Section II: developing a marker-based treatment rule

- Treatment decision rule
- Optimal Treatment Rule
- Estimating optimal treatment decision rule
  - Q-learning (Regression modeling)
  - Direct optimization
  - Super Learning

Treatment Decision Rule

Treatment Decision Rule

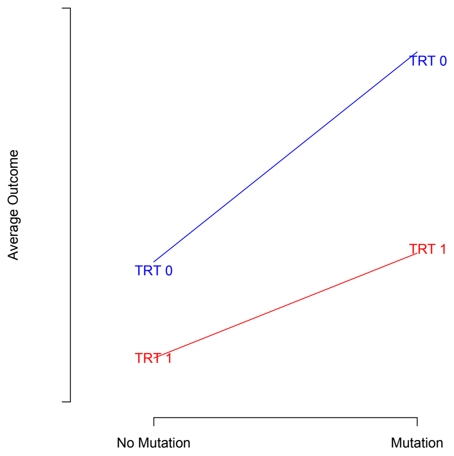
Outcomes are denoted by D,

- Survival time, CD4 count, indicator of no myocardial infarction within 30 days, . . .
- Lower outcomes are better

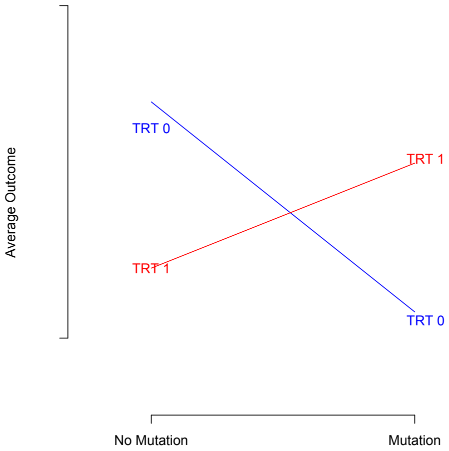
Intuitively: rules should depend on characteristics (variables , covariates), i.e., X, that exhibit a qualitative interaction with treatment

- Tailoring variables/ treatment selection biomarker

Tailoring Variables



Tailoring Variables



Statistical Framework

Simplest setting: A single decision with two treatment options

Observed data: (X_i, A_i, D_i), $i = 1, \dots, n$, independently and identically distributed (iid)

- X_i baseline covariates, $A_i = 0, 1$ treatment received, D_i outcome

Treatment decision rule: A treatment rule

- A function $d : X \to \{0,\ 1\}$

Simple example

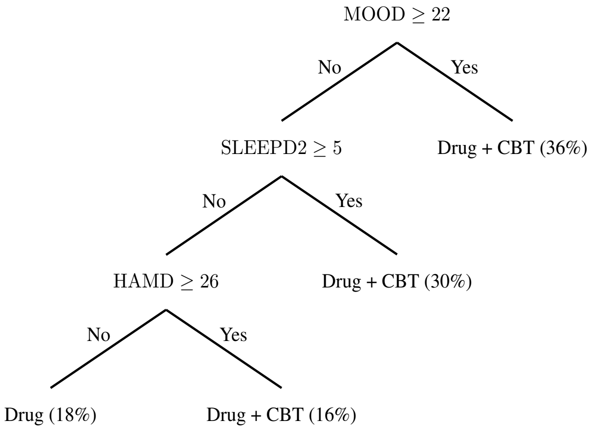
Which treatment to give patients who present with *nonpsychotic Chronic Major Depressive Disorder*?

