Regularizing the Forward Pass

Patrick Power Shomik Ghosh

March 1, 2024

Abstract

In certain applied microeconomic settings, it's typical to view one's dataset as the realization of a stratified cluster randomized control trial: treatment is assigned at the cluster level (such as zip code), and controls vary at both the individual and cluster level. Locally, this makes it more likely that observation will be from the same cluster which can increase the variance for estimators which don't account for the clustered nature of the data. We introduce a framework for partialling out non-parametric cluster effects in a way that generalizes least squares and is inherently compositional even under regularization. We provide a python library based on JAX: https://github.com/pharringtonp19/rfp.

Keywords: Causal Inference, Deep Learning

1 Introduction

In many economic contexts, treatment varies at a level above the unit of interest. As a working example, we'll consider the implementation Right to Counsel policy in Connecticut which was initially introduced in a subset of zip codes in January 2022. The Right to Counsel provides tenants facing eviction with access to free legal representation.

Such settings are attractive in practice because they often balance the desire for identification with the aim of providing insight into the general equilibrium effects. They improve over state level comparisons with respect to identification because they are able to exploit within state variation of the treatment. They are more suitable than individual level randomized control trails at capturing the general equilibrium effects because the density of the treatment incentivizes other individuals to respond. For instance, in our working example, the zip code level roll-out of the Right to Counsel allows us to empirically estimate the extent to which landlords pass the associated costs of this policy onto the unhoused.

The statistical aim in this setting is to estimate treatment effects which generalize across individuals in the unobserved zip codes (clusters). That is, we observe the potential outcome $\tilde{Y}_i(1)$ for those in treated zip codes, and would like to say something about that potential outcome for individuals in the control zip codes. The typical approach is to construct controls such that locally, treatment is as good as randomly assigned. Meaning that conditional on the controls, treatment is not related to the potential outcomes in expectation. In many policy settings, including ours, the clusters which adapt the policy are not randomly selected. This necessitates the inclusion of controls that vary at the cluster level in addition to the individual level. When studying the impacts of the Right to Counsel, we include aggregate eviction counts prior to the implementation.

$$x_i = \Big[\underbrace{\text{Month},\ \text{Plaintiff},\ \text{Gender},\ \text{Rent},}_{\text{Individual level}}\ \underbrace{\text{Zip Code Eviction Count}}_{\text{Cluster level}} \Big]$$

The motivating observation on this paper is that cluter level controls together with cluster treatment assignment change the neighborhood structure of the observations. As figure 1 illustrates, they make it more likely that *local* observations within the treatment/control group will be from the same cluster. Allowing for potentially nonparametric cluster effects, this introduces additional variance into the estimation process. In this paper, we introduce an estimation framework that handles this issue in a conceptually similar way to typical deep learning estimation strategies with i.i.d. data.

Deep Learning models, which we base our framework on, are a subset of machine learning models that can be constructed by composing parameterized functions and trained via gradient descent-like methods. Typical models involve composing linear maps with nonlinear activation functions which can be shown to be universal approximators (Hornik et al. [1989]). In recent years, such models have become easier to train as gradients are

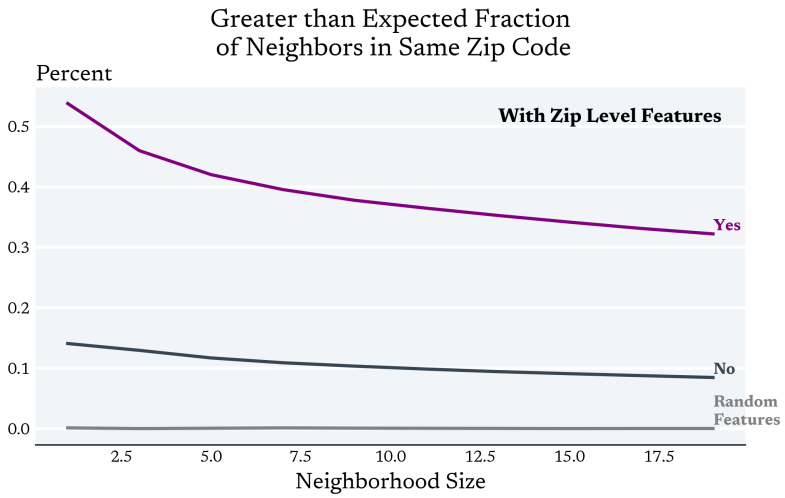


Figure 1: Within the control group, adding cluster level features increases the likelihood that the local neighborhood is overpopulated with observations from the same cluster.

computed "automatically" often via reverse mode automatic differentiation (Griewank and Walther [2008]) and the entire training process can be compiled and scaled across hardware accelerators like GPUs.

