Reproducing the results in *Optimal transport weights for causal inference*

Eric Dunipace

3/27/2022

Basics

This R Markdown document will reproduce the tables and figures in the paper *Optimal transport weights for causal inference*. If this file is compiled in this directory, the tables and figures will be generated from the source files found in the `code` folder.

To generate tables and figures

By default, this document will re-compile the tables and figures from the simulation studies using the already run simulation data from the original paper. This document will also input the tables and figures from the case study. The case study can be re-run by setting the corresponding chunk to `eval = TRUE`.

To re-run simulations

This will be a lengthy process if not on a cluster. The original files used to run the simulations on the cluster are found in the subfolder `code\original_sim` for both the convergence analysis and Hainmueller simulations. How to run these on a cluster is included below.

Required R pacakges

To re-run these simulations and/or recompile tables and figures, there are several libraries needed.

For just compiling tables and figures, here's all of the libraries that should be loaded:

```
library(causalOT)
library(dplyr)
library(ggplot2)
library(scales)
library(ggsci)
library(xtable)
library(cowplot)
library(tidyr)
```

For re-running simulations, the following packages are needed

```
library(causalOT)
library(doRNG)
library(Rmosek)
```

as well as the Python libraries `numpy`, `scipy`, `pykeops`, and `geomloss`.

Simulations studies

Study of bias/RMSE using Hainmueller (2012)

To generate the figures and tables for the simulations, we can use the following code

```
source("code/Hainmueller.R")
```

This will take the simulations for the Hainmueller setting and calculate the summary statistics. Then it generates the Latex table in Table 1. Some of the references in the table will not work since this is not in the larger paper with the .bib file.

Note this will download the simulation data if it's not already found in the data folder of this workflow.

Re-running on a cluster

To re-run this data, I recommend using a cluster. There are several other files in `code/original_sim` to discuss.

1. `hain_setting_array.R` generates a setting array that the cluster run will refer to in order to setup the simulation settings.
2. `generate_seeds.R` generates a list of seeds to be used by the analysis (and also for the convergence simulations)
3. `combine_sim_res.R` will combine the raw simulation results into one folder.

The files in `code/original_sim/Hainmueller` will re-run the raw analysis. The command

```
sbatch --array=1-1500 hain.sh
```

should re-run the results as desired.

Output

At the end we get the following table, which should match the paper.

			Bias			RMSE		
overlap	method	constraint	Hajek	DR	WOLS	Hajek	DR	WOLS
high	GLM	none	**-0.01**	**-0.01**	-0.02	1.18	1.14	1.14
	CBPS	means	0.24	**-0.01**	-0.02	1.12	1.11	1.09
	SBW	means	**-0.01**	**-0.01**	**-0.01**	1.00	1.00	1.00
	SCM	none	0.36	0.27	0.28	1.63	1.57	1.55
	NNM	none	0.43	0.32	0.28	0.69	0.65	0.56
	COT	none	**0.01**	**0.01**	**0.01**	0.61	0.61	0.61
		means	**0.01**	**0.01**	**0.01**	**0.42**	**0.42**	**0.42**
medium	GLM	none	1.12	1.10	1.04	1.72	1.69	1.70
	CBPS	means	1.20	1.06	0.95	1.72	1.64	1.56
	SBW	means	0.63	0.63	0.63	1.20	1.20	1.20
	SCM	none	1.19	1.12	1.10	2.05	1.97	1.95
	NNM	none	0.73	0.65	0.58	0.94	0.91	0.79
	COT	none	0.23	0.23	0.23	0.74	0.74	0.74
		means	**-0.03**	**-0.03**	**-0.03**	**0.43**	**0.43**	**0.43**
low	GLM	none	0.19	0.06	0.02	1.72	1.49	1.51
	CBPS	means	0.45	0.06	0.01	1.42	1.46	1.42
	SBW	means	0.03	0.03	0.03	1.03	1.03	1.03
	SCM	none	0.64	0.42	0.43	1.75	1.69	1.65
	NNM	none	0.81	0.56	0.49	1.02	0.89	0.77
	COT	none	0.05	0.05	0.05	0.85	0.85	0.85
		means	**0.00**	**0.00**	**0.00**	**0.41**	**0.41**	**0.41**

