Optimally Combining Outcomes to Improve Prediction

David Benkeser
On behalf of the HBGD community & HBGDki
benkeser@berkeley.edu

Acknowledgments

Collaborators:
Mark van der Laan, Alan Hubbard, Ben Arnold, Jack Colford, Andrew Mertens, Oleg Sofyrgin, Jonathan French, Aryeh Stein, Shasha Jumbe

Funding:
Bill and Melinda Gates Foundation OPP1147962

Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Motivation

The observed data are n i.i.d. copies of $O = (X, Y)$.

- X = D covariates
- Y = J standardized outcomes

Together Y represents an unmeasured outcome of interest.

- Inflammation in CVD
- Immune response in HIV
- Subject area tests in cognitive development

Researchers are interested in prediction of unmeasured outcome using X.

Motivation

Can we use neonatal information to predict neurocognitive outcomes later in life?

- Early identification of at-risk children.

What covariates are important for prediction?

- Informs what information to collect to screen children.

Motivation: PROBIT

The Promotion of Breastfeeding Intervention Trial enrolled pregnant mothers in 1996–97 (Kramer et al, 2001).

Group name	**Variables**
Breastfeeding	breastfeeding encouraged
Socioeconomic	household animals
Parental	age, height, weight, education, siblings employment status
Birth	gestational age, Apgar score
Growth	WAZ, HAZ, HCAZ (0,1,2,3,6,9,12 months)
Other	mother smoked during pregnancy, mother drank during pregnancy
WASI score (age 6)	Matrix, Block, Vocabulary, Similarities

Motivation: CLHNS

The Cebu Longitudinal Health and Nutrition Survey enrolled pregnant mothers in 1983–84 (Feranil et al, 2008).

Group name	Variables
Health care	health care access, preventive health care
Household	child:adult ratio, child dependency ratio, crowding index, urban score
Socioeconomic	total income, SES
Water/sanitation	sanitation, access to clean water
Parental	mother age, father age, mother height, mother education (years), father education (years), marital status, mother age first child, parity
Growth	WAZ, HAZ (0,6,12,18,24 months)
Other	mother smoked during pregnancy, child's sex, gestational age at birth
Achievement tests (age 11)	Math, Cebuano, English

Motivation

How to combine test scores to measure “neurocognition”?

Give equal weight to all scores?

- What if some scores are noisy or not related to covariates?

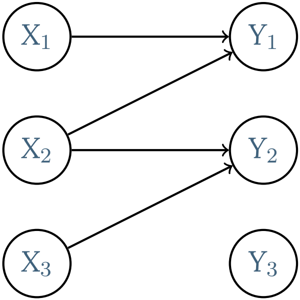
PCA or factor analysis to combine scores?

- Not related to scientific goal.

Use the combination that is predicted most accurately.

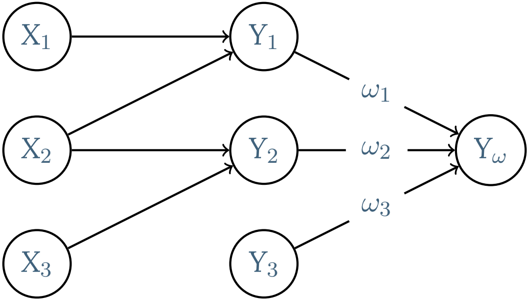
Motivation

Consider this simple associative directed acyclic graph.



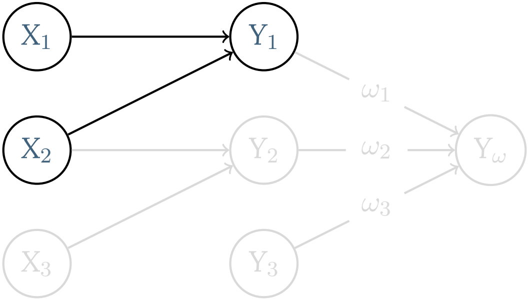
Motivation

Let $Y_\omega = \sum_{j=1}^{J} \omega_j Y_j$, with $\omega_j \geq 0$ for all j and $\sum_{j=1}^{J} \omega_j = 1$.



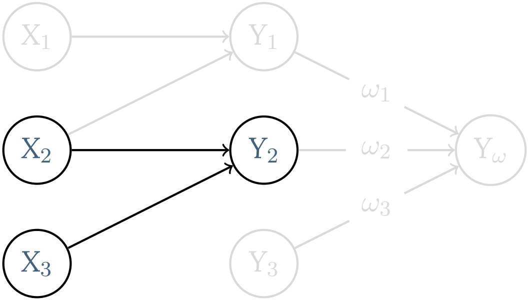
Motivation

Predicting only Y_1 ignores X_3's association with Y.



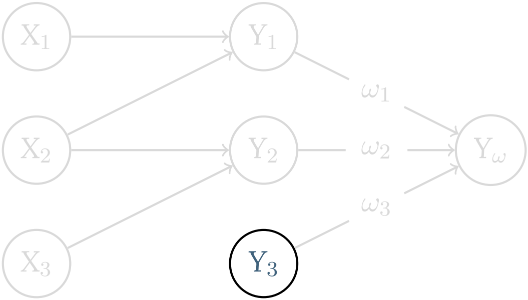
Motivation

Predicting only Y_2 ignores X_1's association with Y.



