Textual Causal Inference about Readers' Interest on News Articles

Yu Yang*

1 Introduction

A major challenge in causal inference is to deal with the biases caused by confounders. A golden method to eliminate the confounding bias is to perform randomized controlled trials, which can often be infeasible or unethical. In face of this, researchers turn to observational data and try to make adjustment for the confounders. Popular methods include regression adjustment, propensity score regression adjustment, inverse propensity score weighting, and matching.

In recent years, researchers have been trying to dig information out of textual data, which they believe can serve as a surrogate for the latent confounders. For example, Olteanu et al., 2017 investigated how people's current word uses would affect their future word uses, where they measured the confounder - past word uses - from Twitter posts. And Veitch et al., 2020 used articles' content to measure the subjects so as to infer how the presence of a theorem affects the rate of acceptance. However, despite of these progress, several challenges remain. (1) Text data by nature is of high dimension and requires a conversion into a meaningful lower-dimensional representation vector. (2) It remains an open question how to evaluate different causal inference methods under the textual scenario (Keith et al., 2020) and there is no solid best-performing inference method from a theoretical perspective.

This paper focuses on how the length of an article affects readers' interest to read it, which is of critical values to publishers, especially news article publishers, in terms of getting more readers and higher click rate for their articles. In particular, we use the CNN/Daily Mail news articles dataset (Nallapati et al., 2016) for this study, and we consider the treatment as whether the article is longer than 800 tokens[1] and the outcome as whether the readers will read it. One confounder could be the topics of the paper - for example, sports news articles tend to be shorter and at the same time may have a large number of fans choosing to read them. Since there is no solid conclusion on which adjustment strategy works the best under the textual scenario, we perform a series of simulations[2] to investigate the most appropriate method for this particular problem.

*School of Statistics, University of Minnesota, yang6367@umn.edu

[1]A token is a sequence of characters in the document that are grouped together as a semantic unit. We may obtain the tokens of a sentence by chopping it up into pieces and throwing away special characters such as punctuation and whitespace. For example, we may process the sentence "I like winter and snow." into 5 tokens: ["I", "like", "winter", "and", "snow"].

[2]Check the code in Github: `https://github.umn.edu/YANG6367/PUBH8485/tree/master/project`.

The rest of the paper goes as follows. Section 2 introduces the simulation mechanism and the modeling methods in details; Section 3 demonstrates the performance of different adjustment methods by simulation metrics; Section 4 concludes the paper with some discussion.

2 Simulation Mechanism and Methods

In this section, we describe the simulation mechanism and the estimation methods in the methodology level and will go into technical details in Section 3.

2.1 Simulation Mechanism

Since we don't know the ground-truth causal effects, it would be difficult to compare different effect estimation methods empirically. Therefore, similar to Veitch et al., 2020, we approach this problem with semi-synthetic data: we use the real news articles but simulate the outcome based on the treatment and the confounders, and then we use the simulated data to estimate ATE.

We use the CNN/Daily Mail dataset (Nallapati et al., 2016) for simulation, which consists of 312,085 online news articles. As in See et al., 2017, we use the non-anonymized version of the data, which is split into 287,225, 13,368, and 11,490 examples for training, validation and testing respectively. For each news article, we can obtain two attributes:

1. length: the total number of tokens in the article.
2. hot topic inclusion: whether the article contains "government", "crime", "economic", "game", "health".

Let t_i denote whether or not the ith article has more than 800 tokens and let binary vectors z_i denote the inclusion of "politics", "crime", "economics", "game" and "health", with $z_{ik} = 1$ indicating the inclusion of the kth topic type in the ith article. Note that 800 is chosen to be close to the median of the document length in the training set in order to keep the treatment balanced.

First, for each stratum of z (in total 32 strata), we calculate the true propensity score $\pi(z)$, which equals to the proportion of cases with $t_i = 1$ in the training set. Then, we simulate Y_i following the model:

$$Y_i \sim Bernoulli(\sigma(\alpha t_i + \beta\pi(z_i) + \gamma)), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function. The parameter α controls the treatment effect. The parameter β is used to control the level of confounding, implying the bias had we not adjust for the confounding. And the parameters γ is to control the outcome probabilities to stay close to the real-life scenario.