- *Options:* Nefazodone (Drug) or Drug + Cognitive Behavioral Therapy (CBT)
- *Data:* 681 patients in the Nefazodone-CBASP clinical trial (Keller et al., 2000)
- *Available information:* 50 prognostic variables, e.g., age, baseline depression score
- *Outcome:* Hamilton Rating Scale for Depression

Keller et al. (*NEJM* 2000)

Simple example

A decision rule example:



Simple example

- Even simpler example: If MOOD $\geq 22 \Rightarrow$ Drug + CBT; otherwise $\Rightarrow$ Drug
- *Mathematically:* The formal rule is

$$d(\text{MOOD}) = \begin{cases} 1, & \text{if MOOD} > 22 \\ 0, & \text{otherwise.} \end{cases}$$

Optimal Treatment Decision Rule

Optimal treatment assignment problem

- Identify covariates X that may be predictive of the effect of treatment on outcome
- Treatment rule $d(X)$: a function of covariates X
- There are many possible rules d:

  $\mathcal{D}$: class of all possible treatment decision rules

- Can we find the optimal treatment decision rule in $\mathcal{D}$?
- Optimal treatment decision rule : If followed by all patients in the population, would lead to smallest expected outcome among all rules in $\mathcal{D}$

Potential Outcomes

Single decision: Possible treatment options $a \in \{0, 1\}$

- Define $D(a)$ as the outcome that a patient would experience if, possibly contrary to fact, s/he were to receive treatment option a
- "*Potential outcome*"
- E.g., $D(1)$= the outcome a patient would have if s/he were given treatment 1, and similarly for $D(0)$

Expected outcomes under treatment rules

- Potential outcome for a rule: $D(d)$ = the outcome a patient would have if s/he received treatment according to a rule $d \in \mathcal{D}$

- E.g., if the patient has information X

$$D(d) = D(1)I\{d(X) = 1\} + D(0)I\{d(X) = 0\}\}$$

- $E[D(d)|X = x]$ is the expected outcome for a patient with information x if s/he were to receive treatment according to rule $d \in \mathcal{D}$.

- $E[D(d)] = E[E\{D(d)|X = x\}]$ is the expected outcome for the population if all patients were to receive treatment according to rule $d \in \mathcal{D}$.

Optimal decision rule

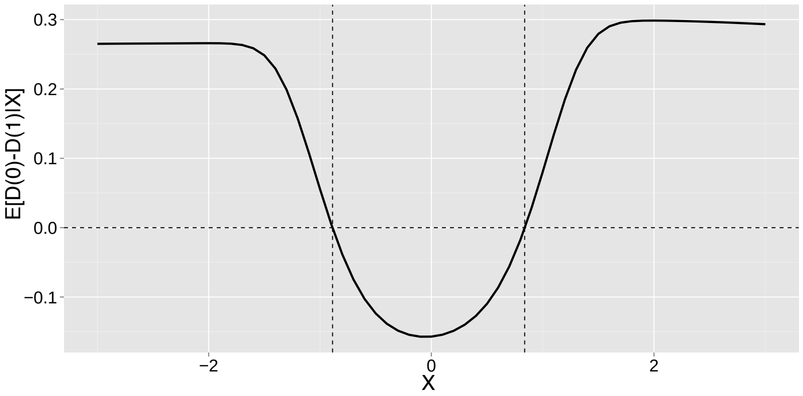
- The optimal treatment decision rule $d^* \in \mathcal{D}$ minimizes the expected outcome

$$d^* = \underset{d \in \mathcal{D}}{\operatorname{argmin}} \, E[D(d)]$$

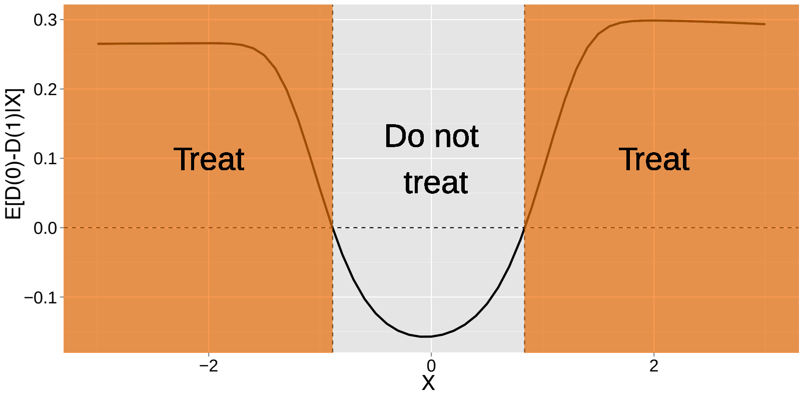
- That is, $E[D(d^*)] \leq E[D(d)]$ for all $d \in \mathcal{D}$
- Also, $E[D(d^*)|X = x] \leq E[D(d)|X = x]$ for all $d \in \mathcal{D}$ and for all patient subgroups defined by x.
- Within each stratum, the optimal rule assigns the treatment with the best average outcome:

