On the Interaction Effects Between Prediction and Clustering

Matt Barnes
Carnegie Mellon University

Artur Dubrawski
Carnegie Mellon University

Abstract

Machine learning systems increasingly depend on pipelines of multiple algorithms to provide high quality and well structured predictions. This paper argues interaction effects between clustering and prediction (e.g. classification, regression) algorithms can cause subtle adverse behaviors during cross-validation that may not be initially apparent. In particular, we focus on the problem of estimating the out-of-cluster (OOC) prediction loss given an approximate clustering with probabilistic error rate p_0. Traditional cross-validation techniques exhibit significant empirical bias in this setting, and the few attempts to estimate and correct for these effects are intractable on larger datasets. Further, no previous work has been able to characterize the conditions under which these empirical effects occur, and if they do, what properties they have. We precisely answer these questions by providing theoretical properties which hold in various settings, and prove that expected out-of-cluster loss behavior rapidly decays with even minor clustering errors. Fortunately, we are able to leverage these same properties to construct hypothesis tests and scalable estimators necessary for correcting the problem. Empirical results on benchmark datasets validate our theoretical results and demonstrate how scaling techniques provide solutions to new classes of problems.

1 Introduction

With the increasing prevalence of machine learning solutions, there is a growing concern over the interactions between algorithms in complex systems [1]. Leveraging multiple learning algorithms is a common technique to optimize performance and incorporate structured prior knowledge. For example, most autonomous vehicles benefit from using separate models for perception of traffic lights, object detection and tracking, localization, predicting actor behavior and ultimately planning an optimal trajectory. Although attempting to directly map from visual inputs to control outputs is simpler, this approach is known to achieve inferior performance. Breaking the larger problem into a sequence of smaller problems may be advantageous for many reasons, but it can create additional challenges which must be addressed.

In this paper, we address the class of interaction effects between clustering and prediction algorithms when attempting to estimate the out-of-cluster (OOC) loss. In the self-driving vehicle example, this encompasses pixel and LIDAR point segmentation (i.e clustering tasks) and prediction tasks based on these segmentations (e.g. object type classification, current and future state regression). We observe this is often also a concern in domains including online shopping, medical systems and census statistics, which are further explored in the experimental section.

To elucidate the potential behavior induced by interaction effects between clustering and prediction algorithms, consider the problem of predicting heart disease from a collection of medical records. Each patient may have several records due to multiple hospital visits but it is unlikely we are able to collect multiple records for every patient. Thus, we must find a learner which generalizes well to new patients not in our training set. The typical approach is to match records belonging to the same individual using some record linkage (i.e. clustering) algorithm. Then the records are split by patient into a training and validation sets, such that all records for a single patient end up in either the training or validation set. This provides an unbiased estimate of the learner's error on new patients, i.e. the out-of-cluster loss.

The underlying challenge in this example is that we do not have access to the oracle clustering (i.e. the

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

mapping from medical records to patients), but only a noisy approximation of it from the record linkage algorithm. Even in relatively low-noise domains like medical and census, these algorithms are known to be imperfect [2, 3, 4]. If we instead take the approach of splitting the dataset according to the *approximated* patient clustering, this effectively causes samples to spill across the true training and validation folds. Some samples which should have been grouped with a validation patient may have ended up with a training patient, and vice versa, without our knowledge. In other words, the training and validation sets are no longer conditionally independent, leading to a problem called *dependency leakage* [5]. This allows the learner to overfit to patient-specific features and optimistically biases our OOC loss estimate. For example, if a patient's records are incorrectly clustered and samples are partitioned into both the training and validation sets, the learner is rewarded for predicting whether a patient has heart disease based on their name – which clearly will not generalize to new patients. This overfitting need not be so blatant. The learner may overfit to subtle patterns in a chest x-ray, a form of bias which is hard to identify even by experienced radiologists.

This interaction between clustering errors and a prediction algorithm is particularly dangerous because our learner may appear to be doing well on the validation set, but does far worse when we deploy it in the real world on new patients. Note that this bias is undetectable during standard cross-validation procedures unless an explicit attempt is made to estimate and correct for it, which is the primary focus of this paper. Saeb et al. note that over half of selected medical studies failed to account for any clustering, allowing records for the same patient to occur in both the training and validation datasets, a significant statistical mistake [6].

The contributions and organization of the remainder of this paper is as follows. We begin in Section 2 by formalizing the problem and notation. In Section 3, we present theoretical properties of out-of-cluster prediction loss given an approximate clustering. In Section 4, we demonstrate how these properties can be used to construct a simple hypothesis test for the presence of bias in cross-validation results.

Computational scalability is a significant barrier to estimate bias in cross-validation results, as previous results typically scale $\mathcal{O}(n^3)$ [5]. In Section 5, we systematically alleviate these concerns by proposing function approximation and matrix sketching techniques which have *constant* computational complexity relative to the dataset size n. Interestingly, our matrix sketching technique is able to reduce the number of columns in a key structured matrix, unlike other matrix sketching techniques which typically reduce the number of rows. Note this does not preclude also applying standard matrix sketching techniques.

