On Supplementing Training Data by Half-Sampling

William D Heavlin, Google, Inc.

September 2019

Abstract

Machine learning (ML) models typically train on one dataset, then assess performance on another. We consider the case of training on a given dataset, then determining which (large batch of) unlabeled candidates to label in order to improve the model further. Each candidate we score by its associated prediction error(s).

We concentrate on the large batch case for two reasons: (1) While choose-1-then-update (batch of size 1) successfully avoids near-duplicates, a choose-N-then-update (batch of size N) needs additional constraints to avoid overselecting near-duplicates. (2) Just as large data volumes enable ML, updates to these large data volumes tend also to come in largish batches.

Model uncertainty we estimate by 50-percent samples without replacement. Using a two-level orthogonal array with $n-1$ columns, the resulting maximally balanced half-samples achieve high efficiency; the result is one model for each column of the orthogonal array. We use the associated n-dimensional representation of prediction uncertainty to choose which N candidates to label.

We illustrate by fitting keras-based neural networks to about 20 percent of the MNIST handwritten digit dataset.

keywords: I-optimality, jackknife, neural networks, orthogonal arrays, prediction covariance matrix, prediction uncertainty

1 Introduction

This work explores the boundary between the rather classical statistical approaches to experimental design and the relatively more modern approaches to predictive modeling, especially those associated with machine learning (ML) and machine intelligence.

1.1 Experimental Design

Statistical experimental design theory posits an experimental domain or space. Factorial designs place points in the corners of this space, achieving high statistical efficiency and enhanced sensitivity to interactions. Box and Hunter (1961a, 1961b) bring two-level factorial designs to substantial maturity, now encapsulated in the classical texts of Box, Hunter, and Hunter (2006) and Montgomery (2013). Plackett and Burman (1948) presented two-level orthogonal arrays for powers other than 2. This line of research and practice emphasizes symmetry properties and highlights orthogonal arrays. Taguchi (1987) expanded the popularity of orthogonal arrays further, especially in the context of estimating variance functions. Daniel (1959) developed the use of orthogonal array contrasts to visualize statistical significant results against background noise.

In their domain of application, two-level designs are statistically optimum. This efficiency comes from three properties: (a) Each coefficient is estimated using all observations. (b) The weights on these observations are equal in absolute value. This implies they are variance-minimizing. (c) The orthogonality implies that each coefficient estimate is uncorrelated with other coefficients. These properties foreshadow a high-efficiency claim we make in section 2.6.

A complementary approach formulates experimental designs as solutions to optimization problems. Wald (1943) introduces D-optimality, which seeks to minimize the volume of the confidence ellipsoid of the coefficients; this volume corresponds to the determinant of the coefficients' variance-covariance matrix. Kiefer and Wolfowitz (1960) propose G-optimality, which corresponds to the largest (greatest) prediction error within a bounded experimental domain; they demonstrate the asymptotic equivalence of D- and G-optimality.

Like G-optimality, Studden (1977) focuses on prediction error, and proposes I-optimality, minimizing the average prediction error — its integral — over a bounded experimental domain. Hardin and Sloane (1993) observe that, relative to D-optimal solutions, I-optimal solutions give rather more preference to points near the center of the experimental domain.

1.2 Machine Learning

The distinction between the coefficient-centric D-optimal criterion and the prediction-centric I-optimal criterion finds some analogy in the difference between statistical models and those of machine learning/artificial intelligence (ML/AI). Breiman (1985) articulates this distinction, which is brought to fruition by Friedman, Hastie, and Tibshirani (2010) and with weak learner theory of Freund et al (2003). Haykin (1998), Bengio et al (2015), and Krizhevshy et al (2012) favor neural network-based models as near-universal interpolators.

The operations research literature has developed approaches to trial-and-error algorithms that work toward online optimization and control. Bellman's classic result articulates the explore-exploit dilemma. For multi-armed bandits, see McMahon and Streeter (2009). Reinforcement learning (RL) adapts more thoroughly to feature spaces, see Sutton and Barto (2018). Characteristic of RL is iteration between proposing a trial and its adaptation to the resulting response. Some versions favor proposals with higher uncertainty, in effect including an explore term into the exploit objective function.

While most of ML/AI presumes a static dataset for training, there has always been an interest in online model updates to streams of data. See, for example, McMahon and Streeter (2014). Sahoo (2017) describe one adaptation of deep learning to the online problem.

It is primarily the field of active learning (AL) that addresses the problem of which candidates for training to label. Settles (2009) offers a still well-cited survey of this field. Lewis and Gale (1994) describes the classic sequential algorithm, and Seung et al (1992) present what has come to be called the query-by-committee algorithm. Cohn et al (1996) implements AL for statistical models, while Schohn and Cohn (2000), Tong and Koller (2001), and Tong and Chang (2001) interweave AL with support vector machines.

For batch labeling problems, Brinker (2003) and Xu et al (2007) build on Tong and Koller (2001) by explicitly incorporating a diversity measure; Brinker uses a minimax correlation, Xu et al a Kullback-Liebler density distance. Working within the framework of logistic regression, Hoi et al (2006) consider the parameters' Hessian matrix. Guo and Schuurmans (2008) propose the entropy of any proposed batch, working around the problem of not knowing labels by an "optimistic" heuristic.

1.3 New Molecule Discovery

In the present work, we focus on proposing data, in part to identify new optima, in part to refine our tentative model. Enabled by high-throughput experimental capabilities, our proposals are required to consist of relatively large batches. As is typical of interesting statistical work, the optima we seek are quite, quite rare — yet-to-be-discovered proteins or drug molecules. For this reason, the current draft of the model is imperfect, a deep neural network of some form, say. The large batch requirement appears to put it outside the reach of RL and exposes the ad hoc selection criteria used by AL. Nonetheless, our effort may be fairly summarized as proposing a modification to the RL and AL objective functions, one that elegantly extends to large-batch supplements.

In this paper, we select from C potential candidates — this is an externally defined data structure. For our purposes, these C candidates have known feature vectors, $\mathbf{x}_c, c = 1, 2, \ldots, C$, but unknown responses/labels/rewards $y_c, c = 1, 2, \ldots, C$. So we want to pick a subset of size C_1 candidates, $C_1 < C$ to label.