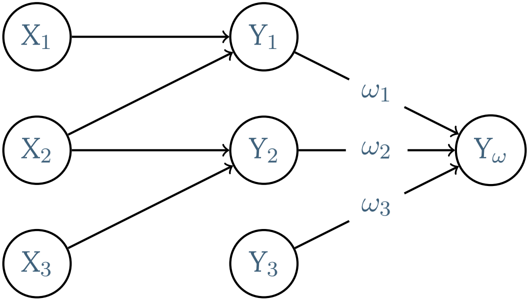
Motivation

Predicting Y_3 adds noise, wastes type-1 error.



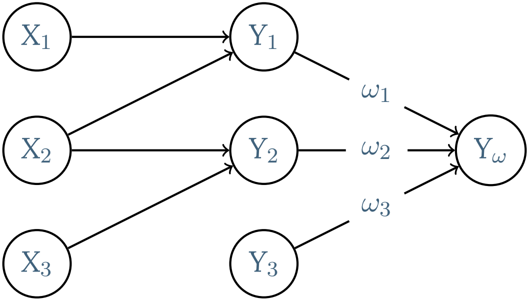
Motivation

Predicting Y_ω uses all of X.



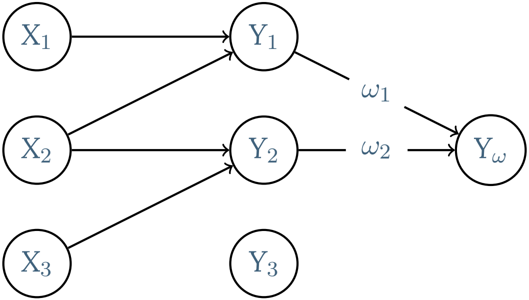
Motivation

We could be clever in choosing weights if we knew the DAG.



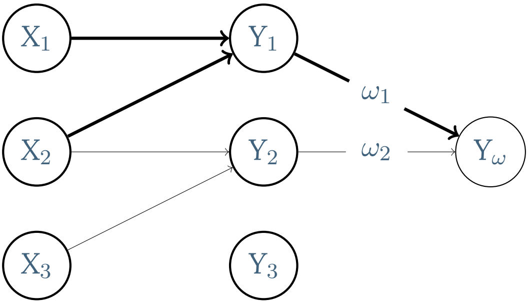
Motivation

Outcome Y_3 gets no weight.



Motivation

Outcomes Y_1 and Y_2 get weight based on accuracy of predictions.



Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Measuring accuracy

Suppose we are given ω and $\psi_\omega : \mathcal{X} \to \mathbb{R}$ to predict Y_ω.

A measure of accuracy of ψ_ω is MSE:

$$\mathcal{E}_0(\psi_\omega) = \mathrm{E}_0 \left[\{\mathrm{Y}_\omega - \psi_\omega(\mathrm{X})\}^2 \right] \ .$$

However, MSE depends on scale and variability of Y_ω.

- Hard to compare across studies, outcomes.

Measuring accuracy

To obtain a scale-free measure, we use a nonparametric version of R^2.

Let $\mu_{0,\omega} = E_0(Y_\omega)$ be predictions made ignoring X.

$$R^2_{0,\omega}(\psi_\omega) = \frac{E_0\left[\{Y_\omega - \mu_{0,\omega}\}^2\right] \; - \; E_0\left[\{Y_\omega - \psi_\omega(X)\}^2\right]}{E_0\left[\{Y_\omega - \mu_{0,\omega}\}^2\right]} \; .$$

Measuring accuracy

To obtain a scale-free measure, we use a nonparametric version of R^2.

Let $\mu_{0,\omega} = E_0(Y_\omega)$ be predictions made ignoring X.

$$R^2_{0,\omega}(\psi_\omega) = \frac{\overbrace{E_0\left[\{Y_\omega - \mu_{0,\omega}\}^2\right]}^{\text{MSE of } \mu_{0,\omega}} - E_0\left[\{Y_\omega - \psi_\omega(X)\}^2\right]}{E_0\left[\{Y_\omega - \mu_{0,\omega}\}^2\right]} \ .$$

Measuring accuracy

To obtain a scale-free measure, we use a nonparametric version of R^2.

Let $\mu_{0,\omega} = E_0(Y_\omega)$ be predictions made ignoring X.

$$R^2_{0,\omega}(\psi_\omega) = \frac{\overbrace{E_0\left[\{Y_\omega - \mu_{0,\omega}\}^2\right]}^{\text{MSE of } \mu_{0,\omega}} - \overbrace{E_0\left[\{Y_\omega - \psi_\omega(X)\}^2\right]}^{\text{MSE of } \psi_\omega}}{E_0\left[\{Y_\omega - \mu_{0,\omega}\}^2\right]} \; .$$

Measuring accuracy

To obtain a scale-free measure, we use a nonparametric version of R^2.

Let $\mu_{0,\omega} = \mathrm{E}_0(\mathrm{Y}_\omega)$ be predictions made ignoring X.

$$R^2_{0,\omega}(\psi_\omega) = \frac{\overbrace{\mathrm{E}_0\left[\{\mathrm{Y}_\omega - \mu_{0,\omega}\}^2\right]}^{\text{MSE of } \mu_{0,\omega}} - \overbrace{\mathrm{E}_0\left[\{\mathrm{Y}_\omega - \psi_\omega(\mathrm{X})\}^2\right]}^{\text{MSE of } \psi_\omega}}{\underbrace{\mathrm{E}_0\left[\{\mathrm{Y}_\omega - \mu_{0,\omega}\}^2\right]}_{\text{MSE of } \mu_{0,\omega}}} \ .$$

Predicting combined outcome

The criteria $R^2_{0,\omega}$ provides a way to compare prediction functions for Y_ω.