Finally, we conducted empirical studies on Parkinson's, heart disease, 1994 US Census and Dota 2 video game data, and provide results in Section 6 which demonstrate the practical behavior of interaction effects closely aligns with our theoretical results. Further, we deploy our scalability techniques to previously intractable problem classes, while maintaining similar error levels on smaller problems.

Broader Impacts Accurate estimates of generalization error are necessary for safely deploying machine learning systems in the real world. Many of the domains where the OOC loss is appropriate involve human records (e.g. medical, census) with extreme consequences (e.g. patient misdiagnosis and misguided public policy decisions).

2 Problem Statement

More formally, let $X = x_1, \ldots, x_{n_x}$ be the n_x observed samples, y be the corresponding labels, and $c : \{1, \ldots, n_x\} \to \{1, \ldots, k\}$ be the oracle clustering algorithm which partitions the data into k clusters (e.g. k is the number of patients, n_x is the number of medical records). Our high level goal is to train a prediction algorithm f which generalizes to new clusters, i.e. has low out-of-cluster loss. The leave-one-cluster-out (LOCO) estimator

$$\widehat{\mathrm{Err}}_{\mathrm{LOCO}} = \frac{1}{|c_1^{-1}|} \sum_{j \in c_1^{-1}} \ell(y_j, f(x_j \mid x_{\bar{c}_1^{-1}}, y_{\bar{c}_1^{-1}})), \quad (1)$$

is an unbiased estimator of the OOC loss[1]. Here, $\mathcal{T} = (X_{\bar{c}_1^{-1}}, Y_{\bar{c}_1^{-1}})$ and $\mathcal{V} = (X_{c_1^{-1}}, Y_{c_1^{-1}})$ denote the training and validation sets, where c_i^{-1} and $\bar{c}_i^{-1}$ denote all sample indices belonging and not belonging to cluster i, respectively. In other words, all samples belonging to one cluster form the validation set, and samples from the remaining clusters form the training set. Without loss of generality, we have arbitrarily chosen to leave the first cluster out.

The key question here is: how will errors in the clustering algorithm $\hat{c}$ effect our ability to train and validate the predictor f? By examining the LOCO estimator used to train and validate f, we see that errors in $\hat{c}$ result in noisy training and validation sets $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$, where some samples have flipped between $\mathcal{T}$ and $\mathcal{V}$.

We assume that clustering errors are made probabilistically with independent rate p_0, an assumption

[1] An unbiased estimate of training on $k - 1$ clusters. It is slightly biased compared to training on all k clusters.

similar to one used in analyzing standard supervised learning with noisy class labels [7, 8]. If the clustering algorithm provides uncertainty estimates (e.g. Bayesian methods), we believe it would be possible to incorporate this uncertainty via importance weighting. Further, we consider the unidirectional leakage scenario where samples move from $\mathcal{V}$ to $\mathcal{T}$ to create $\hat{\mathcal{V}}$ and $\hat{\mathcal{T}}$, such that $\hat{\mathcal{T}} \overset{n}{\sim} M_{P_{\mathcal{T}},P_{\mathcal{V}}}(1-p_0, p_0)$, where $M_{a,b}(w_a, w_b)$ denotes the mixture distribution of a and b with weights w_a and w_b and p_0 is the leakage probability (a function of $\hat{c}$'s error). Our results apply to the other unidirectional leakage scenario where samples move from $\mathcal{T}$ to $\mathcal{V}$, and it may be possible to extend them to the bidirectional leakage scenario using similar techniques as [5].

If the clustering is perfect (i.e. $\hat{c} = c$), then $p_0 = 0$. Let e_i be the expected loss at some other $p = {}^{i}/_{n}$ fraction of corrupted samples (we use the notational shorthand $e(p)$ to denote e_{pn}). The expected OOC loss is equivalent to e_0 (i.e. zero dependency leakage, $p = 0$), but we only observe the empirical loss at some $p_0 > 0$. Thus, our specific goals are to characterize the behavior of the interaction effects e and to efficiently estimate e_0 in order to train and validate f.

3 Theoretical Properties

In this section, we present theoretical results on interaction effects between prediction and clustering algorithms. First, we prove that under mild conditions, the sequence of losses $e = e_0, e_1, \ldots, e_n$ is monotonically decreasing due to dependency leakage. Second, under slightly stronger conditions, the sequence will be convex with respect to p. Intuitively, errors in the clustering algorithm allows the prediction algorithm to 'peak' at samples in the validation distribution, which will improve its performance with diminishing returns. Monotonicity has previously been conjectured, but never proven, and the conditions where it holds were uncertain. To the authors knowledge, no previous work has discussed whether e is convex.

We say a learner f is optimal under its training distribution if

$$f(\cdot|\mathcal{T}) \in \arg\min_{f \in \mathcal{F}} \mathbb{E}_{x,y\sim P_{\mathcal{T}}} \ell(f(x), y). \tag{2}$$

Generally speaking, this tends to be true for large $|\mathcal{T}|$, small model complexity of $\mathcal{F}$ or sufficient regularization in ℓ. This does not imply f is overfit to the training set, but in fact that it generalizes well across $P_{\mathcal{T}}$.