2 Approach

2.1 The Prediction Variance-Covariance Matrix

Our approach combines two ideas. One idea is that if we were able to posit a prediction variance-covariance matrix $\mathbf{V}$, we can judiciously solve for which C_1 candidates to use. The second idea is that we propose a particular approach to estimating this variance-covariance matrix $\mathbf{V}$, one based on carefully balancing half-samples. Along the way, we are sensitive to issues of computational scaling, and consider an option that bypasses computing directly the $C \times C$ matrix $\mathbf{V}$.

Consider two candidates $a \neq b \in \{1, 2, \ldots, C\}$, with feature vectors $\mathbf{x}_a$ and $\mathbf{x}_b$, respectively, and unobserved rewards y_a and y_b. We have a model M by which we make predictions $\hat{y}(\mathbf{x}_a)$ and $\hat{y}(\mathbf{x}_b)$, respectively. Let us suppose we can estimate $\mathbf{V}[a, b] = \mathbb{COV}\{\hat{y}(\mathbf{x}_a), \hat{y}(\mathbf{x}_b)|\mathbf{x}_a, \mathbf{x}_b, M\}$. When $\mathbf{x}_a = \mathbf{x}_b$, $\mathbf{V}[a, b] = \mathbf{V}[a, a]$ and is merely the squared standard error of prediction $\hat{y}(\mathbf{x}_a)$. It is obvious, in a way approaching tautological, that $\mathbf{V}[a, b] \rightarrow \mathbf{V}[a, a]$ as $\mathbf{x}_b \rightarrow \mathbf{x}_a$. In this way, $\mathbf{V}$ captures a sense in which observing $\mathbf{x}_a$ might make observing $\mathbf{x}_b$ unnecessary. This issue we call the *problem of near duplicates*: When $\mathbf{x}_b \approx \mathbf{x}_a$, $\mathbf{V}[a, a] \approx \mathbf{V}[b, b] \approx \mathbf{V}[a, b]$ and $\mathbb{COR}(\hat{y}(\mathbf{x}_a), \hat{y}(\mathbf{x}_b)) \rightarrow 1$, and there is not offer much incremental benefit from observing both a and b beyond that of just observing one of a or b.

For a linear model, we can establish some intuition for $\mathbf{V}$. $\hat{y}(\mathbf{x}_a) = \mathbf{x}_a^\top \mathbf{b}$, so $\mathbb{COV}(\hat{y}(\mathbf{x}_a), \hat{y}(\mathbf{x}_b)) = \mathbf{x}_a^\top \mathbb{COV}(\mathbf{b}, \mathbf{b})\mathbf{x}_b = \mathbf{x}_a^\top (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{x}_b$. So if we have a model fit with matrix of features $\mathbf{X}$ and candidates $\mathbf{Z}$, the variance-covariance matrix of predictions $\mathbb{COV}(\hat{y}(\mathbf{z}_a), \hat{y}(\mathbf{z}_b))_{ab}$ is $(\mathbf{Z}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{Z}^\top)_{ab}$. Now consider updating $\mathbf{X}_0$ with one more row: $\mathbf{X}_1 = (\mathbf{X}_0^\top | \mathbf{x}_1)^\top$. From Sherman-Morrison's (1949) rank-1 update, we have $(\mathbf{X}_1^\top \mathbf{X}_1)^{-1} = (\mathbf{X}_0^\top \mathbf{X}_0)^{-1} - \mathbf{x}_1 \mathbf{x}_1^\top / (1 + \mathbf{x}_1^\top (\mathbf{X}_0^\top \mathbf{X}_0)^{-1}\mathbf{x}_1)$. If we associate $\mathbf{V}_0$ with $\mathbf{X}_0$ and $\mathbf{V}_1$ with $\mathbf{X}_1$, the Sherman-Morrison calculation gives us that $\mathbf{V}_1 = \mathbf{V}_0 - \mathbf{V}_0[, c]\mathbf{V}_0[, c]^\top / (1 + \mathbf{V}_0[c, c])$.

We make three observations regarding $\mathbf{V}$. (1) First, various optimal design criteria are functions of $\mathbf{V}$. In particular, its largest diagonal element, $\texttt{max}(\texttt{diag}(\mathbf{V}))$, corresponds to the G-optimal criterion. In similar vein, $\texttt{ave}(\texttt{diag}(\mathbf{V}))$ gives the I-optimal criterion, $\texttt{trace}(\mathbf{V})/C$. (2) $\mathbf{V}$ is a precision matrix, and if we newly observe $\mathbf{x}_c$, we can update $\mathbf{V}$ by the Sherman-Morrison (1949) rank-1 update:

$$\mathbf{V}_{k+1} \leftarrow \mathbf{V}_k - \mathbf{V}_k[, c]\mathbf{V}_k[, c]^\top / (1 + \mathbf{V}_k[c, c])$$

For $\texttt{trace}(\mathbf{V}_{k+1})$, the (scalar) decrement from $\texttt{trace}(\mathbf{V}_k)$ is $\mathbf{V}_k[, c]^\top \mathbf{V}_k[, c] / (1 + \mathbf{V}_k[c, c])$. This decrement has two components: (a) $\mathbf{V}_k[c, c]^2 / (1 + \mathbf{V}_k[c, c])$ is the influence of candidate c on its own prediction, classical *leverage*. (b) $\sum_{d \neq c} \mathbf{V}_k[d, c]^2 / (1 + \mathbf{V}_k[c, c])$ is the influence of including candidate c on the mean squared prediction error of all the other candidates. When (a) dominates, the I-optimal criterion prescribes essentially *memorizing* the most uncertain candidates. When (b) dominates, the I-optimal criterion prioritizes those candidates whose predictions are more thickly correlated with many other candidates (*interpolation*). (3) Third, note that this Sherman-Morrison update can be calculated without actually needing to observe $y(\mathbf{x}_c)$.