- Large $R^2_{0,\omega}$ indicates accurate predictions.
- $R^2_{0,\omega} = 1$ indicates perfect predictions.
- $R^2_{0,\omega} < 0$ means predictions worse than $\mu_{0,\omega}$!

The maximizer over all $\mathcal{X} \to \mathbb{R}$ is

$$\psi_{0,\omega}(\mathrm{X}) = \mathrm{E}_0(\mathrm{Y}_\omega \mid \mathrm{X}) \ .$$

This fact plays a key role in how we will construct a prediction function.

Optimal weights

For any ω, the function $\psi_{0,\omega}$ gives the most accurate predictions of Y_ω.

We also want weights that lead to most accurate predictions of combined outcome.

Formally, we define

$$\omega_0 = \text{argmax}_\omega R^2_{0,\omega}(\psi_{0,\omega}) \ .$$

The statistical goal is to estimate ψ_{0,ω_0} and ω_0.

Caveats

The sense in which this combination is optimal is strictly related to prediction.

- Not how well combined outcome measures "latent" outcome.

The optimal weights could give zero weight to some outcomes.

- These outcomes may still be important!

The procedure is best viewed as an exploratory analysis.

- However, it can be fully pre-specified!

Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Estimation

The statistical goal is to estimate ψ_{0,ω_0} and ω_0.

At first glance, it looks like a difficult optimization problem,

$$\begin{aligned}\psi_{0,\omega_0} &= \text{argmax}_{\psi} R^2_{0,\omega_0}(\psi) \\ \omega_0 &= \text{argmax}_{\omega} R^2_{0,\omega}(\psi_{0,\omega})\end{aligned}$$

The problem is made easier by recognizing

$$\psi_{0,\omega}(\text{X}) = \text{E}_0(\text{Y}_\omega \mid \text{X}) = \text{E}_0\left(\sum_{j=1}^{J} \omega_j \text{Y}_j \,\middle|\, \text{X}\right) = \sum_{j=1}^{J} \omega_j \text{E}_0(\text{Y}_j \mid \text{X}) \ .$$

Estimation

For any ω, $\psi_{0,\omega}$ is weighted sum of conditional means.

This allows the optimization to be split up:

1. Estimate $\mathrm{E}_0(\mathrm{Y}_j \mid \mathrm{X})$ for $j = 1, \ldots, J$.
2. Combine estimates $\sum_{j=1}^{J} \omega_j \hat{\mathrm{E}}(\mathrm{Y}_j \mid \mathrm{X})$.
3. Optimize over weights using estimated prediction function.

However, we must take care to avoid overfitting!

Outline

- Motivation
- Defining target parameters
  - Measuring accuracy of predictions
  - Optimal predictor and weights
- Estimation
  - Super learning
  - Estimating weights
- Evaluating predictions
  - Estimation and inference
  - Variable importance
- Simulation
- Data Analysis
- Conclusions and future directions

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?
- Gradient boosting?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?
- Gradient boosting? Support vector machines?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?
- Gradient boosting? Support vector machines? Deep neural networks?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?
- Gradient boosting? Support vector machines? Deep neural networks?
- Highly adaptive lasso?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?
- Gradient boosting? Support vector machines? Deep neural networks?
- Highly adaptive lasso?
- Variable selection?

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?
- Gradient boosting? Support vector machines? Deep neural networks?
- Highly adaptive lasso?
- Variable selection?
- Ad infinitum...

Predicting outcomes

How should we estimate $E_0(Y_j \mid X)$ for a given j?

- Linear regression, with interactions, and nonlinear terms, or splines (with different degrees?)
- Penalized linear regression, with different penalties?
- Random forests, with different tuning parameters?
- Gradient boosting? Support vector machines? Deep neural networks?
- Highly adaptive lasso?
- Variable selection?
- Ad infinitum...

The best algorithm for estimating depends on the (unknown) truth!

- Might be different for different outcomes.

Predicting outcomes

We use $\psi : \mathcal{S} \to \Psi$ to denote an algorithm.

- $\mathcal{S}$ is all subsets of $\{1, \ldots, n\}$.
- Ψ is set of $\mathcal{X} \to \mathbb{R}$.

Given a data set, an algorithm:

1. takes as input a subset of observations;
2. uses observations to create a prediction function;
3. returns prediction function.

We refer to this process as training an algorithm.

Predicting each outcome

Say we have M algorithms that could be used to estimate $E_0(Y_j \mid X)$.

How can we evaluate these algorithms?

Predicting each outcome

Say we have M algorithms that could be used to estimate $E_0(Y_j \mid X)$.

How can we evaluate these algorithms?

- Train algorithms on full data, see which has largest empirical R^2.

Predicting each outcome

Say we have M algorithms that could be used to estimate $E_0(Y_j \mid X)$.

How can we evaluate these algorithms?

- Train algorithms on full data, see which has largest empirical R^2.
  - **Overfit!**

Predicting each outcome

Say we have M algorithms that could be used to estimate $E_0(Y_j \mid X)$.

How can we evaluate these algorithms?