2.2 Estimation Methods

Topic Model To start with, we train a topic model to learn the topic embedding vector for each article, which is then used as the surrogate for the confounders. There are many different topic embedding methods, such pLSA (Hofmann, 1999), LDA (Blei et al., 2003), Replicated Softmax (Hinton and Salakhutdinov, 2009), neural-based topic models (Miao et al., 2017) and ETM (Dieng et al., 2020). We consider the ETM model, which builds on top of the recent progress in pre-trained word embedding areas and has been shown to have good topic interpretation as well as robustness to stop words[3] (Dieng et al., 2020). In more detail, this model assumes the generation story of the dth document to be the following ($d = 1, \cdots, D$):

1. Draw the topic proportion vector $\theta_d \sim LN(0, I_K)$, namely, $\delta_d \sim N(0, I_K), \theta_d = \text{softmax}(\delta_d)$.
2. For the nth word w_{dn} in the document:
   (a) Draw the topic assignment from a multinomial distribution characterized by θ_d: $\xi_{dn} \sim \text{Multinomial}(\theta_d)$.
   (b) Draw the word from a multinomial distribution: $w_{dn} \sim \text{Multinomial}(\text{softmax}(\rho^T \alpha_{\xi_{dn}}))$.

where D is the total number of documents in the data set, $K \in \mathbb{R}$ is the number of topics, $\rho \in \mathbb{R}^{L \times V}$ is the word embedding matrix, with L being the word embedding size and V being the vocabulary size, and $\alpha_i \in \mathbb{R}^L$ is the embedding vector of the ith topic, $i = 1, \cdots, K$.

We use variational inference (Jordan et al., 1999; Wainwright and Jordan, 2008) to fit ETM, where we posit the family of the topic proportion distribution $q(\delta_d; \mathbf{w}_d, \nu)$ to be Gaussian and then use amortized variational inference (Kingma and Welling, 2014; Rezende et al., 2014) to estimate the parameters (see Section 5 in Dieng et al., 2020 for details). The inference network takes the normalized bag-of-word representation[4] of the document as the input and outputs the mean and variance of δ_d, which can then be used to generate the topic proportion vector θ_d through sampling and transformation.

Adjustment Methods After getting the topic proportion vectors, we then build a neural classifier as the propensity score model, with the topic proportion vector being the input[5] and whether the length of the article exceeds 800 tokens being the output. Finally, with the propensity score model at hand, we compare the performance of four average treatment effect (ATE) estimation methods: (1) Propensity Score Regression Adjustment (PSReg); (2) Propensity Score Stratification (PSS); (3) Inverse Probability Weighting (IPW); (4) Inverse Probability Weighting variant (IPW2).

[3] Stop words usually carry little useful semantic information but will affect the quality of topic models if not taken care of. Examples include: "a", "the", "is", "are" and etc.

[4] For each word in the vocabulary, count how many times it appears in the document.

[5] I have also tried using the weighted topic vector $\sum_i^K \theta_{di} \alpha_i \in \mathbb{R}^L$ as the representation vector, but the result is not as good. Check `https://github.umn.edu/YANG6367/PUBH8485/tree/master/project/etm-ps` for more details.

3 Experiments and Results

In this section, we describe the technical details of the experiments and the results.

3.1 Simulation Setup

We consider two schemes to set parameters in Equation 1. For both schemes, we fix $\alpha = -1$ and vary β from 1 to 10 to investigate how different methods perform under different confounding levels. Regarding γ, for the first scheme, we vary it along with β to keep $\overline{Y}$ nearly constant at 0.265, while for the second scheme, we vary it along with β to keep the ground-truth ATE (defined later) nearly constant at -0.2. Therefore, we would consider 20 settings in total. And for each setting, we simulate the data with 100 repetitions and report the mean and standard error of the ATE estimation results.

