Handbook of Big Data

Bühlmann Peter, Drineas Petros, Kane Michael, van der Laan Mark

Online Estimation of the Average Treatment Effect

Publication details
Sam Lendle
Published online on: 18 Feb 2016

How to cite :- Sam Lendle. 18 Feb 2016, Online Estimation of the Average Treatment Effect from: Handbook of Big Data CRC Press
Accessed on: 02 Jan 2024

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: https://www.routledgehandbooks.com/legal-notices/terms

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
Online Estimation of the Average Treatment Effect

Sam Lendle

CONTENTS

23.1 Introduction ................................. 429
23.2 Preliminaries ................................. 430
  23.2.1 Causal Parameter and Assumptions ................................. 430
  23.2.2 Statistical Model ................................................. 430
23.3 Batch Estimation of the ATE ................................. 431
23.4 Online One-Step Estimation of the ATE ................................. 432
23.5 Example and Simulation ................................. 433
  23.5.1 Data-Generating Distribution ................................................. 433
  23.5.2 SGD for Logistic Regression ................................................. 434
  23.5.3 Results ................................................. 434
23.6 Discussion ................................................. 436
References ................................................. 437

23.1 Introduction

Drawing causal inferences from observational data requires making strong assumptions about the causal process from which the data are generated, followed by a statistical analysis of the observational dataset. Though we must make causal assumptions, we often know little about the data-generating distribution. This means we generally cannot make strong statistical assumptions so we estimate a statistical parameter in a nonparametric or semiparametric statistical model. Semiparametric efficient estimators, that is, estimators that achieve the minimum asymptotic variance bound, such as augmented inverse probability of treatment weighted (A-IPTW) estimators [11] and targeted minimum loss-based estimators (TMLE) [15,18], have been developed for a variety of statistical parameters with applications in causal inference.

Typically, the computational efficiency and scalability of these estimators are not taken into account. Borrowing language from the large-scale machine learning literature, we call these batch estimators, because they process the entire dataset at one time. They rely on estimation of one or more parts of the data-generating distribution. With traditional statistical methods, estimating each of these parts may require many passes through the data that can quickly become impractical as the size of datasets grow.

In this chapter, we demonstrate an online method for estimating the average treatment effect (ATE) that is doubly robust and statistically efficient with only a single pass through the dataset. In Section 23.2, we introduce the observed data structure, the causal parameter, causal assumptions required for identification of the parameter, and the
statistical parameter. In Section 23.3, we review a batch method for estimating the ATE. In Section 23.4, we describe an online approach to estimating the ATE. In Section 23.5, we demonstrate the performance of the online estimator in simulations. We conclude with a discussion of extensions and future work in Section 23.6.

23.2 Preliminaries

23.2.1 Causal Parameter and Assumptions

We define the ATE using the counterfactual framework [14]. For a single observation, let \( Y_a \) be the counterfactual value of some outcome had exposure \( A \) been set to level \( a \) for \( a \in \{0, 1\} \). These values are called counterfactual because we can only observe a sample's outcome under the observed treatment that it received. Because at least one of \( Y_1 \) or \( Y_0 \) is not observed, we can never calculate for a given observation the value \( Y_1 - Y_0 \), which can be interpreted as the treatment effect for that observation. Under some conditions, however, we can estimate the average of this quantity, \( E_0(Y_1 - Y_0) \). This is known as the ATE, where \( E_0 \) denotes expectation with respect to the true distribution of the counterfactual random variables.

Before attempting to estimate the ATE, we first consider the structure of our observed dataset. Let \( O = (W, A, Y) \) be an observed sample where \( W \) is a vector of covariates measured before \( A \), a binary exposure or treatment, and \( Y \) is the observed outcome. We make the counterfactual consistency assumption that the observed outcome \( Y \) is equal to the counterfactual under the observed treatment \( A \). That is, we assume \( Y = Y_A \).

