Targeted Learning for High-Dimensional Variable Importance

Alan Hubbard, Nima Hejazi, Wilson Cai, Anna Decker

Division of Biostatistics
University of California, Berkeley

July 27, 2016

for Centre de Recherches Mathématiques (CRM) workshop on
"Statistical Causal Inference and its Applications to Genetics"
Outline

1. Introduction
2. Statistical Roadmap
3. Causal Inference
4. Statistical Model and SuperLearning
5. TMLE
6. LIMMA
7. Data Analysis
Study of the mechanisms underlying inflammation, coagulation, and host response to injury (Tompkins 2015).

Data used were a subset of the GLUE cohort (n = 167), who had gene expression measured (Affy U133 plus 2) in WBC’s measured (if possible) within 12 hours, and at 1, 4, 7, 14 and 28 days.

Interested in how expression at various times is related to outcomes (such multiple organ failure; MOF) afterwards.

So, parameter like

\[ \Psi(P)(t) = E(Y_{a_{hi}}(t) - Y_{a_{low}}(t)) \]
Both Expression and Clinical Measurements on Patients

- Heart rate
- Blood pressure
- Oxygen saturation
- Pulmonary artery pressure
- Intracranial pressure
- Temperature
- Cardiac output
- Sedation level
- Inspired oxygen
- Tidal volume
- Peak pressure
- End expiratory pressure
- Respiratory rate
- Ventilation mode
- Tissue oxygen
- Intake Output
- Blood products
- IV fluids
- Medications
- Treatments
- Nursing Documentation

Joint Variable Importance Measures

Use a combination of causal inference to motivate estimand, and

- SuperLearning (van der Laan et al. 2007),
- TMLE (van der Laan and Rose 2011),
- LIMMA (Smyth 2004),

to produce joint inference on variable importance adjusting for other covariates.
Outline

1. Introduction
2. Statistical Roadmap
3. Causal Inference
4. Statistical Model and SuperLearning
5. TMLE
6. LIMMA
7. Data Analysis
Stay on the Road(map)
Roadmap

Design
Science
Causal Model
Identifiability
Cross-validation
Oracle Inequality
Efficiency Theory
Pathwise
Differentiability
Empirical Processes

Targeted Learning
Causal Inference for Observational and Experimental Data
Mark J. van der Laan, Sherri Rose
Springer Series in Statistics

SCM for Point Treatment

- \( X = \{W, A, Y\} \)
- Errors: \( U = (U_W, U_A, U_Y) \)
- Structural Equations (nature):
  - \( W = f_W(U_W) \)
  - \( A = f_A(W, U_A) \)
  - \( Y = f_Y(W, A, U_Y) \)

Distribution of \((U, X)\) generated by:
1. Draw \( U \) from \( P_U \)
2. Generate \( W \) via \( f_W \) with input \( U_W \).
3. Generate \( A \) via \( f_A \) with inputs \((W, U_A)\)
4. Generate \( Y \) via \( f_Y \) with inputs \((W, A, U_Y)\)
Let’s say, keeping the above tentative causal model, we have the equality (skipping identifiability steps):

$$\Psi(P_X) = E(Y_{a_1} - Y_{a_0}) = \text{assumptions}$$

$$E_{0,W}\{E_0(Y \mid A = a_1, W) - E_0(Y \mid A = a_0, W)\} = \Psi(P_0)$$

for discrete $A$ with chosen comparison levels $(a_0, a_1)$.

Now, generalize the data above to situation with $O = (W, A, Y)$, where $A$ is vector of biomarkers: $A = (A_1, \ldots, A_g, \ldots, A_G)$.

Assume estimating $\Psi_g(P_0)$ corresponding to each of the $A_g, g = 1, \ldots, G$.

Estimator based on original work by Robins and Rotnitzky (1992).

Like Chambaz et al. (2012), Ritter et al. (2014), van der Laan (2006), etc., we propose using estimands motivated by causal inference for variable importance measures in high-dimensional contexts.
Outline

1. Introduction
2. Statistical Roadmap
3. Causal Inference
4. **Statistical Model and SuperLearning**
5. TMLE
6. LIMMA
7. Data Analysis
In our example, we wish to estimate: \( Q_0 = E_0(Y | A, W) \).

