JASA ACS Reproducibility Initiative - Author Contributions Checklist Form

January 29, 2019

1 Data

1.1 Abstract

Temporal data of original tweets (originals) and retweets of two hashtags, and Markov chain Monte Carlo (MCMC) output of fitting the models to the data

1.2 Availability

Available in the supplementary files are multiple json files (raw data), csv files (core data) and rds files (MCMC output). The core data are also available from a research data management repository commissioned by the funder (see **Link to data** below).

1.3 Description

Permissions: The json files were collected using tweepy (http://www.tweepy.org), which is a wrapper for the public Application Programming Interface (API) provided by Twitter.

Link to data: https://rdm.ncl.ac.uk/landing/pages/10.17634/154300-57 (For core data csv files only)

DOI: 10.17634/154300-57

File format: json (JavaScript Object Notation), csv (comma-separated values), and rds (for R objects)

Metadata: The list of files and their description are as below:

1. data/20170614.json: the raw data for the hashtag #thehandmaidstale that will be cleaned and converted to the csv files with prefix "20170614". The cleaning and conversion will be explained in Section 3.1.

2. data/20170614_original_a.csv: temporal data for originals with hashtag #thehandmaidstale, unretweeted during data collection period. Its list of variables (data dictionary) is as follows:

    - id_str: unique ID of the original
    - user_followers_count: follower count of the author of the original at its creation
    - retweet_count: retweet count of the original at the end of data collection
    - t_i: creation time of original relative to t_0, in seconds
    - t_ij: creation time of retweet relative to t_0, in seconds; t_ij is always greater than or equal to the corresponding t_i
    - retweeted: whether the original was ever retweeted during data collection
    - log_followers_count: log(1+user_followers_count)
    - log_retweet_count: log(1+retweet_count)

3. data/20170614_original_b.csv: temporal data for retweeted originals with hashtag #thehandmaidstale and their retweets. Its data dictionary is the same as data/20170614_original_a.csv.
4. data/20170614_range.csv: range of data collection period for #thehandmaidstale data. Its data dictionary is as follows:

    - t_0: beginning of data collection
    - t_inf: duration of data collection in seconds; essentially, end of data collection is (t_0 + t_inf)

5. data/20170716.json: the raw data for the hashtag #gots7 that will be cleaned and converted to the csv files with prefix "20170716". The cleaning and conversion will be explained in Section 3.1.
6. data/20170716_original_a.csv: temporal data for originals with hashtag #gots7, unretweeted during data collection period. Its data dictionary is the same as data/20170614_original_a.csv.
7. data/20170716_original_b.csv: temporal data for retweeted originals with hashtag #gots7 and their retweets. Its data dictionary is the same as data/20170614_original_a.csv.
8. data/20170716_range.csv: range of data collection period for #gots7 data. Its data dictionary is the same as data/20170614_range.csv.
9. results/20170614_mcmc_0_single.rds: MCMC output of fitting model 0 (hierarchical model of power law processes) to #thehandmaidstale data
10. results/20170614_mcmc_1_single.rds: MCMC output of fitting model 1 (hierarchical model of hybrid processes) to #thehandmaidstale data

11. results/20170614_rjmcmc_single.rds: MCMC output of model selection for #thehandmaidstale data via reversible jump MCMC (RJMCMC)
12. results/20170614_gvs_cc_single.rds: MCMC output of model selection for #thehandmaidstale data via Gibbs variable selection (GVS)
13. results/20170716_mcmc_0_single.rds: MCMC output of fitting model 0 (hierarchical model of power law processes) to #gots7 data
14. results/20170716_mcmc_1_single.rds: MCMC output of fitting model 0 (hierarchical model of hybrid processes) to #gots7 data
15. results/20170716_rjmcmc_single.rds: MCMC output of model selection for #gots7 data via RJMCMC
16. results/20170716_gvs_cc_single.rds: MCMC output of model selection for #gots7 data via GVS
17. results/sim_0614_mcmc_0_single.rds: MCMC output of fitting model 0 (hierarchical model of power law processes) to simulated data
18. results/sim_0614_mcmc_1_single.rds: MCMC output of fitting model 1 (hierarchical model of hybrid processes) to simulated data
19. results/sim_0614_rjmcmc_single.rds: MCMC output of model selection for simulated data via RJMCMC
20. results/sim_0614_gvs_cc_single.rds: MCMC output of model selection for simulated data via GVS

2 Code

2.1 Abstract

R package and scripts (R and Rnw) for reproducing the figures and numbers in the paper

2.2 Description

How delivered: The essential functions for reproducing the analyses are delivered in the package hybridProcess, while the scripts are provided in the supplementary materials. Once the R package is installed (and loaded), following the instructions in Section 3.1 to run the scripts will yield reproducibility.

Licensing information: Default (MIT License)

Link to code/repository: https://github.com/clement-lee/hybridProcess

Version information: hybridProcess_0.0.1

Metadata: The list of scripts in the supplementary materials is as follows:

1. 20170614_mcmc_0_single.R
2. gots7_prelim.R
3. gots7_results.R
4. initial_values_20170614_single.R
5. initial_values_20170716_single.R
6. initial_values_sim_0614_single.R
7. thehandmaidstale_prelim.R
8. thehandmaidstale_results.R
9. thehandmaidstale_simulate.R
10. reproducibility.Rnw

All R scripts above will be sourced by reproducibility.Rnw when generating the pdf for reproducing the numerical results and figures. The only exception is 20170614_mcmc_0_single.R, which is for illustration in Section 3.1.2.