3.2 Model Details

ETM We use the pre-trained RoBERTa model (Liu et al., 2019) to initialize the word embedding matrix ρ and the tokenizer[6]. The vocabulary size V is 50265 and the word embedding size is 768, and therefore, $\rho \in \mathbb{R}^{50265 \times 768}$. The total number of topics K is set as 300. To perform amortized variational inference, we build a three-layer neural network to model the mean and variance of θ_d and use the reparameterization trick (Kingma and Welling, 2014; Titsias and Lázaro-Gredilla, 2014) and stochastic optimization to update the parameters. Let `FC k-l` denotes a $k \times l$ fully connected layer. The network consists of `FC 50265-800, ReLU, FC 800-800, ReLU, FC 800-300`. The model is trained on the training set for 10 epochs and parameters are tuned using the validation set. The learning rate is initialized as 0.005 and decreases exponentially with rate 0.999. After training, each article in the training set is passed to the trained model to obtain its corresponding topic proportion vector θ_d.

Propensity Score Model The neural propensity score model is a three-layer network consisting of `FC 300-1000, BatchNorm`[7]`, LeakyReLU, FC 1000-300, BatchNorm, LeakyReLU, FC 300-1, Sigmoid`. This model takes the topic proportion vector as the input and outputs the probability that the document is longer than 800 tokens. The training loss function is the Binary Cross Entropy [8] between the target and the estimated probabilities. This model is trained on the training set for 5 epochs and the initial learning rate is 0.02, which then decreases exponentially with rate 0.999. After training, we apply the trained model to the topic proportion vectors and get the corresponding propensity scores.

[6] A tokenizer performs tokenization and transforms a document into a list of tokens. In the paper, we use the one implemented by Hugging Face: `https://huggingface.co/docs/transformers/model_doc/roberta`.

[7] One technique to make the network training faster and more stable.

[8] As defined in PyTorch Documentation `https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html`

ATE Estimation For the PSReg method, we consider the interaction between the treatment and the restricted cubic spline (of order 5) transformation of the estimated propensity score in the logistic regression model. For the PSS method, we divide the data into deciles based on the estimated propensity scores. For the IPW method, $\mathbb{E}[Y^1]$ is estimated by $\frac{1}{D}\sum_{d=1}^{D}\frac{t_i Y_i}{\hat{\pi}_i}$, while for the IPW2 method, $\mathbb{E}[Y^1]$ is estimated by $\frac{1}{D}\sum_{d=1}^{D}\frac{t_i Y_i}{\hat{\pi}_i}/(\frac{1}{D}\sum_{d=1}^{D}\frac{t_i}{\hat{\pi}_i})$.

3.3 Results

For each of the simulation setting, we compute the ground-truth average treatment effect as $\frac{1}{D}\sum_{d=1}^{D}[\sigma(\alpha t_i + \beta\pi(z_i) + \gamma) - \sigma(\beta\pi(z_i) + \gamma)]$ and apply the four ATE estimation methods to the simulated data with the trained propensity score model. The results for Scheme 1 are shown in Table 1 and Figure 1 and the results for Scheme 2 are shown in Table 2 and Figure 2 in the Appendix. We can see that

1. all methods help reduce the confounding bias, and among them, IPW works the best in general under both schemes;
2. the performance of PSReg, PSS, and IPW2 are very similar for all β values under both schemes;
3. if we hold the ground-truth ATE almost constant, then as shown in Figure 2, when β increases, the confounding bias, suggested by the difference between the truth and the unadjusted method, increases, which matches how we simulate the data;
4. still from Figure 2, as β increases, the absolute value of the estimated ATEs given by all four methods become smaller. This implies that large confounding biases tend to eliminate the treatment effect in this case.

4 Discussion

In this paper, we adapt a modern topic model to measure the confounding effect and investigate how articles' lengths affect readers' interest. We have run a simulation study to examine the performance of different confounding adjustment methods and we find that IPW seems to work the best among its cohorts.

