Replication Data for: "A Nontrivial Interplay between Triadic Closure, Preferential, and Anti-Preferential Attachment: New Insights from Online Data"

The dataset includes code and data files that are necessary for reproducing the results reported in the main manuscript. In the dataset, the snapshots of three different networks are marked by the labels “MainCity”, “City(1)”, and “City(2)” correspondingly. Analysis of the network labeled as “MainCity” is given in the Main Manuscript, whereas the second and third networks were used to check the robustness of our results. Each network was measured twice, in March 2019 and September 2019. The corresponding snapshots are stored in the following files (in the table below, the word “City” should be substituted by “MainCity”, “City(1)”, or “City(2)”).

File name	Description
“AdjMatrix(1) – {City}.npz”	The first snapshot of the social graph stored as a (squared) adjacency matrix (csr-format).
“AdjMatrix(2) – {City}.npz”	The second snapshot of the social graph stored as a (squared) adjacency matrix (csr-format).
“UsersAttributes(1) – {City}.csv”	The first snapshot of users’ opinions (+ demographic attributes).
“UsersAttributes(2) – {City}.csv”	The second snapshot of users’ opinions.
“Followees(1) – {City}.npz”	The adjacency matrix that stores the first snapshot of users’ followees (the number of rows corresponds to the number of users in the sample, the number of columns is equal to the number of information sources in the

	feature space of the opinion estimation model) (csr-format).
“Followees(2) – {City}.npz”	The adjacency matrix that stores the second snapshot of users’ followees (the number of rows corresponds to the number of users in the sample, the number of columns is equal to the number of information sources in the feature space of the opinion estimation model) (csr-format).
“Code (1) - create table of observations.ipynb”	This Python code takes all the snapshots of a given social network and transforms the data into a suitable form that is ready for further analysis (see below).
“Code (2) – exploratory data analysis.ipynb”	This Python code performs exploratory analysis of the data (including correlation analysis and organization of homophily).
“Code (3) – analyzing table of observations (tie appearing).ipynb”	This Python code concentrates on analyzing how the probability of tie appearing depends on nodal and edge attributes.
“Code (4) - analyzing table of observations (tie removing).ipynb”	This python code concentrates on analyzing how the probability of tie removing depends on nodal and edge attributes.
“Code (5) - MLE (tie appearing).ipynb”	Python code for performing MLE analysis (tie appearing).
“Code (6) - MLE (tie removing).ipynb”	Python code for performing MLE analysis (removing of ties).

"Code (7) - Cliques.ipynb"	Python code for clique community analysis.

Preliminary work

After running the file "Code (1) - create table of observations.ipynb" and going through all the cells of the code, the folder "City" is created in the working directory, in which the files "UsersAttributesGiant.csv" and "ObservationsGiant.csv" appear. The former is a csv-table that stores user-level information (variables $age, gender, x, f, f^{new}, f^{del}, s, s^{new}, s^{del}$). The following table shows the correspondence between the variables and the column names.

Variable	Column name
age	age
$gender$	gender
$x(t_1)$	OpinionBefore
$x(t_2)$	OpinionAfter
$f(t_1)$	FriendsBefore
$f(t_2)$	FriendsAfter
f^{new}	NewFriends
f^{del}	DeletedFriends
$s(t_1)$	FolloweesBefore
$s(t_2)$	FolloweesAfter
s^{new}	NewFollowees
s^{del}	DeletedFollowees

There are some additional columns in the table "UsersAttributesGiant.csv", but they are not used in this study: "FriendsOpinionBeforeAvg" (average opinion of the user's friends at time

t_1), "FriendsOpinionAfterAvg" (average opinion of the user's friends at time t_2), and "FriendsOpinionBeforeStd" (the standard deviation of friends' opinions at time t_1).

In the directory "City", one can also find the file "ObservationsGiant.csv." This is a csv-table that contains nodal and edge attributes for each pair of nodes. Because the number of such observations is extremely huge (and depends quadratically on the number of nodes), this information is typically stored in several files that are marked with numerical labels like "ObservationsGiant(1).csv", "ObservationsGiant(2).csv", etc. Each of them includes no more than 50,000,000 observations. However, such a configuration of data is not convenient for analysis. Besides, to calculate the probabilities of tie appearing and removing, one needs anyway stack all these matrices along the vertical axis to get the whole table. On this step, some problems could appear due to memory constraints. In this regard, we decided to use an alternative organization of the table "ObservationsGiant.csv" – we store it as separate columns and masks. If we want to calculate a specific metric, then we load only those vectors or masks in the working memory that will be used in calculations.