Code Documentation for Auerbach, Guo, Tabord-Meehan (2025) "The Local Approach to Causal Inference under Network Interference"

This file provides instructions for replicating Tables 1-13, Figures 6-12, and p-values in Section 5 of the paper. Figures 1-5 and Tables 14-16 in the paper are just for graphical display and they are not produced by codes in the replication package.

1 Description of Content

The replication package includes:

1. A README file
2. Folder "Empirical-Application":
   - Folders (Adjacency Matrices, Adjacency Matrix Keys) that contain the data for empirical application
   - PY file "empirical_application.py" that contains python codes for replication of Tables 1, 2, 12 and 13, Figures 8-12, and p-values in Section 5 of the paper
   - wrapper file "empirical_application.sh" with bash codes
   - EPS files that are Figures 8-12 in the paper
   - TXT file ("p-values-empirical-application.txt") that contains p-values in Section 5 in the paper
   - TXT files that contain Latex codes for Tables 1, 2, 12 and 13 in the paper
3. Folder "Figures6-7":
   - PY file "figures6-7.py" that contains python codes for replication of Figures 6-7 in the paper
   - wrapper file "figures6-7.sh" with bash codes
   - EPS files that are Figures 6-7 in the paper
4. Folder "my-virtenv-py38" that contains codes to create virtual environment "my-virtenv-py38" on a bash shell

5. Folder "my-virtenv-py310" that contains codes to create virtual environment "my-virtenv-py10" on a bash shell

6. Folder "Table3":

   - Folder "Quest_codes" that contains codes generating Table 3 on a slurm-managed bash shell:
     - SH files that contain bash codes generating different subsections of the results in Table 3. For example, executing "submit_null_ER_g12.sh" will produce rejection rates for testing $H_0 : g_1 =_d g_2$ with DGP Erdös-Renyi(0.1) in Table 3
     - PY files that contain python codes for the replication, which are inputs of SH files
     - TXT files that contain input arguments to run PY files, which are also inputs of SH files
   - Folder "results" that contains outputs of codes in "Quest_codes"
   - Folder "single_computer_codes" that contains codes generating Table 3 on a non-slurm-managed bash shell:
     - SH files that contain bash codes generating Table 3 ("table3_single_computer_full.sh") and a subsection of Table 3 ("table3_single_computer_partial.sh")
     - PY files that contain python codes for the replication and table generation, which are inputs of SH files
     - TXT files that contain input arguments to run PY files, which are also inputs of SH files
     - TXT file ("Table3.txt") which contains Latex codes for a subsection of Table 3
     - TXT file ("partial_replication_results.txt") which contains a subset of results in Table 3 for a partial replication
   - IPYNB file "Table3.ipynb" that contains the codes to generate Latex codes for Table 3 in the paper
   - TXT file "Table3.txt" that contains Latex codes for Table 3 in the paper

7. Folder "Table11":

   - SH file that contains bash codes generating Table 11
   - PY file "simulation_mse.py" that contains python codes for the replication and table generation, which is input of the SH file
   - TXT file "Table11.txt" that contains Latex codes for Table 11 in the paper

8. Folder "Tables4-5":

   - Folder "Quest_codes" that contains codes generating Tables 4-5 on a slurm-managed bash shell:
     - SH files that contain codes generating different subsections of Tables 4 and 5. For example, executing "submit_test_alt_ER_0.1_1.sh" will produce rejection rates for testing $H_0 : g_3 =_d g_4$ with DGP Erdös-Renyi(0.1), $U_{ic} \sim U[-1, 1]$ in Table 4
     - PY file that contains python codes for the replication, which are inputs of SH files

