Order Book Queue Hawkes-Markovian Modeling: code execution instructions

Qianfan Wu, Shihao Yang and Philip Protter

July 21, 2021

1 Step 1: Environment Setup

Set dictionary on Ubuntu terminal: cd /home/User/Desktop/Research/R/LOB_Scripts (or any other default folders containing the scripts); Set stack size limit: set stack size limit: ulimit -s 15922790 (or unlimited);

2 Step 2: Calculate bigdt

Run **"Rscript Odeyssey_timefactor.R arg"** on terminal. Need adjust "save_dir" and "data_dir" folder for local OS. The maximum support and delta value also have to be adjusted. "save_bigdt_only = TRUE" argument means the code will not execute hawkes estimation and plotting.

Odeyssey_timefactor.R will iterate all lobster data files from the "data_dir" folder and save 3 types of files to the "save_dir" for each day in consideration :

1. 24 "bigdt" files for "nosize". (There are 24 types of event; nosize means all orders of a specific type is considered equivalent. order size is not considered);

2. 24 "bigdt" files for "withsize". (There are 24 types of event; withsize means order size are considered to be addictive);

The defualt command line argument *arg* is set 1. It could be any number which represents file number in the data folder.

Execute the following code to iterate the computation through multiple files (the following example means we have 2-day files in the lobster data folder):

for arg in 1 2; do Rscript odyssey.R $arg; done

3 Step 3: Hawkes estimation for a single event for 1-day

Run **"Rscript calc_hawkes_single_multicore.R arg"** on terminal. The command line argument is two-dimensional: arg = [arg1, arg2]. For arg1, arg1 = 1 is for "withsize" estimations and arg1 = 2 for "nosize" estimations. Besides arg2 represents the model type we want to run. arg2 is defined as the following:

- arg2 = 1: state only;
- arg2 = 2: time only;

- arg2 = 3: state and time;
- arg2 = 4: Hawkes only;
- arg2 = 5: state and Hawkes;
- arg2 = 6: time and Hawkes;
- arg2 = 7: Hawkes, state, and time;
- arg2 = 8: Hawkes and LASSO ($\lambda = 0.001$);
- arg2 = 9: Hawkes, state, time, and LASSO($\lambda = 0.001$);
- arg2 = 10: Hawkes and LASSO ($\lambda = 0.0005$);
- arg2 = 11: Hawkes, state, time, and LASSO($\lambda = 0.0005$);

We obtain the model estimation by executing the following loop: for arg1 in 1 2; do for arg2 in 1 2 3 4 5 6 7 8 9 10 11; do Rscript calc_hawkes_single_multicore.R $arg1 $arg2; done;

For each day and each model type, the execution will output the following in consideration:

1. 24 "hawkes_kernel_single" rds files for "nosize" estimation. 2. 24 "hawkes_kernel_single" files for "withsize" rds estimation.

The "cores = " variable in the code inputs the number of cores used in the parellel computing.

4 Step 4: Aggregate all 1-day events

Run **"Rscript agg_hawkes_single_timefactor.R arg"** on terminal. The command line argument is two-dimensional: arg = [arg1, arg2]. For arg1, arg1 = 1 is for "withsize" estimations and arg1 = 2 for "nosize" estimations. Besides arg2 represents the model type we want to run. arg2 is defined as the following:

- arg2 = 1: state only;
- arg2 = 2: time only;
- arg2 = 3: state and time;
- arg2 = 4: Hawkes only;
- arg2 = 5: state and Hawkes;
- arg2 = 6: time and Hawkes;
- arg2 = 7: Hawkes, state, and time;
- arg2 = 8: Hawkes and LASSO ($\lambda = 0.001$);
- arg2 = 9: Hawkes, state, time, and LASSO($\lambda = 0.001$);
- arg2 = 10: Hawkes and LASSO ($\lambda = 0.0005$);
- arg2 = 11: Hawkes, state, time, and LASSO($\lambda = 0.0005$);

We obtain the model estimation by executing the following loop: for arg1 in 1 2; do for arg2 in 1 2 3 4 5 6 7 8 9 10 11; do Rscript agg_hawkes_single_timefactor.R $arg1 $arg2; done;

For each day and each model type, the execution will output the following:

1. 1 "hawkes_markov_pref_price" rds files for "nosize" estimation;
2. 1 "hawkes_markov_pref_price_withsize" rds files for "withsize" estimation;
3. 1 "hawkes_residuals" rds files for "nosize" estimation residuals;
4. 1 "hawkes_residuals_eventsize" rds files for "withsize" estimation residuals;
5. 1 "hawkes_markov_pref_price_kernel" pdf files for "nosize" plotting;
6. 1 "hawkes_markov_pref_price_withsize_kernel" pdf files for "withsize" plotting;

5 Step 5: Aggregate across many days

Run **"Rscript summarize_ody.R"** on terminal. The choice of model type among 1-11 demonstrated in Step 4 and Step 5 is needed. For each model type, the execution will output:

1. "hawkes_markov_pref_price_kernel" pdf file for "nosize" hawkes estimation plotting across all days.

2. "hawkes_markov_pref_price_kernel_individual" pdf file for "withsize" hawkes estimation plotting across all days. There's no layout and plots are formatted vertically one-by-one.

3. "hawkes_markov_pref_price_eventsize_kernel" pdf file for "withsize" hawkes estimation plotting across all days.

4. "hawkes_markov_pref_price_eventsize_kernel_individual" pdf file for "withsize" hawkes estimation plotting across all days. There's no layout and plots are formatted vertically one-by-one.

6 Step 6: AIC calculation

Run **"calc_aic_realdata_timefactor.R"** on R terminal. The value of maximize support and binsize delta need to be customized. The codes will output daily calculated AIC values and AIC comparisons across different model types.