JASA ACS Reproducibility Initiative - Author Contributions Checklist Form

The purpose of the Author Contributions Checklist (ACC) Form is to document the code and data supporting a manuscript, and describe how to reproduce its main results.

As of Sept. 1, 2016, the ACC Form must be included with all new submissions to JASA ACS.

This document is the initial version of the template that will be provided to authors. The JASA Associate Editors for Reproducibility will update this document with more detailed instructions and information about best practices for many of the listed requirements over time.

Data

Abstract

Our primary data is a mobile app gaming data that holds daily player level gaming information for a free-to-play Robot versus Robot Fighting game based on the movie Real Steel for Windows, iOS and Android devices. The main game-play revolves around fighting and upgrading the robots while the secondary goals are to own as many robots as possible and collect rewards. A key feature of the game is a Lucky Draw which is a card game where players bet on their earnings to earn exciting in-app consumables, virtual currencies for robot upgrades or even robots! There are 38,860 players with first activity date 24-Oct-2014 and for each player, there are 29 variables, apart from the unique player identifier field, across the 60 time points. These variables have been described in table 6 of the supplementary materials. We also have side information about the different retention and promotion strategies that were used across the 60 days. Table 8 and figure 9 in the supplementary materials provide a summary of the 6 different promotion strategies that were used during the 60 days that the players were observed.

Availability

The primary data is proprietary and not available publicly as the authors have signed a Non-Disclosure Agreement (NDA) with the data provider (henceforth referred to as client). The NDA prohibits the authors to disclose the client name and any level of aggregated /dis-aggregated data that can materially be used to identify players, their game patterns or any other strategies used by the client for marketing, sales or brand development. However, the authors have released a MATLAB implementation of the CEZIJ procedure in GitHub at https://github.com/trambakbanerjee/cezij#what-is-cezij. This repository stores the implementation of the CEZIJ procedure and allows users to test the method on a pseudo data set with similar contents but with the variables holding simulated data rather than the actual observed values. A description of the pseudo data is available below while further details on the pseudo data, instructions to create such data and reproduce setting I of the numerical experiment of section C.2 of the supplementary material is available in `cezij help.pdf` at the GitHub public repository https://github.com/trambakbanerjee/cezij.

Pseudo Data Description

The MATLAB implementation of the CEZIJ procedure (see https://github.com/trambakbanerjee/cezij) uses a pseudo data that can be used to replicate setting I of the numerical experiment of section C.2 of the supplementary material. This data holds a sample of $n = 500$ players observed over a period of $m = 30$ days. There are $p = 10$ candidate predictors of which 8 are fixed effects and the remaining are composite effects. For each player i, let $\mathbb{X}_i = (\boldsymbol{X}_{i.1}, \ldots, \boldsymbol{X}_{i.p})$ denote the $m \times p$ matrix of candidate predictors where $\boldsymbol{X}_{i.k} = (x_{i1k}, \ldots, x_{imk})^T$, $m = 30$ denotes the number of time points for which we observe player i. We take $\mathcal{I}_f = \{1, \ldots, 8\}$ as the indices of the fixed effects and, $\mathcal{I}_c = \{9, 10\}$ as the indices of the composite effects so that $p_f = |\mathcal{I}_f| = 8$, $p_c = |\mathcal{I}_c| = 2$. Thus, the first 8 columns of $\mathbb{X}_i$ represent fixed effects while the last 2 represent composite effects. The five responses $[\alpha_i, \mathbb{A}_i, \epsilon_i, \mathbb{E}_i, \mathbb{D}_i]$ corresponding to `AI`, positive Activity, `EI`, Positive Engagement and Dropout are generated from the following models: $\mathsf{logit}(\pi_{ij}) = \beta_0^{(1)} + x_{ij1}\beta_1^{(1)} + b_{i1}$,

