Tangetial Wasserstein Projections

Data Availability and Provenance Statements

Summary of Availability

- [ ] All data used herein are publicly available.

Details on Each Data Source

- For Lego image replication, the data can be downloaded from Kaggle: https://www.kaggle.com/datasets/joosthazelzet/lego-brick-images. In our repository, select `Lego_bricks` in the folder `Data`, and the images used are contained therein. We kept the same file names for the files downloaded from the link above.

Datafiles: `Data/legos`

- For Medicaid expansion application, the data can be downloaded from IPUMS: https://usa.ipums.org/usa/.

We downloaded the variables stated in the main text: HINSCAID, EMPSTAT, UHRSWORK, INCWAGE.

Additionally, we selected RELATE (relationship to household head) to allow us to limit our sample to household head and spouse (if any). We selected PERWT (person weight) to identify how many persons in the U.S. population are represented by a given individual in the ACS sample. We applied further sample selection criteria mentioned in Appendix B.3 of the accompanying paper, using the STATEFIP and AGE variables.

Dataset list

Data file	Source	Notes	Provided
`Data/legos/0001.png`	Listed above	Target image in Section 4.2 of main text	Yes
`Data/legos/0040.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0080.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0120.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0160.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0200.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0240.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0280.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0320.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0360.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/legos/0400.png`	Listed above	Control image in Section 4.2 of main text	Yes
`Data/Medicaid_Data/`	Listed above	ACS data used to obtain optimal weights lambda star	Yes; in Dropbox folder above
`Data/Medicaid_Data/Counterfactual`	Listed above	ACS data used to obtain counterfactual distributions in Section 4.3	Yes; in Dropbox folder above

Computational requirements

Software Requirements

The simulations and applications were ran using Python/3.8.1.

- ot (pip install pot)
- cvxpy
- numpy
- matplotlib
- pandas
- seaborn

Memory and Runtime Requirements

Summary

Approximate time needed to reproduce the analyses on a standard (CURRENT YEAR) desktop machine:

- [ ] 1-3 days

Details

The code was last run on a **8-core Apple-based laptop with MacOS version 12.4**.

Portions of the code were last run on a **36-core Intel server with 180 GB of RAM**. Computation took 2 hours.

Description of programs/code

- Programs in `Python Code` will perform all simulations and estimations described in the main text. These programs will also generate all tables and plots found therein.

Instructions to Replicators

- Download the repository to your local computer.
- Download MNIST to `Data/mnist`.
- Download Medicare data over the pre-intervention time periods, as described in the main text, to `Data/Medicaid_Data`.
- Download Medicare data over the counterfactual time periods, as described in the main text, to `Data/Medicaid_Data/Counterfactual`.
- Move the `Data` folder to the `Python Code` subdirectory.
- Run `Python Code/Multivariate Gaussian Example.py` for the Gaussian simulations.
- Run `Python Code/MNIST Experiment.py` for the MNIST image experiment.
- Run `Python Code/Lego Bricks Application.py` for the Lego Brick image replication.
- Run `Python Code/Medicaid.ipynb` for the Medicaid expansion application.

The .png files will be stored in `Python Code` subdirectory by default. The tables are generated within each .ipynb program. If you wish to save the .png files in alternative directories, please change the corresponding `plt.savefig()` commands in the programs.

Details

- `Python Code/DSC_setup.py`: defines barycentric projection and projection method described in the main text.
- `Python Code/Multivariate Gaussian Example.py`: contains the Gaussian simulations described in the main text.
- `Python Code/Lego Bricks Application.py`: contains the Lego Brick image replication described in the main text.
- `Python Code/MNIST Experiment.py`: contains the MNIST image experiment described in the main text.

- `Python Code/Medicaid.ipynb`: contains the Medicaid expansion application described in the main text.

Apart from `Python Code/Lego Bricks Application.py`, it takes less than 5 minutes to finish running all the programs on a laptop with Apple M1 processor. Running `Python Code/Lego Bricks Application.py` takes 2 hours from start to finish on a cluster computer with 36 cores.

List of tables and programs

The provided code reproduces:

- [ ] All tables in the paper
- [ ] Figures 2 through 7 in the paper

Figure/Table #	Program	Line (Block) Numbe
Table 1	Python Code/Medicaid.ipynb	9
Figure 2	Python Code/Multivariate Gaussian Example.py	188, 189
Figure 3	Python Code/MNIST Experiment.py	296
Figure 4 (Top)	Python Code/Lego Bricks Application.py	143, 200
Figure 4 (Weights)	Python Code/Lego Bricks Application.py	189
Figure 5	Python Code/Medicaid.ipynb	19, 20
Figure 6	Python Code/Medicaid.ipynb	21, 22
Figure 7	Python Code/Medicaid.ipynb	10, 12
Figure 8	Python Code/Medicaid.ipynb	14, 15

References

Steven Ruggles, Sarah Flood, Ronald Goeken, Megan Schouweiler and Matthew Sobek. IPUMS USA: Version 12.0 [dataset]. Minneapolis, MN: IPUMS, 2022.
https://doi.org/10.18128/D010.V12.0