The ATE is a function of the distribution of counterfactuals \( Y_1 \) and \( Y_0 \). To estimate the ATE, we must be able to write it as a function of the distribution of the observed data. When we can do this, the ATE is said to be identifiable. To do this, we need to make some assumptions. The first is the randomization assumption where we assume \( A \perp (Y_1, Y_0) \mid W \). That is, we assume that if there are any common causes of the exposure \( A \) and outcome \( Y \), they are measured and included in \( W \). This is sometimes called the no unmeasured confounders assumption. This is called an untestable assumption, because it is not possible to test if this assumption is true using the observed data. Making the randomization assumption requires expert domain knowledge and careful study design. We also make the experimental treatment assignment assumption or positivity assumption: that for any value of \( W \), there is some possibility of either treatment being assigned. Formally, we assume \( 0 < P_0(A = 1 \mid W) < 1 \) almost everywhere. Under these assumptions, we can write the ATE as [12]

\[
E_0(Y_1 - Y_0) = E_0[E_0(Y \mid A = 1, W) - E_0(Y \mid A = 0, W)]
\]

23.2.2 Statistical Model

Now that we have posed some causal assumptions that allow us to write the ATE as a parameter of the distribution of the observed data, we need to specify a statistical model and target parameter. A statistical model \( \mathcal{M} \) is a set of possible probability distributions of the observed data.

Suppose that we observe a dataset of \( n \) independent and identically distributed observations, \( O_1, \ldots, O_n \), with distribution \( P_0 \). For a distribution \( P \in \mathcal{M} \), let \( p = dP/d\mu \).
be the density of $O$ with respect to some dominating measure $\mu$. We can factorize the density as
\[
p(o) = Q_W(w)G(a \mid w)Q_Y(y \mid a, w)
\]
where:
- $Q_W$ is the marginal density of $W$
- $G$ is the conditional probability that $A = a$, given $W$
- $Q_Y$ is the conditional density of $Y$, given $A$ and $W$

Let $Q = (Q_Y, Q_W)$ and $\bar{Q}(a, w) = E_{Q_Y}(Y \mid A = a, W = w)$. We can parameterize the model as $\mathcal{M} = \{P : Q \in \mathcal{Q}, G \in \mathcal{G}\}$.

The randomization assumption puts no restriction on the distribution of the observed data, and the positivity assumption only requires that $G(1 \mid W)$ be bounded away from 0 and 1. To ensure the true distribution $P_0$ is in $\mathcal{M}$, we make no additional assumptions on $Q$ so $\mathcal{Q}$ is nonparametric. In some cases, we may know something about the treatment mechanism $G$. For instance, we may know that the probability of treatment only depends on a subset of covariates. In that case, we put some restriction on the set $\mathcal{G}$ in addition to assuming $0 < G(1 \mid W) < 1$. In general, our model $\mathcal{M}$ is semiparametric.

We define the parameter mapping $\Psi : \mathcal{M} \to \mathbb{R}$ as
\[
\Psi(P) = E_P[E_P(Y \mid A = 1, W) - E_P(Y \mid A = 0, W)]
\]
where $E_P$ denotes expectation with respect to the distribution $P$. Let $\psi = \Psi(P)$ be the parameter mapping applied to distribution $P$. The target parameter we wish to estimate is $\psi_0 = \Psi(P_0)$, the parameter mapping applied to the true distribution. We note that $\Psi(P)$ only depends on $P$ through $Q$, so recognizing the abuse of notation, we sometimes write $\Psi(Q)$. Throughout, we will use subscript $n$ to denote that a quantity is an estimate based on $n$ observations, and subscript 0 to denote the truth. For example $\bar{Q}_n$ is an estimate of $Q_0$, defined as $E_{\bar{Q}_n}(Y \mid A = a, W = w)$, where $E_{\bar{Q}_n}$ is expectation with respect to the true distribution $P_0$.

## 23.3 Batch Estimation of the ATE

An asymptotically linear estimator is one that can be written as a sum of some mean zero function called an influence curve plus a small remainder. An efficient estimator is an estimator that achieves the minimum asymptotic variance among the class of regular estimators. In particular, an efficient estimator is asymptotically linear with the influence curve equal to the efficient influence curve, which depends on the particular parameter and model. The asymptotic variance of an efficient estimator is the variance of the efficient influence curve [2]. For our model $\mathcal{M}$ and parameter mapping $\Psi$, the efficient influence curve is given by
\[
D^*(P)(O) = \frac{2A - 1}{G(A \mid W)}(Y - Q(A, W)) + Q(1, W) - Q(0, W) - \Psi(Q).
\]

To denote the dependence of $D^*$ on $P$ through $Q$ and $G$, we sometimes also write $D^*(Q, G)$.