By these three properties, $\mathbf{V}$ emerges as a principled object for constructing supplemental batches. The case against using $\mathbf{V}$ is this: as a $C \times C$ matrix, $\mathbf{V}$ becomes an uncomfortably large object as C becomes large, $O(C^2)$. This concern is mitigated in section 3, algorithm 2, which adapts to operate on a $C \times n$ matrix instead, so is rank n not potentially rank C, and computationally therefore is $O(C)$. Viswanathan et al (2019) likewise manage and manipulate a $C \times C$ precision matrix, in their case by using sparse matrix representations; their approach is a qualitatively different from the rank n approach presented here as algorithm 2.

As an aside, consider Sherman-Morrison in reverse: For observation i,

$$\mathbf{V}_k \leftarrow \mathbf{V}_{k+1} + \mathbf{V}_{k+1}[, i]\mathbf{V}_{k+1}[, i]^\top / (1 - \mathbf{V}_{k+1}[i, i])$$

So the increment to $\texttt{trace}(\mathbf{V}_k)$, $\mathbf{V}_{k+1}[, i]^\top \mathbf{V}_{k+1}[, i] / (1 - \mathbf{V}_{k+1}[i, i])$, quantifies the influence of observation i in the current training set on the overall mean prediction error. As before, note that this increment draws attention to observations whose predictions are more thickly correlated with the predictions of other observations. This diagnostic therefore emerges as intrinsically useful, sufficiently so that it presents an additional reason for calculating $\mathbf{V}$ and its equivalents — one applicable even in contexts when supplementing batches is not of primary concern.

2.2 Sample Sizes for Variances

For histograms, a rather well-known rule of thumb for the sample size is $n = 50$. The detailed rationale for this is as follows: (a) The proxy for the precision of a histogram is the precision of its spread, e.g. its standard deviation. (b) Based on Gaussian theory and the chi-square distribution, one achieves for an estimated standard deviation a 95 percent confidence interval of ± 10 percent with about $n = 200$ degrees of freedom; this corresponds to one significant digit. (c) There is a square-root rule in effect, so doubling the tolerance to ± 20 percent requires one fourth the data, hence, $n = 50$.

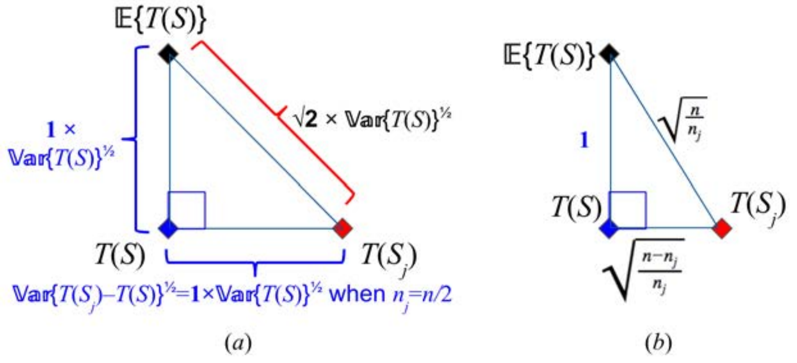
$\mathbb{E}\{T(S)\}$
$1 \times \mathbb{V}\mathrm{ar}\{T(S)\}^{½}$
$\sqrt{2} \times \mathbb{V}\mathrm{ar}\{T(S)\}^{½}$
$T(S)$
$T(S_j)$
$\mathbb{V}\mathrm{ar}\{T(S_j)-T(S)\}^{½}=1\times\mathbb{V}\mathrm{ar}\{T(S)\}^{½}$ when $n_j=n/2$
(a)
$\mathbb{E}\{T(S)\}$
1
$\sqrt{\frac{n}{n_j}}$
$T(S)$
$T(S_j)$
$\sqrt{\frac{n-n_j}{n_j}}$
(b)

Figure 1: Geometrical interpretation of the jackknife correction factors, (a) for half-sampling, (b) when retaining n_j of n.

We now offer a forward reference to section 2.6: By consideration (b), one can reasonably use orthogonal arrays such as $OA(256)$, by consideration (c), $OA(48)$.

2.3 Half-Sampling

We now turn to estimating this prediction variance-covariance matrix $\mathbf{V}$. We use a resampling approach, 50 percent samples without replacement, which we term *half-sampling*. In this regard, it is a particular form of the jackknife of Quenouille (1956) and Tukey (1958).

Half-samples have a minor, if intriguing, property. The variance among random half-samples is an unbiased estimate of the variance of estimates based on the whole sample. This we illustrate in Figure 1a, which pictures a right isosceles triangle.

Denote the sample by S, which has n observations and denote the j-th half-sample by S_j, which has $n_j = n/2$ observations. Denote the estimand by $\tau = \mathbb{E}\{T(S)\} = \mathbb{E}\{T(S_j)\}$. Of primary interest is the standard error of $T(S), \mathbb{VAR}\{T(S)\}^{1/2}$, where $\mathbb{VAR}(T(S)) = \mathbb{E}\{(T(S)-\tau)^2\}$. This is not straightforward to estimate because τ is unknown. However, $\mathbb{E}\{(T(S_j)-\tau)^2\} = \mathbb{E}\{(T(S_j)-T(S))^2\} + \mathbb{E}\{(T(S)-\tau)^2\}$. When $n_j = n/2$, then $\mathbb{E}\{(T(S_j)-\tau)^2\} = 2 \times \mathbb{E}\{(T(S)-\tau)^2\}$, so $\mathbb{E}\{(T(S_j)-T(S))^2\} = \mathbb{E}\{(T(S)-\tau)^2\}$. The significance of this is that the left-hand side is estimable, and of course, estimates $\mathbb{E}\{(T(S)-\tau)^2\}$, which is exactly the standard error we desire. This calculation is presented geometrically in Figure 1a, essentially a right isosceles triangle.

