A Nonparametric Test for Instantaneous Causality with Time-varying Variances

Jilin Wu[a], Ruike Wu*[a], and Zhijie Xiao[b]

[a]WISE & School of Economics, Xiamen University
[b]Department of Economics, Boston College, Chestnut Hill, MA, USA.

June 26, 2023

Abstract

This manual introduces the main functions of paper "A Nonparametric Test for Instantaneous Causality with Time-varying Variances" and provide a simple example to show the usages of these functions.

Keywords: Instantaneous Causality; Time-varying Variances; U-statistic; Wild Bootstrap.

Contents

*Corresponding author at: Department of Finance, School of Economics, Xiamen University, Xiamen, China. Email: wrkworld@163.com.

1 Main Functions

1.1 $Test\_inst\_causal\_by\_nonpara.m$

Description: This function computes the proposed test statistic J_T in Eq.(11) of section 3.1 in the main paper.

Usage:

$$[p\_val, J\_T] = Test\_inst\_causal\_by\_nonpara(Y, p, d_1, h)$$

Input:

(1). Y is $d \times T$ input data, d is the dimension, T is sample size.

(2). p is lag length of vector. autoregressive model.

(3). d_1 is the dimension of first group u_{1t}.

(4). $h0$ is the bandwidth coefficient, the bandwidth applied is $h0 \times T^{-1/5}$.

Output:

(1). p_val outputs the p-value of test statistic using standard normal distribution.

(2). J_T outputs the value of the test statistic.

1.2 $Test\_inst\_causal\_by\_npbs.m$

Description: This function executes the proposed tests with wild bootstrap algorithm developed in section 3.3.

Usage:

$$[pval, J\_T, ifa] = Test\_inst\_causal\_by\_npbs(Y, p, B, d_1, cf, h0)$$

Input:

(1). Y is $d \times T$ input data, d is the dimension, T is sample size.

(2). p is lag length of vector autoregressive model.

(3). B is the replication of bootstrap algorithm.

(4). d_1 is the dimension of first group u_{1t}.

(5). cf is confidence level.

(6). $h0$ is the bandwidth coefficient, the bandwidth applied is $h0 \times T^{-1/5}$.

Output:

(1). p_val outputs the bootstrapped p-value of test statistic.

(2). J_T outputs the value of proposed test statistic.

(3). ifa is a logistic variable, it takes 1 if the test successfully rejects the null hypothesis, and takes 0 if not.

1.3 $Test\_inst\_causal\_by\_npbs\_cv.m$

Description: This function executes the proposed tests for two-dimensional data with wild bootstrap algorithm and cross-validation based bandwidth developed in section 3.3.
Usage:

$$[pval, J\_T, ifa] = Test\_inst\_causal\_by\_npbs\_cv(Y, p, B, d_1, cf)$$

Input:

(1). Y is $2 \times T$ input data, T is sample size.

(2). p is lag length of vector autoregressive model.

(3). B is the replication of bootstrap algorithm.

(4). d_1 is the dimension of first group u_{1t}.

(5). cf is confidence level.

Output:

(1). p_val outputs the bootstrapped p-value of test statistic.

(2). J_T outputs the value of proposed test statistic.

(3). ifa is a logistic variable, it takes 1 if the test successfully rejects the null hypothesis, and takes 0 if not.

1.4 $cv\_for\_band\_bt.m$

Description: The cross-validation procedure developed in section 3.3 for selecting bandwidth.
Usage:

$$h\_opt = cv\_for\_band\_bt(cov\_x\_set, T, p)$$

Input:

(1). cov_x_set is the series of $u_{1t}u_{2t}$ (or its estimator).

(2). T is sample size.

(3). p is the lag length.

Output:

(1). h_opt outputs the optimal bandwidth selected by cross-validation.