- Train algorithms on full data, see which has largest empirical R^2.
  - **Overfit!**
- Train algorithms on full data, collect more data to evaluate R^2.

Predicting each outcome

Say we have M algorithms that could be used to estimate $E_0(Y_j \mid X)$.

How can we evaluate these algorithms?

- Train algorithms on full data, see which has largest empirical R^2.
  - **Overfit!**
- Train algorithms on full data, collect more data to evaluate R^2.
  - **Expensive!**

Predicting each outcome

Say we have M algorithms that could be used to estimate $E_0(Y_j \mid X)$.

How can we evaluate these algorithms?

- Train algorithms on full data, see which has largest empirical R^2.
  - **Overfit!**
- Train algorithms on full data, collect more data to evaluate R^2.
  - **Expensive!**
- Cross validation!

Cross validation

Consider randomly splitting the data into K different pieces.

S_1
S_2
S_3
S_4
S_5

Cross validation

Define first training, T_1, and validation, V_1, sample.

V_1
T_1
T_1
T_1
T_1

Cross validation

Train M algorithms using T_1.

- $\psi_{j,m}(T_1)$, for $m = 1, \ldots, M$

Withhold validation sample V_1 from training process.

- As though we did another experiment of size $|\mathrm{V}_1|$!

Use validation sample to estimate MSE of each algorithm

$$\hat{\mathcal{E}}_{j,m,1}(\psi_{j,m}) = \frac{1}{|\mathrm{V}_1|} \sum_{i \in \mathrm{V}_1} \{\mathrm{Y}_{j,i} - \psi_{j,m}(T_1)(\mathrm{X}_i)\}^2 \ .$$

Cross validation

Define second training, T_2, and validation, V_2, sample.

T_2
V_2
T_2
T_2
T_2

Cross validation

Train M algorithms using T_2.

- $\psi_{j,m}(T_2)$, for $m = 1, \ldots, M$

Withhold validation sample V_2 from training process.

- As though we did another experiment of size $|V_2|$!

Use validation sample to estimate MSE of each algorithm

$$\hat{\mathcal{E}}_{j,m,2}(\psi_{j,m}) = \frac{1}{|V_2|} \sum_{i \in V_2} \{Y_{j,i} - \psi_{j,m}(T_2)(X_i)\}^2 \ .$$

Cross validation

Continue until each split has been validation once.

T_3
T_3
V_3
T_3
T_3

Cross validation

Continue until each split has been validation once.

T_4
T_4
T_4
V_4
T_4

Cross validation

Continue until each split has been validation once.

T_5
T_5
T_5
T_5
V_5

Cross-validation selector

The cross-validated MSE of algorithm m is

$$\hat{\mathcal{E}}_{j,m}(\psi_{j,m}) = \frac{1}{K} \sum_{k=1}^{K} \hat{\mathcal{E}}_{j,m,k}(\psi_{j,m}) \ .$$

We call the algorithm m^* with the lowest MSE the cross-validation selector.

We might use $\psi_{j,m^*}(F_n)$ as estimate of $\mathrm{E}_0(\mathrm{Y}_j \mid \mathrm{X})$, where $F_n = \{1, \ldots, n\}$.

Ensemble estimator

What if $\psi_{j,1}$ and $\psi_{j,2}$ capture different features?

Using $\psi_{j,\mathrm{SL}} = 0.5\psi_{j,1} + 0.5\psi_{j,2}$ might be better than $\psi_{j,1}$ and $\psi_{j,2}$ alone.

More generally, consider an ensemble prediction function

$$\psi_{j,\mathrm{SL}} = \sum_{m=1}^{M} \alpha_{j,m}\psi_{j,m} \ , \ \alpha_{j,m} \geq 0 \text{ for all } m \ , \ \text{and} \sum_{m=1}^{M} \alpha_{j,m} = 1 \ .$$

Easy to find α_j that minimizes cross-validated MSE.

Super learner

Stacked regression originally proposed in Breiman, 1996.

Referred to as a super learner due to oracle inequality results (van der Laan and Dudoit, 2003).

- MSE of the super learner is asymptotically equivalent to the oracle estimator.
- Even when one considers many estimators.
- Often seen to have good finite-sample performance (van der Laan et al, 2007).

Proving new oracle results is an open area of research!

- Recent work on Big Data oracle inequalities.

Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Statistical goal

Difficult optimization problem,

$$\begin{aligned}
\psi_{0,\omega_0} &= \text{argmax}_{\psi} R^2_{0,\omega_0}(\psi) \\
\omega_0 &= \text{argmax}_{\omega} R^2_{0,\omega}(\psi_{0,\omega}) \ ,
\end{aligned}$$

made easier because

$$\psi_{0,\omega} = E_0\left(\sum_{j=1}^{J} \omega_j Y_j \,\middle|\, X\right) = \sum_{j=1}^{J} \omega_j E_0(Y_j \mid X) \ .$$

For any ω, combine super learners to estimate $\psi_{0,\omega}$,

$$\psi_{n,\omega} = \sum_{j=1}^{J} \omega_j \psi_{j,SL}(F_n) \ .$$

Estimating optimal weights

How do we estimate ω_0?

Estimating optimal weights

How do we estimate ω_0?