Table 1: Performance of various weighting methods under the simulation settings of [**?**]. Bold values are the values with the lowest bias or root mean-squared error (RMSE) of the methods under the same conditions. GLM refers to weighting by the inverse of the propensity score as calculated from a logistic regression model, CBPS is the covariate balancing propensity score, SBW is the stable balancing weights, SCM is the synthetic control method, and COT is the optimal transport formulation proposed in this paper. The estimators are Hajek weights (Hajek), doubly-robust augmented IPW (DR), and weighted least squares (WOLS). All weights are normalized to sum to 1. Constraints refer to balancing constraints and are one of “none” for no constraints or “mean” for mean constraints.

Convergence and confidence intervals

Similarly, the convergence and confidence interval simulations can be compiled with the following code chunk

```
source("code/convergence.R")
```

Note this will download the simulation data if it’s not already found in the data folder of this workflow.

Re-running on a cluster

To re-run this data, I again **recommend** using a cluster. There are several other files in `code/original_sim` as mentioned previously:

1. `generate_seeds.R` generates a list of seeds to be used by the analysis (and also for the convergence simulations)
2. `combine_sim_res.R` will combine the raw simulation results into one folder.

Then the files in `code/original_sim/Convergence` will re-run the analysis. On a Slurm-based cluster running

```
sbatch --array=1-1000 conv.sh
```

should be sufficient.

Outputs for convergence

Then we can look at the plots demonstrating convergence in terms of Sinkhorn divergence and L_2 norm.

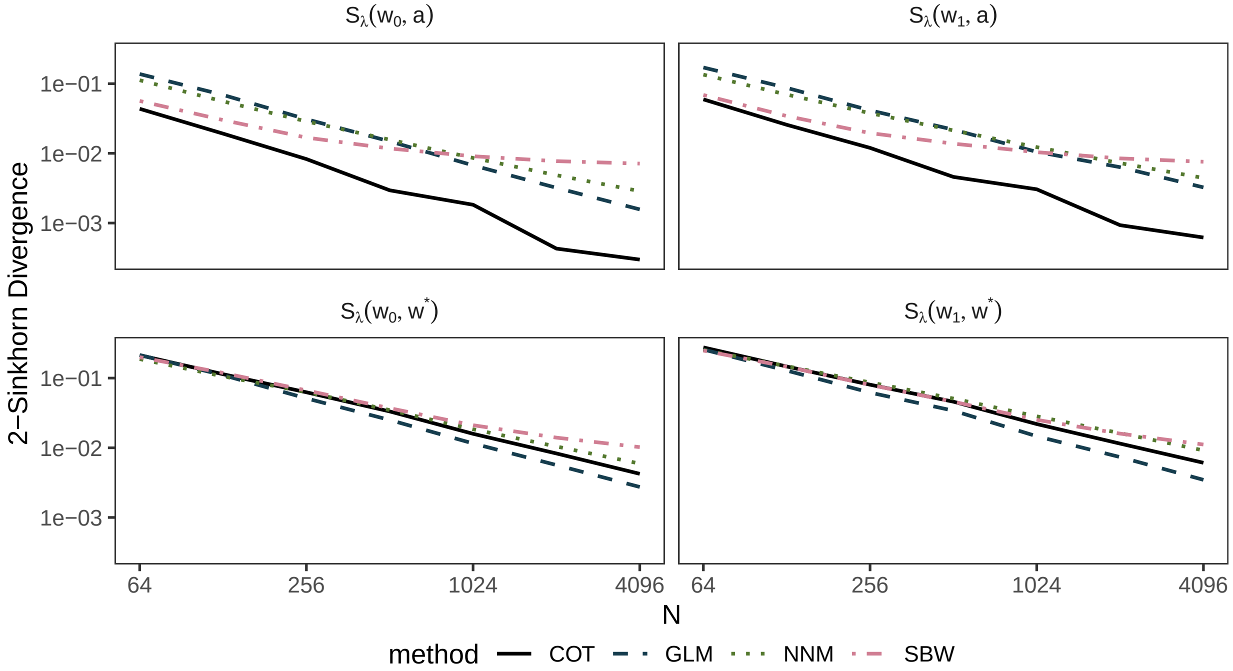

Figure 1: Convergence of the weights to the distributions specified by the empirical distributions (top) and the distributions specified by the true propensity score/Radon-Nikodym derivatives (bottom). Weights are a Causal Optimal Transport (COT), Nearest Neighbor Matching (NNM), a Probit model (GLM), and Stable Balancing Weights (SBW). Lines denote means across 1000 simulations. Both axes are on the log scale.