While often touted for their ability to drastically improve their performance with the size of the data, we leverage them because they allow us to encode meaningful priors in a controllable fashion. This quasi-Bayesian viewpoint of learning/estimation in the finite sample is in line with modern deep learning works (Jacot et al. [2018], Nagarajan and Kolter [2019], Wilson [2020], Belkin [2021], Zhang et al. [2021]) which emphasize the importance of "understanding the nature of the inductive bias"[1] of the estimator and differentiates this work from recent theoretical econometric work that seeks to reduce the influence of regularization in the estimation process (Chernozhukov et al. [2018]).

Our high level aim is an empirical estimation framework that can seamlessly account for clustered data in a flexible way, has a clear inductive bias, can be easily extended, and is conceptually simple. While the details of the framework are further explained in section 3, the key ideas are captured by the equations below. As indicated in the first line, the framework generalizes ordinary least squares (fitting a linear model to the data). Furthermore, as represented by the clusterMap, the model allows for nonparameteric cluster effects by allowing the parameters of the neural network to vary across clusters during the forward pass.[2] That is by equating clusters of the data to tasks in a meta-learning context, we partial out the cluster effects by allowing the parameters to adopt in the forward pass to the clusters, thereby "locally" correcting for the presence of clustered data in a global way. And finally, as indicated by the fish operator '>=>', the model remains inherently

[1] Belkin [2021] "the properties that make some solutions preferable to others despite all of them fitting the training data equally"

[2] The clusterMap is the cluster specific map that takes the parameters of the neural network that define the conditional expectation function and adopts them to better fit the individual cluster.

compositional, even under regularization.

$$
\begin{aligned}
&\text{linearModel } \text{data} && \text{(OLS)}\\
&\text{linearModel} \circ \text{ identityMap } \text{data}\\
&\text{linearModel} \circ (\text{featureMap } \text{data}) \text{ params}\\
&\text{linearModel} \circ (\text{featureMap } \text{data}) \circ \text{identityMap params}\\
&\text{linearModel} \circ (\text{featureMap } \text{data}) \circ (\text{clusterMap } \text{data}) \text{params}\\
&\text{linearModel} >=> (\text{featureMap } \text{data}) >=> (\text{clusterMap } \text{data}) \text{params}
\end{aligned}
$$

For visual clarity we assume that composition has a higher precendance than function application. Using Haskell-like notation, the fish operator denotes the composition of 'embellished' functions. The functions are embellished in the sense that not only do they return the transformed data in the case of the featureMap, or the cluster specific parameters as in clusterMap, but to each function call they also return a data dependent penalty value which is the regularization term in *regularizing the forward pass*.

The outline of this paper is as follows. First, we dive into why cluster level covariates with cluster level treatment assignment can increase the variance in the estimation process. Second, we introduce our approach for tackling this problem which can be understood in an intuitively simple way: Early Stopping at the Cluster Level. Third, we provide empirical simulation results demonstrating how our approach differs from other methods.

2 Motivation

Our approach to causal inference follows the potential outcome framework as articulated by *Mostly Harmless Econometrics*. Under a selection on observable assumption, that is treatment is as good as randomly assigned, the corresponding conditional expectation function has a causal interpretation - $\mathbb{E}[Y|X, D]$[3] .