- Maximize empirical R^2 over weights.

$$\omega_n = \text{argmax}_\omega \left(1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\{\text{Y}_{\omega,i} - \psi_{n,\omega}(\text{X}_i)\}^2}{\frac{1}{n}\sum_{i=1}^{n}\{\text{Y}_{\omega,i} - \bar{\text{Y}}_\omega\}^2} \right)$$

Estimating optimal weights

How do we estimate ω_0?

- Maximize empirical R^2 over weights.

$$\omega_n = \text{argmax}_\omega \left(1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\{Y_{\omega,i} - \psi_{n,\omega}(X_i)\}^2}{\frac{1}{n}\sum_{i=1}^{n}\{Y_{\omega,i} - \bar{Y}_\omega\}^2} \right)$$

  - **overfit!**

Estimating optimal weights

How do we estimate ω_0?

- Maximize empirical R^2 over weights.

$$\omega_n = \text{argmax}_\omega \left(1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\{Y_{\omega,i} - \psi_{n,\omega}(X_i)\}^2}{\frac{1}{n}\sum_{i=1}^{n}\{Y_{\omega,i} - \bar{Y}_\omega\}^2} \right)$$

  - **overfit!**
- Collect more data, maximize R^2 on new data.

Estimating optimal weights

How do we estimate ω_0?

- Maximize empirical R^2 over weights.

$$\omega_n = \text{argmax}_\omega \left(1 - \frac{\frac{1}{n}\sum_{i=1}^n \{Y_{\omega,i} - \psi_{n,\omega}(X_i)\}^2}{\frac{1}{n}\sum_{i=1}^n \{Y_{\omega,i} - \bar{Y}_\omega\}^2} \right)$$

  - **overfit!**
- Collect more data, maximize R^2 on new data.
  - **expensive!**

Estimating optimal weights

How do we estimate ω_0?

- Maximize empirical R^2 over weights.

$$\omega_n = \text{argmax}_\omega \left(1 - \frac{\frac{1}{n}\sum_{i=1}^n \{Y_{\omega,i} - \psi_{n,\omega}(X_i)\}^2}{\frac{1}{n}\sum_{i=1}^n \{Y_{\omega,i} - \bar{Y}_\omega\}^2} \right)$$

  - **overfit!**
- Collect more data, maximize R^2 on new data.
  - **expensive!**
- Cross validation!

Nested cross validation

Define first training, T_1, and validation, V_1, sample.

V_1
T_1
T_1
T_1
T_1

Cross validation

Train J super learners using T_1.

- $\psi_{j,SL}(T_1)$, for $j = 1, \ldots, J$

For any ω, we can construct combined super learner

$$\psi_{\omega,SL}(T_1) = \sum_{j=1}^{J} \omega_j \psi_{j,SL}(T_1) \ .$$

Withhold validation sample V_1 from super learner fitting.

- As though we did another experiment of size $|V_1|$!

For any ω, use validation sample to estimate MSE

$$\hat{\mathcal{E}}_{\omega,1}(\psi_{\omega,SL}) = \frac{1}{|V_1|} \sum_{i \in V_1} \{Y_{\omega,i} - \psi_{\omega,SL}(T_1)(X_i)\}^2 \ .$$

Nested cross validation

Define second training, T_2, and validation, V_2, sample.

T_2
V_2
T_2
T_2
T_2

Nested cross validation

Train J super learners using T_2.

- $\psi_{j,\text{SL}}(\text{T}_2)$, for $j = 1, \ldots, J$

For any ω, we can construct combined super learner

$$\psi_{\omega,\text{SL}}(\text{T}_2) = \sum_{j=1}^{J} \omega_j \psi_{j,\text{SL}}(\text{T}_2) \ .$$

Withhold validation sample V_2 from super learner fitting.

- As though we did another experiment of size $|\text{V}_2|$!

For any ω, use validation sample to estimate MSE

$$\hat{\mathcal{E}}_{\omega,2}(\psi_{\omega,\text{SL}}) = \frac{1}{|\text{V}_2|} \sum_{i \in \text{V}_2} \{\text{Y}_{\omega,i} - \psi_{\omega,\text{SL}}(\text{T}_2)(\text{X}_i)\}^2 \ .$$

Nested cross validation

Continue until each split has been used once.

T_3
T_3
V_3
T_3
T_3

Nested cross validation

Continue until each split has been used once.

T_4
T_4
T_4
V_4
T_4

Nested cross validation

Continue until each split has been used once.

T_5
T_5
T_5
T_5
V_5

Estimating weights

For any ω, we have cross-validated estimate of R^2,

$$R^2_{n,\omega}(\psi_{\omega,\text{SL}}) = 1 - \frac{\frac{1}{K}\sum_{k=1}^{K} \hat{\mathcal{E}}_{\omega,k}(\psi_{\omega,\text{SL}})}{\frac{1}{n}\sum_{i=1}^{n}\{Y_{\omega,i} - \bar{Y}_{\omega}\}^2} \ .$$

Estimate of optimal weights is

$$\omega_n = \text{argmax}_{\omega} R^2_{n,\omega}(\psi_{\omega,\text{SL}}) \ .$$

Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Estimating predictive performance

How accurate is ψ_{n,ω_n} in predicting Y_{ω_n}?

Estimating predictive performance

How accurate is ψ_{n,ω_n} in predicting Y_{ω_n}?

- Report $R^2_{n,\omega_n}(\psi_{\omega_n,\mathrm{SL}})$, call it a day.

Estimating predictive performance

How accurate is ψ_{n,ω_n} in predicting Y_{ω_n}?