    - TXT files that contain input arguments to run PY files, which are inputs of SH files
  - Folder "results" that contains outputs of codes in "Quest_codes"
  - Folder "single_computer_codes" that contains codes generating Tables 4-5 on a non-slurm-managed bash shell:
    - SH files that contain bash codes generating Tables 4-5 ("tables4-5_single_computer_full.sh") and a subsection of Tables 4-5 ("tables4-5_single_computer_partial.sh")
    - PY files that contain python codes for the replication and table generation, which are inputs of SH files
    - TXT files that contain input arguments to run PY files, which are also inputs of SH files
    - TXT files ("Table4.txt" and "Table5.txt") which contain Latex codes for a subsection of Tables 4-5
    - TXT file ("partial_replication_results.txt") which contains a subset of results in Tables 4 and 5 for a partial replication
  - IPYNB file "Tables4-5.ipynb" that contains the codes to generate Latex codes for Tables 4-5 in the paper
  - TXT files that contains Latex codes for Tables 4-5 in the paper

9. Folder "Tables6-10":
  - Folder "Quest_codes" that contains codes generating Tables 6-10 on a slurm-managed bash shell:
    - SH files that contain codes generating the results in Tables 6-10. More specifically, "submit_test_node_level_treatment.sh" contains the codes generating Tables 6-10 at once, and the rest of the .sh files contain the codes generating a subsection of Tables 6-10
    - PY file that contains python codes for the replication, which are inputs of SH files
    - TXT files that contain input arguments to run PY files, which are inputs of SH files
  - Folders "results-full" and "results-partial" that contain outputs of codes in "Quest_codes" for full and partial replication of Tables 6-10
  - Folder "single_computer_codes" that contains codes generating Tables 6-10 on a non-slurm-managed bash shell:
    - SH files that contain bash codes generating Tables 4-5 ("tables6-10_single_computer_full.sh") and a subsection of Tables 4-5 ("tables6-10_single_computer_partial.sh")
    - PY files that contain python codes for the replication and table generation, which are inputs of SH files
    - TXT files that contain input arguments to run PY files, which are also inputs of SH files
    - TXT files ("Table6.txt", "Table7.txt", "Table8.txt", "Table9.txt", and "Table10.txt") which contain Latex codes for a subsection of Tables 6-10
    - TXT file ("partial_replication_results.txt") which contains a subset of results in Tables 6-10 for a partial replication

- IPYNB files "Tables6-10-full.ipynb" and "Tables6-10-partial.ipynb" that contain the codes to generate Latex codes for Tables 6-10 in the paper and a subsection of Tables 6-10 in the paper
- TXT files that contain Latex codes for Tables 6-10 in the paper

2 Instructions

Codes replicating Figures 8-12, Tables 1, 2, 12, and 13, and all p-values in the folder "Empirical-Application" were run on a MacBook Air (Apple M4 Chip). Codes fully replicating Tables 3-10 were run on Northwestern's High-Performance Computing cluster (Quest), and codes partially replicating Tables 3-10 were run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. The rest of the results were all run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. **Note that results can differ by machine, as "np.argsort()" function may exhibit platform-dependent behavior**.

Step 1 (Creating/Checking Virtual Environment) Anaconda/Miniconda is required to create a virtual environment with Conda version 25.5.1. Replication of Tables 3-10 requires virtual environment "my-virtenv-py38", and replication of the rest of the results require virtual environment "my-virtenv-py310". Here are instructions creating the desired virtual environments on computer cluster managed by slurm and single computer not managed by slurm:

- **Creating "my-virtenv-py38"**

  Change the working directory to "Replication_Package/my-virtenv-py38", then execute the following codes on a machine with operating system that supports Bash shell:

  ```
  $ bash create-my-virtenv-py38.sh
  ```

- **Creating "my-virtenv-py10"**

  Change the working directory to "Replication_Package/my-virtenv-py310", then execute the following codes on a machine with operating system that supports Bash shell:

  ```
  $ bash create-my-virtenv-py310.sh
  ```

Note that this step will create a virtual environment and install all the required Python packages. If there is already a virtual environment with the same name on the machine, the existing virtual environment with this name will be removed. Note that this step only needs to be executed once before executing codes for replicaiton of results.