We can generalize to n_j as some other fraction of n: As before, $\mathbb{E}\{(T(S_j)-\tau)^2\} = \mathbb{E}\{(T(S_j)-T(S))^2\}+\mathbb{E}\{(T(S)-\tau)^2\}$. Now, $\mathbb{E}\{(T(S_j)-\tau)^2\} = (n/n_j)\mathbb{E}\{(T(S)-\tau)^2\}$, so $((n/n_j)-1)\mathbb{E}\{(T(S)-\tau)^2\} = \mathbb{E}\{(T(S_j)-T(S))^2\}$ or $\mathbb{E}\{\left[(n_j/(n-n_j))^{1/2}(T(S)-\tau)\right]^2\} = \mathbb{E}\{(T(S_j)-T(S))^2\}$. This calculation is presented geometrically in Figure 1b, picturing a right triangle. The right angles in Figures 1a and 1b result from the fact that $\mathbb{E}\{(T(S)-\tau)(T(S_j)-T(S))\} = 0$, a result of the unbiasedness of $T(S_j)$ and that S_j is interpreted as a random half-sample.

2.4 File shards

When implemented in a computer file system, large datasets sometimes consist of multiple physical files called *shards*. A computationally convenient interpretation of half-sampling is that it consists of half the shards. A recognizably natural practice is for the number of shards to be 2^k for some integer k. $k = 9$ to 12 correspond to shard counts of 256 to 4096. One can use orthogonal arrays to implement half-sampling and enforce the property that each shard is used exactly half the time.

2.5 Orthogonal arrays

Figure 2 presents $OA(48)$, an orthogonal array consisting of $n = 48$ shards; this array is due to Plackett and Burman (1948). $OA(48)$ gives 47 half-samples. The orthogonality property ensures that these 47 columns have correlation zero, that is, for $j \neq k$ corresponding to different columns of $OA(48)$, $\mathbb{E}\{(T(S_j)-T(S))(T(S_k)-T(S))\} = 0$.

Intuitively, the careful balancing among shards achieved by orthogonal arrays should be somehow better than half-sampling induced randomly. We assess this question by simulation. We construct orthogonal arrays of size $n = 2^k, k \in \{4,5,6,7,8\}$, which we assess for $p \in \{0,1,2,4,8\}$

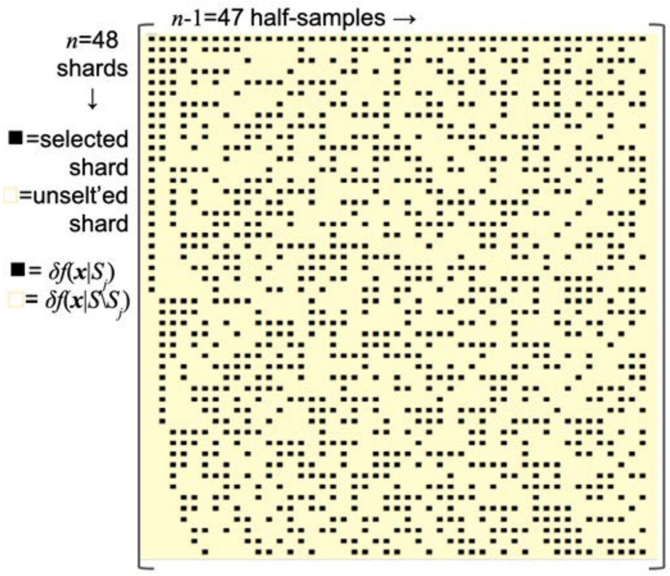

Figure 2: Plackett and Burman's $OA(48)$, a 48×47 orthogonal array.

two-level features. The results are presented in Figure 3. Note that the largest p considered is 8, while the smallest n is 16; this boundary is chosen to reduce the probability of random half-samples becoming fatally collinear.

For $80,000$ simulations, Figure 3 plots the relative statistical efficiencies of orthogonal arrays versus random half-sampling. The estimand is the prediction variance at the $\mathbf{q} = (1, 1, \ldots, 0)$-corner, where $\texttt{sum}(\mathbf{q}_i) = p$. The group labeled $p = 1 + 0$ corresponds to an intercept and no two-level features, $p = 1 + 1$ to an intercept and one two-level feature, and so on. Figure 3 plots the median prediction variance from random halves to that estimated from orthogonal array-based halves.

Note that when p/n is small, the benefit of orthogonal arrays is minimal. When p/n is larger, by eye one can see the relative efficiency is roughly $1+p/n$. A more detailed numerical analysis, plotted by a red "|" symbol, suggests the relative statistical efficiency is approximately $(n-1)/(n-1-p) = \sum_{k=0}^{\infty}[p/(n-1)]^k$. From the uppermost right point pair, this appears to underestimate the relative efficiency gain for the highest p/n, where it approaches a relative efficiency of $2\times$. Note that conventional neural networks fit models exactly in this domain: high p-to-n ratios.

2.6 Contrast Matrix $\boldsymbol{\Delta}$

For any given column of an orthogonal array, we have two half-samples, so two predictive models, $f(\mathbf{x}|S_j)$ and $f(\mathbf{x}|S\backslash S_j)$, say. A natural common estimate is their average, $[f(\mathbf{x}|S_j) + f(\mathbf{x}|S\backslash S_j)]/2$. Their (signed) difference, $[f(\mathbf{x}|S_j) - f(\mathbf{x}|S\backslash S_j)]/\sqrt{2}$, estimates with one degree of freedom the standard error for $f(\mathbf{x}|S) \approx [f(\mathbf{x}|S_j) + f(\mathbf{x}|S\backslash S_j)]/2$. (In much of what follows, the $\sqrt{2}$ factor is inessential.)

For a typical candidate c with features $\mathbf{x}_c$, we have the $n-1$ half-sample contrasts $\Delta[\mathbf{x}_c, j], j = 1, 2, \ldots, n-1$. The $C \times (n-1)$ matrix $\boldsymbol{\Delta}$ has four properties: (1) It estimates $\mathbb{VAR}\{f(\mathbf{x}|S)\}$ unbiasedly: $\mathbb{E}\{\sum_j^{n-1} \Delta^2[\mathbf{x}, j]/(n-1)\} = \mathbb{VAR}\{f(\mathbf{x}|S)\}$. (2) For each j, $\Delta[\mathbf{x}, j]$ make use of all the data in S, it is in this sense of high statistical efficiency. (3) For disjoint half-samples, $j \neq k, \mathbb{COR}(\Delta[\mathbf{x}, j], \Delta[\mathbf{x}, k]) = 0$. In this second sense, the matrix $\boldsymbol{\Delta}$ is of high statistical efficiency. (4) A modest generalization of (1), $\mathbb{E}\{\sum_j^{n-1} \Delta[\mathbf{x}_c, j]\Delta[\mathbf{x}_d, j]/(n-1)\} = \mathbb{COV}\{f(\mathbf{x}_c|S), f(\mathbf{x}_d|S)\}$. That is, $\boldsymbol{\Delta}\boldsymbol{\Delta}^\top/(\mathbf{n}-\mathbf{1})$ estimates $\mathbf{V}$ of section 2.1 Proposition (1) asserts this for the diagonal elements; here we recognize in (4) the more general result for prediction covariances; from section 2.1, we know that estimating $\mathbf{V}$ is important.

