Author Contributions Checklist Form

This form documents the artifacts associated with the article (i.e., the data and code supporting the computational findings) and describes how to reproduce the findings.

Part 1: Data

☐ This paper **does not** involve analysis of external data (i.e., no data are used or the only data are generated by the authors via simulation in their code).

☒ I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.

Abstract

We analyze the following two datasets in Section 6.

HCP dataset: This is a 68 × 68 × 136 binary tensor consisting of structural connectivity patterns among 68 brain regions for 136 individuals from Human Connectome Project (HCP). The tensor entries encode the presence or absence of fiber connections between 68 brain regions.

Nations dataset: This is a 14 × 14 × 56 binary tensor consisting of 56 political relations among 14 countries between 1950 and 1965. The tensor entry indicates the presence or absence of a political action, such as “treaties”, “sends tourists to”, between the nations.

Availability

☒ Data **are** publicly available
☐ Data **cannot be made** publicly available

If the data are publicly available, see the *Publicly available data* section. Otherwise, see the *Non-publicly available data* section, below.

Publicly available data

☐ Data are available online at:
☒ Data are available as part of the paper’s supplementary material.
☐ Data are publicly available by request, following the process described here:

☐ Data are or will be made available through some other mechanism, described here:

HCP and nations datasets can be accessed through the R package, tensorregress. Use data (HCP) and data(nations) to load the data.

Non-publicly available data

Discussion of lack of publicly available data:

Description

File format(s)

☐ CSV or other plain text:
☒ Software-specific binary format (.Rda, Python pickle, etc.): HCP.Rdata, nations.Rdata
☐ Standardized binary format (e.g., netCDF, HDF5, etc.):
☐ Other (described here):

Data dictionary

☐ Provided by the authors in the following file(s): readme.txt
☒ Data file(s) is (are) self-describiing (e.g., netCDF files)
☐ Available at the following URL:

Additional information (optional)

Part 2: Code

Abstract

We build an R package, tensorregress, that implements the Algorithm 1 in Section 4.2. Functions to generate random tensor data and select the tensor rank by BIC are also included in the package. The R package can be accessed through “tensorregress.tar.gz” or downloading from https://cran.r-project.org/web/packages/tensorregress/index.html. We use this R package for all the simulations and real data analyses in the paper.

We provide the R scripts for reproducing the Figures 2-8, S1, S2 and Tables 2-3 in the main text and supplementary notes. The file names of the R scripts are consistent with the number of figures (e.g. Figure2.R reproduces Figure 2, Table2.R reproduces Table 2.)

Scripts for simulations (Figure2.R, Figure3.R, Figure4.R, Figure5.R) depends on the source script “simulation.R”. The script “Figure5.R” also depends on the source script “compare.R”.

We produce the data for Figure 6 via R and visualize the data via a Matlab software, BrainNet Viewer, version 1.7. The software is available at https://www.nitrc.org/projects/bnv. Readers should import the data generated by Figure6.R, output_HCP.Rdata, to the BrainNet Viewer to reproduce Figure 6.

Table 2 and Table 3 are organized from the data generated by Table2.R and Table3.R.

Description

Code format(s)

☒ Script files

☒ R ☐ Python ☐ Matlab

☐ Other:

☒ Package

☒ R ☐ Python ☐ MATLAB toolbox

☐ Other:

☐ Reproducible report

☐ R Markdown ☐ Jupyter notebook

☐ Other:

☐ Shell script

☐ Other (described here):

Supporting software requirements

Version of primary software used

R version 3.5.1

Libraries and dependencies used by the code

Libraries for R package, tensorregress:
MASS version 7.3-50
speedglm version 0.3-2
pracma version 2.2.5

Libraries for figures:
ggplot2 version 3.3.2
patchwork verison 1.0.1.9000
lattice version 0.20-35
RColorBrewer version 1.1-2

Supporting system/hardware requirements (optional)

Parallelization used

☒ No parallel code used
☐ Multi-core parallelization on a single machine/node
Number of cores used:
☐ Multi-machine/multi-node parallelization
Number of nodes and cores used:

License

☒ MIT License (default)
☐ BSD
☐ GPL v3.0
☐ Creative Commons
☐ Other (described here):
Click or tap here to enter text.