Theorem 3.1. *The sequence $e_0, e_1, \ldots, e_n$ is monotonically decreasing if f is optimal under its training distribution.*

Proof. See Appendix A.1. □

Remark This theorem implies that the interaction will always *optimistically* bias our cross-validation results. This is in fact the most dangerous type of bias, as our heart disease classifier will perform well on the off-line hold-out set, but then perform worse when we deploy it in the real world on new patients or at new hospitals.

If f is not optimal among $\mathcal{F}$, it is possible to construct counterexamples such that $e_0, \ldots, e_n$ is not monotonically decreasing.

In our second theoretical result, we show that the expected loss is convex with respect to the strength of interaction effect p. Let $\ell_P(f) = \mathbb{E}_{x,y\sim P}\ell(f(x), y)$ be the expected loss of the learner f under distribution P. Then the following theorem holds.

Theorem 3.2. *The sequence $e_0, e_1, \ldots, e_n$ is convex if f is optimal under its training distribution and $\ell_{P_{\mathcal{T}}}$ and $\ell_{P_{\mathcal{V}}}$ are strictly convex and differentiable over f.*

Proof. See Appendix A.2 □

Strictly convex and differentiable loss functions hold for a wide class of problems, including support vector machines and linear or ridge regression.

Remark The convexity of e compounds the monotonic behavior in Theorem 3.1, as it implies that even a small amount of error in our clustering $\hat{c}$ can cause large amounts of cross-validation bias in f.

In Section 6, we empirically demonstrate both these properties hold on all examined datasets.

4 Hypothesis Testing

A principal question for data scientists is whether an interaction effect exists between their clustering and prediction algorithms. Here, we show how to use the theoretical properties from Section 3 to quickly construct a two-sample t-test for dependency leakage, which avoids the complexity of constructing an estimator for the OOC loss $\hat{e}_0$.

Consider the alternative hypothesis $H_a : e_0 > e(p_0)$, where $p_0 > 0$ is the unknown leakage probability and e_0 is the OOC loss with zero leakage (i.e. no interaction effect). By Theorem 3.1, we can use a one sided test because $e(p_0) \geq e(p_n)$. First, form $n_{\mathcal{T}}$ training folds each of size n' from $\hat{\mathcal{T}}$. Additionally, form $n'_{\mathcal{T}}$ training folds of size n' and $n_{\mathcal{T}} + n'_{\mathcal{T}}$ validation folds of size $n_{\mathcal{V}}$ from $\hat{\mathcal{V}}$.

Train and validate f on the disjoint $n_{\mathcal{T}} + n'_{\mathcal{T}}$ training folds and corresponding validation folds. Let $z = z_1, \ldots, z_{n_{\mathcal{T}}}$ and $z' = z'_1, \ldots, z'_{n'_{\mathcal{T}}}$ be the validation loss of f trained on the folds from $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$, respectively. Let $\bar{z}$ and $\bar{z}'$ be the mean of these two sequences. Then

$$\bar{z} - \bar{z}' \sim N(e(p_0) - e(p_n), \sigma^2(\bar{z}) + \sigma^2(\bar{z}'))$$

and the two-sample t-test statistic is

$$T = \frac{\bar{z} - \bar{z}'}{\sqrt{\frac{s_1^2}{n_{\mathcal{T}}} + \frac{s_2^2}{n'_{\mathcal{T}}}}} \tag{3}$$

where s_1^2 and s_2^2 are the sample variances of z and z', respectively.

Rejecting the null hypothesis $H'_0 : e(p_0) \leq e(p_n)$ when $T > t_{1-\alpha,v}$ is a level α test, where $t_{1-\alpha,v}$ is the critical value of the t-distribution with v degrees of freedom. Further, by Theorem 3.1 and Theorem 3.2, $e(p_0) \neq e(p_n) \Rightarrow e_0 \neq e(p_0)$ so long as $p_0 > 0$. Thus, rejecting the null hypothesis $H_0 : e_0 \neq e(p_0)$ when $T > t_{1-\alpha,v}$ is also a level α test.

There are two takeaways to consider when using this test. The first powerful property is that it does not require actually knowing the clustering error or leakage probability p_0 a priori, only that it is not perfect (a very weak assumption). Second, the Type II error rate of this test largely depends on the convexity of e. If $p_0 < 0.5$ and e is linear, then $e(p_0) - e(p_n) > e_0 - e(p_0)$ and the Type II error rate will actually be *lower* than if we could directly test $e_0 \neq e(p_0)$. Conversely, the Type II error rate becomes larger as e becomes more strongly convex.

5 Scalable Estimators

Existing asymptotically unbiased estimators for the OOC loss are limited by their need to solve a linear system of n variables, where n is the size of the bootstrap training set [5]. In this section, we present two approaches for dramatically improving the computational efficiency of the unidirectional, known p_0 variant of the Binomial Block Bootstrap (B3) estimator, which is the core method of other variants.

