Statistica Sinica

On Regression Tables for Policy Learning

Comment on a Paper by Jiang, Song, Li and Zeng

Stefan Wager

Stanford University

Policy Learning. The problem of policy learning (or learning optimal treatment regimes) has received considerable attention across several fields, including statistics, operations research, and economics. In its simplest setting, we observe a sequence of independent and identically distributed samples (X_i, A_i, R_i), where $X_i \in \mathbb{R}^p$ is a vector of features, $A_i \in \{-1, +1\}$ is a randomly assigned action, and $R_i \in \mathbb{R}$ is a reward. We then seek to learn a good decision rule $d : \mathbb{R}^p \to \{-1, +1\}$ that can be used to assign actions in the future. Following the Neyman–Rubin causal model [Imbens and Rubin, 2015], we assume potential outcomes $R_i(-1)$ and $R_i(+1)$, corresponding to the reward that the i-th subject would have experienced had it been assigned action -1 or $+1$ respectively, such that $R_i = R_i(A_i)$. We write the conditional average treatment effect as $\tau(x) = \mathbb{E}\left[R_i(+1) - R_i(-1) \,\middle|\, X_i = x\right]$.

Given this setting, the expected reward from deploying a decision rule d is $V(d) = \mathbb{E}\left[R_i(d(X_i))\right]$; we refer to this quantity as the value of d. Furthermore, assuming randomization such that $\{R_i(-1), R_i(+1)\} \perp\!\!\!\perp A_i$, we have [Kitagawa and Tetenov, 2018, Qian and Murphy, 2011]

$$V(d) = \mathbb{E}\left[\frac{1\left(\{A_i = d(X_i)\}\right) R_i}{A_i\pi + (1 - A_i)/2}\right], \quad \pi = \mathbb{P}\left[A_i = 1\right], \tag{01}$$

and it is natural to consider learning a decision rule $\hat{d}$ by maximizing an empirical estimate $\hat{V}(d)$ of $V(d)$ over a class $\mathcal{D}$ of candidate decision rules. Contributions to this problem from the statistics literature, including those of Qian and Murphy [2011] and Zhao et al. [2012], are reviewed by Jiang et al. [2019]; related results from neighboring fields, including extensions to observational studies and multi-action settings, are developed by Athey and Wager [2017], Dudík et al. [2011], Hirano and Porter [2009], Kallus [2018], Kallus and Zhou [2018], Kitagawa and Tetenov [2018], Manski [2004], Stoye [2009, 2012], Swaminathan and Joachims [2015], and Zhou et al. [2018].

Jiang et al. [2019] present a thought-provoking approach to statistical inference for the policy learning problem. They start by observing that the empirical analogue $\hat{V}(d)$ of the objective function in (01) is discontinuous in the decision rule $d(\cdot)$, and so exact asymptotic analysis is complicated. To avoid this difficulty, they propose first solving a surrogate problem with a smooth loss function (they assume all rewards R to be positive),

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{R_i}{A_i \pi + (1 - A_i)/2} \left(-\frac{(A_i + 1)}{2} f(X_i) + \log\left(1 + e^{f(X_i)}\right) \right) \right\}, \tag{02}$$

and then obtain a decision rule by thresholding $\hat{f}$, that is, a rule of the form $\hat{d}(x) = \operatorname{sign}(\hat{f}(x))$. Jiang et al. [2019] show that the loss function used in (02) is Fisher-consistent; this implies that that, under reasonable conditions and if the class $\mathcal{F}$ in (02) is unrestricted, then a regularized variant of their procedure is consistent for the maximizer of the value function $V(\cdot)$ in large samples. These results are also extended to the dynamic decision-making context.

Parametrizing Policy Learning. Relative to existing methods, the main advantage of the approach of Jiang et al. [2019] is that, because the "entropy loss" minimized in (02) is smooth, we can provide an exact characterization of the asymptotic behavior of $\hat{f}$ using classical

	intercept	beta 1	beta 2	beta 3	beta 4	beta 5
point estimate	0.429	−0.177	0.031	−0.022	0.070	0.022
standard error	0.096	0.089	0.069	0.083	0.082	0.078
p-value	**0.000**	**0.047**	0.652	0.794	0.392	0.779

Table 1: Regression table for the optimal linear rules in the simulation design (05), following the entropy learning approach of Jiang et al. [2019]. Here, $n = 4,000$, $p = 5$, standard errors are obtained via the bootstrap, and p-values less than 0.05 are indicated in bold.

second-order theory. Such results are particularly intriguing when we restrict ourselves to a parametric class $f(x) = c + x\beta$, as then we can use the results of Jiang et al. [2019] to quantify the uncertainty in the parameters $\hat{c}$ and $\hat{\beta}$ that underlie the learned decision rule $\hat{d}(x) = \text{sign}(\hat{c} + x\hat{\beta})$.