There are several future work directions. First, we may try some other topic models and check how the choice of topic models affects the treatment effect estimation. Second, we only consider adjustment methods based on propensity scores in this paper and we may include more, such as regression adjustment and matching. Finally, we focus on the CNN/Daily Mail Dataset in this paper and it is worthwhile to investigate whether our findings apply to the other data sets as well.

References

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *the Journal of machine Learning research* 3, pp. 993–1022.

Dieng, Adji B, Francisco JR Ruiz, and David M Blei (2020). "Topic modeling in embedding spaces". In: *Transactions of the Association for Computational Linguistics* 8, pp. 439–453.

Hinton, Geoffrey E and Russ R Salakhutdinov (2009). "Replicated softmax: an undirected topic model". In: *Advances in neural information processing systems* 22, pp. 1607–1614.

Hofmann, Thomas (1999). "Probabilistic latent semantic indexing". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57.

Jordan, Michael I et al. (1999). "An introduction to variational methods for graphical models". In: *Machine learning* 37.2, pp. 183–233.

Keith, Katherine, David Jensen, and Brendan O'Connor (2020). "Text and Causal Inference: A Review of Using Text to Remove Confounding from Causal Estimates". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5332–5344.

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1312.6114.

Liu, Yinhan et al. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692. arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

Miao, Yishu, Edward Grefenstette, and Phil Blunsom (2017). "Discovering discrete latent topics with neural variational inference". In: *International Conference on Machine Learning*. PMLR, pp. 2410–2419.

Nallapati, Ramesh et al. (2016). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290.

Olteanu, Alexandra, Onur Varol, and Emre Kiciman (2017). "Distilling the outcomes of personal experiences: A propensity-scored analysis of social media". In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 370–386.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). "Stochastic backpropagation and approximate inference in deep generative models". In: *International conference on machine learning*. PMLR, pp. 1278–1286.

See, Abigail, Peter J Liu, and Christopher D Manning (2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083.

Titsias, Michalis and Miguel Lázaro-Gredilla (2014). "Doubly stochastic variational Bayes for non-conjugate inference". In: *International conference on machine learning*. PMLR, pp. 1971–1979.

Veitch, Victor, Dhanya Sridhar, and David Blei (2020). "Adapting text embeddings for causal inference". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR, pp. 919–928.

Wainwright, Martin J and Michael I Jordan (2008). "Graphical Models, Exponential Families, and Variational Inference". In: *Foundations and Trends® in Machine Learning* 1.1-2, pp. 1–305.

5 Appendix

	α	β	γ	$\overline{Y}$	Truth	Unadjusted	PSReg	PSS	IPW	IPW2
Setting 1	-1	1	-1	0.2793	-0.1971	-0.1873	-0.1886	-0.1886	-0.2107	**-0.1887**
				(0.00084)		(0.00176)	(0.00191)	(0.00192)	(0.00196)	(0.00193)
Setting 2	-1	2	-1.6	0.2658	-0.19	-0.1704	-0.1731	-0.1731	**-0.1941**	-0.1732
				(0.00086)		(0.00167)	(0.00183)	(0.00183)	(0.00188)	(0.00185)
Setting 3	-1	3	-2.1	0.2721	-0.1907	-0.1606	-0.1648	-0.1648	**-0.1862**	-0.165
				(8e-04)		(0.00169)	(0.00179)	(0.00181)	(0.00182)	(0.0018)
Setting 4	-1	4	-2.7	0.2606	-0.1826	-0.1432	-0.1488	-0.1488	**-0.1693**	-0.149
				(0.00075)		(0.00168)	(0.00179)	(0.0018)	(0.00182)	(0.00179)
Setting 5	-1	5	-3.2	0.2684	-0.1818	-0.1321	-0.1394	-0.1393	**-0.1604**	-0.1397
				(0.00077)		(0.00178)	(0.00188)	(0.00189)	(0.00191)	(0.00189)
Setting 6	-1	6	-3.75	0.2677	-0.1766	-0.1175	-0.1264	-0.1262	**-0.1472**	-0.1266
				(0.00073)		(0.00175)	(0.00183)	(0.00184)	(0.00186)	(0.00183)
Setting 7	-1	7	-4.3	0.2675	-0.171	-0.103	-0.1133	-0.1131	**-0.1341**	-0.1136
				(0.00079)		(0.00174)	(0.00179)	(0.0018)	(0.00182)	(0.0018)
Setting 8	-1	8	-4.85	0.2678	-0.1652	-0.0887	-0.1005	-0.1003	**-0.1212**	-0.1009
				(0.00073)		(0.00164)	(0.00174)	(0.00174)	(0.00177)	(0.00175)
Setting 9	-1	9	-5.4	0.2686	-0.1592	-0.0748	-0.0881	-0.0878	**-0.1088**	-0.0885
				(0.00068)		(0.00161)	(0.00184)	(0.0018)	(0.00188)	(0.00186)
Setting 10	-1	10	-6	0.2619	-0.1508	-0.0599	-0.0745	-0.0741	**-0.0946**	-0.0749
				(7e-04)		(0.00149)	(0.00174)	(0.0017)	(0.00179)	(0.00177)