$$d^*(X) = \begin{cases} 1, & \text{if } E[D(1)|X] < E[D(0)|X] \\ 0, & \text{otherwise.} \end{cases}$$

For which values of X would the optimal rule recommend treatment?



For which values of X would the optimal rule recommend treatment?



Identifying the optimal treatment decision rule

- We need to learn (nearly) optimal rules based on data.
- The optimal rule is defined in terms of potential outcomes, not the observed data
- It is possible to learn (nearly) optimal rules based on the observed data under certain assumptions

Assumptions Allowing Identification of the Optimal Rule

Positivity: $P(A = a|X = x)$ strictly positive for all x

- E.g., if never assign elderly to strenuous exercise, then data is not informative about expected survival for elderly under strenuous exercise

Consistency: $D(a) = D$ whenever treatment a is actually received

No unmeasured confounders:

$$D(0) \perp A|X \qquad \text{and} \qquad D(1) \perp A|X$$

- X contains all information used to assign treatments

Identifiability Assumptions Plausible in a Randomized Trial

Positivity: $P(A = a|X = x)$ strictly positive for all x

- **Randomized Trial:** controlled by the investigator

Consistency: $D(a) = D$ whenever treatment a is actually received, usually satisfied in a randomized trial

- Requires that there is both
  (i) only one version of treatment,
  (ii) no interference, i.e. an individual's potential outcome is not impacted by the treatment received by other individuals
- **Randomized Trial:** both of these assumptions often plausible

No unmeasured confounders:

$$D(0) \amalg A|X \qquad \text{and} \qquad D(1) \amalg A|X$$

- **Randomized Trial:** automatic if A is randomly assigned based on covariates in X (or, completely at random)

Potential Outcomes

- Randomization ensures that
$$E[D(1)|X=x] \quad = \quad E[D(1)|A=1, X=x],$$
i.e. to learn about the counterfactual mean outcome under treatment 1 within a stratum x, enough to look at individuals who actually received treatment 1 in stratum x

- Consistency ensures that, among those who received treatment 1, the counterfactual outcome is the observed outcome, i.e.
$$E[D(1)|A=1, X=x] \quad = \quad E[D|A=1, X=x]$$

- Putting these together, we see that
$$E[D(1)|X=x] \quad = \quad E[D|A=1, X=x],$$
and similarly for $E[D(0)|X=x]$

Optimal Rule in Terms of Observed Outcomes

- Recall that the optimal rule is

$$d^*(X) = \begin{cases} 1, & \text{if } E[D(1)|X] < E[D(0)|X] \\ 0, & \text{otherwise.} \end{cases}$$

- Using the results from the last slide, we see that

$$d^*(X) = \begin{cases} 1, & \text{if } E[D|A = 1, X] < E[D|A = 0, X] \\ 0, & \text{otherwise.} \end{cases}$$

Mean Outcome under Any Rule d in Terms of Observed Outcomes

- By the law of total expectation:

$$\mathbf{E}[\mathbf{D}(\mathbf{d})] = E[E[D(d)|X]] \tag{1}$$

- By consistency,

$$E[D(d)|X] = d(X)E[D(1)|X] + \{1 - d(X)\}E[D(0)|X]$$

- By our earlier result:

$$E[D(d)|X] = d(X)E[D|A = 1, X] + \{1 - d(X)\}E[D|A = 0, X]$$

- Plugging this back into (1),

$$\mathbf{E}[\mathbf{D}(\mathbf{d})] = E\Big[d(X)E[D|A = 1, X] + \{1 - d(X)\}E[D|A = 0, X]\Big]$$