Jiang et al. [2019] go further still, and propose using a regression table to summarize the uncertainty in $\hat{c}$ and $\hat{\beta}$; Table 1 shows an example based on a simple simulation study described at the end of this note. Looking at Table 1, we may feel inclined to cautiously conclude that the first feature X_1 matters for treatment personalization. Jiang et al. [2019] present a similar table to quantify the value of personalized depression treatments in the context of the STAR*D study [Sinyor et al., 2010], and argue that gender, age, and other features are significant in determining the best treatment options. Such regression tables have the potential to have a large impact on practice as they present information about optimal treatment rules in a familiar, easy-to-read format.

This regression table approach presents a marked departure from the standard approach to policy learning based on utilitarian regret [Manski, 2004]. For example, using the latter approach, Athey and Wager [2017] and Kitagawa and Tetenov [2018] consider the case where the class $\mathcal{D}$ of allowable decision rules has a finite Vapnik–Chervonenkis dimension $\text{VC}(\mathcal{D})$, and show that the policy $\hat{d}$ learned by empirical maximization of the objective (01) over $\mathcal{D}$ has regret

bounded by

$$\mathcal{R}\left(\hat{d}\right) = \mathcal{O}_P\left(\sqrt{\frac{\mathrm{VC}\left(\mathcal{D}\right)}{n}}\right), \quad \mathcal{R}\left(d\right) = \sup\left\{V(d') : d' \in \mathcal{D}\right\} - V\left(d\right). \tag{03}$$

The key distinction between the results of Jiang et al. [2019] that underlie their regression tables and those of Athey and Wager [2017] and Kitagawa and Tetenov [2018] presented above is that the latter do make any optimality claims about the functional form of $\hat{d}$. Rather, they are only focused on high-level properties of $\hat{d}$, in particular the expected reward from deploying it $\hat{d}$.

Interpreting Regression Tables. A major question left open in the above discussion is how the p-values in Table 1 ought to be used in applied data analysis. If a coefficient in the learned rule has a significant p-value, as in Table 1, how should we take this into account in practice? As discussed in Jiang et al. [2019], we should in general not expect the population minimizer of (02) to actually be linear in applications; however, given reasonable model-free assumptions, the confidence intervals above ought to cover the population minimizers

$$\{c^*,\, \beta^*\} = \underset{c,\,\beta}{\mathrm{argmin}} \left\{\mathbb{E}\left[\frac{R_i}{A_i\pi + (1-A_i)/2}\left(-\frac{(A_i+1)}{2}\left(c + X_i\beta\right) + \log\left(1 + e^{c + X_i\beta}\right)\right)\right]\right\}. \tag{04}$$

The question then becomes one of understanding how to interpret c^* and β^* in a general non-parametric setting. A first encouraging result is that, if there is no treatment effect, then the null model minimizes the above loss:

Proposition 1. *If $\tau(x) = 0$ for all $x \in \mathbb{R}^p$ and A_i is randomized, then $\{c^* = 0,\, \beta^* = 0\}$ is a minimizer of the population entropy loss (04).*

In precision medicine, we are often interested in the more subtle question of whether personalized treatment is useful. One might then hope for a result of the following type: if

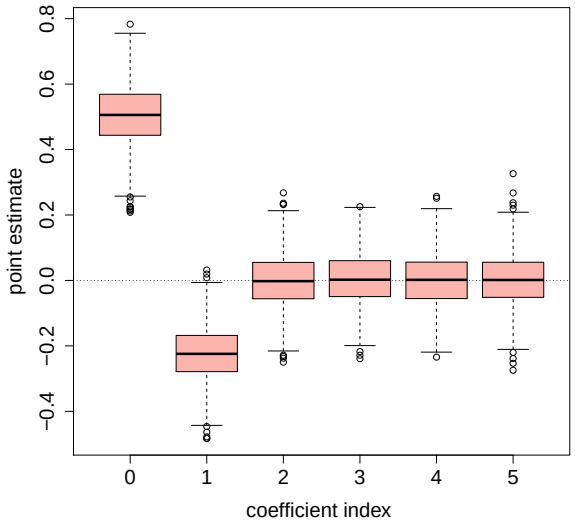

Figure 1: Point estimates for $\hat{c}$ and $\hat{\beta}_1$, ..., $\hat{\beta}_5$ on the design (05), aggregated across 1,000 simulation replications.