Algorithm 1 B3: Binomial Block Bootstrap

```
procedure B3($f, \hat{\mathcal{T}}, \hat{\mathcal{V}}, p, n', t$)
    $\bar{b} \leftarrow \vec{0}$
    for $p_i$ in $p$ do
        $p' \leftarrow \frac{p_i - p_0}{1 - p_0}$
        for $j \leftarrow 1$ to $t$ do
            $\mathcal{T}'_j \overset{n'}{\sim} M_{\hat{\mathcal{T}},\hat{\mathcal{B}}}(1 - p', p')$
            $\mathcal{V}'_j \leftarrow \hat{\mathcal{V}} \setminus \mathcal{T}'_j$
            $\hat{b}_i \leftarrow \frac{1}{|\mathcal{V}'_j|} \sum_{(x,y) \in \mathcal{V}'_j} \ell(y, f(x \mid \mathcal{T}'_j))$
            $\bar{b}_i \leftarrow \bar{b}_i + \frac{\hat{b}_i}{t}$
        end for
    end for
    $A_{ij} \leftarrow \mathbb{P}(\text{Binomial}(n', p_i) = j)$
    $\hat{e}, residual \leftarrow A(A^{\intercal}A)^{-1}A^{\intercal}\bar{b}$
    return $\hat{e}_0, residual$
end procedure
```

We begin by recapping the Binomial Block Bootstrap (B3) estimator for the OOC loss, shown in Algorithm 1 [5]. The method leverages the fact that a resample with replacement from $\hat{\mathcal{T}}$ can be written as a binomial expectation over e (see row 1 of Eq. (4)), by definition of the binomial distribution. Second, by adding additional corruption into $\hat{\mathcal{T}}$ in the form of samples from $\hat{\mathcal{V}}$, it effectively increases the clustering error p_0 to p_1 (see row 2 of Eq. (4)). Repeated operation of these principles allows constructing the fully defined linear system

$$\begin{array}{cc} & \begin{array}{ccccc} 0 & 1 & \cdot & \cdot & n \end{array} \\ \begin{array}{c} p_0 \\ p_1 \\ \cdot \\ \cdot \\ 1 \end{array} & \begin{pmatrix} \leftarrow \text{Binomial pmf} \rightarrow \\ \cdot \\ \cdot \\ \cdot \\ \leftarrow \text{Binomial pmf} \rightarrow \end{pmatrix} \end{array} \begin{pmatrix} \\ e \\ \\ \end{pmatrix} = \begin{pmatrix} \\ b \\ \\ \end{pmatrix} \tag{4}$$

$$A(p_0) \qquad\qquad e \quad = \quad b$$

where matrix $A \in \mathbb{R}^{m \times (n+1)}$ is defined by

$$A_{ij} = \mathbb{P}(\text{Binomial}(n, p_i) = j). \tag{5}$$

Vector b is formed by the average empirical loss of repeatedly sampling with replacement from $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ at m increasing levels of corruption $p_0, p_1, \ldots, 1$. The insight of this method is that by *increasing* dependency leakage by further mixing $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ and thus increasing p from p_0 towards 1, one can extrapolate the loss at zero clustering error, using the structured matrix A. Although the B3 estimator is asymptotically unbiased, solving the linear system has $\mathcal{O}(n^3)$ cost, and forming the loss estimate b has $\mathcal{O}(n)$ computational cost. If the prediction algorithm f has an expensive training procedure (e.g. deep neural networks), the latter term may outweigh the former due to a large fixed constant.

5.1 Basis Function Approximation

Perhaps the most straightforward approach to scaling these estimators is through function approximation, which also conveniently provides a natural form of regularization. We parameterize e by a set of s basis functions $\psi_1, \ldots, \psi_s$, such that

$$e_i = \xi_1\psi_1(i) + \xi_2\psi_2(i) + \ldots + \xi_s\psi_s(i) \tag{6}$$

where $\xi_1, \ldots, \xi_s = \xi \in \mathbb{R}^s$ are the s parameters. Then $e = \Psi\xi$ where $\Psi \in \mathbb{R}^{(n+1)\times s}$ is the matrix of basis values.

Instead of solving the linear system $Ae = b$, where $A \in \mathbb{R}^{m\times(n+1)}$ and we choose $m \geq n$, we can now solve $A'\Psi\xi = b'$ where $A' \in \mathbb{R}^{m'\times(s+1)}$ and we choose $m' \geq s$. Note the size of this system no longer depends on the number of samples n. Instead, it depends on the number of parameters in our approximation of e, which will be a fixed constant. This new linear system is well behaved, depending on the choice of basis function ψ.

Theorem 5.1. *Let $\psi_0, \ldots, \psi_s$ be a set of s unisolvent, bounded and continuous functions over $[0, 1]$ and let*

$$e_i = \xi_0\psi_0\left(\frac{i}{n}\right) + \ldots + \xi_s\psi_s\left(\frac{i}{n}\right).$$

Then $A\Psi$ is invertible as $n \to \infty$.

Proof. See Appendix A.3. □

5.2 Matrix Sketching

Second, we propose a new matrix sketching technique which reduces the number of columns in the structured matrix A. Unlike typical matrix sketching techniques, which reduce the number of rows, we are able to reduce the number of columns and thus the dimensionality of the solution e by leveraging the structure in A and properties of e from Theorem 3.1. After reducing the number of columns, one could further apply standard matrix sketching techniques to also reduce the number of rows. Our algorithm guarantees recovering e_0 within a linear factor of the true value.