Estimating optimal treatment decision rule

- **Q-learning (Regression modeling)**
- Direct optimization
- Super Learning

Q-learning (Regression modeling)

- If we had a sample of data $(X_i, A_i, D_i), i = 1, \ldots, n$, we can posit a regression model

$$E(D|A, X) = \mu(A, X; \beta)$$

and estimate $\hat{\beta}$ using e.g. least squares/logistic regression.

- The estimate of the optimal treatment decision rule is:

$$\hat{d}_n(x) = \begin{cases} 1, & \text{if } \mu(1, x; \hat{\beta}_n) \leq \mu(0, x; \hat{\beta}_n) \\ 0, & \text{otherwise.} \end{cases}$$

Alternatives

- Use flexible models for the outcome.
- Other methods, e.g., modeling contrast
  - A more robust method for estimating the optimal treatment decision rule
  - One does not need to know the entire function $E(D|A, X)$.
  - It suffices to only consider the contrast function

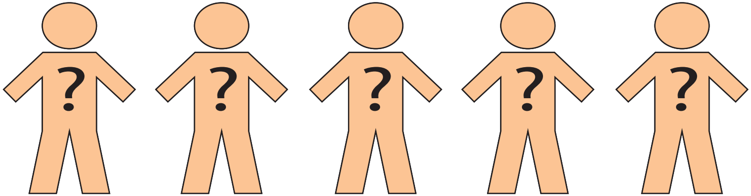
$$\Delta(X) = E(D|A = 0, X) - E(D|A = 1, X)$$

  - $d^*(x) = I\{\Delta(x) \geq 0\}$.

Murphy (*JRSSB*, 2003); Tian et al (*JASA*, 2014)

Contrast function specifies optimal resource-constrained allocation

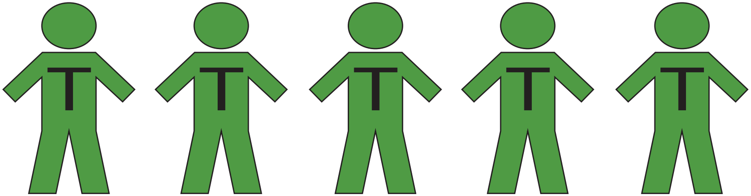
- Suppose treatment is beneficial to everyone...



Luedtke & van der Laan (Int J Biostat, 2016); vanderWeele et al. (arXiv 1802.09642, 2018)

Contrast function specifies optimal resource-constrained allocation

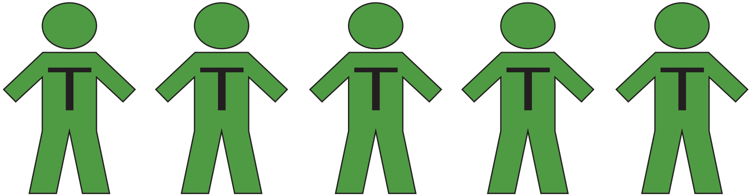
- Suppose treatment is beneficial to everyone...



Luedtke & van der Laan (Int J Biostat, 2016); vanderWeele et al. (arXiv 1802.09642, 2018)

Contrast function specifies optimal resource-constrained allocation

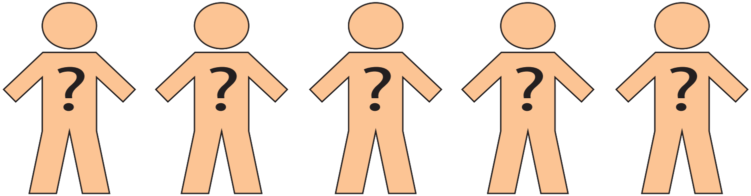
- Suppose treatment is beneficial to everyone...
- But resources are limited so can only treat 40% of population