There are many ways to construct an efficient estimator, for example TMLE or A-IPTW. We now review the A-IPTW estimator in the batch setting. An A-IPTW estimate is calculated as
\[
\psi_n = \frac{1}{n} \sum_{i=1}^{n} \frac{2A_i - 1}{G_n(A_i \mid W_i)}(Y_i - Q_n(A_i, W_i)) + Q_n(1, W_i) - Q_n(0, W_i)
\]
where $\bar{Q}_n$ and $G_n$ are estimates of $Q_0$ and $G_0$, respectively. The A-IPTW estimator treats $D^*(P)$ as an estimating function in $\psi$ and nuisance parameters $Q$ and $G$ and solves for $\psi$. The A-IPTW is also a one-step estimator, which starts with a plug in estimator for $\psi_0$ and takes a step in the direction of the efficient influence curve. That is,

$$\psi_n = \Psi(Q_n) + \frac{1}{n} \sum_{i=1}^{n} D^*(Q_n, G_n)$$

where

$$\Psi(Q_n) = \frac{1}{n} \sum_{i=1}^{n} Q_n(1, W_i) - Q_n(0, W_i).$$

Under regularity conditions, the A-IPTW estimator is efficient if both $\bar{Q}_n$ and $G_n$ are consistent for $Q_0$ and $G_0$, respectively. Additionally, the A-IPTW estimator is doubly robust, meaning that if either of $\bar{Q}_n$ or $G_n$ is consistent, then $\psi_n$ is consistent.

### 23.4 Online One-Step Estimation of the ATE

The batch A-IPTW in Section 23.3 has some nice statistical properties. In particular, it is efficient and doubly robust. Our goal now is to construct an estimator of $\psi_0$ that has these same properties, but we only want to make a single pass through the dataset. Additionally, we only want to process a relatively small number of observations, a minibatch, at one time.

Let $0 = n_0 < n_1 < \cdots < n_K = n$. Here $n = n_K$ represents the total sample size, and $n_j$ is the sample size accumulated up to minibatch $j$. Suppose $n_j - n_{j-1}$ is bounded for all $j$, and for simplicity let $n_j - n_{j-1} = m$ be constant. Let $O_{n_i:n_j}$ denote the observations $O_{n_{i+1}}, O_{n_{i+2}}, \ldots, O_{n_j}$ for $i < j$.

In Section 23.3, we computed an estimate of $\psi_0$ with estimates $\bar{Q}_n$ and $G_n$, which were fit on the full dataset. Now suppose we have estimates $\bar{Q}_{n_{j-1}}$ and $G_{n_{j-1}}$ for $Q_0$ and $G_0$, respectively, that are based on observations $O_{n_0:n_{j-1}}$. We will return to the problem of computing $\bar{Q}_{n_{j-1}}$ and $G_{n_{j-1}}$ later. Using those estimates of $\bar{Q}_0$ and $G_0$, we compute a new estimate of $\psi_0$ on the next minibatch as

$$\psi_{n_{j-1}:n_j} = \frac{1}{m} \sum_{i=n_{j-1}+1}^{n_j} \left[ \frac{2A_i - 1}{G_{n_{j-1}}(A_i, W_i)} (Y_i - \bar{Q}_{n_{j-1}}(A_i, W_i)) + Q_{n_{j-1}}(1, W_i) - Q_{n_{j-1}}(0, W_i) \right].$$

That is, $\psi_{n_{j-1}:n_j}$ is a one-step estimator computed on minibatch $j$ using initial estimates of $\bar{Q}_0$ and $G_0$ from the previous minibatches. We compute the final estimate of $\psi_0$ by taking the mean of estimates $\psi_{n_{j-1}:n_j}$ from each minibatch. Let

$$\psi_{n_K} = \frac{1}{K} \sum_{j=1}^{K} \psi_{n_{j-1}:n_j},$$

and call this procedure the online one-step (OLOS) estimator.
Under regularity conditions and if $\bar{Q}_{nK}$ and $G_{nK}$ converge faster than rate $n^{-1/4}_K$ and are both consistent for $Q_0$ and $G_0$, then the OLOS estimator is asymptotically efficient as $K \to \infty$. Additionally, the OLOS estimator has the double robustness property, so if either of $\bar{Q}_{nK}$ or $G_{nK}$ is consistent, then $\psi_{nK}$ is consistent [16, Theorem 1].

We now turn to estimating $Q_0$ and $G_0$, both of which are conditional means. To be truly scalable, ideally we want an estimation procedure that has a constant computational time and storage per minibatch up to $K$, but we need that the estimates converge fast enough as $K \to \infty$. This rules out estimates of $Q_0$ and $G_0$ that are fit on data from one or a fixed number of minibatches. Stochastic gradient descent-based methods, however, can achieve an appropriate rate of convergence in some circumstances [8,20].