Consider the setting where $m \leq n$ and the system $Ae = b$ is underdetermined. This is especially relevant for large datasets, where it is computationally infeasible to sample at $m > n$ levels of leakage or perhaps even solve for n unknowns. Let $S \in \mathbb{R}^{m\times(k+1)}$, $m > k$ be our sketching matrix, where S is formed such that the first column of S equals the first column of A, i.e. $S_0 = A_0$. Partition the remaining n columns of A into k sets, for example using k-medoids or simply grouping adjacent columns together (since by the definition of A, these will be close together). Let $r : \{0, \ldots, n\} \to \{0, \ldots, k\}$ be the resulting partition, where $r(0) = 0$ is the singleton partition of the first column. Finally, form the remaining columns of S from the medoids of the $k + 1$ sets. Each column in A is within an ϵ-ball of at least one column in S, i.e.

$$\epsilon = \max_{i\in\{0,\ldots,n\}} \|A_i - S_{r(i)}\|$$

Theorem 5.2. *Let e' be the solution to the sketched system $Se' = b$ and s be the first row of s^{-1}. The error between the true and sketched solution is bounded by*

$$|e'_0 - e_0| \leq \epsilon n \|s'\| e_0. \tag{7}$$

Proof. See Appendix A.4 □

5.3 Connection to Bézier curves and Bernstein polynomials

The B3 estimator in Eq. (4) has close ties to the Bernstein basis and Bézier curves which have not previously been realized. Notice that each column of A corresponds to a Bernstein basis function evaluated at at $p_0, \ldots, 1$. Thus, the B3 estimator is equivalent to solving for the Bernstein coefficients or Bézier control points e, where the system is constructed through the B3's bootstrapping process. A more detailed mathematical connection is provided in Appendix B.

6 Empirical study

Finally, we conducted an empirical, finite-sample study which validates the theoretical properties in Section 3 and demonstrates the computational speed-ups provided in Section 5. For comparison, we consider three benchmark estimators for the OOC loss – IID, LOCO and the B3 estimator with a fourth order trend filter and monotonic regularizers (T4+mono). The latter is the empirical state-of-the-art method, though suffers from computational scalability issues. IID is the typical cross-validation split, where samples are uniformly randomly split into training and validation sets, which does not account for the latent clustering. LOCO is the leave-one-cluster-out estimator described in Eq. (1) using an approximated clustering $\hat{c}$ with an error of $p_0 = 0.1$.

In all experiments, we used a linear SVM as the predictor f. This is a best-case scenario, as interaction effects depend on the predictor f's ability to overfit to mistakes from the clustering algorithm $\hat{c}$. Thus, as the complexity of the predictor class increases, the interaction effect worsens.

Note that in order to compute the error of our estimators, we are required to use a dataset where the oracle clustering is indeed available. For many of these experiments, we used data collected in very controlled settings to guarantee no clustering error in the ground truth. In more practical scenarios, this information would not be available. We formed the training and validation sets by splitting the approximate clusters according to Section 2, and we controlled clustering errors by flipping samples from $\mathcal{V}$ to $\mathcal{T}$ with uniform, i.i.d. rate according to p_0. Complete experimental details are provided in Appendix D.

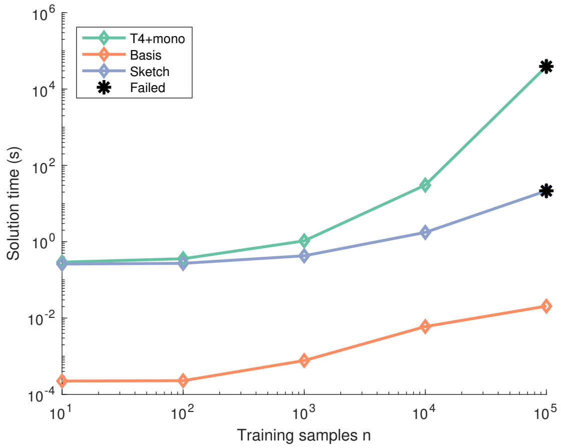


Figure 1: Computational scalability results on synthetically generated datasets. Our methods (Sketch, and in particular, Basis) are significantly faster than existing methods (T4+mono). "Failed" indicates the SDPT3 solver failed to find an accurate solution.

Table 1: Computational timing results demonstrate our methods, and in particular the basis function approximation technique, are significantly faster than the previous state-of-the-art B3 estimator with fourth order trend filter and monotonicity constraint (T4+mono). Results shown in seconds.

	Method		
Dataset	T4+mono	Sketching	Basis
1994 US Census	0.5662	0.4059	**7.822e-5**
Heart	0.5847	0.4105	**6.582e-5**
Parkinson's	0.6194	0.4338	**2.043e-5**
Dota 2	1.0965	0.4678	**1.946e-5**

Computational Scalability The proposed approximation techniques, and especially the basis function approximation technique, are faster than existing OOC estimators and are tractable on larger problem classes. To compare performance across a large range of dataset sizes, we generated increasingly large synthetic training sets and compared solution times in Section 6. All methods used only 10 corruption levels (i.e. the number of rows in A), the smallest reasonable number required to find an accurate solution. We observed that increasing the number of rows in A exponentially increased solution times. Thus, these results are likely the largest datasets appropriate for existing methods. In particular, notice that the solver failed to find accurate solutions on the largest problem class for all methods except for with the basis approximation technique.