Luedtke & van der Laan (Int J Biostat, 2016); vanderWeele et al. (arXiv 1802.09642, 2018)

Contrast function specifies optimal resource-constrained allocation

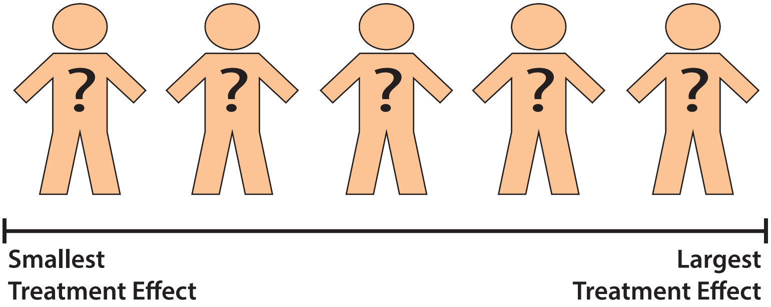
- Suppose treatment is beneficial to everyone...
- But resources are limited so can only treat 40% of population



Luedtke & van der Laan (Int J Biostat, 2016); vanderWeele et al. (arXiv 1802.09642, 2018)

Contrast function specifies optimal resource-constrained allocation

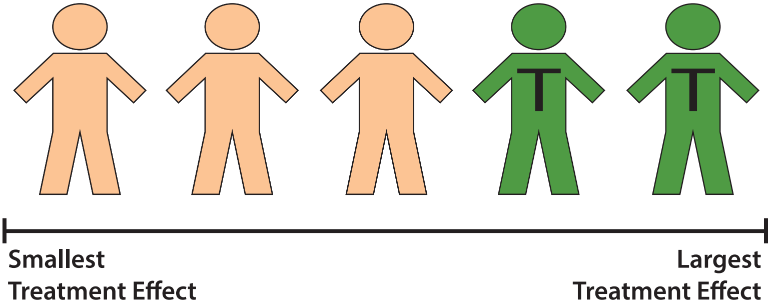
- Suppose treatment is beneficial to everyone...
- But resources are limited so can only treat 40% of population



Luedtke & van der Laan (Int J Biostat, 2016); vanderWeele et al. (arXiv 1802.09642, 2018)

Contrast function specifies optimal resource-constrained allocation

- Suppose treatment is beneficial to everyone...
- But resources are limited so can only treat 40% of population

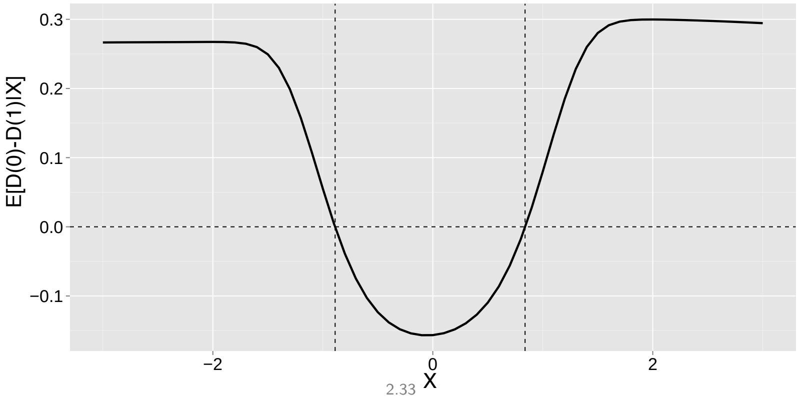


Luedtke & van der Laan (Int J Biostat, 2016); vanderWeele et al. (arXiv 1802.09642, 2018)

If our regression model is misspecified, is our rule reasonable?