the treatment effect is constant, i.e., $\tau(x) = \tau$ for all $x \in \mathbb{R}^p$, then $\beta^* = 0$. However, this is *not* true in general. It is possible to design data-generating distributions with no treatment heterogeneity, but where the minimizer β^* in (04) is nonzero. Furthermore, it is possible to design settings where where $\mathbb{E}\left[\text{Cov}\left[\tau(X),\, X_j \,\middle|\, X_{-j}\right]\right]$ is positive but β^*_j is negative, etc.

A Simulation Study. To investigate the extent to which nonzero β^* may arise in a problem without any treatment heterogeneity, we consider a simple simulation example. We generate data as follows, with $n = 4,000$ and $p = 5$:

$$X_{ij} \overset{\text{iid}}{\sim} \text{Bernoulli}\left(\frac{1}{2}\right), \quad A_i \sim \text{Bernoulli}\left(\frac{1}{3}\right), \quad R_i = X_{i1} + \frac{A_i + 1}{3} + E_i, \tag{05}$$

where the treatment is randomized, and E_i is an exogenous standard exponential random variable. The treatment effect is obviously constant ($\tau = 2/3$), but a simple calculation shows that β_1^* is nonzero. Figure 1 reports numerical results, and we observe that the point estimate for $\hat{\beta}_1$ is in fact systematically negative. What's going on here is that the baseline expected reward $\mathbb{E}\left[R_i(-1) \,\middle|\, X_i\right]$ changes with X_{i1}, and this affects β_1^* even when there is no treatment heterogeneity.

This simulation was also the basis for the results given in Table 1, which presents bootstrap confidence intervals obtained on a single simulation run. When aggregated over 400 simulation replications, these 95% confidence intervals for β_1^*, ..., β_5^* cover 0 with probabilities 22%, 94%, 96%, 94%, and 96%, respectively. The upshot is that any interpretation of p-values such as those in Table 1 is rather delicate, and a significant p-value for β_j^* cannot necessarily be taken as evidence that variable j is needed for designing optimal personalized treatments.

Closing Thoughts. Providing simple and interpretable insights about optimal personalized treatment rules is a challenging task. Existing approaches to policy learning provide utilitarian regret bounds as in (03). These bounds require no assumptions on the functional form of the optimal treatment assignment rule. However, one downside of the utilitarian regret approach is that it does not provide much information about the functional form of good treatment assignment rules—rather, in the tradition of learning theory [e.g., Vapnik, 2000], it only seeks to show that $\hat{d}$ is not much worse than the best rule in the class $\mathcal{D}$.

The discussed paper proposes a contrasting approach based on hypothesis testing that allows for simple summaries. However, as discussed above, the resulting p-values are difficult to interpret. In particular, the fact that $\hat{\beta}_j$ is significantly different from 0 does not necessarily imply that X_j is useful for personalized treatment assignment, or that there is any treatment heterogeneity at all. In a general non-parametric setting, results on Fisher consistency of the

entropy objective do not translate into a simple characterization the limiting parameters β^* of linear policies obtained via entropy learning.

Interpretable, flexible, and robust significance assessment for policy learning remains an important problem. In a recent advance, Rai [2018] built on the empirical maximization approach (03) and proposed confidence sets $\widehat{\mathcal{D}} \subset \mathcal{D}$ for the optimal policy $d^* \in \text{argmax}\{V(d') : d' \in \mathcal{D}\}$; however, these confidence sets have a generic shape (they are obtained by inverting a hypothesis test), and so cannot be summarized in a simple way as in Table 1. Finally, Nie and Wager [2017], Zhao et al. [2017], and others have studied flexible estimation of the treatment effect function $\tau(x)$; however, this statistical task is only indirectly linked to the problem of inference about optimal policies.

References

Susan Athey and Stefan Wager. Efficient policy learning. *arXiv preprint arXiv:1702.02896*, 2017.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1097–1104, 2011.