Timing results on real world datasets (described in the following sections) are reported in Table 1. Similarly, we find the basis approximation technique is the fastest by several orders of magnitude.

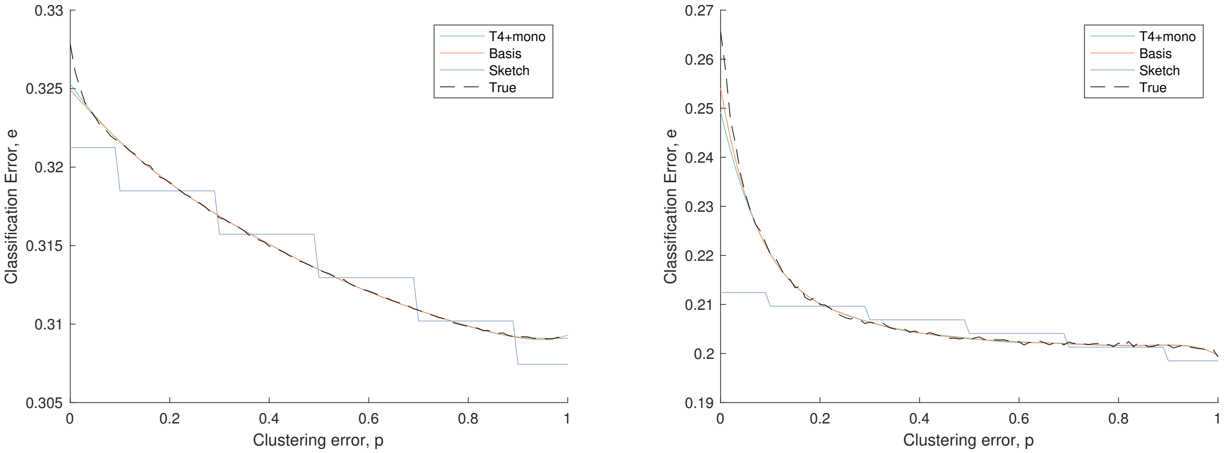
Constrained linear programs (e.g. T4+mono, sketching) were solved using SDPT3's infeasible path-following algorithm, for unconstrained linear systems we took advantage of fast QR solvers (a major reason the basis method is so efficient). All optimizations were performed using an Xeon Gold 6152 CPU @ 2.10GHz and 754 GB RAM. We found that T4+mono, and to a lesser extent, the sketching approximation, required the majority of this memory for the largest problem classes.

Interaction Characteristics Fig. 2 demonstrates that interaction effects between the clustering and prediction algorithm cause the cross-validation error e to decay monotonically and convexly, as predicted by Theorem 3.1 and Theorem 3.2. This visually demonstrates the expected adverse behavior – if our clustering algorithm makes even a few mistakes, we may think our predictor has a low error rate, but when we deploy it in the real world on new clusters, it will perform far worse. Empirically, the interaction biases cross-validation results by upwards of 25%, but our methods are largely able to correct for this bias. Note our methods not only recover the true OOC loss e_0, but also the entire curve $e_1, e_2, \ldots, e_n$.

Estimator Error Finally, we empirically test whether the approximations introduce any additional error into the OOC estimate. Fig. 3 shows that the approximations perform comparably to the previous state-of-the-art T4+mono estimator, at significantly reduced computational cost. The specific experiments are briefly described below, see Appendix D for details.

Parkinson's In the first experiment, we attempted to predict whether a patient has Parkinson's disease based on multiple voice recordings featurized according to doctor specifications [9]. Here, each cluster corresponds to an individual, and each cluster contains multiple voice recordings. The OOC error corresponds to the ability to predict Parkinson's on new individuals not in the training set.

Heart Disease In the second experiment, we use medical records from four hospitals in Switzerland; Hungary; Cleveland and Long Beach, USA [9]. Given a patient medical record, including vital signs and demographics, the task is to train a heart disease classifier which performs well at new, previously unseen hospitals. Since we do not have access to multiple records per patient, we instead choose the related task of generalizing across hospital clusters.



(a) 1994 US Census

(b) Heart Disease

Figure 2: Empirical results show the loss is indeed convex and monotonically decreasing, validating our theoretical results in Section 3. Note our methods are able to recover the full loss in addition to the true OOC loss e_0. Plots for the remaining experiments are provided in Appendix C
.

1994 US Census In the third experiment, we consider the issue of machine learning bias against certain populations in the 1994 US Census due to interaction effects [9]. Given a person's occupational, educational and demographic information, our task is to predict whether a person's income is greater than US$50k per year (finer resolution income data cannot be publicly disclosed). In particular, we wish to train a classifier which performs well across people from all origin countries. We arbitrarily chose Indian and Canadian immigrants as our leave-one-out clusters, and natural born citizens, Salvadoran, German, Mexican, Filipino and Puerto Rican immigrants as our training set[2].