$\mu_{ij} = \beta_0^{(2)} + x_{ij2}\beta_1^{(2)} + b_{i2}$ with $\sigma_1 = 0.5$, $\mathsf{logit}(q_{ij}) = \beta_0^{(3)} + x_{ij3}\beta_1^{(3)} + b_{i3}$, $\gamma_{ij} = \beta_0^{(4)} + x_{ij4}\beta_1^{(4)} + b_{i4}$ with $\sigma_2 = 0.5$ and $\mathrm{logit}(\lambda_{ij}) = \beta_0^{(5)} + x_{ij5}\beta_1^{(5)} + \eta_1 b_{i1} + \ldots + \eta_4 b_{i4}$ where the true values of the fixed effect coefficients are: $\boldsymbol{\beta}^{(1)} = (1, -1.5)$, $\boldsymbol{\beta}^{(2)} = (3.5, -2)$, $\boldsymbol{\beta}^{(3)} = (1, -1)$, $\boldsymbol{\beta}^{(4)} = (3, -3)$, $\boldsymbol{\beta}^{(5)} = (-1, 2)$ and, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_4) = (-0.1, 0.2, 0.1, -0.2)$. Thus this setting presents a scenario wherein there are no composite effects in the true model. The random effects $\boldsymbol{b}_i = (b_{i1}, \ldots, b_{i4})$ are sampled from $N_4(\mathbf{0}, \boldsymbol{\Sigma})$, independently for each i, where

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.2 & 0.4 & 0.5 \\ 0.2 & 3 & 0.9 & 0.7 \\ 0.4 & 0.9 & 0.8 & 0.5 \\ 0.5 & 0.7 & 0.5 & 4 \end{pmatrix}$$

Finally, to complete the specification, we sample $(x_{ij1}, \ldots, x_{ij4})$ from $N_4(\mathbf{0}, 4\boldsymbol{I}_4)$, independently for each $i = 1, \ldots, n$, $j = 1, \ldots, m$. To ensure that the generated sample contains players that have not churned for at least the first 7 to 10 days, we let $\boldsymbol{X}_{i.5}$ to be an m dimensional ordered sample from $\texttt{Unif}(-1, 1)$ so that $\boldsymbol{X}_{i.5} = (x_{i15} \leq \cdots \leq x_{im5})$ and, generate the remaining predictors independently from $\texttt{Unif}(-1, 1)$. Please see section 3 of `cezij help.pdf` (https://github.com/trambakbanerjee/cezij) for further details regarding the pseudo data including instructions to generate it.

Code

Abstract

The authors have released a MATLAB implementation of the CEZIJ procedure in GitHub at https://github.com/trambakbanerjee/cezij#what-is-cezij. To install this toolbox, simply download `cezij.mltbx` (available at https://github.com/trambakbanerjee/cezij) in your computer and double click to install it. For a successful installation, please make sure that the following system requirements are met.

- Access to 32GB RAM and at least 8 CPU cores for parallel computing

- MATLAB 2016b or higher with the following toolboxes (and their dependencies):
  - Statistics toolbox
  - Optimization toolbox
  - Parallel Computing toolbox
  - Data Acquisition toolbox
- CVX for MATLAB (version 2.1 or higher)

Description

In this section, we indicate which files in the toolbox should be edited post installation to reproduce the results of setting I of the numerical experiment discussed in section C.2 of the supplementary material. Further details can be found in section 3 of `cezij help.pdf` available at https://github.com/trambakbanerjee/cezij. Figure 1 presents three MATLAB scripts that should be edited to prepare the cezij toolbox for analyses. In what follows, we discuss these scripts.