Table 1: **Scheme 1**(holding $\overline{Y}$ nearly constant at 0.265) Comparison results of ATE estimation methods. The bold numbers represent the one that is the closet to the ground-truth ATE and the numbers in the parentheses are the standard errors from 100 repetitions (with different seeds).

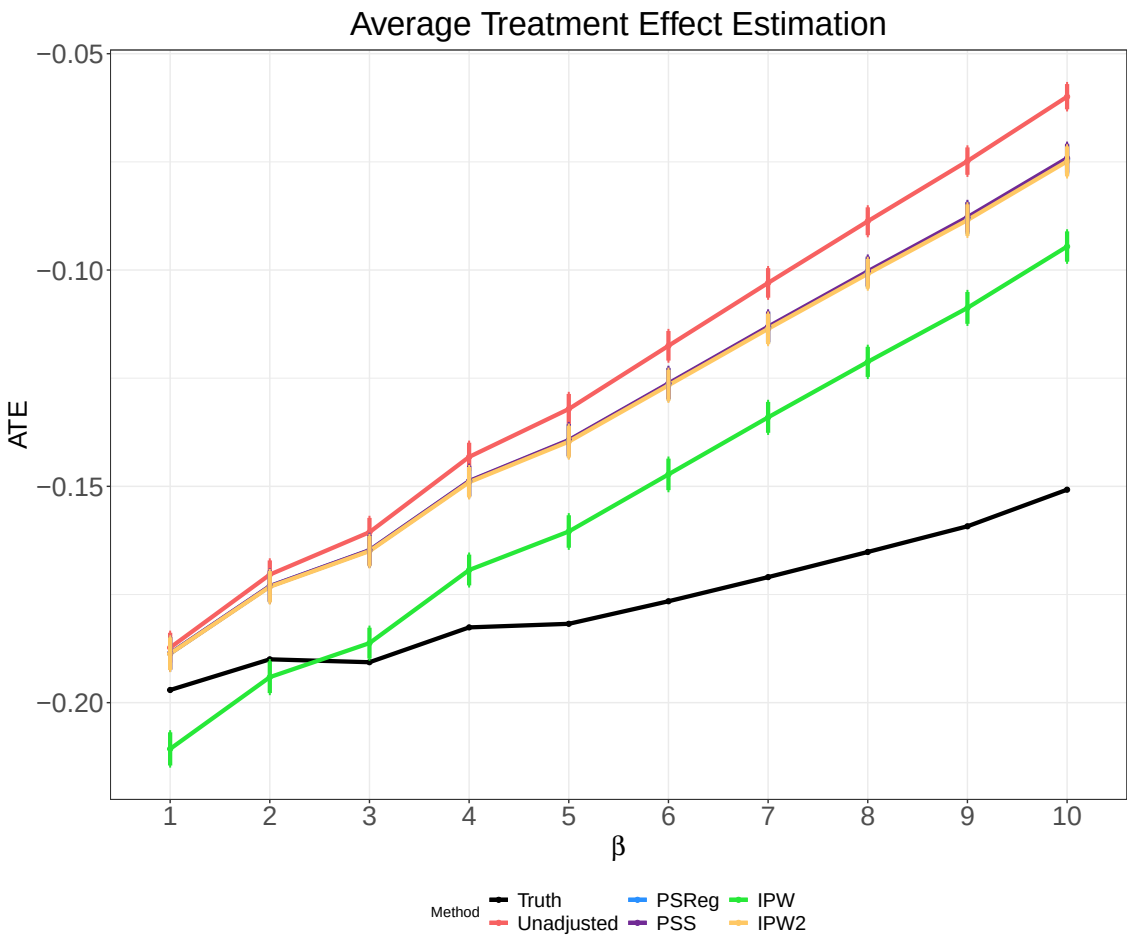

Figure 1: **Scheme 1** (holding $\overline{Y}$ nearly constant at 0.265) ATE Estimation Comparison. The vertical bars are the 95% confidence intervals.

	α	β	γ	$\overline{Y}$	Truth	Unadjusted	PSReg	PSS	IPW	IPW2
Setting 1	-1	1	-1	0.2793	-0.1971	-0.1873	-0.1886	-0.1886	-0.2107	**-0.1887**
				(0.00084)		(0.00176)	(0.00191)	(0.00192)	(0.00196)	(0.00193)
Setting 2	-1	2	-1.5	0.2848	-0.1983	-0.178	-0.1808	-0.1808	**-0.2033**	-0.1809
				(0.00083)		(0.00175)	(0.00186)	(0.00187)	(0.0019)	(0.00187)
Setting 3	-1	3	-2	0.2912	-0.1985	-0.1677	-0.1721	-0.172	**-0.1949**	-0.1722
				(8e-04)		(0.00164)	(0.00175)	(0.00176)	(0.00178)	(0.00176)
Setting 4	-1	4	-2.47	0.3042	-0.1998	-0.1579	-0.1639	-0.1638	**-0.1877**	-0.1641
				(0.00085)		(0.00175)	(0.00192)	(0.00192)	(0.00194)	(0.00192)