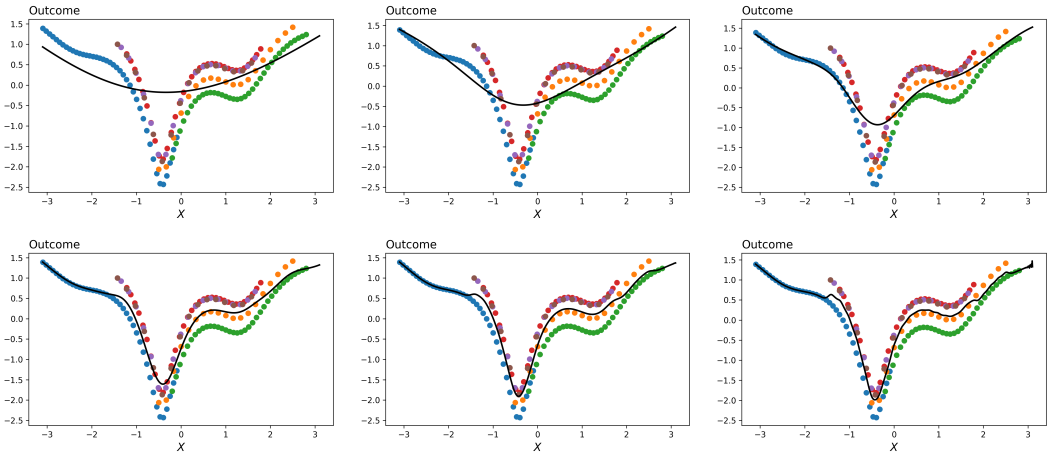
Estimating the condition expectation function can be challenging when we (1) observe only a subset of the clusters in both the treated (control) group, (2) the distribution of covariates differ across clusters, and (3) the distribution of outcomes conditional on covariates differ across clusters. Figure 2 captures the behavior of a typical smoothing estimator in this context. A standard nonparametric model (black line) with a "smoothing" hyperparameter struggles: in order to fit the 'v'-shaped component of the data where there is general

[3]We abuse terminology here by writing "the" conditional expectation function.

$$
\begin{aligned}
\omega &\longmapsto \mathbb{E}[Y|X, D](\omega)\\
\int Y d\mathbb{P}_A &= \int \mathbb{E}[Y|X, D] d\mathbb{P}_A, \quad \forall A \in \sigma(X \times D)
\end{aligned}
$$

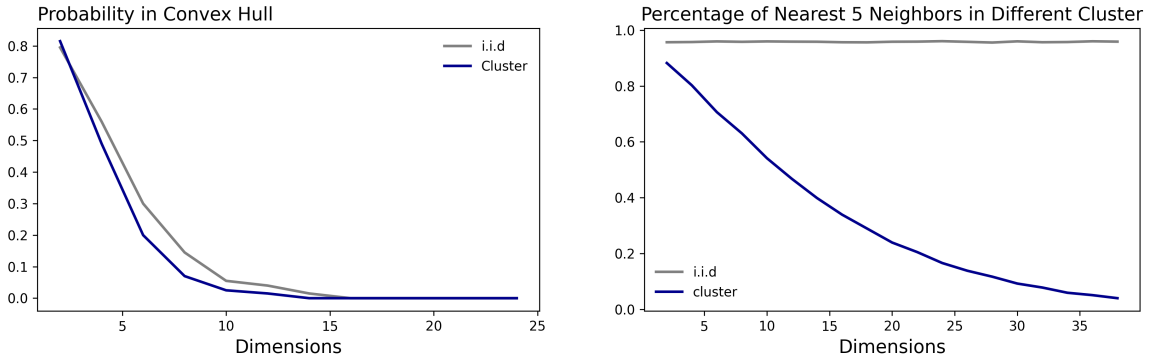
consensus across the clusters, the bandwidth of the estimator must be small. In doing so, though, it over fits the tails. Intuitively what's needed in this context is an estimator that is "locally" aware of the cluster structure of the data.

Figure 2: The Tragic Triad of Clustered Data



Reproduced Here: In this figure, we assume away within-cluster variation. Each dot corresponds to an observation. The different colors highlight the various clusters.

It's interesting to note that these issues are perhaps only magnified as we increase the dimensionality of the data. Extending the work of Balestriero et al. [2021], we illustrate in figure 4a that clustered sampling doesn't change the fundamental issue of learning in high dimensions (extrapolation) so much as it motivates us to reconsider how we go about it, as highlighted in figure 4b.