Stochastic gradient descent (SGD) is an optimization procedure similar to traditional batch gradient descent, where the gradient of the objective function for the whole dataset is replaced by the gradient of the objective function at a single observation (or a minibatch). The convergence rate to the empirical optimum of the objective function in terms of number of iterations is very poor relative to batch gradient descent. However, a single iteration of SGD or minibatch gradient descent takes constant time regardless of the size of the dataset, while a single iteration of batch gradient descent takes $O(n)$ time [4].

SGD can be used to fit the parameters of generalized linear models (GLMs). Despite the slow convergence rate, with an appropriately chosen step size, the parameters of a GLM fit with SGD can achieve $\sqrt{n_K}$ consistency in a single pass [8]. If curvature information is taken into account, parameters fit by so-called second-order SGD can achieve the same variance as directly optimizing the empirical objective function [8], but this is often computationally infeasible and rarely done in practice [4]. We note that the class of models for which SGD will obtain $\sqrt{n_K}$ consistency is larger than just generalized linear models that we use as an example here [6].

Averaged stochastic gradient descent (ASGD) is a variant of SGD where parameter estimates are computed with SGD and then averaged. With an appropriate step size, parameters fit by ASGD have been shown to achieve the same variance in a single pass as those fit by directly optimizing the objective function [10,20]. ASGD is much simpler to implement than second-order SGD but has not been popular in practice. This may be because it takes a very large sample size to reach the asymptotic regime [20].

Some variants of SGD allow for a step size for each parameter, such as SGD-QN [3], Adagrad [7], and Adadelta [21], which tend to work well in practice. For information about other variants and implementation details, see [5]. We provide a simple concrete implementation of SGD in Section 23.5.2.

Despite the drawbacks, if $Q_0$ and $G_0$ can be well approximated by GLMs, SGD-based optimization routines are a good way to compute the estimates in one pass.

23.5 Example and Simulation

23.5.1 Data-Generating Distribution

We evaluate the statistical performance of the OLOS estimator and discuss practical implementation details in the context of a simulation study. For each observation, $W$ is generated by making $p = 2000$ independent draws from a uniform distribution on $[-1, 1]$. Given $W$, $A$ is drawn from a Bernoulli distribution with success probability

$$\frac{1}{1 + \exp(-0.75Z)}$$
where $Z$ is the sum of the first four components of $W$. Finally, $Y$ is drawn from a Bernoulli distribution with success probability

$$\frac{1}{1 + \exp(1 + 0.5Z - 0.3A)}.$$ 

The first four components of $W$ are related to both $A$ and $Y$ so they are confounders. In general, failing to adjust for confounding variables will result in a biased estimate of the ATE. The other components of $W$ are not confounders and just add noise to the data. The true parameter $\psi_0$ is approximately 0.0602. A naive estimate of $\psi_0$ that fails to adjust for $W$ is approximately $-0.026$. In this simulation, confounding is strong enough that a naive estimate of $\psi_0$ that does not adjust for $W$ will be quite biased.

### 23.5.2 SGD for Logistic Regression

We estimate both $G_0$ and $\bar{Q}_0$ with main term logistic regression with parameters computed with minibatch gradient descent.

We estimate $G_0$ with a logistic regression model with main terms for each component of $W$, and $\bar{Q}_0$ with a logistic regression model with main terms for $A$ and each component of $W$. To investigate the double robustness of the estimator, we misspecify models for $G_0$ and $\bar{Q}_0$ by leaving out $W$. The parameters for both $\bar{Q}_0$ and $G_0$ are computed with minibatch gradient descent.

For concreteness, we describe the SGD algorithm for logistic regression to estimate $\bar{Q}_0$. Estimating $G_0$ is analogous. Let $X_i$ be a $d$-dimensional vector of predictor variables including a constant, $A_i$, and when the correctly specified, $W_i$. Let $\theta$ be a vector of the same dimension of $X_i$. For a particular $\theta$, $\bar{Q}(A_i,W_i)$ is computed as

$$\frac{1}{1 + \exp(-\theta'X_i)}.$$ 

Initializing $\theta_0$ to a vector of 0s, $\theta_j$ at minibatch $j$ is computed as

$$\theta_j = \theta_{j-1} - \eta_j \frac{1}{m} \sum_{i=n_{j-1}+1}^{n_j} \left( Y_i - \frac{1}{1 + \exp(-\theta_{j-1}'X_i)} \right) X_i$$

where $\eta_j$ is the step size.