Step 2 Here are instructions for replicating different tables and figures after creating the required virtual environments:

- **Table 3**

We ran the codes on Northwestern's High-Performance Computing cluster (Quest) managed by slurm. For replicators which do not have access to computing cluster managed by slurm, we also provide codes for full and partial replication on a machine with operating system that supports Bash shell.

- **On a single computer not managed by slurm:**
  Note that codes that can partially replicate Table 3 were run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. Results can differ by machine, as "np.argsort()" function may exhibit platform-dependent behavior.
  Since full replication of Table 3 on a single computer not managed by slurm would take an unreasonably long time and parallel computing may affect random seed control and compromise reproducibility, we recommend partial replication and here is the instruction:
  1. Change the working directory to "Table3/single_computer_codes"
  2. We provided a partial replication of Table 3 with the following results:
     * Rejection rates for the null $H_0 : g_1 =_d g_2$ (Erdös-Renyi(0.1)) when $C = 20$ and $q = 4, 7$
     * Rejection rates for the null $H_0 : g_3 =_d g_8$ (Watts-Strogatz(2,0.2)) when $C = 20$ and $q = 5, 8$
     * Rejection rate for the null $H_0 : g_6 =_d g_7$ (Watts-Strogatz(2,0.8)) when $C = 20$ and $q = 6$
     * Rejection rates for the null $H_0 : g_1 =_d g_2$ (Watts-Strogatz(2,0.275)) when $C = 20$ and $q = 7, 9$
     * Rejection rates for the null $H_0 : g_4 =_d g_5$ (Random Geometric(0.2)) when $C = 20$ and $q = 8$
     * Rejection rates for the null $H_0 : g_1 =_d g_2$ (Random Geometric(0.2)) when $C = 20$ and $q = 6, 9$

     If the replicator wants to replicate the results above, then just move to the next step. If the replicator wants to replicate rejection rates not included above, here is the instruction:
     Edit the file "input_arguments_partial.txt" so that the first element of each row is the value of C, second element is the value of q, and the third element is the null hypothesis. Here is a list of choices of the null hypothesis:
     * ERg12: $H_0 : g_1 =_d g_2$ (Erdös-Renyi(0.1))
     * WS02: $H_0 : g_3 =_d g_8$ (Watts-Strogatz(2,0.2))
     * WS08: $H_0 : g_6 =_d g_7$ (Watts-Strogatz(2,0.8))
     * WS0275: $H_0 : g_1 =_d g_2$ (Watts-Strogatz(2,0.275))
     * RG45: $H_0 : g_4 =_d g_5$ (Random Geometric(0.2))
     * RG12: $H_0 : g_1 =_d g_2$ (Random Geometric(0.2))

For example, if the following lines are included in the "input_arguments_partial.txt" file:

$$20 \quad 4 \quad RGg12$$
$$20 \quad 5 \quad RGg12$$

then the rejection rates of testing $H_0 : g_1 =_d g_2$ (Random Geometric(0.2)) for $C = 20$ and $q = 4, 5$ will be generated.

3. Execute

```
$ bash table3_single_computer_partial.sh
```

After the steps above, here are the outputs:

1. File "partial_replication_results.txt" which contains the rejection rates for each of the prespecified null hypotheses and (C, q) values.
2. File "Table3.txt" which contains the Latex codes of Table 3 with the rejection rates generated from the steps above and other positions being empty.

Note that to reproduce one rejection rate with $C = 20$, the computation time is approximately 2-3 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5. Runtime may differ for different machines.

Although we do not recommend full replication due to the runtime, we provide codes for it. To execute, run

```
$ bash table3_single_computer_full.sh
```

We have not run this script, so we do not have an estimate of its runtime.