(a) (b)

Figure 4: (a) Probability that a observation in the test set is in the convex hull formed by the training set (b) Fraction of the five nearest neighbors in a different cluster. Data consists of 25 cluster and 25 observations per cluster. Clusters differ only in the mean which is drawn from an isotropic gaussian distribution: Reproduced Here and Here

These three issues, which together we term the *Tragic Triad of Clustered Data*[4] can

[4]Borrowing the "Tragic Triad" component from Yu et al. [2020]

all occur when treatment is assigned at the cluster level and the conditional expectation function involves controls which vary at the cluster level. In this case, cluster fixed effects are collinear with treatment which means we cannot fit within clusters, but must partial out the potentially nonparametric cluster effects.

We highlight in figure 5 that subject to the typical caveats of hyperparameter tuning, our model fits the 'v'-shaped nature of the data where there is consensus across the clusters without overfitting in the tails." That is, our model, which we formally define in the next section, is implicitly locally aware of the cluster structure of the data.

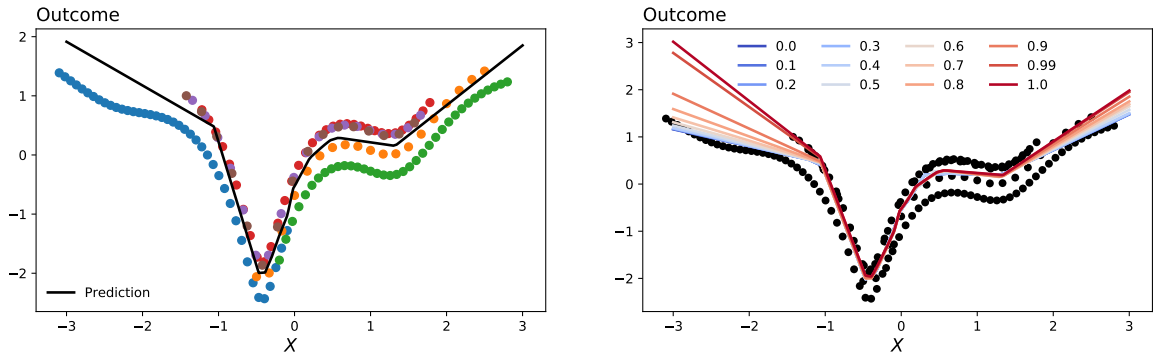


Figure 5

3 Method

3.1 Motivation

Our approach is motivated in part by Domingos [2020] which illustrates how neural networks can be understood within the framework of kernel machines. Specifically, Domingos [2020] shows that in the gradient flow regime, neural networks can be understood as kernel machines where the kernel is formed by integrating the inner product of gradient of the neural network evaluated at the corresponding points of the domain.

Path Kernel

$$K_\theta(x, x') = \int_0^1 \left\langle \nabla_\theta f_x\big(\theta(t)\big), \nabla_\theta f_{x'}\big(\theta(t)\big) \right\rangle dt \tag{1}$$

This insight suggest that one way of partially out the influence of the cluster in a nonparametric manner, is to allow the neural network to fit to each cluster at the end of the training period.

$$K_{\theta_c^*(\theta)}(x, x') = K_\theta(x, x') + \int_1^{1+\varepsilon} \Big\langle \nabla_\theta f_x\big(\theta(t)\big), \nabla_\theta f_{x'}\big(\theta(t)\big) \Big\rangle dt \tag{2}$$

At a high level, it suggests early stopping at the cluster level. Building off the popular meta-learning algorithm MAML (Finn et al. [2017]), we train a neural network via bi-level gradient descent. In the inner level, we allow the parameters of the neural network to adopt to each cluster in parallel. By adding a regularization penalty which differentiates our approach from MAML, the model intuitively learns a representation which has a low prediction loss, but where given a few steps of gradient descent can significantly improve its loss on any cluster. In this way, we partial out the potentially nonparametric effects.

3.2 Description

We can express our objective function as the weighted average of a loss function evaluated at two different parameter values.

$$R(\theta) := \frac{1}{N} \sum_i \alpha L_i(\theta) + (1 - \alpha) L_i(\theta_c^*(\theta) \tag{3}$$

The first term captures the typical squared prediction loss. Here we assume that the outcome is a real-valued, but the same approach works for discrete outcomes as well.

$$L_i(\theta) := (y_i - f(\theta, x_i))^2 \tag{4}$$

The second term contains the same loss function, but the parameter corresponds to the learnt parameter based on k steps of gradient descent on observations from the cluster.