We use minibatches of size 100 and step size

$$\eta_j = \frac{0.05}{1 + j0.05\alpha}$$

for minibatch $j$ where $\alpha$ is 0.01 for $\bar{Q}_0$ and 0.1 for $G_0$.

### 23.5.3 Results

We constructed datasets of size $n = 10^6$. Using double precision floating point numbers for storage, a single dataset is more than 15 GB uncompressed, so computing a batch estimate of $\psi_0$ on a typical personal computer is non-trivial. For each simulated dataset, the OLOS estimator is run using either correctly or incorrectly specified initial estimators of $\bar{Q}_0$ and $G_0$. The current estimate $\psi_{n_j}$ is recorded at each minibatch $j$. 
To investigate the performance of the estimator as a function of sample size, we compute the observed bias and variance at each minibatch across 1000 simulated datasets. Figure 23.1 shows the bias as a function of sample size scaled by the square root of the accumulated sample size.

For an estimator to be asymptotically linear, the bias should be \( o_P(1/\sqrt{n}) \). This means that we want to see that the scaled bias converge to zero as \( n \) increases. We see that in all cases, bias is quite high when sample size is small. This is because estimates of \( \bar{Q}_0 \) and \( G_0 \) are far from the truth for the first few hundred minibatches. When only one of \( Q_0 \) or \( G_0 \) is correctly specified, bias moves toward zero slowly, because we are relying on the double robustness of the OLOS. When both \( \bar{Q}_0 \) and \( G_0 \) are estimated by a correctly specified model, the scaled bias approaches 0 reasonably quickly.

Bias is high in early minibatches and it takes many steps to recover. To avoid some of the initial bias, it is sometimes helpful to discard estimates of \( \psi_0 \) in early minibatches and compute the final estimate of \( \psi_0 \) as

\[
\frac{1}{K-K_0} \sum_{j=K_0+1}^{K} \psi_{n_{j-1:n_j}}
\]

where \( K_0 \) is the number of discarded minibatches. This allows estimates of \( \bar{Q}_0 \) and \( G_0 \) to get warmed up with the first \( K_0 \) minibatches, and in experiments we see this can reduce bias substantially. (Results not shown.)

Figure 23.2 plots the variance of the OLOS estimates scaled by accumulated sample size. The red line denotes the variance of the efficient influence curve at \( P_0 \), which is approximately 0.95 in this simulation. If an estimator is efficient, its variance scaled by the sample size will approach the variance of the efficient influence curve as \( n \to \infty \). As expected, we see that when both \( \bar{Q}_0 \) and \( G_0 \) are consistently estimated, the OLOS estimator is efficient. When only one of \( \bar{Q}_0 \) or \( G_0 \) is consistently estimated, the OLOS estimator is not efficient in general, but in this simulation we see that the variance is close to the efficiency bound.
23.6 Discussion

The OLOS estimator gives us a way to estimate the ATE in a single pass through a dataset, getting running estimates along the way. Although bias can be high when a relatively small amount of data is processed, asymptotically we can do as well as we would with a batch estimator that typically requires many passes through the dataset.

In the example in Section 23.5, we used main term logistic regression to estimate both $Q_0$ and $G_0$. This can easily be extended to any user-specified basis function of $W$. However, generalized linear models will not include the true $Q_0$ and $G_0$ in general, and more flexible online estimators are needed.

SGD-based optimization is a natural way to turn batch algorithms into online algorithms and is applicable to a wide range of estimation methods. One such class of estimators is multilayer neural networks, which are incredibly flexible. Typically, because of the computational complexity of neural networks, they are fit with SGD in the batch setting with multiple passes through the dataset. In principle, one could fit a neural network as an online estimator, though a single pass through even very large datasets may not be enough for the neural network to reach the asymptotic regime.

There are other examples in the literature of estimators that have been extended to the online setting that are not based on SGD. Examples include bagging and boosting [9], random forests [1,13], and generalized additive models [19].

It is not clear when one estimator should be chosen over another. There are also typically some choices of tuning parameters, for example level of regularization. For online methods, fitting procedures may have their own set of choices. For example, there are a number of variants of SGD for optimization, each of which has some tuning parameters. In the batch
setting, cross-validation can be used to select between many estimators, or a model stacking approach such as the super learner algorithm [17] can be used to choose a combination of estimators. A similar approach can be taken in an online setting by using each new minibatch as an independent validation set to estimate out-of-sample performance before updating an estimator with that minibatch. In this way, many estimators with different choices of tuning parameters can be fit concurrently, and the best or a combination can be selected based on the estimated out-of-sample performance.

References