Suppose we use the model $\mu(a, x; \hat{\beta}) = \beta_0 + \beta_1 X + \beta_2 A + \beta_3 XA$, so that our rule takes the form

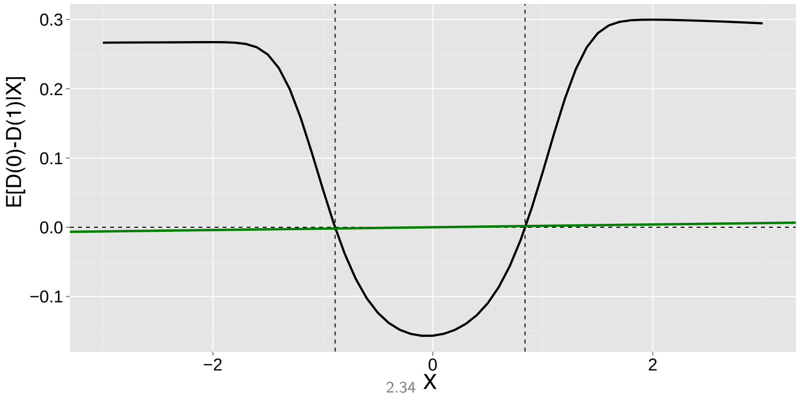
$$\hat{d}_n(x) = \begin{cases} 1, & \text{if } \mu(0, x; \hat{\beta}) - \mu(1, x; \hat{\beta}) = -\hat{\beta}_2 - \hat{\beta}_3 X \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$



If our regression model is misspecified, is our rule reasonable?

Suppose we use the model $\mu(a, x; \hat{\beta}) = \beta_0 + \beta_1 X + \beta_2 A + \beta_3 XA$, so that our rule takes the form

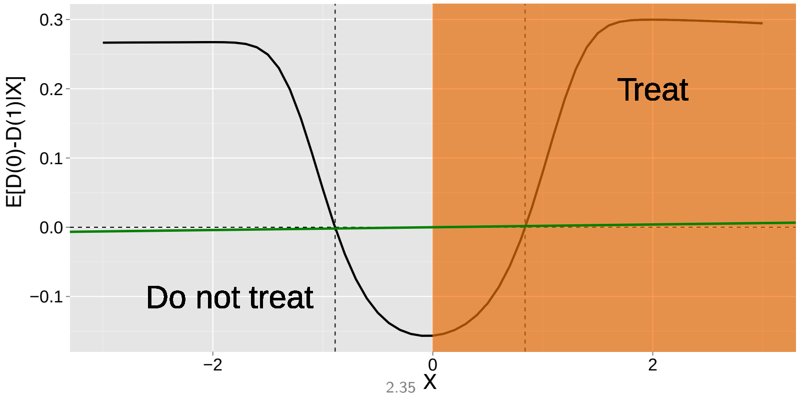
$$\hat{d}_n(x) = \begin{cases} 1, & \text{if } \mu(0, x; \hat{\beta}) - \mu(1, x; \hat{\beta}) = -\hat{\beta}_2 - \hat{\beta}_3 X \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$



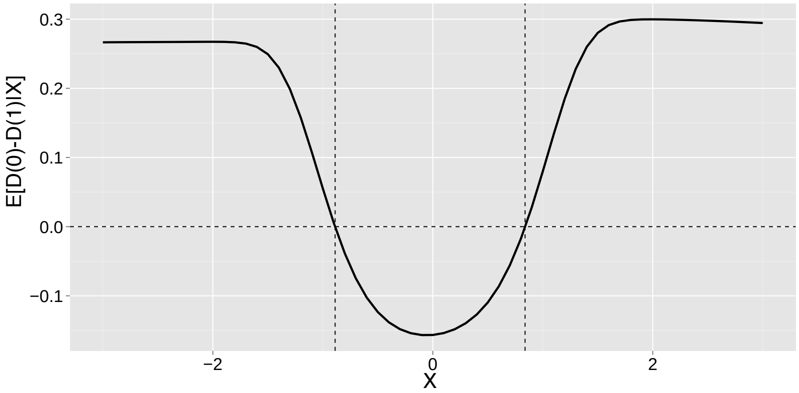
If our regression model is misspecified, is our rule reasonable?

