Generalizing Across Clusters

Patrick Power **Shomik Ghosh** **Markus Schwedeler**

Boston University

Abstract

Rarely is there a pre-established estimator that addresses most of the issues competing for "first-order" importance in observational studies. Because of this, it can be helpful to have econometric methods that are **well-targeted** (i.e. address a specific issue) and **composable** (i.e. the components fit together) so that researchers can adjust their models to their specific context. With this aim in mind, we illustrate that a regularized version of MAML offers a conceptually simply, model-agnostic way to adjust one's estimator for the presence of clustered data. Conceptually, the approach can be understood as a gradient correction that favors early stopping at the cluster level.

The github repository can be found here, and related work by the same authors can be found here.

1 Overview

There is often a difference between the econometrics taught in graduate school courses and the econometrics emphasized in applied seminars. This difference is sometimes mistakenly attributed to a gap in mathematical backgrounds. The better explanation, though, is that the difference exist because in seminars, the data is messy: clusters of individuals receive the same treatment; people drop out of the sample; outcomes get censored; selection into treatment is unknown.

With multiple issues jockeying for "first-order" consideration, it's rare that a pre-existing estimator addresses most of the issues competing for "first-order" importance. Because of this, it can be helpful to have econometric methods that are **well-targeted** (i.e. address a specific issue) and **composable** (i.e. the components fit together) so as to allow researchers to adjust their models for their specific context. With this aim in mind, we illustrate that a regularized version of MAML offers a conceptually simply, model-agnostic way to adjust one's estimator for the presence of clustered data. Conceptually, the approach can be understood as a gradient correction that favors early stopping at the cluster level.

2 The Problem

2.1 Context

To keep things simple, we describe our approach in the specific context of cluster-level randomized control trials where we're interested in estimating treatment heterogeneity.[1] With a binary treatment variable, such a problem can be formulated as below where f the objective is minimized over the treatment and control samples separately.

$$\inf_{f \in \sigma(x)} E\big[(Y - f)^2\big] \tag{1}$$

Such experiments are common in development, education, and health settings because they are (A) generally easier to implement, (B) better adhere to the potential outcome framework[2] and perhaps most importantly[3] (C) allow us to understand the the effects of scaling the treatment.[4]

2.2 Challenge (The Tragic Triad)[5]

Under the potential outcome framework, clustered level treatment assignment can be roughly thought of as forming the treatment and controls groups via random clustered sampling. From an estimation standpoint, this poses a few challenges because in each treatment group:

1. We observe only a subset of the clusters
2. The distribution of covariates can differ across clusters
3. The distribution of outcomes conditional on covariates may differ across clusters

The above issues are perhaps only magnified as we increase the dimensionality of the data

2.3 Addressing the Problem

The central challenge is how to incorporate a cluster indicator in the training phase so that the function adaptively pools information across clusters, without using the cluster indicator in the inference phase. To highlight this, we construct a toy data set where the average within cluster outcome value

[1] Cluster-level randomized control trials are randomized control trials where treatment varies at a level above the unit of interest.

[2] Reduce the chance of spillover effects between treated and non-treated individuals.

[3] See John Lists's book, 'The Voltage Effect' which highlights this importance in great detail

[4] Many large scale studies such as HIE prefer to include many control variables in their regression specification: size of family, age categories, education level, income, self-reported health status, and use of medical care in the year prior to the start of the experiment, kind of insurance (if any) the person had prior to the experiment, whether family members grew up in a city, suburb, or town, and spending on medical care and dental care prior to the experiment

[5] The expression "tragic triad" is taken from Gradient Surgery for Multi-Task Learning

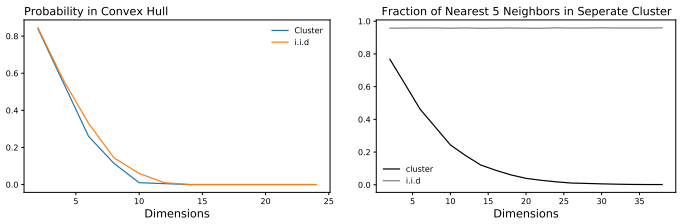


Figure 1: The general pattern that these figures try to highlight is that in order to fit the the ‘v‘-shaped valley in the function, the model overfits the tails of the function –> Reproduced Here

is zero (i.e. adding cluster specific fixed effects would not improve the fit to the data). The central challenge is how “addaptively” share information across clusters. That is, when there are a lot of clusters present, we would intuitively prefer a small bandwidth. When there are few clusters present, we would prefer a larger bandwidth. And of course, we would like to extend this to high dimensions!