– **On a computing cluster managed by slurm:**

To replicate Table 3 in the paper, first copy the folder "Table3/Quest_codes" to a computing cluster managed by slurm, and then run all the files with suffix .sh in the folder "Tables4-5/Quest_codes" by executing

```
$ sbatch submit_null_ER_g12.sh
$ sbatch submit_null_RG_g12.sh
$ sbatch submit_null_RG_g45.sh
$ sbatch submit_null_WS_0.2.sh
$ sbatch submit_null_WS_0.8.sh
$ sbatch submit_null_WS_0.275.sh
```

where each line above will produce a subsection of Table 3. For example, excecuting "sbatch submit_null_ER_g12.sh" will produce rejection rates for testing $H_0 : g_1 =_d g_2$ with DGP Erdös-Renyi(0.1). Note that the .sh files provided in the replication package were originally written for Northwestern's High-Performance Computing cluster (Quest), and to run those .sh files on other computing clusters, some parameters in the .sh files such as account name and directory should be changed accordingly. Save the ".txt" outputs under the folder "Table3/results", and then take the results in the folder "Table3/results" as given, execute

"Table3/Table3.ipynb" to generate "Table3.txt", which includes the Latex codes for Table 3 in the paper.

Since the runtime is long, for a partial replication, replicators can choose to run a subset of the .sh files on computing cluster. Each .sh file produces a proportion of results in Table 3. For instance, executing "submit_null_ER_g12.sh" will produce all the rejection probabilities of testing $H_0 : g_1 =_d g_2$ (Erdös-Renyi(0.1)) in Table 3. Follow the same steps afterwards as the full replication, then the Latex codes for the specified proportion of Table 3 will be generated.

- **Tables 4-5**
  - **On a single computer not managed by slurm:**
    Note that codes that can partially replicate Tables 4-5 were run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. Results can differ by machine, as "np.argsort()" function may exhibit platform-dependent behavior.

    Since full replication of Tables 4-5 on a single computer not managed by slurm would take an unreasonably long time and parallel computing may affect random seed control and compromise reproducibility, we recommend partial replication and here is the instruction:

    1. Change the working directory to "Tables4-5/single_computer_codes"
    2. We provided a partial replication of Tables 4-5 with the following results:
       * Rejection rates for the null $H_0 : g_3 =_d g_4$ (Erdös-Renyi(0.1), $U_{ic} \sim U \in [-1, 1]$) when $C = 20$ and $q = 4, 5, 6, 7, 8$
       * Rejection rates for the null $H_0 : g_3 =_d g_4$ (Random Geometric(0.1), $U_{ic} \sim U \in [-1, 1]$) when $C = 20$ and $q = 5, 6, 7, 8, 9$

       If the replicator wants to replicate the results above, then just move to the next step. If the replicator wants to replicate rejection rates not included above, here is the instruction:

       Edit the file "input_arguments_partial.txt" so that the first element of each line is the value of C, the second element is the value of q, the third element is the upper bound of U_{ic}), the fourth element of each row is the name of DGP, the fifth element of each row is the parameter of DGP, and the sixth element is the number of Monte Carlo simulations. For example, if the following line is included in the "input_arguments_partial.txt" file:

$$\begin{array}{cccccc} 20 & 4 & 1 & ER & 0.1 & 2000 \\ 20 & 7 & 2 & RG & 0.1 & 2000 \end{array}$$

       then the rejection rate of testing $H_0 : g_3 =_d g_4$ (Erdös-Renyi(0.1), $U_{ic} \sim U \in [-1, 1]$) with $(C, q) = (20, 4)$ and the rejection rate of testing $H_0 : g_3 =_d g_4$ (Random Geometric(0.1), $U_{ic} \sim U \in [-2, 2]$) with $(C, q) = (20, 7)$ will be generated.
    3. Execute

```
$ bash tables4-5_single_computer_partial.sh
```

After the steps above, here are the outputs:

1. File "partial_replication_results.txt" which contains the rejection rates for each of the prespecified null hypotheses and (C, q) values.
2. Files "Table4.txt" and "Table5.txt" which contain the Latex codes of Tables 4 and 5 with the rejection rates generated from the steps above and other positions being empty.