Before we can choose a "best" algorithm to estimate this regression function, we must have a way to define what "best" means, made explicit by loss function.

Data structure: \( O = (W, A, Y) \sim P_0 \) → distribution \( P_n \) which places probability \( 1/n \) on each observed \( O_i, i = 1, \ldots, n \).

"Best" algorithm defined in terms of a loss function (for candidate function \( Q \) when applied to an observation \( O \)).

\[
L : (O, Q) \rightarrow L(O, Q) \in \mathbb{R}.
\]

It is a function of the random variable \( O \) and parameter value \( Q \).

Example: \( L_2 \) squared error (or quadratic) loss function:

\[
L(O, Q) = (Y - Q(A, W))^2.
\]
We define our parameter of interest, $\bar{Q}_0 = E_0(Y \mid A, W)$, as the minimizer of the expected squared error loss:

$$\bar{Q}_0 = \arg \min Q E_0 L(O, \bar{Q}),$$

where $L(O, \bar{Q}) = (Y - \bar{Q}(A, W))^2$.

$E_0 L(O, \bar{Q})$ (the risk), is minimized at the optimal choice of $\bar{Q}_0$. 

Unbiased Estimate of Risk: Cross-Validation

- \( V \)-fold cross-validation: partition the sample of \( n \) observations \( O_1, \ldots, O_n \) in training and corresponding validation sets.
- Cross-validation is used to select our “best” algorithm among a library.
- In addition, we also use cross-validation to evaluate the overall performance of the super learner itself.
• In $V$-fold cross-validation, our observed data $O_1, \ldots, O_n$ is referred to as the learning set.

• Estimate our parameter of interest by partitioning into $V$ sets of size $\approx \frac{n}{V}$.

• For any given fold, $V - 1$ sets will comprise the training set and the remaining 1 set is the validation set.

• The observations in the training set are used to construct (or train) the candidate estimators.

• The observations in the validation set are used to assess the performance (i.e., risk) of the candidate algorithms.
The validation set rotates $V$ times such that each set is used as the validation set once.
Suppose a researcher is interested in using three different parametric statistical models to estimate $E_0(Y \mid A, W)$.

We can use these algorithms to build a library of algorithms and select the one with the smallest (honest) cross-validated risk.
1. Input data and a collection of algorithms.

2. Split data into 10 blocks.

3. Fit each of the 3 algorithms on the training set (non-shaded blocks).

4. Predict the estimated probabilities of death (Z) using the validation set (shaded block) for each algorithm, based on the corresponding training set fit.

5. Calculate estimated risk within each validation set for each algorithm using Z. Average the risks across validation sets resulting in one estimated cross-validated risk for each algorithm.

6. The discrete super learner algorithm selects the algorithm with the smallest cross validated risk.
Oracle Properties

- Assume $K$ algorithms in the library of algorithms.
- The oracle selector chooses the algorithm with the smallest risk under true data-generating distribution, $P_0$.
- However, the oracle selector is unknown since it depends on both the observed data and $P_0$.
- Oracle Inequality (van der Laan et al. (2004)) suggests discrete super learner performs as well as the oracle selector, up to a second order term.
- The loss function must be bounded, and then we will perform as well as the algorithm that is the risk minimizer of the expected loss function.
- The number of algorithms, $K$, in the library can grow with sample size.
Stacking algorithm that takes multiple algorithms as inputs can outperform a single algorithm in realistic non-parametric and semi-parametric statistical models.

- Define parameter of interest, \( \bar{Q}_0 = E_0(Y \mid A, W) \), as the minimizer of the expected squared error loss:

  \[
  \bar{Q}_0 = \arg \min_{\bar{Q}} E_0 L(O, \bar{Q}),
  \]

  where \( L(O, \bar{Q}) = (Y - \bar{Q}(A, W))^2 \).