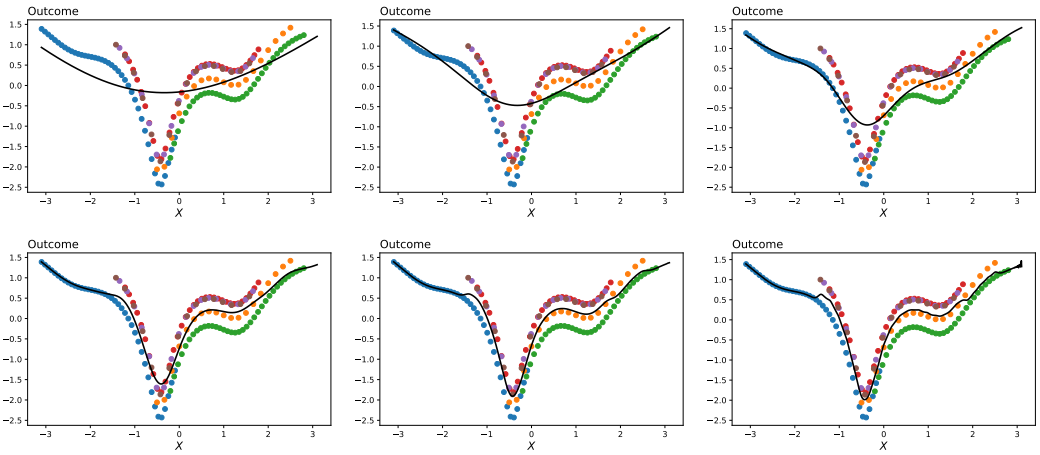


Figure 2: The general pattern that these figures try to highlight is that in order to fit the the ‘v‘-shaped valley in the function, the model overfits the tails of the function –> Reproduced Here

In constrast, subject to the appropriate hyperparameters, our method produces a reasonable estimate.

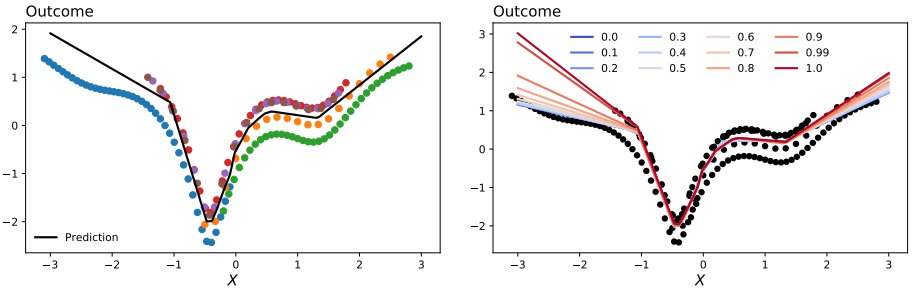


Figure 3: The general pattern that these figures try to highlight is that in order to fit the the ‘v‘-shaped valley in the function, the model overfits the tails of the function –> Reproduced Here

3 Approach

As applied microeconomists, we are accustomed to writing our problem as a bi-level optimization problem so as to better distinguish between the parameters of interest and the nuisance parameters.

In this context, the nuisance parameters are cluster specific parameters that are "fit" during the inner optimization process.

$$\mathcal{L}(\theta) := \sum_c \mathcal{L}_c(\theta), \quad \mathcal{L}_c(\theta) := F(\theta, \theta_c^*(\theta)), \quad \theta_c^*(\theta) := \underset{\theta_c}{\text{argmin}}\, F(\theta, \theta_c)$$

$$\begin{aligned}
\theta^* &= \underset{\theta}{\text{argmin}}\, \mathcal{L}(\theta) \\
&= \underset{\theta}{\text{argmin}} \sum_c \mathcal{L}_c(\theta) \\
&= \underset{\theta}{\text{argmin}} \sum_c F(\theta, \theta_c^*(\theta)) \\
&= \underset{\theta}{\text{argmin}} \sum_c F(\theta, \underset{\theta_c}{\text{argmin}}\, F(\theta, \theta_c))
\end{aligned}$$

With "Classical" under-parameterized models, as in the case of linear regression, F, the clustere-specific empirical loss function is exactly what you would expect.

$$F(\theta, \theta_c^*(\theta)) := \sum_{i \in c} \left(y_i - \theta^T d_i - \theta_c^*(\theta)^T x_i\right)^2$$

With"Modern" over-parameterized models, though, like the ones that we target in this paper, we make the following adjustments

1. We restrict the objective function that is used to implicitly define the cluster specific maps. Without some form of augmentation/regularization, we can lose the learning signal as these models are capable of perfectly interpolating the data.
2. We generalize the above set-up by allowing the cluster specific parameters to be in one-to-one correspondance with the parameters of interest.
3. We add a penalty term to the cluster specific loss function to ensure that adaptation happens in the right space.