We emphasize that points (1), (2), and (3) reiterate the efficient design properties of $OA(2^k)$ listed in section 1.1.

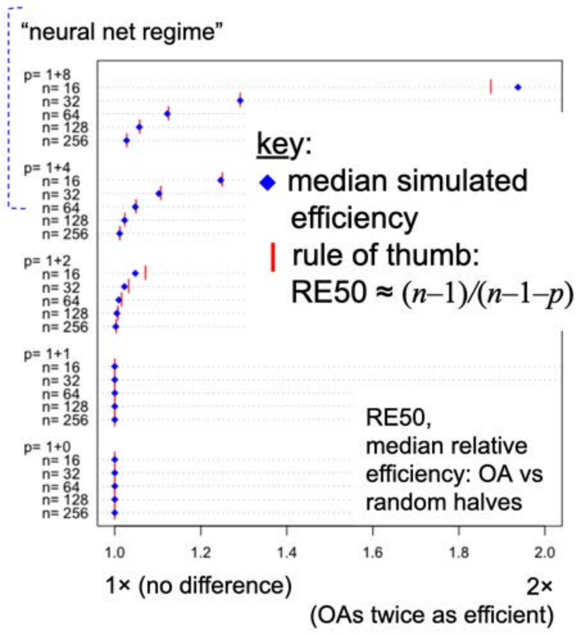

Figure 3: The simulated relative efficiency of orthogonal arrays relative to random half-sampling. Blue diamonds plot the simulation-estimated RE50 (median relative efficiency), | plots the heuristic approximation $(n-1)/(n-1-p)$.

3 Algorithms

Note that $\mathbf{\Delta\Delta}^\top/(n-1)$ is an unbiased estimate of $\mathbf{V} = (V_{ih})_{ih} = \mathbb{COV}\{\hat{y}_i, \hat{y}_h|\mathbf{x}_i, \mathbf{x}_h\}$, first encountered in section 2. *Assuming the feasibility* of computing such a $C \times C$ matrix and holding it in memory, we can iteratively select candidates and update $\mathbf{V}$ as follows:

Algorithm 1: extract candidates using $C \times C$ precision covariance matrix $\mathbf{V}$

```
Initially :
1    $\mathcal{C} \leftarrow \{\}$
2    $\mathbf{V}_0 \leftarrow \mathbf{\Delta\Delta}^\top/(n-1)$
3    $k \leftarrow 0$
Loop in $k$ :
4    $c_k \leftarrow \texttt{argmin}\ f(\mathbf{V}_k - \mathbf{V}_k[,c]\mathbf{V}_k[c,]/(1+\mathbf{V}_k[c,c]))$
5    $v_k \leftarrow \mathbf{V}_k[c_k, c_k]$
6    $\mathbf{V}_{k+1} \leftarrow \mathbf{V}_k - \mathbf{V}_k[,c_k]\mathbf{V}_k[c_k,]/(1+v_k)$
7    $\mathcal{C} \leftarrow \mathcal{C} \cup \{c_k\}$
8    $k \leftarrow k+1$
Returning:
9    $\mathcal{C}$
```

When $f(\mathbf{V}) = \texttt{max diag}(\mathbf{V})$, algorithm 1 corresponds to G-optimality; when $f(\mathbf{V}) = \texttt{trace}(\mathbf{V})$, it corresponds to I-optimality.

Typically, it is not at all convenient to compute or maintain the $C \times C$ matrix $\mathbf{V}$ in memory. The following (equivalent) algorithm requires only maintaining the $C \times (n-1)$ matrix Δ_k:

As before, choosing $f(\cdot)$ as $\texttt{max diag}(\cdot)$ corresponds to G-optimality while taking $f(\cdot)$ as $\texttt{trace}(\cdot)$ gives the I-optimal criterion.

Note that $\texttt{trace}(\mathbf{\Delta}_k\mathbf{H}_c\mathbf{H}_c^\top\mathbf{\Delta}_k^\top) = \texttt{trace}(\mathbf{H}_c\mathbf{H}_c^\top\mathbf{\Delta}_k^\top\mathbf{\Delta}_k)$, and the latter form merely involves the multiplication of two $(n-1)\times(n-1)$ matrices, $\mathbf{H}_c\mathbf{H}_c^\top$ and $\mathbf{\Delta}_k^\top\mathbf{\Delta}_k$.

Finally, note that one can form from new candidates another $\mathbf{\Delta}$-matrix, $\mathbf{\Delta}^{(1)}$, say, and re-enter algorithm 2 by initializing $\mathcal{C} \leftarrow \mathcal{C}$, $\mathbf{H} \leftarrow \mathbf{H}$, and $\mathbf{\Delta}_0 \leftarrow \mathbf{\Delta}^{(1)}\mathbf{H}/(n-1)^{1/2}$. This enables the construction of a series of supplemental batches, each of which would present its own list of candidates. The effect of including previous candidates $\mathcal{C}$ is recorded in the $(n-1)\times(n-1)$ matrix