- \( E_0 L(O, \bar{Q}) \) (risk), evaluates the candidate \( \bar{Q} \), and it is minimized at the optimal choice of \( \bar{Q}_0 \).
Super Learner: How it works

- Provides a family of weighted combinations of the algorithms in the library of learners, indexed by the weight vector $\alpha$.

- The family of weighted combinations includes only those $\alpha$-vectors that have a sum equal to one, and where each weight is positive or zero.

- Example for binary outcome:

$$P_n(Y = 1 \mid Z) = \text{expit} \left( \alpha_{a,n}Z_a + \alpha_{b,n}Z_b + \ldots + \alpha_{p,n}Z_p \right)$$

where $Z$ are the candidate learner predictions.

- The (cross-validated) probabilities of death ($Z$) for each algorithm are used as inputs in a working (statistical) model to predict the outcome $Y$.

- Deriving weights: formulated as a regression of the outcomes $Y$ on the predicted values of the algorithms ($Z$).
Ensemble Super Learner: How well it works

The super learner improves asymptotically on the discrete super learner by working with a larger library.

- Asymptotic results: in realistic scenarios (where none of the algorithms are a correctly specified parametric model), the discrete super learner performs asymptotically as well as the oracle.
- When collection of algorithms contains correctly specified parametric statistical model, the super learner will approximate the truth as fast as the parametric statistical model, although it will be more variable.
A maximum likelihood estimator for a parametric statistical model \( \{ p_\theta : \theta \} \) is defined as a maximizer over all densities in the parametric statistical model of the empirical mean of the log density:

\[
\theta_n = \arg \max_\theta \sum_{i=1}^n \log p_\theta(O_i).
\]

This discussion can be equally applied to the case where \( L(p)(O) \) is replaced by any other loss function \( L(Q) \) for a relevant part \( Q_0 \) of \( p_0 \), satisfying that \( E_0 L(Q_0)(O) \leq E_0 L(Q)(O) \) for each possible \( Q \).
MLE - Substitution Estimator

\[ \Psi(P_0) \] for the causal risk difference can be written as the g-formula:

\[
\Psi(P_0) = \sum_w \left[ \sum_y y P_0(Y = y \mid A = a_h, W = w) \right. \\
- \left. \sum_y y P_0(Y = y \mid A = a_l, W = w) \right] P_0(W = w),
\]

where

\[
P_0(Y = y \mid A = a, W = w) = \frac{P_0(W = w, A = a, Y = y)}{\sum_y P_0(W = w, A = a, Y = y)}
\]

is the conditional probability distribution of \( Y = y \), given \( A = a \), \( W = w \), and

\[
P_0(W = w) = \sum_{y,a} P_0(W = w, A = a, Y = y).
\]
**MLE - Substitution**

- Maximum-likelihood-based substitution estimators of the g-formula are obtained by substitution of a maximum-likelihood-based estimator of $Q_0$ into the parameter mapping $\Psi(Q_0)$.

- The marginal distribution of $W$ can be estimated with the nonparametric maximum likelihood estimator, which happens to be the empirical distribution that puts mass $1/n$ on each $W_i$: $i = 1, \ldots, n$.

\[
\psi_n = \Psi(Q_n) = \frac{1}{n} \sum_{i=1}^{n} \{ \bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_i) \},
\]

where this estimate is obtained by plugging in $Q_n = (\bar{Q}_n, Q_{W,n})$ into the mapping $\Psi$. 
MLE - fitting $Q_0$

- **MLE using regression in a parametric working model.** $\bar{Q}_0(A, W)$ is estimated using regression in a parametric working (statistical) model and plugged into the formula given previously.

- **ML-based super learning.** Estimate $\bar{Q}_0$ with the super learner, in which the collection of estimators may include stratified maximum likelihood estimators, maximum likelihood estimators based on dimension reductions implied by the propensity score, and maximum likelihood estimators based on parametric working models, beyond many other machine learning algorithms for estimation of $\bar{Q}_0$. 


TMLE (van der Laan and Rubin 2006) Produces a well-defined, unbiased, efficient substitution estimator of target parameters of a data-generating distribution.

- It is generally an iterative procedure (though there is now a general one-step; van der Laan and Gruber (2016)) that updates an initial (super learner) estimate of the relevant part $Q_0$ of the data generating distribution $P_0$, possibly using an estimate of a nuisance parameter $g_0$.

