User's guide for the R codes implementing "Simulated Likelihood Estimators for Discretely Observed Jump-Diffusions" by Kay Giesecke and Gustavo Schwenkler

François Guay and Gustavo Schwenkler

Version: October 25, 2014

Abstract

1 References

The codes in this zip file implement in R the density and simulated maximum likelihood estimators for one-dimensional jump diffusions developed in the paper "Simulated Likelihood Estimators for Discretely Observed Jump-Diffusions" by Kay Giesecke and Gustavo Schwenkler.

2 How to use the code

2.1 What is inside the zip file

The zip file contains the following R files:

Name	Usage
GS_JD_algorithms.R	Implementation of the algorithms
GS_JD_modelinput.R	Model input to be specified by the user
GS_JD_OC.R	Example file implementing the optimal configuration algorithm
GS_JD_randomelement.R	Random number generation
GS_JD_simulated_likelihood.R	Example file implementing the MLE estimation
GS_JD_UD.R	Example file implementing the computation of the density

The file "GS_JD_algorithms.R" contains all the algorithms and routines necessary to evaluate the density estimator of Giesecke & Schwenkler (2014). This file should not be changed by the user.

The file "GS_JD_modelinput.R" should be filled by the user with the model input. In this file, the user can specify several functions such as the drift, the volatility, the jump intensity, and jump magnitude of X, as well as some other quantities needed to evaluate the density estimator. The original file contains the model input for the numerical example in Section 7 of the paper.

The file "GS_JD_OC.R" is an example file that solves the optimal configuration problem detailed Section 5 of the paper. It computes optimal values for the parameters L, l and ρ that yield a computationally efficient implementation of the density estimator.

The file "GS_JD_randomelement.R" contains the routines that generate the random element R detailed in Section 5 of the paper. The random element R is repeatedly used to evaluate the density estimator at many different parameters and arguments of the density.

The file "GS_JD_simulated_likelihood.R" is an example file that computes simulated maximum likelihood estimators (SMLE) as detailed in Section 6 of the paper. More details will follow in next section.

The file "GS_JD_UD.R" computes the transition density estimator as in Figure 1 of the paper (Section 7.1).

2.2 An example

The file "GS_JD_simulated_likelihood.R" provides an example of how to compute simulated maximum likelihood estimators (SMLE). The original file implements the numerical parameter inference problem of Section 7.3 of the paper.

First, you need to modify the file "GS_JD_modelinput.R" and indicate the model that you wish to estimate. The original file "GS_JD_modelinput.R" contains the model used in Section 7 of the paper. If you wish to estimate a different model, you will need to specify the following functions in the file "GS_JD_modelinput.R":

- $\mu_X(x;\theta)$ and its first order derivative
- $\sigma_X(x;\theta)$ and its first and second order derivatives
- $\Lambda_X(x;\theta)$ and $\Gamma_X(x;\theta)$
- The Lamperti transform $F(w;\theta)$ and its first order derivative
- The inverse of the Lamperti transform $F^{-1}(w;\theta)$ and its first order derivative
- The integral of $\mu_Y(y;\theta)$
- The functions m_i and M_i (step (i) of algorithm 5.2)

Next, you need to input some values in the file "GS_JD_simulated_likelihood.R." You start by entering values for the tuning parameters L, l and ρ. Optimal values for these parameters can be computed using the file "GS_JD_OC.R". Then, enter the name of the data set used in the estimation. In this example, we provide some simulated data in the file "exact_paths_das_600.txt".

You also need to enter an initial parameter guess, and some lower and upper bounds for the parameter space. Finally, you must specify the time period between consecutive observations of the data (e.g. 1/12 for monthly data, 1/52 for weekly data, etc.), the maximum number of iterations for the optimization algorithm (here 5000), and the number of Monte Carlo samples to use when estimating the transition density. In accordance with Theorem 6.2 of the paper, the ratio between the number of observations of the data and the number of Monte Carlo samples has to converge to some finite constant. In order to achieve this condition, Giesecke & Schwenkler (2014) set the number of Monte Carlo samples equal to 10 times the number of observations of the data.

Once executed, the algorithm will return the maximum value of the log-likelihood, a simulated maximum likelihood estimator, and an estimator of the asymptotic standard errors computed in accordance with Theorem 6.2 of the paper.

Note that the algorithm may stop and ask the end-user to modify two quantities "maxk" and "maxp." These quantities can be modified in lines 6 and 7 of the file "GS_JD_algorithms.R." This error message occurs when the random element R is too small and more random variables are needed to estimate the transition density.

2.3 Parallel computing version of the code

We also provide a parallelized version of our code. To use it, you will first need to install the R package "parallel." The use of the files is exactly the same as described above, after specifying the number of parallel cores you want to use.