Suppose we use the model $\mu(a, x; \hat{\beta}) = \beta_0 + \beta_1 X + \beta_2 A + \beta_3 XA$, so that our rule takes the form

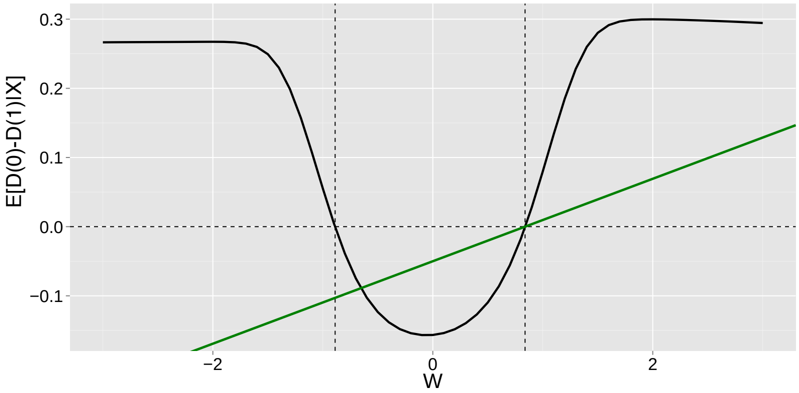
$$\hat{d}_n(x) = \begin{cases} 1, & \text{if } \mu(0, x; \hat{\beta}) - \mu(1, x; \hat{\beta}) = -\hat{\beta}_2 - \hat{\beta}_3 X \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$



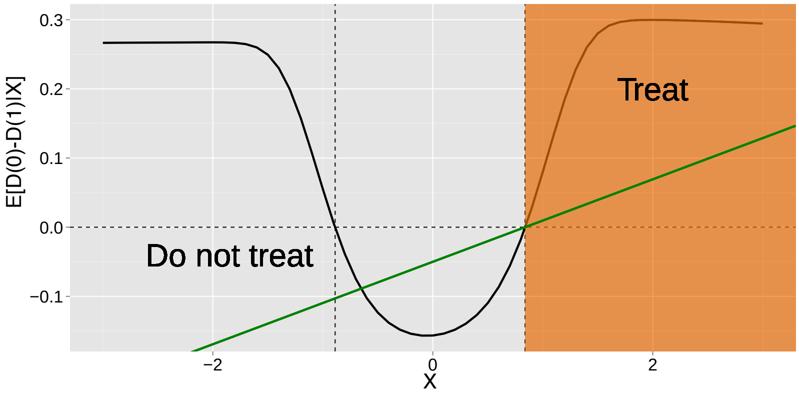
Is there a better linear rule?



Is there a better linear rule?



Is there a better linear rule?



Yes - but how do we learn this rule from the data?

Estimating optimal treatment decision rule

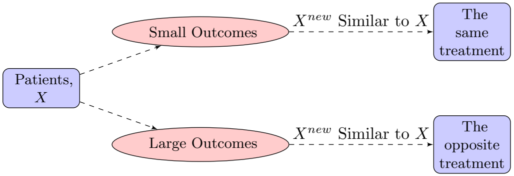
- Q-learning (Regression modeling)
- **Direct optimization**
- Super Learning

Direct Optimization: Classification Perspective

Intuition: Classification

Given a new observation X^{new}, predict the class label $d^{*,\text{new}}$.

- No direct information on the true class labels, d^*.
- Can we assign the right treatment based on the observed information?