- Like corresponding A-IPTW estimators (Robins and Rotnitzky 1992), removes asymptotic residual bias of initial estimator for the target parameter, if it uses a consistent estimator of $g_0$, thus Doubly Robust.

- If initial estimator was consistent for the target parameter, the additional fitting of the data in the targeting step may remove finite sample bias, and preserves consistency property of the initial estimator.
Natural use of machine learning methods for the estimation of both $Q_0$ and $g_0$.

Focuses effort to achieve minimal bias and asymptotic semi-parametric efficiency bound for the variance, but still get inference (with some assumptions).
TMLE Algorithm

This chapter focuses on understanding targeted maximum likelihood estimation. Recall that TMLE is a two-step procedure where one first obtains an estimate of the data-generating distribution \( \Psi \) and the treatment mechanism. The double robustness of TMLE has important implications in both randomized controlled trials and observational studies, with potential reductions in bias and gains in efficiency.

The procedure is double robust and can incorporate data-adaptive likelihood based estimation procedures to estimate variance trade-offs. The second stage updates this initial fit in a step targeted towards making an optimal bias- and variance trade-off.

The initial estimator of the probability distribution of the data is denoted by \( P_n \) and the true probability distribution is denoted by \( P_0 \). The targeted estimator of the probability distribution of the data is denoted by \( \tilde{P}_n \).

INPUTS

- Observed data random variables \( O_1, \ldots, O_n \)
- Target parameter map \( \Psi() \)

STATISTICAL MODEL

Set of possible probability distributions of the data

VALUES OF TARGET PARAMETER

Values mapped to the real line with better estimates closer to the truth

True value (estimand) of target parameter

Initial estimator of the probability distribution of the data

True probability distribution

Targeted estimator of the probability distribution of the data
Example: TMLE for the Average Causal Effect

NPSEM/SCM for a point treatment data structure with missing outcome

\[ W = f_W(U_W), \]
\[ A = f_A(W, U_A), \]
\[ \Delta = f_A(W, A, U_\Delta), \]
\[ Y = f_Y(W, A, \Delta, U_Y). \]

We can now define counterfactuals \( Y_{1,1} \) and \( Y_{0,1} \) corresponding with interventions setting \( A \) and \( \Delta \).

The additive causal effect \( EY_1 - EY_0 \) equals:
\[
\psi(P) = E[E(Y \mid A = 1, \Delta = 1, W) - E(Y \mid A = 0, \Delta, 1, W)]
\]
Example: TMLE for "Causal" Risk Difference:

\[ E_{0,W}\{ E_0(Y \mid A = a_h, W) - E_0(Y \mid A = a_l, W) \} \]

- Generate an initial estimator of \( P^0_n \) of \( P \); we estimate \( E(Y \mid A, \Delta = 1, W) \).
- Fluctuate this initial estimator with a logistic regression:

\[
\logit P^0_n(\epsilon)(Y = 1 \mid A, \Delta = 1, W) = \logit P^0_n(Y = 1 \mid A, \Delta = 1, W) + \epsilon h
\]

where

\[
h(A, W) = \frac{1}{\Pi(A, W)} \left( \frac{I(A = a_h)}{g(a_h \mid W)} - \frac{I(A = a_l)}{g(a_l \mid W)} \right)
\]

and

\[
g(a \mid W) = P(A = a \mid W) \text{ Treatment Mechanism}
\]
\[
\Pi(A, W) = P(\Delta = 1 \mid A, W) \text{ Missingness Mechanism}.
\]
Let $\epsilon_n$ be the maximum likelihood estimator and

$$P_n^* = P_n^0(\epsilon_n).$$

The TMLE is given by $\Psi(P_n^*)$.