- Report $R^2_{n,\omega_n}(\psi_{\omega_n,\mathrm{SL}})$, call it a day.
  - **overfit!**

Estimating predictive performance

How accurate is ψ_{n,ω_n} in predicting Y_{ω_n}?

- Report $R^2_{n,\omega_n}(\psi_{\omega_n,\mathrm{SL}})$, call it a day.
  - **overfit!**
- Collect more data to evaluate predictions.

Estimating predictive performance

How accurate is ψ_{n,ω_n} in predicting Y_{ω_n}?

- Report $R^2_{n,\omega_n}(\psi_{\omega_n,SL})$, call it a day.
  - **overfit!**
- Collect more data to evaluate predictions.
  - **expensive!**

Estimating predictive performance

How accurate is ψ_{n,ω_n} in predicting Y_{ω_n}?

- Report $R^2_{n,\omega_n}(\psi_{\omega_n,\mathrm{SL}})$, call it a day.
  - **overfit!**
- Collect more data to evaluate predictions.
  - **expensive!**
- More cross validation!!!

Doubly nested cross validation

Pictures omitted for the sanity of audience.

Estimating predictive performance

Cross-validate the entire procedure to estimate performance.

- Compute ω_n and ψ_{n,ω_n} in training sample
- Estimate R^2 in validation sample
- Average over splits

Formally, we define $\omega : \mathcal{S} \to \Omega$ and $\psi_{\omega,SL} : \mathcal{S} \to \Psi$.

The cross-validated R^2 estimate is

$$R_n^2(\omega_n, \psi_{n,\omega_n}) = 1 - \frac{\sum_{k=1}^{K} \frac{1}{|V_k|} \sum_{i \in V_k} \{Y_{\omega(T_k),i} - \psi_{\omega(T_k),SL}(X_i)\}^2}{\sum_{k=1}^{K} \frac{1}{|V_k|} \sum_{i \in V_k} \{Y_{\omega(T_k),i} - \bar{Y}_{\omega(T_k)}\}^2} .$$

Inference for predictive performance

In spite of the highly adaptive estimation procedure,

$$n^{1/2}\left\{\mathrm{R}^2_n(\omega_n, \psi_{n,\omega_n}) - \mathrm{R}^2_0(\omega_n, \psi_{n,\omega_n})\right\} \to \mathrm{Normal}(0, \sigma^2)\ .$$

A sufficient condition is that ω_0 is unique.

- Possible to relax this condition (Luedtke et al, 2016).

Variance derived via delta method for influence functions.

- Consistently estimated with closed-form estimator.

Variance estimates used to construct closed-form confidence intervals and hypotheses tests.

- Machine learning with inference!

Variable importance

Define $\mathrm{R}_0^2(\omega_n^d, \psi_{n,\omega_n}^d)$ as the true R^2 of the estimated super learner that leaves out $\mathrm{X}_d,\ d = 1, \ldots, D$.

The "importance" of X_d could be quantified by

$$\Delta_{0n}^d = \mathrm{R}_0^2(\omega_n, \psi_{n,\omega_n}) - \mathrm{R}_0^2(\omega_n^d, \psi_{n,\omega_n}^d)\ .$$

How much did X_d improve predictions of combined outcome?

- Similar to random forest variable importance measures

Variable importance

Variable importance can be estimated as

$$\Delta_n^d = \mathrm{R}_n^2(\omega_n, \psi_{n,\omega_n}) - \mathrm{R}_n^2(\omega_n^d, \psi_{n,\omega_n}^d) \ .$$

We can still establish

$$n^{1/2}(\Delta_n^d - \Delta_{0n}^d) \to \mathrm{Normal}(0, \sigma_d^2) \ .$$

Variance can again be derived using delta method for influence functions.

Did X_d significantly improve predictions of combined outcome?

Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Simulation

Covariates:
$X_1, \ldots, X_6 \sim \text{Uniform}(0, 4),\ X_7, \ldots, X_9 \sim \text{Bernoulli}(0.5)$

Outcomes:

$$\begin{aligned}
Y_1 &= X_1 + 2X_2 + 4X_3 + X_7 + 2X_8 + 4X_9 + 2X_4 + \epsilon_1\ , \\
Y_2 &= X_1 + 2X_2 + 4X_3 + X_7 + 2X_8 + 4X_9 + 2X_5 + \epsilon_2\ ,\ \text{and} \\
Y_3 &= X_1 + 2X_2 + 4X_3 + X_7 + 2X_8 + 4X_9 + 2X_6 + \epsilon_3\ ,
\end{aligned}$$

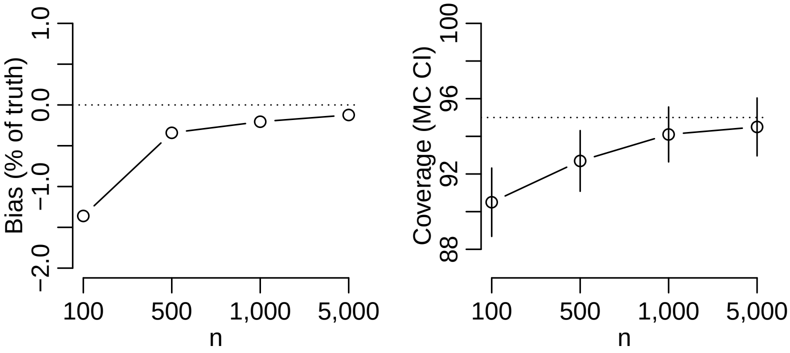
with $\epsilon_j \sim \text{Normal}(0, 5^2),\ j = 1, 2, 3$.