Setting 5	-1	5	-2.95	0.3158	-0.199	-0.1463	-0.154	-0.1539	**-0.1786**	-0.1542
				(0.00082)		(0.00174)	(0.00198)	(0.00197)	(0.00202)	(0.002)
Setting 6	-1	6	-3.4	0.3334	-0.1986	-0.1352	-0.1446	-0.1444	**-0.1704**	-0.1449
				(0.00085)		(0.00177)	(0.002)	(0.00198)	(0.00203)	(0.00201)
Setting 7	-1	7	-3.8	0.361	-0.1992	-0.1254	-0.1364	-0.1362	**-0.1642**	-0.1368
				(0.00084)		(0.00183)	(0.00203)	(0.002)	(0.00205)	(0.00204)
Setting 8	-1	8	-4.15	0.3979	-0.1997	-0.1164	-0.1289	-0.1286	**-0.1592**	-0.1292
				(0.00079)		(0.00176)	(0.00196)	(0.00193)	(0.00196)	(0.00195)
Setting 9	-1	9	-4.45	0.4436	-0.1991	-0.1083	-0.1217	-0.1214	**-0.1553**	-0.122
				(8e-04)		(0.00183)	(0.00203)	(0.00198)	(0.00201)	(0.002)
Setting 10	-1	10	-4.5	0.5372	-0.1978	-0.1055	-0.1182	-0.118	**-0.1586**	-0.1186
				(0.00083)		(0.00188)	(0.00204)	(0.002)	(0.00206)	(0.00205)

Table 2: **Scheme 2** (holding ground-truth ATE nearly constant at -0.2) Comparison results of ATE estimation methods. The bold numbers represent the one that is the closet to the ground-truth ATE and the numbers in the parentheses are the standard errors from 100 repetitions (with different seeds).