Additional information (optional)

Part 3: Reproducibility workflow

Scope

The provided workflow reproduces:
☐ Any numbers proviided in text in the paper
☐ All tables and figures in the paper
☒ Selected tables and figures in the paper, as explained and justified here:

Our package tensorregress is used in all the analyses in the paper. Before reproducing our results, you will need to install the following packages to run our package tensorregress:

MASS version 7.3-50
speedglm version 0.3-2
pracma version 2.2.5

To reproduce the figures, you will need to install the following packages for particular figures:

ggplot2 version 3.3.2 (Figure2-5, S1, S2)
patchwork verison 1.0.1.9000 (Figure 2,5)
lattice version 0.20-35 (Figure 8)
RColorBrewer version 1.1-2 (Figure 8)

If only plotting is needed, then the numbers for Figures 3-6 and Table 2 are pre-saved as .Rdata files, with files names consistent with the figure or the table (e.g. Figure5.RData are the data used for Figure 5). We provide these dataset for readers to compare the reproduced results with our results. Use load() function to load the data (e.g. load(Figure5.RData)).

If a new simulation run is desired, we also provide the simulation code (with same random seed as ours) to reproduce the results in Figures 3-6 and Tables.

Figures 2-8, S1, S2 and Tables 2, 3 in the main text and supplementary notes can be reproduced by the following workflow.

Workflow details

Format(s)

☐ Single master code file
☐ Wrapper (shell) script(s)
☐ Self-contained R Markdown file, Jupyter notebook, or other literate programming approach
☐ Text file (e.g., a readme-style file) that documents workflow
☐ Makefile
☒ Other (more detail in 'Instructions' below)

Instructions

Install and load the R package tensorregress and other necessary packages.
For Figure 2, run the file “Figure2.R” directly.
For Figures 3-5, and Table 2,
— to use the pre-saved data, run the plotting codes in the former part of the scripts;
— to reproduce the results, run the simulation codes in the latter part of the scripts and then run the plotting codes.
The plotting part and simulation part can be distinguished by the annotation in the scripts.
For Figure 6, run the file “Figure6.R” to obtain the result “output_HCP.RData”. Export the slices of the coefficient tensor as .txt files. Import these .txt files to the BrainNet Viewer to reproduce Figure 6. The preserved .txt files are in data_for_Figure6
For Figure 7, put the result “output_HCP.RData” generated by “Figure6.R” under the same folder of “Figure7.R” and run the file “Figure7.R” directly.
For Figure 8, run the file “Figure8.R” directly.

Expected run-time

Approximate time needed to reproduce the analyses on a standard desktop machine:

☐ <1 minute
☐ 1-10 minutes
☐ 10-60 minutes
☐ 1-8 hours
☐ >8 hours
☒ Not feasible to run on a desktop machine, as described here:

The total run-time depends on the combination of number of simulation replicates, rank, dimension, signal level, data type in consideration. For example, our Figure 3 considers 30*3*7*2*3 = 3,780 possible combinations. Because each model fitting is separate from the other, we run these simulations in parallel in server with four cores.

Additional documentation (optional)

Regarding the run-time of the main function, tensor_regress, in our package tensorregress, the rough seconds for one iteration are below:
— 0.6 second, under the normal model with tensor dimension d1 = d2 = d3 = 30, side information dimension p1 = p2 = p3 = 12, and tensor rank r1 = r2 = r3 = 3.
— 3.5 second, under the normal model with tensor dimension d1 = d2 = d3 = 30, side information dimension p1 = p2 = p3 = 12, and tensor rank r1 = r2 = r3 = 6.