True parameters:
$\omega_0 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $R^2_{0,\omega_0} = 0.80$, $\Delta^2_0 = 0.12$.

Super learner:
intercept only, main terms, and stepwise linear model.

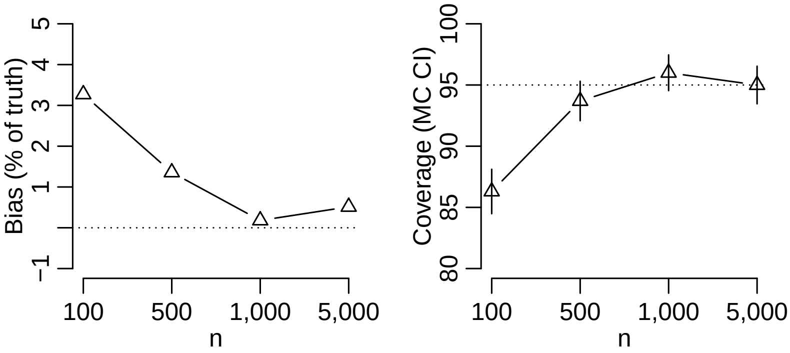
Simulation

Bias and coverage for R^2_{n,ω_n} (1000 replications).



Simulation

Bias and coverage for Δ_n^2 (1000 replications).



Outline

Motivation

Defining target parameters

- Measuring accuracy of predictions
- Optimal predictor and weights

Estimation

- Super learning
- Estimating weights

Evaluating predictions

- Estimation and inference
- Variable importance

Simulation

Data Analysis

Conclusions and future directions

Data analysis

Can we use neonatal information to predict neurocognitive outcomes later in life?

- Early identification of at-risk children.

What covariates are important for making predictions?

- Informs what information to collect to screen children.

Motivation: PROBIT

The Promotion of Breastfeeding Intervention Trial enrolled pregnant mothers in 1996–97.

Group name	**Variables**
Breastfeeding	breastfeeding encouraged
Socioeconomic	household animals
Parental	age, height, weight, education, siblings employment status
Birth	gestational age, Apgar score
Growth	WAZ, HAZ, HCAZ (0,1,2,3,6,9,12 months)
Other	mother smoked during pregnancy, mother drank during pregnancy
WASI score (age 6)	Matrix, Block, Vocabulary, Similarities

Future directions

Effect estimation for discovery in high dimensions.

- Maximize effect instead of R^2?
- Cross-validated TMLE (Zheng and van der Laan, 2010)

The method could be extended to binary outcomes, other performance metrics, and dependent data.

Nonlinear combinations of outcomes are also of interest.

- Alternating conditional expectations (Breiman and Friedman, 1985)

Software

R packages:

r2weight

- Available on GitHub: https://github.com/benkeser/r2weight

SuperLearner (Polley et al, 2016)

- Demonstration – http://benkeser.github.io/sllecture/

References I

[1] Michael S Kramer, Beverley Chalmers, Ellen D Hodnett, Zinaida Sevkovskaya, Irina Dzikovich, Stanley Shapiro, Jean-Paul Collet, Irina Vanilovich, Irina Mezen, Thierry Ducruet, et al.
Promotion of breastfeeding intervention trial (PROBIT): a randomized trial in the Republic of Belarus.
Journal of the American Medical Association, 285(4):413–420, 2001.

[2] AB Feranil, SA Gultiano, and LS Adair.
The Cebu Longitudinal Health and Nutrition Survey: Two Decades Later.
Asia-Pacific Population Journal, 23(3), 2008.

[3] Leo Breiman.
Stacked regressions.
Machine Learning, 24(1):49–64, 1996.

[4] Mark J van der Laan and Sandrine Dudoit.
Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples.
UC Berkeley Division of Biostatistics Working Paper Series, 2003.

[5] Mark J van der Laan and Eric C Polley.
Super learner.
Statistical Applications in Genetics and Molecular Biology, 6(1):1–23, 2007.

[6] Alexander R Luedtke, Mark J Van Der Laan, et al.
Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy.
The Annals of Statistics, 44(2):713–742, 2016.

[7] Wenjing Zheng and Mark J van der Laan.
Asymptotic theory for cross-validated targeted maximum likelihood estimation.
UC Berkeley Division of Biostatistics Working Paper Series, 2010.

[8] Leo Breiman and Jerome H Friedman.
Estimating optimal transformations for multiple regression and correlation.
Journal of the American Statistical Association, 80(391):580–598, 1985.

References II

[9] Eric Polley, Erin LeDell, Chris Kennedy, and Mark van der Laan.
SuperLearner: Super Learner Prediction, 2016.
R package version 2.0-21.