Keisuke Hirano and Jack R Porter. Asymptotics for statistical treatment rules. *Econometrica*, 77(5):1683–1701, 2009.

Guido W Imbens and Donald B Rubin. *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press, 2015.

Binyan Jiang, Rui Song, Jialiang Li, and Donglin Zeng. Entropy learning for dynamic treatment regimes. *Statistica Sinica, forthcoming*, 2019.

Nathan Kallus. Balanced policy evaluation and learning. *Advances in Neural Information Processing Systems*, pages 8909–8920, 2018.

Nathan Kallus and Angela Zhou. Confounding-robust policy improvement. *Advances in Neural Information Processing Systems*, pages 9289–9299, 2018.

Toru Kitagawa and Aleksey Tetenov. Who should be treated? Empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.

Charles F Manski. Statistical treatment rules for heterogeneous populations. *Econometrica*, 72 (4):1221–1246, 2004.

Xinkun Nie and Stefan Wager. Learning objectives for treatment effect estimation. *arXiv preprint arXiv:1712.04912*, 2017.

Min Qian and Susan A Murphy. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180, 2011.

Yoshiyasu Rai. Statistical inference for treatment assignment policies. *Unpublished Manuscript*, 2018.

Mark Sinyor, Ayal Schaffer, and Anthony Levitt. The sequenced treatment alternatives to relieve depression (star* d) trial: a review. *The Canadian Journal of Psychiatry*, 55(3): 126–135, 2010.

Jörg Stoye. Minimax regret treatment choice with finite samples. *Journal of Econometrics*, 151 (1):70–81, 2009.

Jörg Stoye. Minimax regret treatment choice with covariates or with limited validity of experiments. *Journal of Econometrics*, 166(1):138–156, 2012.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16:1731–1755, 2015.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer Information Science and Statistics, 2000.

Qingyuan Zhao, Dylan S Small, and Ashkan Ertefaie. Selective inference for effect modification via the lasso. *arXiv preprint arXiv:1705.08020*, 2017.

Yingqi Zhao, Donglin Zeng, A John Rush, and Michael R Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Zhengyuan Zhou, Susan Athey, and Stefan Wager. Offline multi-action policy learning: Generalization and optimization. *arXiv preprint arXiv:1810.04778*, 2018.

Algorithm 1 Replication script for simulation results

```
rm(list = ls()); set.seed(1)
# Assume that treatment A is coded as +/- 1.
entropy_treat = function(R, A, X) {
    X.with.intercept = cbind(1, X)
    prob = mean(A == 1)
    loss = function(beta) {
        theta = X.with.intercept %*% beta
        mean(R / (A * prob + (1 - A) / 2) *
                 (-(A + 1) / 2 * theta + log(1 + exp(theta))))
    }
    nlm.out = nlm(loss, rep(0, ncol(X) + 1))
    nlm.out$estimate
}
boot_se = function(R, A, X, B = 100) {
    boot.out = replicate(B, {
        bidx = sample.int(length(A), length(A), replace = TRUE)
        entropy_treat(R[bidx], A[bidx], X[bidx,])
    })
    boot.var = var(t(boot.out))
    boot.se = sqrt(diag(boot.var))
}
n = 4000; p = 5; pi = 1/3
all.results = lapply(1:400, function(idx) {
    A = 2 * rbinom(n, 1, pi) - 1
    X = matrix(rbinom(n * p, 1, 0.5), n, p)
    R = X[,1] + (A + 1) / 3 + rexp(n)
    beta.hat = entropy_treat(R, A, X)
    se.hat = boot_se(R, A, X)
    rbind(beta.hat, se.hat)
})
# For Table 1
beta.hat = all.results[[1]][1,]
standard.errors = all.results[[1]][2,]
pvalues = 2 * pnorm(-abs(beta.hat) / standard.errors)
# For aggregate coverage
zscores = Reduce(rbind, lapply(all.results,
                 function(lll) (lll[1,] / lll[2,])))
round(colMeans(abs(zscores) < qnorm(0.975))[-1], 2)
# For Figure 1
point.estimates.raw = lapply(1:1000, function(idx){
    A = 2 * rbinom(n, 1, pi) - 1
    X = matrix(rbinom(n * p, 1, 0.5), n, p)
    R = X[,1] + (A + 1) / 3 + rexp(n)
    beta.hat = entropy_treat(R, A, X)
    data.frame(est=beta.hat, coef=0:p)
})
```