Note that to reproduce one rejection rate with $C = 20$, the computation time is approximately 2-3 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5. Runtime may differ for different machines.

Although we do not recommend full replication due to the runtime, we provide codes for it. To execute, run

```
$ bash tables4-5_single_computer_full.sh
```

We have not run this script, so we do not have an estimate of its runtime.

- **On a computing cluster managed by slurm:**

  To replicate Tables 4 and 5 in the paper, first copy the folder "Tables4-5/Quest_codes" to a computing cluster managed by slurm, and then run all the files with suffix .sh in the folder "Tables4-5/Quest_codes" by executing

  ```
  $ sbatch submit_test_alt_DGP_para_err.sh
  ```

  with DGP = ER, RG, para = 0.1, and err = 1,2,3,4,5. Each line above will produce a subsection of Tables 4 and 5. For example, "sbatch submit_test_alt_ER_0.1_1.sh" will produce rejection rates for testing $H_0 : g_3 =_d g_4$ with DGP Erdös-Renyi(0.1), $U_{ic} \sim U[-1, 1]$ in Table 4. Note that the .sh files provided in the replication package were originally written for Northwestern's High-Performance Computing cluster (Quest), and to run those .sh files on other computing clusters, some parameters in the .sh files such as account name and directory should be changed accordingly. Save the ".txt" outputs under the folder "Tables4-5/results", and then take the results in the folder "Tables4-5/results" as given, execute "Tables4-5/Tables4-5.ipynb" to generate "Table4.txt" and "Table5.txt", which include the Latex codes for Tables 4 and 5 in the paper.

  Since the runtime is long, for a partial replication, replicators can choose to run a subset of the .sh files on a computing cluster. Each .sh file produces a proportion of results in Tables 4 and 5. For instance, executing "submit_test_alt_ER_0.1_5.sh" will produce all the rejection probabilities of testing $H_0 : g_3 =_d g_4$ (Erdös-Renyi(0.1), $U_{ic} \sim U \in [-5, 5]$) in Table 4, and executing "submit_test_alt_RG_0.1_5.sh" will produce all the rejection probabilities of testing $H_0 : g_3 =_d g_4$ (Random Geometric(0.1), $U_{ic} \sim U \in [-5, 5]$) in Table 5. Follow the same steps afterwards as the full replication, then the Latex codes for the specified proportion of Tables 4 and 5 will be produced.

- **Tables 6-10**

– **On a single computer not managed by slurm:**
Note that codes that can partially replicate Tables 6-10 were run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. Results can differ by machine, as "np.argsort()" function may exhibit platform-dependent behavior.
Since full replication of Tables 6-10 on a single computer not managed by slurm would take an unreasonably long time and parallel computing may affect random seed control and compromise reproducibility, we recommend partial replication and here is the instruction:

1. Change the working directory to "Tables6-10/single_computer_codes"
2. We provided a partial replication of Tables 4-5 with the following results:
   * Rejection rate for the null $H_0 : \tilde{g}_1 =_d \tilde{g}_5$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,3,2,1)$) when $C = 20$ and $q = 5$
   * Rejection rate for the null $H_0 : \tilde{g}_3 =_d \tilde{g}_7$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,3,2,1)$) when $C = 20$ and $q = 7$
   * Rejection rate for the null $H_0 : \tilde{g}_5 =_d \tilde{g}_8$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,3,2,1)$) when $C = 20$ and $q = 6$
   * Rejection rate for the null $H_0 : \tilde{g}_1 =_d \tilde{g}_4$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,3,2,1)$) when $C = 20$ and $q = 8$
   * Rejection rate for the null $H_0 : \tilde{g}_1 =_d \tilde{g}_3$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,4,3,2)$) when $C = 20$ and $q = 6$
   * Rejection rate for the null $H_0 : \tilde{g}_5 =_d \tilde{g}_7$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,4,3,2)$) when $C = 20$ and $q = 8$
   * Rejection rate for the null $H_0 : \tilde{g}_1 =_d \tilde{g}_2$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,5,4,3)$) when $C = 20$ and $q = 9$
   * Rejection rate for the null $H_0 : \tilde{g}_5 =_d \tilde{g}_6$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,5,4,3)$) when $C = 20$ and $q = 5$
   * Rejection rate for the null $H_0 : \tilde{g}_1 =_d \tilde{g}_2$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,6,5,4)$) when $C = 20$ and $q = 6$
   * Rejection rate for the null $H_0 : \tilde{g}_1 =_d \tilde{g}_4$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4,4]$, $\theta = (0,2,0,6,5,4)$) when $C = 20$ and $q = 6$