Algorithm 2: extract candidates using $C \times (n-1)$ half-sample matrix $\mathbf{\Delta}$

```
Initially :
1       $\mathcal{C} \leftarrow \{\}$
2       $\mathbf{H} \leftarrow \mathbf{I}_{n-1}$
3       $\mathbf{\Delta}_0 \leftarrow \mathbf{\Delta}/(n-1)^{1/2}$
4       $k \leftarrow 0$
Loop in $k$ :
5       $c_k \leftarrow \texttt{argmin}\ f(\mathbf{\Delta}_k \mathbf{H}_c \mathbf{H}_c^\top \mathbf{\Delta}_k^\top)$,
6           where $\mathbf{H}_c \equiv \mathbf{I}_{n-1} - \mathbf{\Delta}_k[c,]\mathbf{\Delta}_k[c,]^\top/(1+\|\mathbf{\Delta}_k[c,]\|^2)$
7       $v_k \leftarrow \|\mathbf{\Delta}_k[c_k,]\|^2$
8       $\mathbf{H}_k \leftarrow \mathbf{I}_{n-1} - \mathbf{\Delta}_k[c_k,]\mathbf{\Delta}_k[c_k,]^\top/(1+v_k)$
9       $\mathbf{H} \leftarrow \mathbf{H}\mathbf{H}_k$
10      $\mathbf{\Delta}_{k+1} \leftarrow \mathbf{\Delta}_k \mathbf{H}_k$
11      $\mathcal{C} \leftarrow \mathcal{C} \cup \{c_k\}$
12      $k \leftarrow k+1$
Returning:
13      $\mathcal{C}, \mathbf{H}$
```

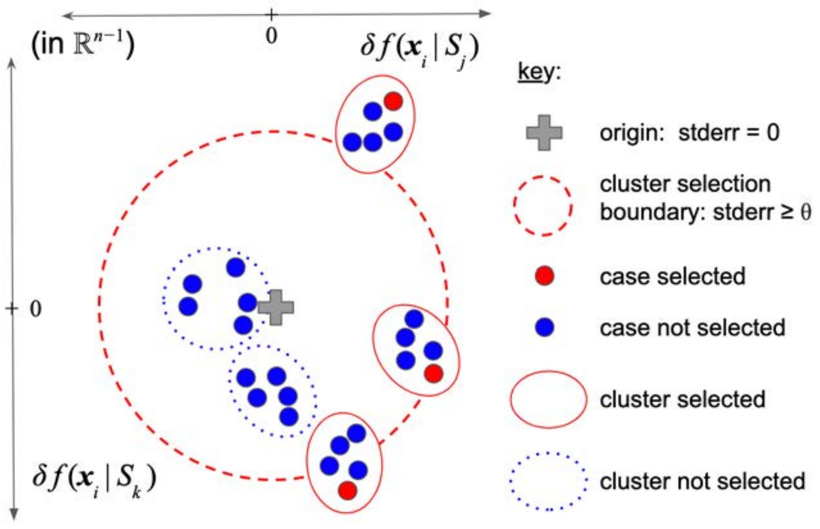

Figure 4: Geometry of the candidates (the rows) of $\mathbf{\Delta}$.

$\mathbf{H}$.

3.1 Geometry and Clustering

The $C \times (n-1)$ candidate×half-sample matrix $\mathbf{\Delta}$ has an interesting (and Euclidean!) geometry (Figure 4). In $\mathbb{R}^{n-1}$, the lengths of each row $\|\mathbf{\Delta}[c,]\|$ are the estimated standard errors of the corresponding predictions $f(\mathbf{x}|S)$ and distances from the origin. Candidates near one another have predictions that correlate highly. From this geometry, it is natural to consider any given candidate as giving substantial information about the peers of its clusters. The computational benefit is that rather than pick candidates one at a time, one can pick them one per cluster for a pre-specified number of clusters — in other words, as a batch. (The cluster(s) near the origin may not need to be selected.) For a given cluster, a natural selection rule would be to select that one farthest from the origin — the one with the largest standard error.

This cluster-based approach should be somewhat less efficient relative to using the prediction variance-covariance matrix. It also has reduced computational requirements — $O(C^2\texttt{log}(C))$ vs $O(C^3)$.

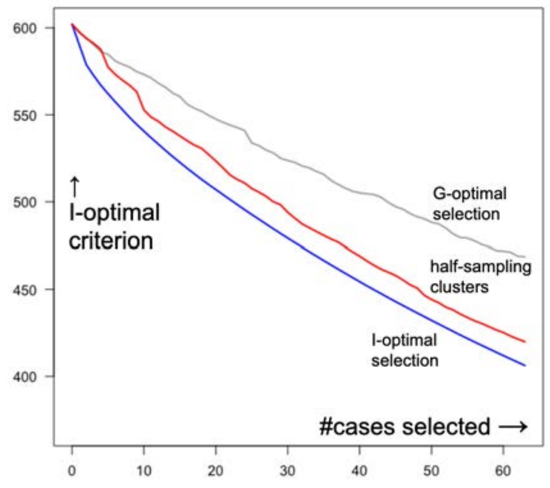

Figure 5: As a function of the number of candidates selected, the (lower-is-better) I-optimal criterion for three different selection algorithms.

4 Example

For an example, we consider the handwritten MNIST digits (LeCun, et al 1995). These digit images are 28×28 256-valued grayscale matrices. Following the template of Chollet (2018), we flatten this representation to 784-tuple vectors of floats, scaled into $[0, 1]$. The ground truths of the digits are represented by 10-tuples of mutually exclusive booleans: The digit 0 is $(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, the digit 1 as $(0, 1, 0, 0, 0, 0, 0, 0, 0, 0)$, etc. The MLP consists of two hidden layers, respectively of 256 and 128 fully connected nodes. We set the cross-validation rate at 0.50.

For training, we use the first $12,288 (= 48 \times 256)$ digits, and for candidates the second $12,288$ digits of the $60,000$-digit training set. The orthogonal array we use is $OA(48)$, due to Plackett and Burman, 1948. Following Chollet (2018), we fit an initial keras model using 64 iterations to all 48 shards; this model $M(S)$ defines the warm start for all subsequent half-sample models $M(S_j)$ and $M(S \backslash S_j), j = 1, 2, ..., n - 1$. The result is for each of the $12,288$ candidates, the 47 contrasts sketched in Table 2a.

We apply hierarchical clustering to obtain 64 clusters, of which one holds 97.4 percent of all candidates: these all have small prediction variances — these images are so easy to classify that their predictions are insensitive to changes in the underlying training data.