2 Example

In this section, we first compute the empirical sizes and powers of J_T^B with bandwidth $0.75T^{-1/5}$ and $J_T^{cv,B}$ with cross-validation bandwidth, both of tests are based on bootstrap algorithm developed in section 3.3 of the main paper. Then we plot the monotonic power curves of both J_T^B and $J_T^{cv,B}$

tests with respect to the deviation parameter c. We generate the virtual data based on following data generating process employed in simulation section of main paper with sample size $T = 200$. Specifically,

$$\begin{pmatrix} Y_{1t} \\ Y_{2t} \end{pmatrix} = \begin{pmatrix} 0.64 & -1 \\ -0.01 & 0.44 \end{pmatrix} \begin{pmatrix} Y_{1t-1} \\ Y_{2t-1} \end{pmatrix} + \begin{pmatrix} u_{1t} \\ u_{2t} \end{pmatrix},$$

and

$$\Sigma(r) = \begin{pmatrix} \Sigma^{11}(r) & \Sigma^{12}(r) \\ \Sigma^{12}(r) & \Sigma^{22}(r) \end{pmatrix}, \forall r \in [0,1],$$

where $\Sigma^{11}(r) = 1.1 - \cos(11r)$, $\Sigma^{12}(r) = c\sin(2\pi r)$ and $\Sigma^{22}(r) = 1.1 + \sin(11r)$. When $c = 0$, data is generated under the null hypothesis of no instantaneous causality. When $c \neq 0$, the data is generated under the alternative, the larger c means the larger deviation from the null. The replication for bootstrap is set to 299, and total replication is 1000. The complete codes can be found in "Example1.m".

We firstly generate the virtual data under the null.

```
clear; clc;
T = 200; % set sample size
cf = 0.95; % set significance level
B = 299; % set bootstrap replications
LOOP = 1000; % set total replications

% generate data under the null
c = 0;
rng('default');
rng(2023);% set the random seed
Y_set = cell(LOOP, 1);
for ct  =  1 : LOOP
    x_0 = [0; 0];
    Y = [];
    for r =  1 : T
        sigma = [1.1 - cos(11 * r/T), c * sin(2 * pi * r/T); c * sin(2 * pi * r/T), 1.1 + sin(11 * r/T)];
        u_1 = (sigma^0.5) * randn([2, 1]);
        x_1 = [0.64, -1; -0.01, 0.44] * x_0 + u_1;
        Y(:, r) = x_1;
        x_0 = x_1;
    end
% Y is 2 × T matrix
    Y = Y(1 : 2, 1 : T);
    Y_set{ct} = Y;
end
```

For data under alternative with $c = 0.5$, user only needs to change the eight line "$c = 0$;" to "$c = 0.5$;". With 1000 data sets $\{Y\}_{t=1}^{T}$ (under the null or alternative) in hand, we continue to conducting the J_T^B and $J_T^{cv,B}$ to make statistical inference. Taking the situation of computing empirical sizes as example:

```
size_count = [];
size_count_cv = [];
for ct = 1 : LOOP
    tic; % set timer
    Y = Y_set{ct};
    pval = Test_inst_causal_by_npbs(Y, 1, B, 1, cf, 0.75);
    pval_cv = Test_inst_causal_by_npbs_cv(Y, 1, B, 1, cf);
    size_count(ct) = pval;
    size_count_cv(ct) = pval_cv;
    toc;
end
% Calculate bootstrapped p-values at 1%, 5% and 10% significance level.

size099 = sum(size_count < 0.01)/LOOP;
size099_cv = sum(size_count_cv < 0.01)/LOOP;
size095 = sum(size_count < 0.05)/LOOP;
size095_cv = sum(size_count_cv < 0.05)/LOOP;
size09 = sum(size_count < 0.1)/LOOP;
size09_cv = sum(size_count_cv < 0.01/LOOP;
Final_size = [size099, size095, size09; size099_cv, size095_cv, size09_cv];
```

Simple input "*Final_size*" into command window and we can see the final result as below.

```
>> Final_size
Final_size =
        0.0140 0.0550 0.0930
        0.0120 0.0550 0.0930
```