Our results, presented in Fig. 3c, validates our claim that interaction effects can bias our learner against certain populations. The SVM classifier learns attributes specific to the corrupted samples which spilled from the validation set into the training set, even though they do not generalize to all immigrants.

Dota 2 In the final experiment, we attempt to predict the winner of a Dota 2 video game based on the heroes each team selects at the beginning of the game. This is equivalent to learning an undiscounted value function for a binary, sparse reward function in reinforcement learning. Here, clusters correspond to the type of game played, and we wish to learn a predictor f which generalizes across new game types.

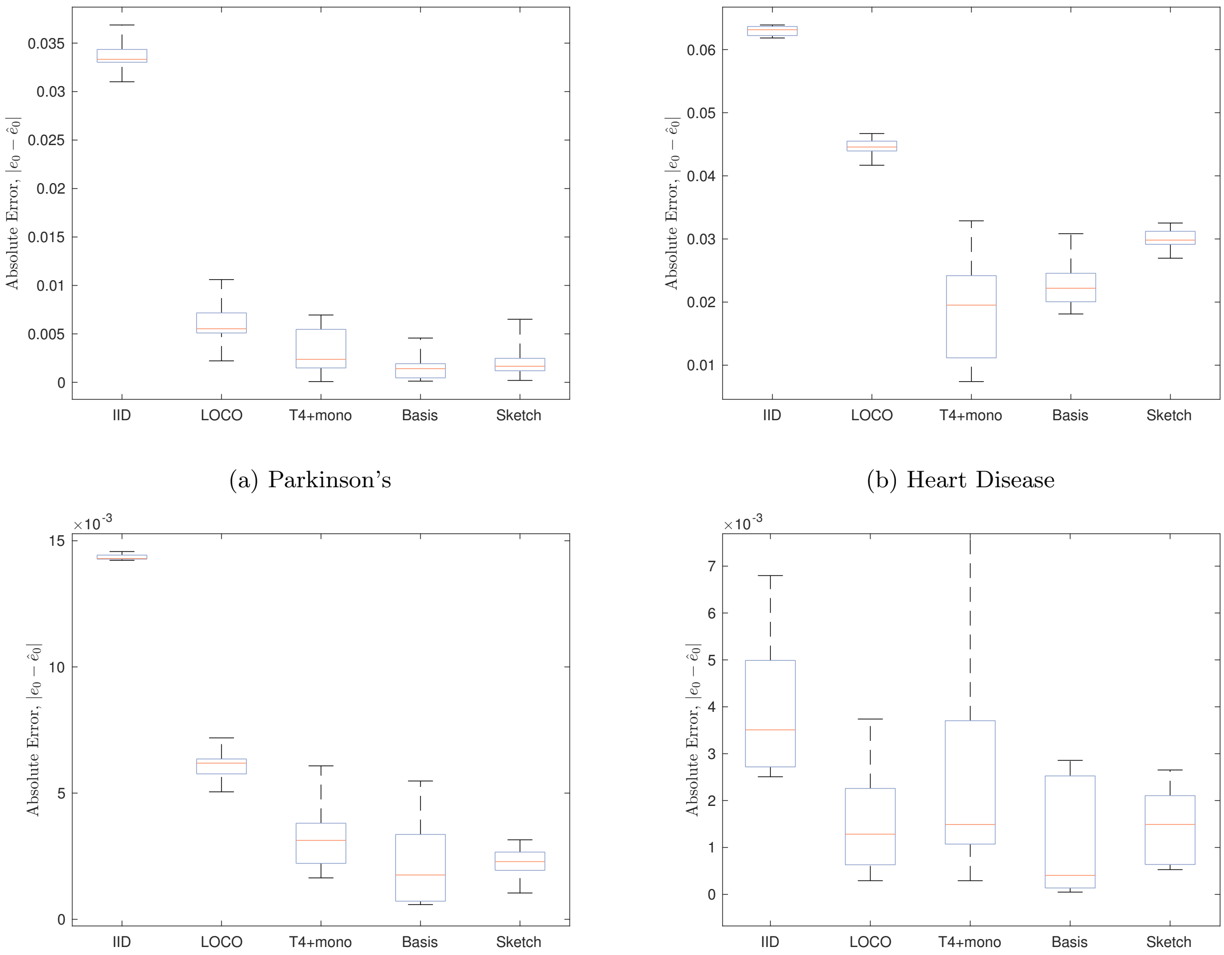
7 Previous Work

Previous work has studied various aspects of learning with dependent data, beginning with the necessity of independence for the naive bootstrap [10]. Subsequent work has proceeded along two directions: most prominently for time-series data, but also for cluster data. In time-series data, a stochastic process defines the data dependency, which usually decreases over larger time intervals [11, 12]. The common approach to limiting dependency and thus controlling estimator bias and variance is to form blocks of data which are sufficiently spaced in the time domain [13].