Outputs for confidence intervals

This analysis also gets the following figures for confidence interval coverage.

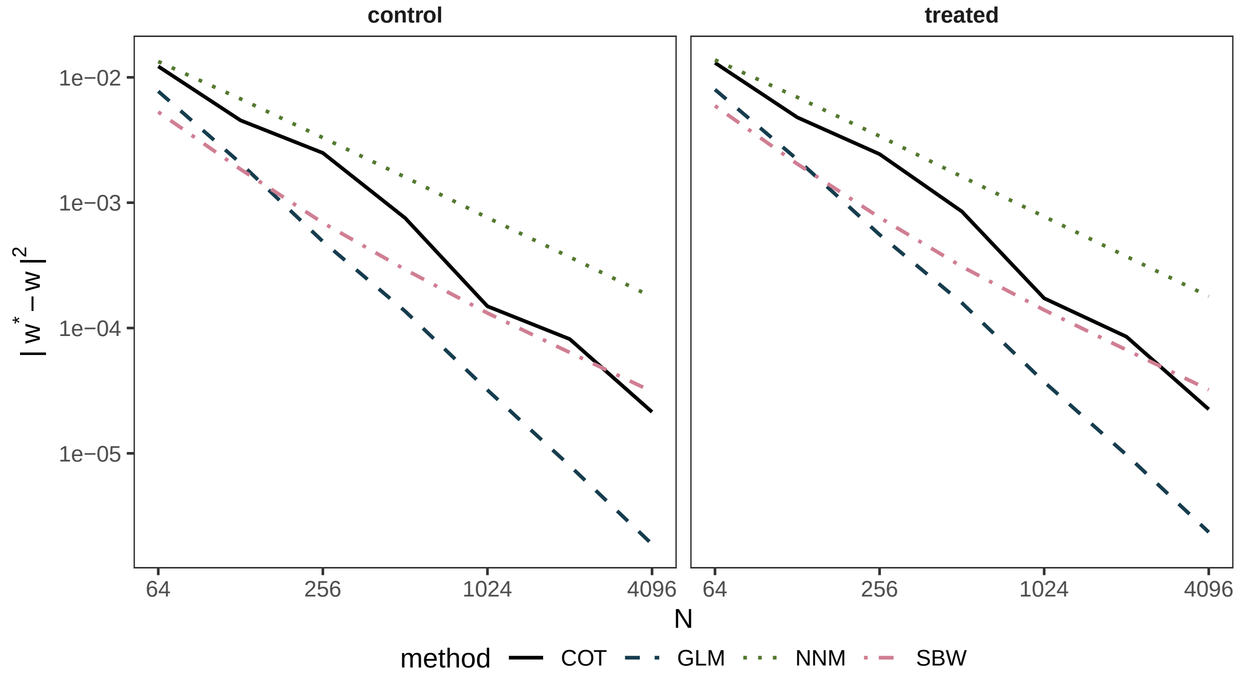

Figure 2: Convergence of the estimated weights to the values of the true inverse propensity score in terms of the L_2 norm. Weights are a Causal Optimal Transport (COT), Nearest Neighbor Matching (NNM), a Probit model (GLM), and Stable Balancing Weights (SBW). Lines denote means across 1000 simulations. Both axes are on the log scale.

