Data and Code for "Co-citation and Co-authorship Networks of Statisticians" by Ji, Jin, Ke and Li

This short note contains the instructions to access the data and code for the paper (Ji et al., 2021). If you have questions, please contact Zheng Tracy Ke (zke@fas.harvard.edu) and Jiashun Jin (jiashun@cmu.edu).

The data set was collected and cleaned by Ji et al. (2021). Real author names are used, but all information shown in this data set was collected from public sources.

Contents

1 A list of ready-to-use data matrices **2**

2 Organization of the code files **3**

3 Some instructions on running the code **5**

1. A list of ready-to-use data matrices

All these data matrices are in the folder *Ready-to-use data matrices*.

- The full data are in the file `AuthorPaperInfo.RData`. It has two variables.
  - `AuPapMat`: This matrix summarizes the bibtex data. It has 4 columns, where *idxAu* is author ID, *idxPap* is paper ID, *year* is publication year, and *journal* is publication journal.
  - `PapPapMat`: This matrix summarizes the citation data. It has 5 columns, where *FromPap* and *ToPap* are the paper IDs, *FromYear* and *ToYear* are the publication years of two papers, and *SelfCite* is an indicator whether this is a self citation.

  Additionally, the file `author_name.txt` contains all author names, arranged in the same order as their IDs

- The adjacency matrices of co-citation networks
  - `CiteeAdjFinal.mat`: The adjacency matrix of the citee network (1991-2000). This is the network used to produce the Statistics Triangle and Research Map. It has 2831 authors.
    To get the names of the 2831 authors, use the variable *keepNodeID* in `CiteeAdjFinal.mat`, as well as the file `author_name.txt`.
  - `CiteeDynamicFinal.mat`: The adjacency matrices of the 21 citee networks (1991-2015). These are the networks used to produce the Research Trajectories.

- The adjacency matrices of co-authorship networks
  - `CoauAdjFinal.mat`: The adjacency matrix of the co-authorship network (36 journals). This is the network used to obtain the community tree. It has 4383 authors.
    The names of the 4383 authors are given by the variable *authorNames* in `CoauAdjFinal.mat`.
  - `CoauSankeyFinal.mat`: The adjacency matrices of the 3 co-authorship networks (4 journals). This is the network used to produce the Sankey diagram.
    These sparse matrices are stored on the original 47311 authors. The variable *authorNames* contains the names of all 47311 authors. The variable V contains the indices of 1687 nodes used to draw the Sankey diagram. Restricting each adjacency matrix on V gives the data matrices used in the paper.

2. Organization of the code files

The data and code are arranged into 5 parts, stored in 5 separate folders. Each folder contains:

- A *data* sub-folder: It contains all the data matrices needed to run the code of this part. These data files were produced by either pre-processing of raw data or the other parts of the code.
- An *output* folder: It contains the output generated by the code of this part. The output files include figures and R/Matlab data files.
- Code files: These code files implement the analysis and produce the results in the paper. Please see detailed instructions below.

Each of these 5 folders is self-contained: Files in the other folders are not needed to run the code of this part.

Folder: 1.Statistics Triangle.

It contains code for mixed membership estimation on the citee network. It generates Figure 1 in the paper.

- `CreateAdj.m`: It creates the adjacency matrix of the citee network (1991-2000).
- `Mixed_Membership_Estimation.R`: It computes the Statistics Triangle and Research Map.
- `plotResearchTriangle.R`: It plots Figure 1. It also prints the names of representative authors in each of the 15 clusters in Research Map.
- `Mixed-SCORE.R`: It contains all the functions needed to run `Mixed_Membership_Estimation.R`.

Folder: 2.Trajectory.

It contains code for dynamic analysis of the 21 citee networks. It generates Figure 2 in the paper.

- `CreateAdjSeqs.m`: It creates the adjacency matrices of the 21 citee networks.
- `Get_Trajectories.R`: It computes the Research Trajectories. It also plots Figure 2.

Folder: 3.Coauthorship Communities.

It contains code for hierarchical community detection on the co-authorship network (36 journals,1975-2015). It generates Figure 4 and Table 3 in the paper.

- `community_first_layer.m`: It applies SCORE to obtain the 6 first-layer communities.
- `community_hierarchy.m`: It runs a recursive algorithm to obtain the community tree. It also prints the representative nodes of each leaf community.
- `NetworkPartition.m`: Matlab function.
- `SCORE.m`: Matlab function.
- `SgnQ.m`: Matlab function.

Note: Figure 4 was produced by a different software, based on the output of `community_hierarchy.m`.

Folder: 4.Sankey.

It contains code for dynamic analysis of the 3 co-authorship networks (4 journals, 1975-2015). It generates Figure 5 in the paper.

- `CreateAdj3Periods.m`: It creates the adjacency matrices of the 3 co-authorship networks.
- `dynamic_comm_detect.m`: It runs community detection on the 3 co-authorship networks and prints the list of representative nodes in Figure 5.
- `SCORE.m`: Matlab function.

Note: Figure 5 was produced by a different software, based on the output of `dynamic_comm_detect.m`.

Folder: 5.Diversity.

It contains code for analysis of personalized networks. The `R.Matlab` package can be used to convert data matrices It generates Figures 6-7 and Table 4.

- `person_coauthor.m`: It computes the SgnQ p-values of personalized coauthorship networks of 1000 selected nodes.
- `person_citation.m`: It computes the SgnQ test statistics of personalized citer and citee networks of 1000 selected nodes.
- `q_stat.m`: It draws the two histograms in Figure 6 and the scatter plot in Figure 7.
- `carroll_net`: It draws the example of Raymond Carrol's personalized network in Figure 7.
- `FuncGetNetwork.m`: Matlab function.
- `SCORE.m`: Matlab function.
- `SgnQ.ma`: Matlab function.

3. Some instructions on running the code

The code files are in Matlab or R. The `R.matlab` package can be used to transfer data between R and Matlab. Use the function `readMat` to read *.mat* files into R. Use the function `writeMat` to save data into *.mat* files in R environment.

Some R code requires installing additional packages. You will find the required packages in each separate R code file. The following code installs all required packages automatically.

```
# install and load packages automatically
   package_list =  c("xx", "yy", "zz")
for(package in package_list) {
          if (!require(package))
          install.packages('package')
          library(package) }
```

Each code file may have the data analysis part and the plotting part. The plotting part usually requires more packages. If you cannot run the whole code file, you can delete the plotting part and run the data analysis part only. Please check the annotation of each code file.

Sometimes, the output in your computer may be slightly different from that in the paper. This is usually because the algorithm of eigen-decomposion is different in different environments.

References

Pengsheng Ji, Jiashun Jin, Zheng Tracy Ke, and Wanshan Li. Co-citation and co-authorship networks of statisticians. *Journal of Business & Economic Statistics*, (just-accepted):1–32, 2021.