In the clustering setting, bootstrap methods have been proposed for a variety of problem formulations, roughly categorized into model-based and model-free methods [14]. The first, model-based line of work directly models the within-cluster error correlation, a relatively strong data assumption. The second, model-free line of work performs post-estimation bias-correction for least squares [15], small or unbalanced number of clusters [16] and non-linear settings [17]. Other authors have shown asymptotic analysis results for the residual bootstrap [18], randomized cluster bootstrap, two-stage bootstrap [19] and multi-way bootstrap [20, 21]. The fundamental difference in our work is we do not assume samples in different clusters $\hat{c}$ are independent.

In Section 5, we built off the work of [5], who introduced the first asymptotically exact estimator of the OOC loss with clustering errors. However, they failed to characterize the behavior of the interaction effect,

[2] We left out two clusters due to a small number of immigrants from some countries in the dataset.



(a) Parkinson's

(b) Heart Disease

(c) 1994 US Census

(d) Dota 2

Figure 3: Estimating the OOC loss e_0. Our function approximation and novel matrix sketching techniques perform comparably to existing methods at significantly reduced computational cost.

and their estimator scaled $\mathcal{O}(n^3)$.

8 Conclusion

We argued that interaction effects between clustering and prediction algorithms can cause dangerous and elusive behavior when estimating the out-of-cluster loss in machine learning systems. We theoretically characterized when and how this interaction behavior is exhibited, and demonstrated these properties hold in practice on all examined datasets. In particular, we showed the out-of-cluster loss bias is convex and monotonically decreasing – implying that even a small clustering error can significantly and optimistically bias cross-validation results. Further, these theoretical properties are necessary to construct the statistical hypothesis test in Section 4, an important practical takeaway to detect for OOC bias. Our newly introduced estimators are able to correct for this bias at significantly reduced computational cost compared to existing estimators, making the proposed approaches scalable to a wide range of practical applications.

The interaction between clustering and prediction algorithms is one common instance of an interaction effect. [1] discussed several other issues in complex machine learning systems, including hidden feedback loops and undeclared data dependencies, which may warrant further exploration.

Acknowledgements

We thank Kin Gutierrez Olivares for the insightful connection to Bézier curves. This work was partially supported by DARPA (FA8750-14-2-0244 and FA8750-17-2-0130) and AFRL (FA8750-17-2-0212). Matt was partially supported by an NSF GRFP.

References

[1] D Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young. Machine Learning : The High-Interest Credit Card of Technical Debt. *NIPS 2014 Workshop on Software Engineering for Machine Learning (SE4ML)*, 2014.

[2] Rebecca C. Steorts, Rob Hall, and Stephen E. Fienberg. A Bayesian Approach to Graphical Record Linkage and Deduplication. *Journal of the American Statistical Association*, 111(516):1660–1672, 2016.

[3] William E. Winkler and Yves Thibaudeau. An application of the Fellegi-Sunter model of record linkage to the 1990 US decennial census. Technical report, U.S. Census Bureau, 1990.

[4] William E. Winkler. Overview of record linkage and current research directions. Technical report, U.S. Census Bureau, 2006.

[5] Matt Barnes and Artur Dubrawski. The Binomial Block Bootstrap Estimator for Evaluating Loss on Dependent Clusters. *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[6] Sohrab Saeb, Luca Lonini, Arun Jayaraman, David C Mohr, and Konrad P Kording. Voodoo Machine Learning for Clinical Predictions. *bioRxiv*, 2016.

[7] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013.

[8] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2016.

[9] Moshe Lichman. UCI Machine Learning Repository, 2013.

[10] Kesar Singh. On the Asymptotic Accuracy of Efron's Bootstrap. *The Annals of Statistics*, 9(6):11877–1195, 1981.

[11] Peter Hall. Resampling a coverage pattern. *Stochastic Processes and their Applications*, 20(2):231–246, 1985.

[12] Regina Y. Liu and Kesar Singh. Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the Limits of Bootstrap*, pages 225–248. Wiley-Interscience, 1992.

[13] Soumendra Nath Lahiri. *Resampling methods for dependent data*. Springer Science & Business Media, 2003.

[14] R. C. Cameron and Douglas L. Miller. A Practitioner's Guide to Cluster-Robust Inference. *Journal of Human Resources*, 50(2):317–372, 2015.

[15] Halbert White. *Asymptotic theory for econometricians*. Academic Press, 1984.

[16] James G. MacKinnon and Matthew D. Webb. Wild Bootstrap Inference for Wildly Different Cluster Sizes. *Journal of Applied Econometrics*, 32:233–254, 2017.

[17] Kung-Yee Liang and Scott L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986.

[18] Michael K Andersson and Sune Karlsson. Bootstrapping error component models. *Computational Statistics*, 16(2):221–231, 2001.

[19] Anthony Christopher Davison and David Victor Hinkley. *Bootstrap methods and their application*, volume 1. Cambridge University Press, 1997.

[20] C. A. Field and A. H. Welsh. Bootstrapping clustered data. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 69(3):369–390, 2007.

[21] Douglas Miller, A. Cameron, and Jonah Gelbach. Robust Inference with Multi-way Clustering. *Journal of Business & Economic Statistics*, 29(2), 2011.