Specifically in the case of the risk difference:

$$\logit \bar{Q}_n^1(A, W) = \logit \bar{Q}_n^n(A, W) + \epsilon_n H_n^*(A, W).$$

This parametric working model incorporated information from $g_n$, through $H_n^*(A, W)$, into an updated regression.
Inference (Standard errors) via Influence Curve (IC)

- Influence curve for estimator is:

\[ IC_n(O_i) = \left( \frac{I(A_i = a_h)}{g_n(a_h \mid W_i)} - \frac{I(A_i = a_l)}{g_n(a_l \mid W_i)} \right) (Y - \bar{Q}_n^{1}(A_i, W_i)) 
\]

\[ + \bar{Q}_n^{1}(a_h, W_i) - \bar{Q}_n^{1}(a_l, W_i) - \psi_{TMLE,n} \]

- Sample variance of the estimated influence curve:

\[ S^2(IC_n) = \frac{1}{n} \sum_{i=1}^{n} (IC_n(o_i))^2. \]

- Use sample variance to estimate the standard error of our estimator:

\[ SE_n = \sqrt{\frac{S^2(IC_n)}{n}}. \]

- Use this to derive uncertainty measures (p-values, confidence intervals, etc.).
Consider this is repeated for $j = 1, \ldots, J$ different biomarkers, so that one has, for each $j$:

$$\Psi_j(Q^*_j,n), S^2_j(IC_j,n)$$

or estimate of variable importance and standard error for all $J$.

Propose an existing joint-inferential procedure that can add some finite-sample robustness to an estimator that can be highly variable.
Outline

1. Introduction
2. Statistical Roadmap
3. Causal Inference
4. Statistical Model and SuperLearning
5. TMLE
6. LIMMA
7. Data Analysis
Thus, one can define a standard t-test statistic for a general (asymptotically linear) parameter estimate (over $j = 1, \ldots, J$) as:

$$t_j = \frac{\sqrt{n}(\Psi_j(P_n) - \psi_0)}{S_j(IC_{j,n})}$$

Consider the moderated t-statistic proposed by Smyth (2005):

$$\tilde{t}_j = \frac{\sqrt{n}(\Psi_j(P_n) - \psi_0)}{\tilde{S}_j^2}$$

where the posterior estimate of the variance of the influence curve is

$$\tilde{S}_j^2 = \frac{d_0 S_0^2 + d_j S_j^2(IC_{j,n})}{d_0 + d_j}$$
Treat like one-sample problem estimate of parameter with associated SE from IC.

Just need to get estimate for each $j$ as well as the plug-in IC for every observation for that $j$ and repeat for all $j$.

Transform data original $Jxn$ matrix where new entries are:

$$Y_{j,i}^* = IC_{j,n}(O_i; P_n) + \Psi_j(P_n)$$

Since the average of the $IC_{j,n}$ across the columns (units) for a single $j$ will be 0, the average of this transform will be the original estimate $\Psi_j(P_n)$.

For simplicity assume the null value is $\psi_0 = 0$ for all $j$. Then, running *limma* package on this transform, $Y_{j,i}^*$, will generate multiple testing corrections based on presented above for $\tilde{t}_j$. 
Why LIMMA approach in this context?

- Often times these analyses based on relatively small samples.
- want data-adaptive estimate but at least with standard implementation of these estimates (estimation equation, substitution,...), the SE’s can be non-robust.
- practically, one can get "significant" estimates of variable importance measures that are driven by poorly and underestimated $S_j^2(\text{IC}_{j,n})$.
- LIMMA shrinks these $S_j^2(\text{IC}_{j,n})$ by making them bigger and thus takes biomarkers with small parameter estimates but very small $S_j^2(\text{IC}_{j,n})$ out of statistical significance.
- Also, just seems to work very well...
- **Goal:** Ascertain relative variable importance based on estimand for adjusted risk difference discussed above.

- Because expression generally fairly low, and wanted to discretize expression, used a cut-off that separated (roughly) some expression from background.

- Though there are thousands of expression values, just focus on a few genes known to be involved in coagulation.
GLUE Study Expression Data - coagulation pathway genes

Box plots for genes involved in the coagulation pathway, showing the distribution of expression levels. The box plots are labeled with gene names such as `f2`, `f3`, `f7`, etc. A heatmap is also present, indicating the correlation between different genes.
Substitution based on Simple Parametric Model vs. SL


References