Table 2b reports the scores and cluster names for those images in Table 2a. The reported score of each image (row) is the root mean square of the half-sample contrasts for that row. Table 2a displays 5 rows extracted from the $C \times (n - 1) = 12,288$ row $\times 47$ column matrix $\mathbf{\Delta}$.

Figure 5 compares three selection rules for selecting a supplement of size 64. The y-axis of the plot is the lower-is-better I-optimality criterion, the trace of the matrix $\mathbf{V}_{k+1}$, which is proportional to average prediction variance. This is somewhat similar in spirit to AUC-type scoring, in that both are overall scalar metrics of predictive performance.

In Figure 5, we have three curves, gray, red, and blue. Gray selects by the G-optimal criterion, then updates the matrix $\mathbf{V}_k$. Gray is sort of a proxy for reinforcement learning, except that it looks ahead to the post-update impact on $\mathbf{V}_{k+1}$.

Red selects by clustering the rows of $\mathbf{\Delta}$; this selection is more-or-less in parallel, with no operations on $\mathbf{V}_k$. Blue selects by the I-optimal criterion, and like G-optimality, also updates the matrix $\mathbf{V}_k$ to $\mathbf{V}_{k+1}$.

It is not too surprising that the blue curve does better; it's the curve associated with a greedy algorithm keying the y-axis's very own criterion. It's a little bit interesting that the red curve does better than gray also. These half-sampling clusters do not depend on or update $\mathbf{V}_k$; and instead prioritizes the parallelism that hierarchical clustering provides.

One can calculate the ratios of the last values plotted; these can be interpreted as statistical efficiencies: I-optimal-based selection is 15 percent more efficient than G and half-sample-based clusters are 11 percent more efficient. With respect G-optimality, neither of these values has reached an asymptote, and their advantages relative to G should increase as we add candidates. The ratios between I-optimal and half-sample clusters seems more stable, about a 3 percent efficiency deficit for the latter.

5 Summary

The problem we consider is that of supplementing training data for a machine learning model. One family of motivating examples is discovering molecules for an intended function. The space of potential molecules is combinatorially large, and any model predicting the suitability of new molecules is likely to be perpetually incomplete.

For our purposes, there is a pre-existing model, of a modern ML type, a deep neural network, say, fit using a current dataset, so there is some ability to make model predictions. For candidate c drawn from C potential candidates, we have available the associated feature vector $\mathbf{x}_c$ but not the associated response/reward y_c. Consistent with many big data applications, any data supplement likely needs to consist of a batch of substantial size, drawn from an even larger set of potential candidates.

Common also with many ML models, we balance statistical considerations with those of computational feasibility, both in terms of computing complexity and data object size.

In general, we pursue a paradigm of multivariate model uncertainty, recognizing early that merely adding error bars to our model predictions is insufficient. This is because of the near-duplicate problem: Two molecules close together in molecule space may share about the same prediction and also larger error bars. If we observe only one of them, we may gain enough information about the other that observing both becomes wasteful. This redundancy can be detected, for instance, were we to have available the prediction variance-covariance matrix; this would signal near-duplicates by the high correlation between the two candidate molecules' predictions.

Our approach therefore has two parts. The first consists in recognizing the theoretical centrality of the $C \times C$ prediction covariance matrix $\mathbf{V}$. $\mathbf{V}$ is a precision matrix, and quite suitable for this batch supplement problem, even for, especially for, modern ML models such as deep neural networks, for four reasons: (1) Consistent with such semi-parametric models, $\mathbf{V}$ is defined in the prediction domain. (2) $\mathbf{V}$ does not require observing the response/reward y_c in order to be updated by the prospect of including candidate c, with features $\mathbf{x}_c$, in the supplement. (3) When constructed for training set observations, $\mathbf{V}$ can be used to quantify for each observation the impact on overall model fit. (4) The underlying theory for $\mathbf{V}$ is comfortably compatible with linear model theory and classical experimental design ideas.

Actually estimating $\mathbf{V}$ and its kin defines the second element of our approach, *half-sampling.* Half-sampling strongly resembles a fifty-percent jackknife. In the context of deep neural networks, which act almost as interpolators, such without-replacement sampling seems preferable to with-replacement bootstrap resampling. As presented here, half-sampling depends on defining mutually exclusive, equally sized data shards, then enforcing careful balance properties by using a two-level orthogonal array. (Note that when shards correspond to physical computer files, half-sampling cuts file input reads by half.) The use of the orthogonal array buys something substantial: relative efficiencies approaching $2\times$ when the ratio of parameter count to sample size becomes large — exactly the case inhabited by deep neural networks. Thus, half-sampling adapts to ML-type models three ways: without-replacement sampling, file sharding, and orthogonal-array-based sample balancing.

The resulting matrix of prediction contrasts, $\mathbf{\Delta}$, inherits the statistical efficiency properties of two-level orthogonal arrays: (1) For every candidate, each half-sample contrast uses all the training data. (2) The (signed) weights on each shard are equal in absolute value. (3) Between any two half-samples, the weights are uncorrelated. In this sense, the construction of $\mathbf{\Delta}$ has high statistical efficiency, and therefore estimates $\mathbf{V} = \mathbf{\Delta}\mathbf{\Delta}^\top/(n-1)$ with $n-1$ degrees of freedom. This is reflected in the blue curve in Figure 5.

We give due diligence to computational issues: (1) Fitting by half-sampling can be accelerated by warm starts, reduced file reads, and parallel computation. (2) By using algorithm 2, we can avoid direct calculation of the $C \times C$ prediction variance-covariance matrix $\mathbf{V}$. (3) We provide the alternative batch construction method C, based on non-iterative clustering, which proceeds with a three percent loss in statistical efficiency.

Finally, half-sampling sets forth a parallel algorithm for estimating a sampling distribution. For the subset of data and models to which it applies, half-sampling therefore offers an estimation paradigm that is computationally more elegant than that based on (the fundamentally single-threaded) Monte Carlo Markov chains.

6 Acknowledgments

The author benefited from the comments and questions of Patrick Riley, Anton Geraschenko, and John Platt.

7 References

Bengio, Y, LeCun, Y, Hinton, G. (2015). "Deep learning," *Nature* 521(7553): pp 436-444.

Bishop, CM (2006). *Pattern Recognition and Machine Learning*, Springer.