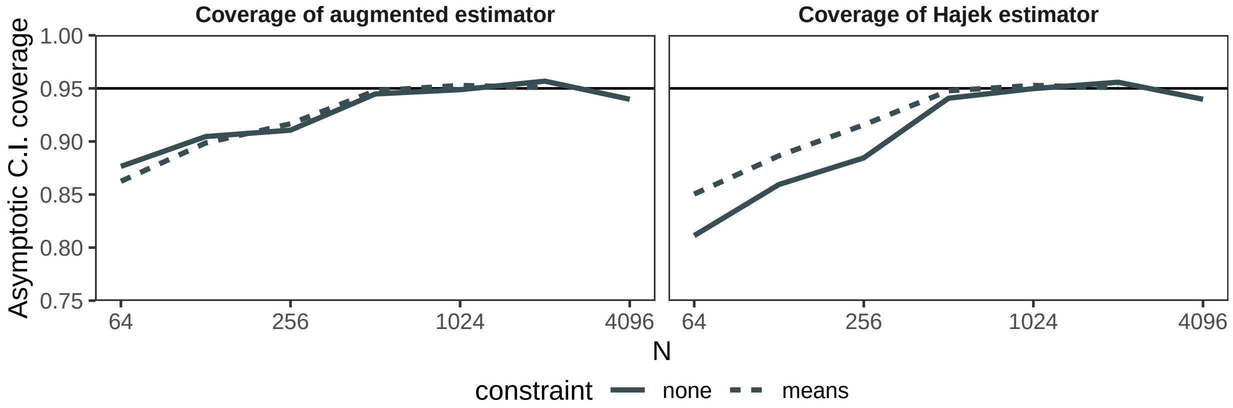

Figure 3: Coverage of the true treatment effect

Algorithm check

In the paper, we offer an algorithm for tuning the hyperparameters of the optimal transport distances. We can generate the figures for the algorithm check with this code

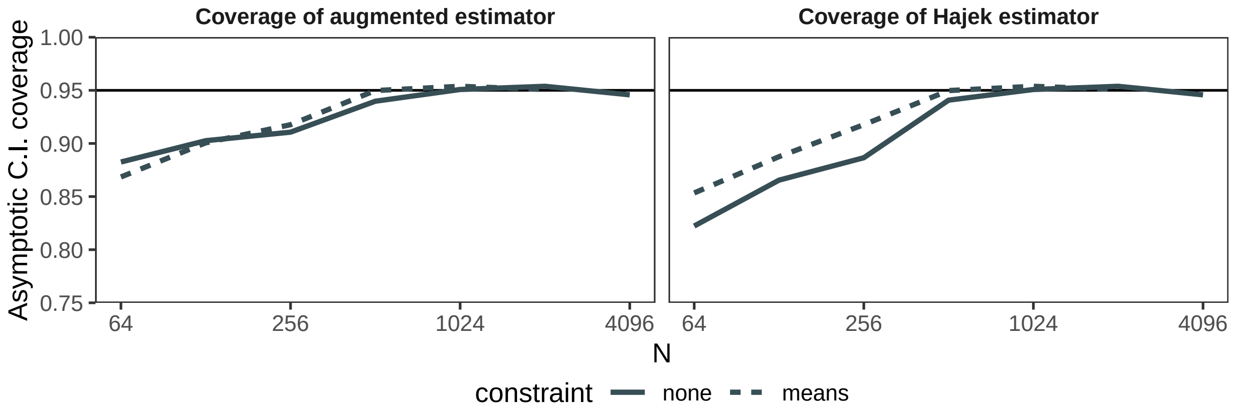

Figure 4: Coverage of the estimated average treatment effect

```r
source("code/algorithm.R")
```

Re-running on a cluster

To re-run this data, I used a cluster. There are several other files in `code/original_sim` as mentioned previously:

1. `generate_seeds.R` generates a list of seeds to be used by the analysis (and also for the convergence simulations)
2. `combine_sim_res.R` will combine the raw simulation results into one folder.

Then the files in `code/original_sim/Algorithm` will re-run the analysis. On a Slurm-based cluster run

```
sbatch --array=1-1000 algor.sh
```

Outputs for algorithm

This generates the following figures.