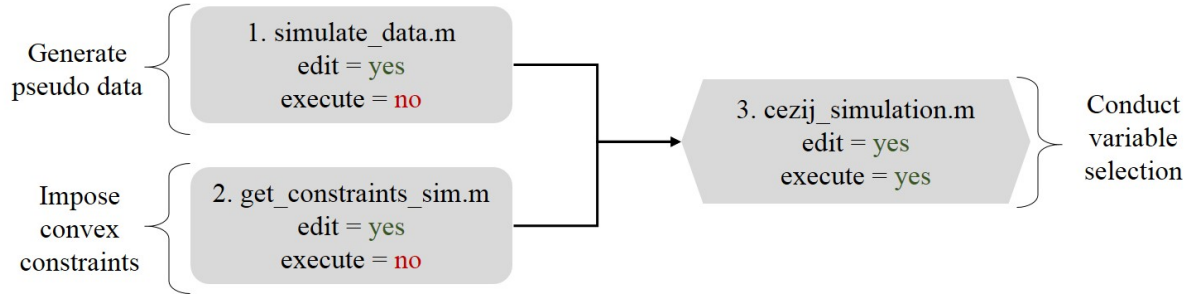

Figure 1: There are three MATLAB scripts, numbered 1-3, that must be edited to prepare the `cezij` toolbox for analyses. To reproduce table 4 in the supplementary file or table 1 in `ceziz help.pdf`, script 3 must be executed without making any changes to the default parameters in scripts 1-3.

1. `simulate_data.m`

To run the `cezij` variable selection algorithm on a dataset of your own choice, please edit the m file `simulate_data.m` and ensure that your data is in the same format as the output of this m file. In the default setting, `simulate_data.m` generates a simulated dataset that is described in the `Pseudo Data Description` section of this form.

2. `get_constraints_sim.m`

   The CEZIJ framework can incorporate convexity constraints on the fixed effect coefficients and this MATLAB file stores the following default constrains:

$$\beta_0^{(1)} > 0,\ \beta_1^{(1)} < 0;\ \beta_1^{(3)} < 0;\ \beta_0^{(4)} > 0,\ \beta_1^{(4)} < 0.$$

   Please modify this file to enforce constraints specific to your application or leave this file unchanged to reproduce table 4 of the supplementary materials or table 1 in `cezij help.pdf`.

3. `cezij_simulation.m`

   Recall that CEZIJ uses a split-and-conquer approach to split the full set of n players into K non-overlapping groups and conducts variable selection separately in each group by solving K parallel maximization problems represented by equation (9) of the main paper. The selected fixed and random effects are then determined using a majority voting scheme across all the K groups as described in Section 5 of the main paper.

   In the MATLAB file `cezij_simulation.m`, lines `15-27` can be used to specify a number of user defined parameters. For example in the default setting, we fix $K = 5$ so that n/K is 100 while $q = 3$ indicates the number of random effects including a random intercept. In the default setting, we generate `nsets` $= 25$ datasets and run `cezij_simulation.m` to assess the model selection performance in terms of the average False Negatives (in-model predictors falsely identified as being out of model) and average False Positives (out of model predictors falsely identified as being in-model)

for the fixed effects (composite or not) and the random effects. To evaluate the hierarchical selection property of our framework, the code also reports the percentage of datasets where our method conducted non-hierarchical selection and chose predictors with random effects only. Please see table 4 in the supplementary materials or table 1 in `cezij help.pdf` for the results of the simulation exercise under this default setting (setting I of the numerical experiment discussed in section C.2 of the supplementary material).

Instructions for Use

To install the `cezij` MATLAB toolbox, simply download `cezij.mltbx` (available at https://github.com/trambakbanerjee/cezij) in your computer and double click to install it. For a successful installation, please ensure that your system meets the minimum hardware and software requirements listed in the `Code` section of this form. Alternatively, see section 2 of `cezij help.pdf`(https://github.com/trambakbanerjee/cezij).

Reproducibility

To start using the toolbox, run `cezij_simulation.m` to reproduce the results of simulation experiment I discussed in section C.2 of the supplementary material. For using a different simulation setting or a different dataset altogether, please refer to section 3 of `cezij help.pdf`.