Taken together, these three points illustrate that approach is simply a regularized version of MAML. Although, as we highlight through extensive similations, the regularization part is key.

$$\hat{\theta}_c^*(\theta) := \underset{\theta_c}{\text{argmin}}\, G(\theta, \theta_c)$$

$$G(\theta, \theta_c) =$$

Define our empirical cluster parameters in relation to the parameters of interest

$$F(\theta) := \sum_{i \in c} \left(y_i - f(\theta_c^*(\theta), x_i)\right)^2$$

As well as introduce an auxilliary term to the cluster specific loss function:

$$\mathcal{L}_c(\theta) = F(\theta) + H(\text{Path}(\theta, \hat{\theta}_c^*(\theta)))$$

4 Theory

- Highlight why MAML is insufficient: MAML is in some sense overparameterized.
- How does this improve in any way over the bootstrap (i.e. dropping clusters would also act as a gradient correction)
- Formalize that this is early stopping at the cluster level based on the work of Neural Tangent Kernel!
- Some analysis/mention of the scale of the network based on NTK and lazy training paper

4.1 Interpretation

- Show that this is roughly/implicitly equivalent to projected gradient descent (in some sense) – highlight the fragility of the method

5 Conclusion

5.0.1 Regularizing in the Right Space

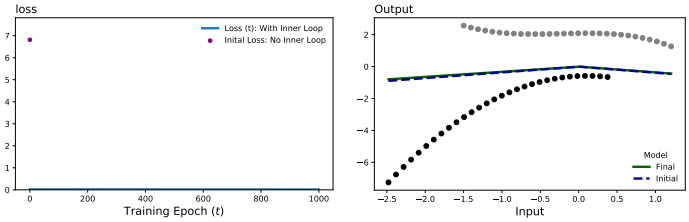


Figure 4: The general pattern that these figures try to highlight is that in order to fit the the ‘v‘-shaped valley in the function, the model overfits the tails of the function –> Reproduced Here

- Repeated composition of the same function can be expressed recursively
- Writing the clusterMap as an implicit function, though, allows for us to interpret

$$\text{Network}_\theta(x) = \text{Linear Map}_{\theta_L} \circ \text{Non-Linear Activation} \circ \text{Linear Map}_{\theta_{L-1}} \cdots \circ \text{Linear Map}_{\theta_1}(x)$$

$$\text{ClusterMap}(\theta) = \text{Update} \circ \text{Update} \cdots \circ \text{Update}(\theta)$$

better ideas[6]

$$\begin{aligned}
&\text{ClusterMap} : \text{Nat} \to (\text{Params} \to \text{Params}) \to \text{Params} \to \text{Params} \\
&\text{ClusterMap } n\ f\ p = \\
&\quad \mid n == 0 \qquad = p \\
&\quad \mid \text{otherwise} \qquad = \text{ClusterMap } (n-1)\ f\ (f\ p)
\end{aligned}$$

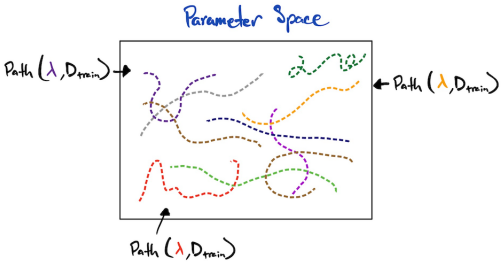
[6] Link

6 Preamble/Caution

6.1 Deep Learning is Different

- Over-parameterization enables interpolation and provides flexibility to select a right interpolating model.
- Thus, the statistical question is understanding the nature of the inductive bias – the properties that make some solutions preferable to others despite all of them fitting the training data equally well.

6.2 Learning is fundamentally a Bi-level Optimization Problem



$$
\begin{aligned}
\theta^*(\lambda, \mathcal{D}_{\text{train}}) &= \underset{\theta \in \text{Path}(\lambda, \mathcal{D}_{\text{train}})}{\text{argmin}} \; \mathcal{L}(\theta, \mathcal{D}_{\text{train}}) \\
\lambda^*(\mathcal{D}) &= \underset{\lambda \in H_{\text{researcher}}}{\text{argmin}} \; \mathcal{L}\big(\theta^*(\lambda, \mathcal{D}_{\text{train}}), \mathcal{D}_{\text{val}}\big) \\
\hat{\theta}(\mathcal{D}) &= \underset{\theta \in \text{Path}(\lambda^*(\mathcal{D}), \mathcal{D})}{\text{argmin}} \; \mathcal{L}(\theta, \mathcal{D})
\end{aligned}
\tag{2}
$$

6.3 Solver

- the solution chosen by the algorithm depends on the specifics of the optimization process

```
gradient_transform = chain(
   cluster_sampling_adj(inner_lambdas),
   clip_by_global_norm(max_norm),
   scale_by_adam(eps=1e-4),
   scale(-learning_rate)
```