   If the replicator wants to replicate the results above, then just move to the next step. If the replicator wants to replicate rejection rates not included above, here is the instruction:
   Edit the file "input_arguments_partial.txt" so that the first element of each row is the value of C, the second element is the value of q, the third element of each row is the fourth element of parameter θ, the fourth element of each row is the fifth element of parameter θ, the fifth element of each row is the sixth element of parameter θ, the sixth element of each row is the index of the first graph in the null (for example, 1 represents $\tilde{g}_1$), the seventh element of each row is the index of the second graph in the null (for example,

2 represents $\tilde{g}_2$), and the eighth element is the number of Monte Carlo simulations. For example, if the following line is included in the "input_arguments_partial.txt" file:

```
20 4 3 2 1 1 2 2000
20 5 4 3 2 5 6 2000
```

then the rejection rate of testing $H_0 : \tilde{g}_1 =_d \tilde{g}_2$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4, 4]$, $\theta = (0, 2, 0, 3, 2, 1)$) with $(C, q) = (20, 4)$ and the rejection rate of testing $H_0 : \tilde{g}_5 =_d \tilde{g}_6$ (Watts-Strogatz(2,0.3), $U_{ic} \sim$ Unif$[-4, 4]$, $\theta = (0, 2, 0, 4, 3, 2)$) with $(C, q) = (20, 5)$ will be generated.

3. Execute

```
$ bash tables6-10_single_computer_partial.sh
```

After the steps above, here are the outputs:

1. File "partial_replication_results.txt" which contains the rejection rates for each of the prespecified null hypotheses and (C, q) values.
2. Files "Table6.txt", "Table7.txt", "Table8.txt", "Table9.txt", and "Table10.txt", which contain the Latex codes of Tables 6-10 with the rejection rates generated from the steps above and other positions being empty.

Note that to reproduce one rejection rate with $C = 20$, the computation time is approximately 2-3 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5. Runtime may differ for different machines.

Although we do not recommend full replication due to the runtime, we provide codes for it. To execute, run

```
$ bash tables6-10_single_computer_full.sh
```

We have not run this script, so we do not have an estimate of its runtime.

– **On a computing cluster managed by slurm:**

To replicate Tables 6-10 in the paper, first copy the folder "Tables6-10/Quest_codes" to a computing cluster managed by slurm, and then run all the files with suffix .sh in the folder "Tables6-10/Quest_codes" by executing

```
$ sbatch submit_test_node_level_treatment.sh
```

Note that the .sh files provided in the replication package were originally written for Northwestern's High-Performance Computing cluster (Quest), and to run those .sh files on other computing clusters, some parameters in the .sh files such as account name and directory should be changed accordingly. Save the outputs under the folder "Tables6-10/results", and then take the results in the folder "Tables6-10/results" as given, execute "Tables6-10/Tables6-10.ipynb" to generate "Table6.txt", "Table7.txt", "Table8.txt", "Table9.txt", and "Table10.txt", which include the Latex codes for Tables 6-10 in the paper.