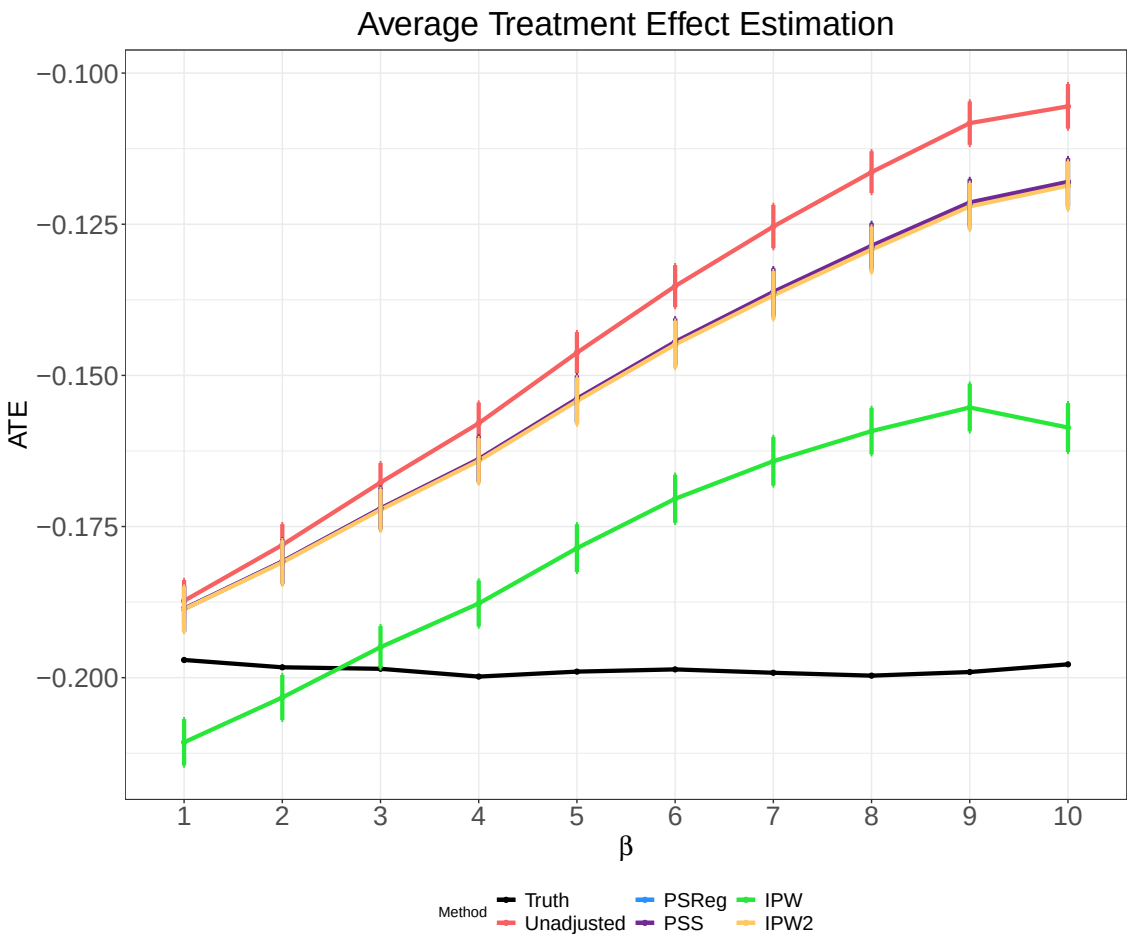

Figure 2: **Scheme 2** (holding ground-truth ATE nearly constant at -0.2) ATE Estimation Comparison. The vertical bars are the 95% confidence intervals.