Oracle Inequality

Let $L : \Psi \times \mathcal{O} \to \mathbb{R}$ be a loss function for ψ_0 in the sense that

$$\psi_0 = \mathrm{argmin}_{\psi \in \Psi} E_0\{L(\psi)(O)\} .$$

Define $d_0(\psi, \psi_0) := E_0\left\{L(\psi)(O) - L(\psi_0)(O)\right\}$ and let p be the proportion of observations in the validation sample.
Assume

1. $\psi_{n,m} \in \Psi$ with probability tending to one for $m = 1, \ldots, M$.
2. For some $C_0 < \infty$, $\sup_{\psi \in \Psi} d_0(\psi, \psi_0) < C_0$ almost surely.
3. For some $C_1 < \infty$, $E_0\left\{L(\psi)(O) - L(\psi_0)(O) - d_0(\psi, \psi_0)\right\}^2 \leq C_1 d_0(\psi, \psi_0)$ for all $\psi \in \Psi$.

For every $\lambda > 0$ and $C(\lambda) := \frac{2}{3}(1 + \lambda)^2(C_0 + C_1)$,

$$\begin{aligned} &E_{B_n}\left\{d_0(\psi_{n,SL}, \psi_0)\right\} \\ &\quad \leq (1 + 2\lambda)E_{B_n}\left\{d_0(\psi_{n,OR}, \psi_0)\right\} + 2C(\lambda)\left(\frac{1 + \log K(n)}{np}\right) \end{aligned}$$

Asymptotics

The cross-validated R^2 estimate is

$$R_n^2(\omega_n, \psi_{n,\omega_n}) = 1 - \frac{\sum_{k=1}^{K} \frac{1}{|V_k|} \sum_{i \in V_k} \{Y_{\omega(T_k),i} - \psi_{\omega(T_k),SL}(X_i)\}^2}{\sum_{k=1}^{K} \frac{1}{|V_k|} \sum_{i \in V_k} \{Y_{\omega(T_k),i} - \bar{Y}_{\omega(T_k)}\}^2}$$
$$= 1 - \frac{\theta_{1,n}}{\theta_{2,n}}$$

For $k = 1, \ldots, K$, define

$$D_{0n,k}(\psi_\omega)(O) := \{Y_{\omega(T_k)} - \psi_{\omega(T_k)}(T_k)(X)\}^2 - E_0\left[\{Y_{\omega(T_k),i} - \psi_{\omega(T_k),SL}(X_i)\}^2\right] ,$$
$$D_{0n,k}(\bar{Y}_\omega)(O) := \{Y_{\omega(T_k)} - \bar{Y}_{\omega(T_k)}\}^2 - E_0\left[\{Y_{\omega(T_k),i} - \bar{Y}_{\omega(T_k)}(X_i)\}^2\right] .$$

Asymptotics

$n^{1/2}(\theta_{1,n} - \theta_{1,0}) \rightarrow \text{Normal}(0, \sigma_1^2)$, with

$$\sigma_1^2 = \frac{1}{K} \sum_{k=1}^{K} \text{E}_0\{\text{D}_{0n,k}(\psi_{\omega_0})(\text{O})^2\} \; .$$

$n^{1/2}(\theta_{2,n} - \theta_{2,0}) \rightarrow \text{Normal}(0, \sigma_2^2)$, with

$$\sigma_2^2 = \frac{1}{K} \sum_{k=1}^{K} \text{E}_0\{\text{D}_{0n,k}(\bar{\text{Y}}_{\omega_0})(\text{O})^2\} \; .$$

Let $\text{D}_{0n,k}(\psi_\omega, \bar{\text{Y}}_\omega) = \left(\text{D}_{0n,k}(\psi_\omega), \text{D}_{0n,k}(\bar{\text{Y}}_\omega)\right)^T$, $g(\theta) = log(\theta_1/\theta_2)$, and $\triangledown g(\theta) = (1/\theta_1, -1/\theta_2)^T$.

Asymptotics

We have $n^{1/2}\{g(\theta_n) - g(\theta_0)\} \to \mathrm{Normal}(0, \sigma_3^2)$, where σ_3^2 is

$$\nabla g(\theta_0)^{\mathrm{T}} \frac{1}{\mathrm{K}} \sum_{k=1}^{\mathrm{K}} \mathrm{E}_0\{D_{0n,k}(\psi_{\omega_0}, \bar{\mathrm{Y}}_{\omega_0})(\mathrm{O}) D_{0n,k}(\psi_{\omega_0}, \bar{\mathrm{Y}}_{\omega_0})(\mathrm{O})^{\mathrm{T}}\} \nabla g(\theta_0)\,.$$

Canonical correlation

Let $Y_\alpha = \sum_{j=1}^{J} \alpha_j Y_j$ and $X_\beta = \sum_{d=1}^{D} \beta_d X_d$.

The first-order canonical variate of X and Y is found by maximizing

$$\frac{E_0\{(Y_\alpha - \mu_{0,\alpha})(X_\beta - \mu_{0,\beta})\}}{E_0\{(Y_\alpha - \mu_{0,\alpha})^2\}E_0\{(X_\beta - \mu_{0,\beta})^2\}}$$

over α and β under constraint that variances equal one.

The canonical correlation is the correlation between $X_{\beta,0}$ and Y_{α_0}.

Canonical correlation

If $\psi_{0,j} = X_\beta$ for all $j = 1, \ldots, J$, the optimal R^2 equals the squared first-order canonical correlation.

To illustrate difference, consider

- $X_d \sim \text{Normal}(0, 1)$, $d = 1, 2$
- $Y_j = X_j^2$ for $j = 1, 2$

Canonical correlation measures linear association between X and Y.

- Canonical correlation equals zero.

Optimal R^2 measures how well we predict Y using X.

- Optimal R^2 equals one.