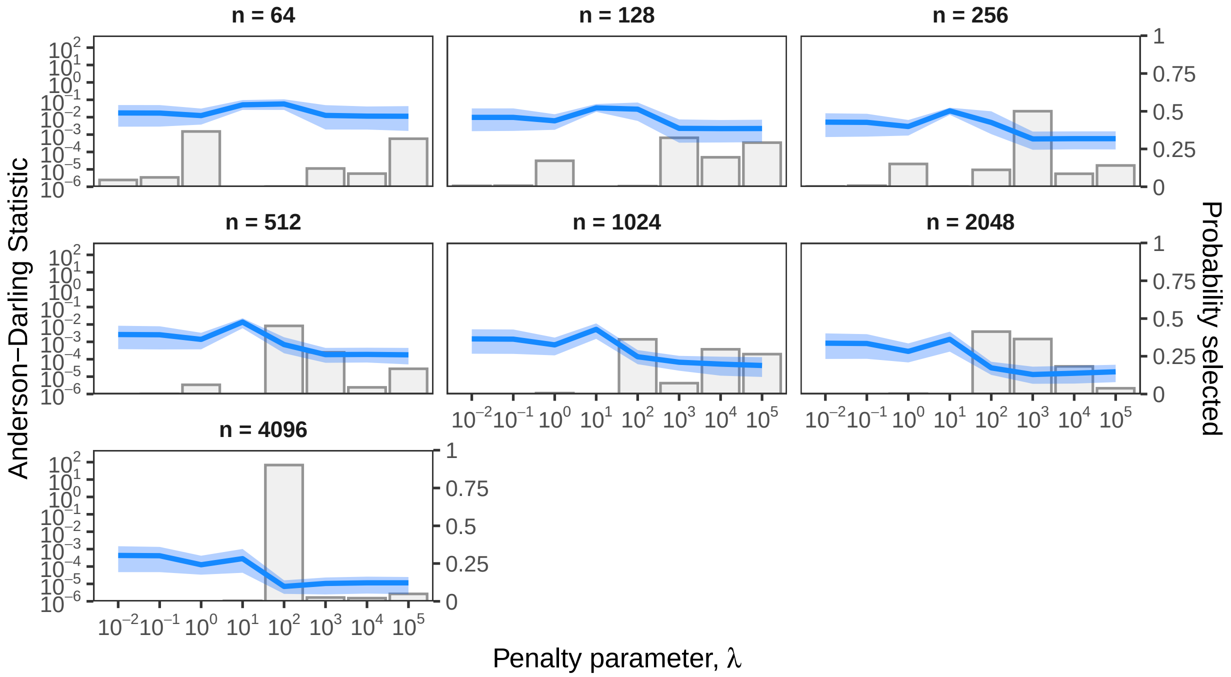

Figure 5: Selection of penalty parameter for the treated

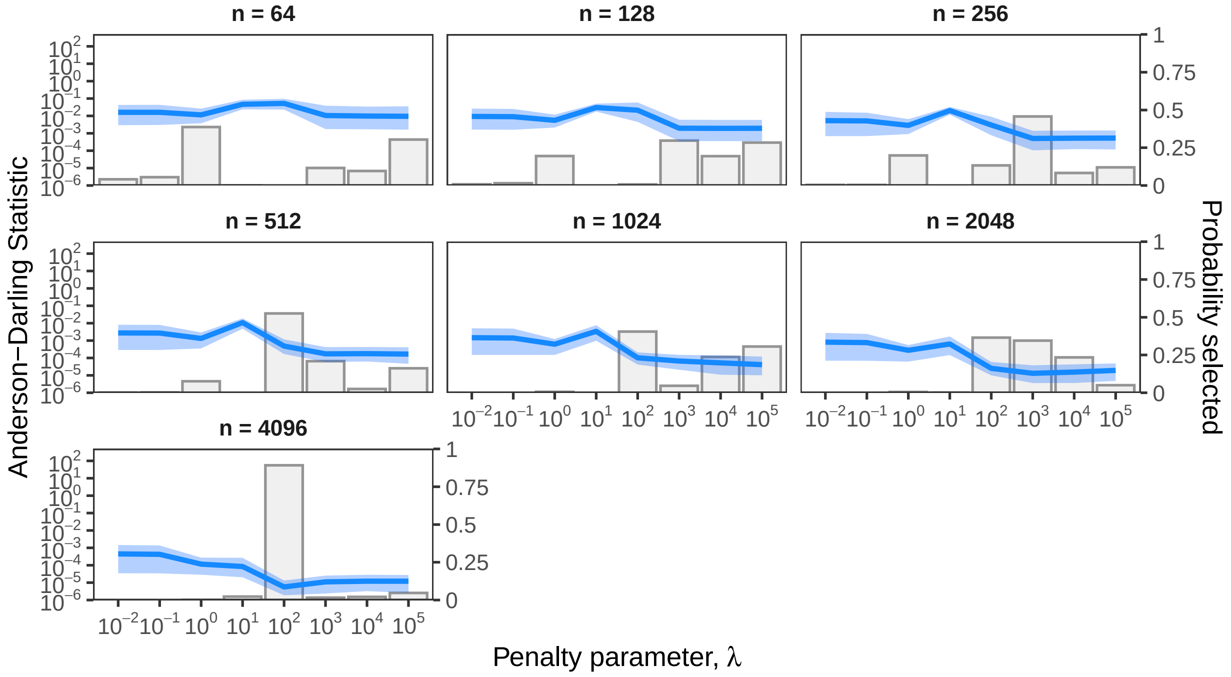

Figure 6: Selection of penalty parameter for the control

A few notes. The Anderson-Darling statistic in this case measures the closeness of the estimated weights, w, to the Radon-Nykodim derivatives, $w^\star$:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{(w_i - w_i^\star)^2}{w_i^\star(1-w_i^\star)}.$$

This gives us a sense of how well chosen penalty parameter, λ, gives weights that approximate the true Radon-Nykodim derivatives. As we can see, the algorithm finds the optimal hyperparameter on average as the sample size grows.

Case study

The data

The data come from a study by Blum et al. looking at the effect of misoprostol vs. oxytocin at stopping post-partum hemorrhage. The data is described in the documents for the `causalOT` package which can be accessed by running `?causalOT::pph`. A more detailed description is found in the paper *Optimal transport weights for causal inference.*

Running the analysis

The analysis can be run by running the following chunk with `eval = TRUE`

```
source("code/misoprostol.R")
```

Figures/Tables

We now turn to the tables and figures generated by this analysis. We can see that the effect estimates look good for the optimal transport methods!

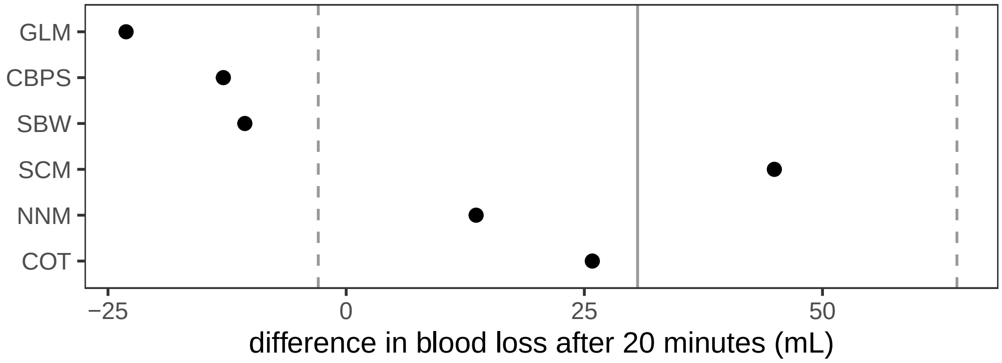

