3,3,64);
K=(-10:10);
sumf=0;
for i= 1:length(theta)
    sumf=0;
    for j=1:length(K)
        suma=0;
        for l=1:length(yt)
            suma= suma+Um(2^m*yt(l)-K(j))*2^(m/2);
        end
        amk=suma/length(yt);
        phimk=2^((m/2))*interp1(x,phi,2^m*theta(i)-K(j),'spline');
        sumf=sumf+amk*phimk;
    end
    f(i)=sumf;
end
set(gcf,'Color','w');
plot(theta,f,theta,normpdf(theta,0,1),'--');
legend('Estimate','Exact Density');
title('Devonvolution using Meyer wavelets')
end
Bibliography