By conducting the same procedure for 1000 data sets generated under alternative, we can obtain the corresponding empirical powers. Suppose all powers are saved in variable *Final_power*, then input it into command window, it has

```
>> Final_power
Final_power =
        0.7030 0.8900 0.9370
        0.7180 0.8920 0.9340
```

To plot the power curves with respect to ascending c, we only need to repeat above procedure for computing empirical power with c from 0 to 1 (interval is 0.1). Let the powers of J_T^B and $J_T^{cv,B}$ be saved in *power_local* and *power_local_cv*, then we can plot the curves by following codes:

```
plot(c_set, power_local, 'm-.', 'MarkerSize', 4, 'LineWidth', 1);
hold on;
plot(c_set, power_local_cv, 'b-.x', 'MarkerSize', 4, 'LineWidth', 1);
legend('0.75', 'CV');
xlabel('The deviation c from the null');
ylabel('Power');
```

The power curves of two tests are shown in following Figure 1.

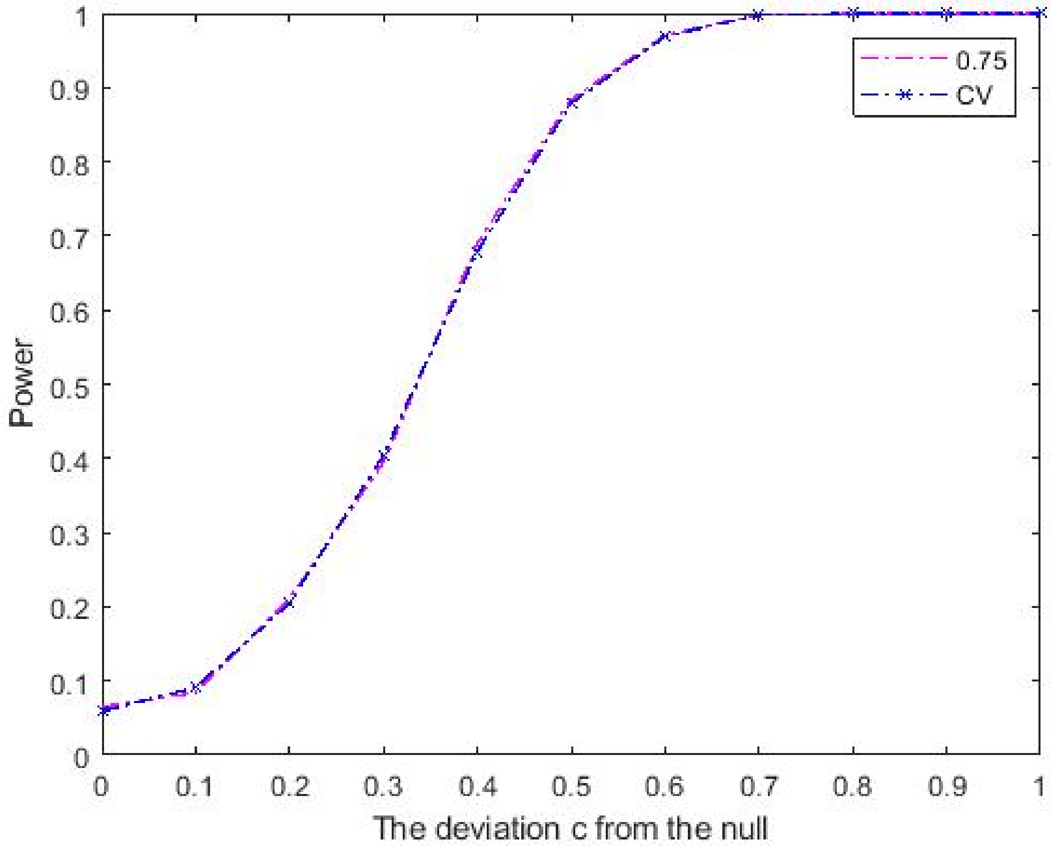

Figure 1: Empirical powers of J_T^B and $J_T^{cv,B}$.