Since the runtime is long and the number of cores required is large, for a partial replication, replicators can choose to run a subset of the .sh files on computing cluster. Here we provide instructions for the replication of the first subsection of Table 6 (i.e. all the rejection probabilities for testing $H_0 : \tilde{g}_1 =_d \tilde{g}_5$ (Watts-Strogatz(2,0.3), $U_{ic} \sim U \in [-4, 4]$, $\theta = (0, 2, 0, 3, 2, 1)$)), and the replication of the rest of the subsections follows similarly:

First copy the folder "Tables6-10/Quest_codes" to a computing cluster managed by slurm, and then run the following .sh file:

```
$ sbatch submit_test_node_level_treatment_coeff321_g15.sh
```

Note that the .sh file provided in the replication package were originally written for Northwestern's High-Performance Computing cluster (Quest), and to run the .sh file on other computing clusters, some parameters in the .sh files such as account name and directory should be changed accordingly. Also note that to replicate other subsections of Tables 6-10, the name of the input argument file in line 19 should be changed accordingly, and all the input argument files can be found in the folder "Tables6-10/Quest_codes". Save the outputs under the folder "Tables6-10/results-partial", and then take the results in the folder "Tables6-10/results-partial" as given, execute "Tables6-10/Tables6-10-partial.ipynb" to generate the Latex codes for the specified proportion of Tables 6-10 in the paper.

- **Tables 1, 2, 12, 13, Figures 8-12, and p-values in Section 5 in the paper**

  Note that codes that can replicate these results were run on a Macbook Air (Apple M4 Chip). Results can differ by machine, as "np.argsort()" function may exhibit platform-dependent behavior.

  Change the working directory to "Empirical-Application", and then execute the following codes on a machine with operating system that supports Bash shell:

  ```
  $ bash empirical-application.sh
  ```

  This command will generate EPS and TXT files in this folder, which are Latex codes for Tables 1, 2, 12 and 13, EPS files for Figures 8-12, and p-values ("p-values-empirical-applicaiton.txt") in Section 5 in the paper. It will take approximately 5-6 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5. Runtime may differ for different machines.

- **Figures 6-7**

  Note that codes that can replicate Figures 6-7 were run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. Results can differ by machine, as "np.argsort()" function may exhibit platform-dependent behavior.

  Change the working directory to "Figures6-7", and then execute the following codes on a machine with operating system that supports Bash shell:

  ```
  $ bash figures6-7.sh
  ```

This command will generate EPS files in this folder, which are Figures 6-7 in the paper. It will take approximately 1 hour on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5. Runtime may differ for different machines.

- **Table 11**

  Note that codes that can replicate Table 11 were run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. Results can differ by machine, as "np.argsort()" function may exhibit platform-dependent behavior.

  Change the working directory to "Table11", and then execute the following codes on a machine with operating system that supports Bash shell:

  ```
  $ bash simulation_mse.sh
  ```

  This command will generate "Table11.txt", which contains Latex codes for Table 11 in the paper. It will take approximately 40 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5. Runtime may differ for different machines.

3 Hardware Specifications

Codes replicating Figures 8-12, Tables 1, 2, 12, and 13, and all p-values in empirical applications were run on a MacBook Air (Apple M4 Chip). Codes fully replicating Tables 3-10 were run on Northwestern's High-Performance Computing cluster (Quest), and codes partially replicating Tables 3-10 were run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory. The rest of the results were all run on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5 and 16 GB 3733 MHz LPDDR4X memory.

The runtime for reproducing results in folder "Empirical-Application" is approximately 5-6 hours. The runtime for reproducing results in folder "Figures6-7" is approximately 1 hour. The runtime for reproducing results in folder "Table11" is approximately 40 hours. The runtime of IPYNB files in the other folders are short (less than 1 minute).