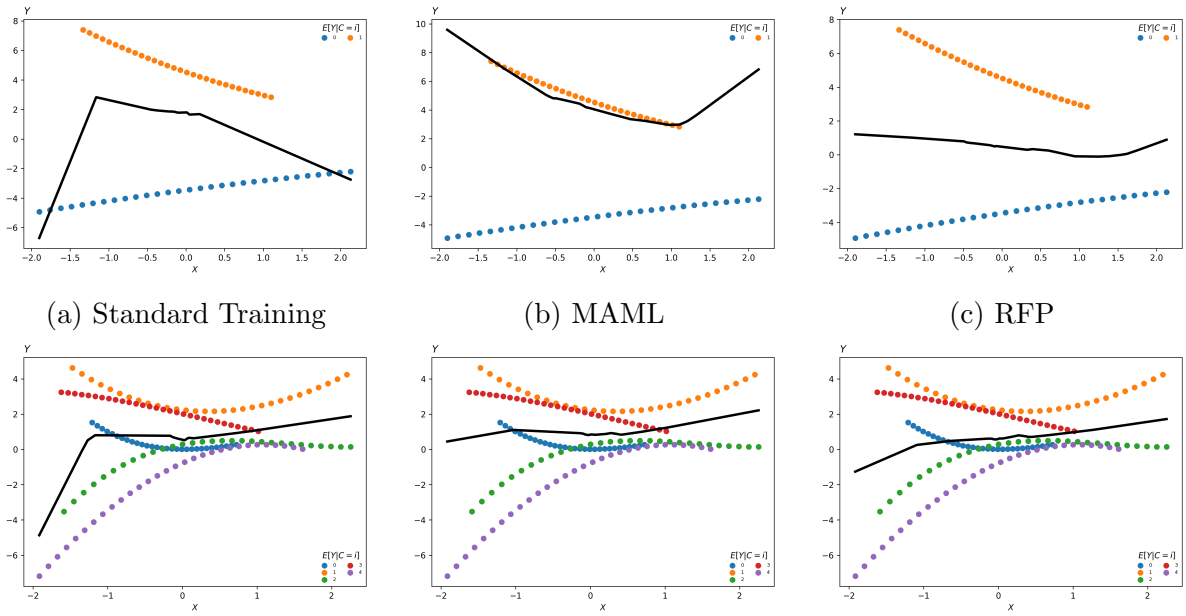
$$\theta_c^*(\theta) := \theta^t - \alpha \nabla_\theta \frac{1}{n_c} \sum_{i \in c} L_i(\theta^{t-1}), \quad \theta^0 = \theta \tag{5}$$

Together, this can be understood as a regularized version of the popular metal-learning algorithm MAML. We highlight the importance of the regularization parameter in the results section.

4 Results

We illustrate the relative importance of our design choices in figure 6. First, standard methods ignore the clustered nature of the data and therefore tend to overfit in the tails where the local observations are from the same cluster. Second, MAML - which is bi-level

gradient descent with out the regularization terms - learns parameters values from which it can "quickly" minimize the cluster specific loss values. This learnt initialization though has no guarantee of in sample performance. As figure 6e illustrates, MAML essentially ignores one of the clusters. Importantly though, this issue isn't apparent when there are multiple clusters with significant overlap. Therefore, the regularization parameter appears important when the clusters don't overlap significantly, which we showed previously can occur with clustered data.



(a) Standard Training (b) MAML (c) RFP

(d) Standard Training (e) MAML (f) RFP

Figure 6: The grey and black dots represent data from separate clusters. Each figure corresponds to fitting a neural network to this data under different training algorithms

5 Conclusion

In this paper we show that clustered treatment assignment together with cluster level controls can increase the variance of standard smoothing estimators. We illustrate that a regularized version of the popular meta-learning algorithm (MAML) is a potentially attractive estimation method in this context in that it partials out the potentially nonparametric cluster effects via bi-level gradient descent. Intuitively, this is akin to early stopping at the cluster level.

References

Randall Balestriero, Jerome Pesenti, and Yann LeCun. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*, 2021.

Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

Pedro Domingos. Every model learned by gradient descent is approximately a kernel machine. *arXiv preprint arXiv:2012.00152*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Andrew Gordon Wilson. The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*, 2020.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.