2.3 Optional Information

Version numbers for R & the R libraries used:
R 3.4.4
dplyr_0.7.8
purrr_0.2.5
readr_1.1.1
tidyr_0.7.2
tibble_1.4.2
ggplot2_3.1.0
lubridate_1.7.1
magrittr_1.5
glue_1.3.0
rjson_0.2.20
lazyeval_0.2.1
anytime_0.3.0
Rcpp_1.0.0

These libraries and their version numbers are dependencies of the R package hybridProcess, and are documented in the file DESCRIPTION. The package can be installed via an R command (provided that the package devtools is installed):

```
# install.packages("devtools")
devtools::install_github("clement-lee/hybridProcess")
### if dependencies are not wanted to be upgraded automatically,
### set the argument upgrade_dependencies = FALSE (default TRUE)
```

MCMC code: All MCMC algorithms mentioned in the paper are coded in C++, and are made usable in R, specifically being wrapped by the function `run_mcmc()`, via the R package Rcpp. The C++ code is available in the sub-directory 'src' of the **source** code of the R package hybridProcess.

3 Instructions for Use

3.1 Reproducibility

There are three levels of reproducibility:

1. from raw data (json files) to core data (csv files),
2. from core data to MCMC output (rds files), and
3. from MCMC output to the figures and numbers in the paper.

While each level can be run on its own and will be explained separately below, in principle the final results i.e. the figures and numbers in the paper can be reproduced from the raw data. The two sets of intermediate results are provided here to save computation time.

3.1.1 From raw data to core data

What is to be reproduced: the csv data files in the subdirectory 'data'

How to reproduce analyses: To reproduce the csv files for #thahandmaidstale data from the json file obtained on 2017-06-14, type

```
library(hybridProcess); extraction("20170614")
```

in an R session. The csv files will be written in the sub-directory 'data' (and potentially overwriting the existing ones). Similarly, to reproduce the csv files for the #gots7 data from the raw json file obtained on 2017-07-16, type

```
library(hybridProcess); extraction("20170716") # takes minutes
```

in an R session. A user may want to move or rename the existing csv files first before the generation of the new ones, in order to check that the two sets of csv files are identical.

3.1.2 From core data to MCMC output

What is to be reproduced: the rds files in the subdirectory 'results'

How to reproduce analyses: We take #thehandmaidstale data and model 0 (defined in the paper) for example here. Running the MCMC and saving the output in an rds file can be reproduced by running the following in R:

```
source("20170614_mcmc_0_single.R")
```

This will run the required MCMC algorithm, and write the output to the file results/20170614_mcmc_0_single.rds, potentially overwriting an existing one that contains identical results. If the user wants to check reproducibility, it is suggested that the existing file is renamed before running the above line. If the user does not want to re-run the MCMC if there already is an output file, they can change the `rebuild` argument in `run_mcmc()` to `FALSE` in the script 20170614_mcmc_0_single.R.

As by-products there will be three csv files generated with prefix "20170614_mcmc_0", in the sub-directory 'results', when running the MCMC. They contain the chains of the estimated and simulated retweet counts, the latter of which are used in computing the 95% predictive intervals of retweet counts, which are in turn plotted in Figure 9 of the paper. However, due to their sheer sizes, they are not included in the supplementary files. As they are not compulsory for reproducibility here, their generation can be disabled, by setting the argument `write` to `FALSE` in the function `run_mcmc()`.

In reproducibility.Rnw there is a line

```
run_mcmc(
    date_str.thmt, 1234L, mh_etas, N.thmt, t.thmt, b.thmt, f.thmt,
    a0.thmt, b0.thmt, 1.0, p01.thmt, p10.thmt, write = TRUE,
    l0.thmt, beta.thmt, kappa.thmt, lambda.thmt, phi.thmt,
    psi.thmt, tau.thmt, theta.thmt, s_beta.thmt, s_kappa.thmt,
    s_lambda.thmt, s_psi.thmt, s_theta.thmt, s_e_init.thmt
)
```

which is the same as the corresponding line in 20170614_mcmc_0_single.R, except that the argument `rebuild` is default to `FALSE` above. In the same fashion, for any combination of algorithm (individual model/RJMCMC/GVS) and data (#thehandmaidstale/#gots7/simulated), the corresponding code chunk can be found in reproducibility.Rnw and run individually, with the argument `rebuild` changed to `TRUE`, for checking with the respective MCMC output.

3.1.3 From MCMC output to figures and numbers

What is to be reproduced: All figures and numerical results in the paper

How to reproduce analyses: There are two ways of reproducing the figures, the second of which also reproduces the numerical results. The first way is to separately run the corresponding code chunk for each figure in the script reproducibility.Rnw, in an interactive R session. The second way is via dynamic document generation using reproducibility.Rnw as the source file. In order to do so, first knit the Rnw file into a tex file by a command in R:

```
# install.packages("knitr")
library(knitr); knit("reproducibility.Rnw")
```

Then, generate a pdf file from reproducibility.tex in a terminal (**not R**):

```
pdflatex \\nonstopmode\\input reproducibility
```

The file reproducibility.pdf will now contain all the numerical results and figures in the paper. The only omission is the dotted red lines in Figure 9 in the paper. Their exclusion is due to, as mentioned before, the sheer sizes of the csv files required to compute the prediction intervals of the retweet counts.

3.2 Replication

Without the user running the MCMC as explained in Section 3.1.2, the computation times in reproducibility.pdf generated from reproducibility.Rnw will be based on the output *provided by the authors*. All the MCMC runs were performed on a Linux machine with Intel Core i5-4690S Processor (3.2GHz). Replacing the output by the results obtained by the user will give different computation times to those reported in the paper, albeit identical Bayes factors and posterior predictive values.

4 Notes

None