Codes in folder "Table3/Quest_codes" were run on Northwestern's High-Performance Computing cluster (Quest). There are 6 SH files to run, each of which takes 48 hours on a computer server with 80 cores and 1 RAM memory. Following the instructions in Section 2, the code can be run on any computing cluster managed by slurm, and we expect similar runtime to Quest. For partial replication of Table 3 on a slurm-managed computer, the replicator can run a subset of all the SH files, and each SH file should take approximately 48 hours with 80 cores and 1 RAM memory. For partial replication of Table 3 on a non-slurm-managed computer, the runtime of the sample codes is about 12 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5, and we do not have an estimate for the full replication.

Codes in folder "Tables4-5/Quest_codes" were run on Northwestern's High-Performance Computing cluster (Quest). There are 10 SH files to run, each of which takes 48 hours on a computer server with 68 cores and 1 RAM memory. Following the instructions in Section 2, the code can be run on any computing cluster managed by slurm, and we expect similar runtime to Quest. For partial replication of Tables 4 and 5, the replicator can run a subset of all the .sh files, and each .sh file will take approximately 48 hours with 68 cores and 1 RAM memory. For partial replication of Tables 4 and 5 on a non-slurm-managed computer, the runtime of the sample codes is about 12 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5, and we do not have an estimate for the full replication.

Codes in folder “Tables6-10/Quest_codes” were run on Northwestern’s High-Performance Computing cluster (Quest), with 2720 cores and 1 RAM memory for each core. The runtime is approximately 48 hours. For partial replication of Tables 6-10 on a slurm-managed computer, we divided the computation into 30 pieces and the replicator can run a subset of all the SH files[1]. Each SH file should take approximately 48 hours on Quest with 68 cores and 1 RAM memory. For partial replication of Tables 6-10 on a non-slurm-managed computer, the runtime of the sample codes is about 20 hours on a MacBook Pro (13-inch, 2020) with a 2 GHz Quad-Core Intel Core i5, and we do not have an estimate for the full replication.

4 Data Source

The data required for empirical application, “Empirical-Application/Adjacency Matrices” and “Empirical-Application/Adjacency Matrix Keys” are the data under the folder “Data/1. Network Data” of the replication package Banerjee et al. (2013b) for the paper Banerjee et al. (2013a).

The data is from a social network survey conducted in 75 villages in rural southern Karnataka, India. In this survey, individuals were asked detailed questions about their relationships with others in the village. The folder “Empirical-Application/Adjacency Matrices” contains adjacency matrices representing various types of social relationships between households within each village, including “borrows money from” and “gives advice to”, among others. These matrices do not include row or column headers, and the folder “Empirical-Application/Adjacency Matrix Keys” provides mapping files that associate each row and column with a specific individual or household.

5 Software and Package Versions

All IPYNB codes were executed using JupyterLab (version 4.2.3) with Python version 3.10.6.

Here is a list of python packages required (with corresponding versions) and python version for codes run on virtual environment “my-virtenv-py38” (replication of Tables 3-10):

python version: 3.8.19

numpy version: 1.24.4

networkx version: 3.1

[1] We provided all the input arguments files needed, but only provided the SH file for replication of the first subsection of Table 6; replicators should change the .sh file accordingly for different subsections.

pandas version: 2.0.3

matplotlib: 3.7.5

scipy version: 1.10.1

Here is a list of python packages required (with corresponding versions) and python version for codes run on virtual environment “my-virtenv-py310” (replication of all results except for Tables 3-10):

python version: 3.10.6

numpy version: 1.26.4

networkx version: 3.3

pandas version: 2.2.3

matplotlib version: 3.5.3

scipy version: 1.11.4

igraph version: 0.11.8

References

Banerjee, A., Chandrasekhar, A. G., Duflo, E. and Jackson, M. O. (2013a). The diffusion of microfinance. *Science*, **341** 1236498.

Banerjee, A., Chandrasekhar, A. G., Duflo, E. and Jackson, M. O. (2013b). The Diffusion of Microfinance. URL https://doi.org/10.7910/DVN/U3BIHX.