Box, G.E.P. and Hunter, J.S. (1961a). The 2^{k-p} fractional factorial designs, Pt. I. *Technometrics*, 3(3): 311-351. (1961b). Pt. II. 3(4): 449-458, 1961.

Box, G.E.P., Hunter, J.S., and Hunter, W.G. (2005), *Statistics for Experimenters: design, discovery, and innovation.* New York: Wiley.

Breiman, L. (2001). "Statistical Modeling: The Two Cultures (with comments)." *Statistical Science* 16:3, pp 199–231.

Brinker, K (2003). "Incorporating diversity in active learning with support vector machines," *Proceedings of the International Conference on Machine Learning*, pp 59–66. AAAI Press.

Chollet, F, with Allaire, JJ, (2018). *Deep learning with R*, Shelter Island: Manning.

Cohn, DA, Ghahramani, Z, and Jordan, MI (1996). "Active learning with statistical models." *Journal of artificial intelligence research* 4: pp 129-145.

Daniel, C (1959). "Use of Half-Normal Plots in Interpreting Factorial Two-Level Experiments," *Technometrics* 1:4, pp 311-341.

Freund, Y, Iyer, R, Schapire, RE, and Singer, Y (2003). "An Efficient Boosting Algorithm for Combining Preferences," *Journal for Machine Learning Research* 4: 933-969.

Friedman, JF, Hastie, T, and Tibshirani, R (2010). *The Elements of Statistical Learning*, Springer.

Hardin, RH and Sloane, NJA (1993). A new approach to the construction of optimal designs, *Journal of Statistical Planning and Inference*, 37:3, pp 339-369.

Guo, Y and Schuurmans, D (2008). "Discriminative batch mode active learning." *Advances in Neural Information Processing Systems* 20: pp 593–600. MIT Press: Cambridge, MA.

Hastie, T. and Tibshirani, R. (1993). Varying-coefficient models (with discussion), *Journal of the Royal Statistical Society*, Series B, 55: 757-796.

Haykin, S (1998). *Neural Networks: a comprehensive formulation*, Prentice Hall: Upper Saddle River, NJ.

Hoi, SCH, Jin, R, and Lyu, MR (2006). "Large-scale text categorization by batch mode active learning." *Proceedings of the International Conference on the World Wide Web*, pp 633–642. ACM Press.

Kiefer, J.C. and Wolfowitz, J. (1960). The equivalence of two extremum problems, *Canadian Journal of Mathematics*, 12, 363-366.

Krizhevshy, A, Sutskever, I, Hinton, G (2012). "ImageNet classification with deep convolutional neutal networks," *NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevado*

LeCun, Y, Jackel, LD, Bottou, L, Brunot, A, Cortes, C, Denker, JS, Drucker, H, Guyon, I, Muller, UA, Sackinger, E, Simard, P, and Vapnik, V. (1995). "Comparison of learning algorithms for handwritten digit recognition" in Fogelman, F. and Gallinari, P. (Eds), *International Conference on Artificial Neural Networks*, pp 53-60, Paris.

Lewis, D and W. Gale, W (1994). "A sequential algorithm for training text classifiers." *Proceedings of the ACM SIGIR*, Conference on Research and Development in Information Retrieval, pp 3–12. ACM/Springer.

McMahan, HB, and Streeter, M (2014). "Delay-tolerant Algorithms for Asynchronous Distributed Online Learning," *Advances in Neural Inforamtion Processing Systems (NIPS).*

McMahan, HB and Streeter, M (2009). "Tighter Bounds for Multi-Armed Bandits with Expert Advice," *Proceedings of the 22nd Annual Conference on Learning Theory.*

Matusugu, M, Mori, K, Mitaro. Y, Kaneda, Y (2003). "Subject independent facial expression recognition with robust face detection using a convolutional neural network, *Neural Networks*, 16(5): pp 555-559.

Montgomery, D.C. (2013). *Design and Analysis of Experiments*, New York: Wiley.

Quenouille, MH (1956). "Notes on bias in estimation," *Biometrika* 43(3-4): pp 353-360.

Sahoo, D, Pham, Q, Lu, J, and Hoi, SCH (2017). "online Deep Learning: Learning Deep Neural Networks on the Fly," `arxiv` 1711.03705.

Schohn, G, and Cohn, D (2000). "Less is more: Active learning with support vector machines." *International Conference on Machine Learning*, 2:4.

Settles, B (2009). "Active learning literature survey." Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Seung, HS, Opper, M, and Sompolinsky, H (1992). "Query by committee." *Proceedings of the ACM Workshop on Computational Learning Theory*, pp 287–294.

Studden, W. J. (1977) Optimal designs for integrated variance in polynomial regression, in *Statistical Decision Theory and Related Topics* II, eds. S. S. Gupta and D. S. Moore, New York: Academic Press.

Sutton, RS and Barto, AG. (2018). *Reinforcement learning: an introduction*, 2nd edition. Cambridge, MA: MIT Press.

Taguchi, G (1987). *System of experimental design.* UNIPUB/Kraus.

Tong, S and Chang, E (2001), "Support vector machine active learning for image retrieval." *Proceedings of the Ninth ACM International Conference on Multimedia*, ACM.

Tong, S, and Koller, D. (2001). "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research* 2: pp 45-66.

Tukey, JW (1958). "Bias and confidence in not quite large samples (abstract)." *The Annals of Mathematical Statistics* 29(2): pp 614.

Viswanathan, K, Sachdeva, S, Tomkins, A, and Ravi, S (2019). Improved Semi-Supervised Learning with Multiple Graphs. *Proceedings of Machine Learning Research* 89: pp 3032-3041.

Wald, A. (1943) On the efficient design of statistical investigations. *Annals of Mathematical Statistics* 14(2), pp 134-140.

Xu, Z, Akella, R and Zhang, Y (2007). "Incorporating diversity and density in active learning for relevance feedback," *Proceedings of the European Conference on IR Research*, pp 246–257. Springer-Verlag.

Zhang, W (1988). "Shift-invariant pattern recognition neural network and its optical architecture," *Proceedings of Annual Conference of the Japan Society of Applied Physics.*