Figure 7: Results for treatment effect estimation averaged across treatment groups and study sites. The grey vertical line is the original treatment effect estimate for the entire study while the dotted vertical lines are the original confidence interval. The weighting methods under examination are logistic regression (GLM), Covariate Balancing Propensity Score (CBPS), Stable Balancing Weights (SBW), Synthetic Control Method (SCM), Nearest Neighbour Matching (NNM), and Causal Optimal Transport (COT).

COT also has the best performance in terms of coverage!

Method	% C.I. covering original effect	% of estimates in original C.I.
GLM	20	20
CBPS	30	30
SBW	30	40
SCM	50	50
NNM	20	20
COT	60	60

Table 2: For each method, the table displays the percentage of times that the calculated 95% confidence interval (C.I.) covered the true treatment effect and whether the estimated treatment effect was inside the original C.I. from the study. The weighting methods under examination are logistic regression (GLM), Covariate Balancing Propensity Score (CBPS), Stable Balancing Weights (SBW), Synthetic Control Method (SCM), Nearest Neighbor Matching (NNM), and Causal Optimal Transport (COT).

Conclusions

This document reproduces the results for the paper *Optimal transport weights for causal inference* by Eric Dunipace. Any questions or concerns can be addressed by filing an issue at https://github.com/ericdunipace/causalOT/issues.