Directly Estimating the Optimal Rule

Thought: Minimize a "*good*" estimator for $E[D(d)]$

- $\pi(X) = P(A = 1|X)$ is the propensity score for treatment
- $\pi(X)$ known in a randomized study; Can also be estimated using the data $(A_i, X_i), i = 1, \ldots, n$, e.g., logistic regression $\pi(X; \gamma)$ and estimate γ by $\hat{\gamma}$.
- The propensity of receiving treatment consistent with $d(X)$

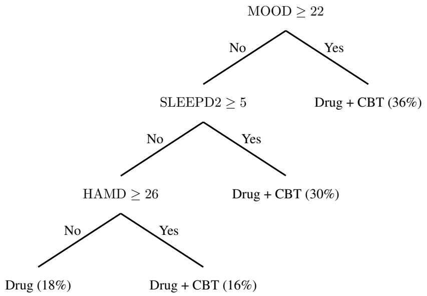
$$P\{d(X)|X\} = \begin{cases} \pi(X), & \text{if } d(X) = 1 \\ 1 - \pi(X), & \text{if } d(X) = 0. \end{cases}$$

Direct Optimization: Optimal Restricted Rule

- Optimize the objective within a restricted class of rules, e.g.
  - Linear rules

$$d_\eta(x) = \begin{cases} 1, & \text{if } \eta_0 + \eta_1 X_1 + \eta_2 X_2 > 0 \\ 0, & \text{otherwise.} \end{cases}$$

  - Binary decision trees of depth at most 3, each decision parameterized by a linear rule



Zhang et al. (*Biometrics* 2012)

Inverse Probability Weighted Estimator for Mean Outcome of Rule

Identify estimators for $E[D(d)]$:

- Using that

$$E[E\{D(d)|A = d(X), X = x\}] = E\left[\frac{I\{A = d(X)\}}{P\{d(X)|X\}} D\right],$$

we arrive at the inverse probability weighted estimator

$$IPWE(d) = n^{-1} \sum_{i=1}^{n} \frac{I\{A_i = d(X_i)\} D_i}{P\{d(X_i)|X, \hat{\gamma}\}}. \qquad (2)$$

- Consistent for $E[D(d)]$ if $\pi(X; \gamma)$, and hence $P\{d(X_i)|X, \hat{\gamma}\}$, is correctly specified

Outcome Weighted Learning (OWL)

- Minimize $IPWE(d)$ (2)
- For any rule d, $2d(X) - 1 = \text{sign}\{f(X)\}$ for some function f.
- Hence, minimize:

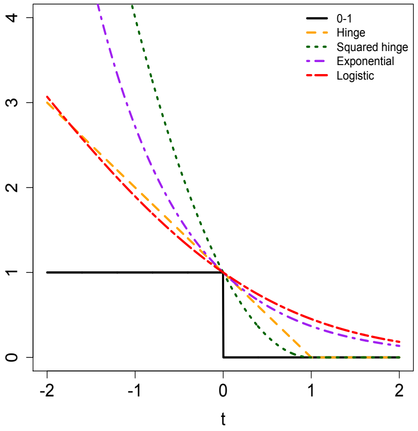
$$n^{-1} \sum_{i=1}^{n} \frac{-D_i}{P\{d(X_i)|X, \hat{\gamma}\}} I\{(2A_i - 1) \neq \text{sign}(f(X_i)\}.$$

- Can be treated as recoding $\mathcal{A} = \{-1, 1\}$

Zhao et al. (*JASA* 2012)

Computational challenges: non-convexity and discontinuity of 0-1 loss

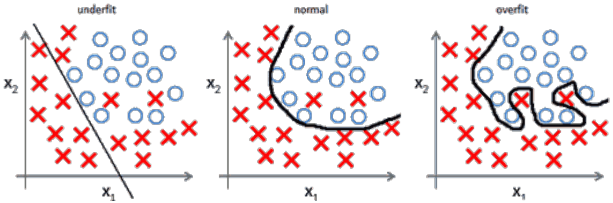
- Solution: replace the indicator that $(2A_i - 1) \neq \text{sign}(f(X_i)\}$ by a smoother function ϕ



Directly estimating the optimal rule is prone to overfitting

Consider the simple case that D is a binary event indicator and treatment probability is always 1/2 so that we minimize

$$-\frac{1}{n}\sum_{i=1}^{n} D_i I\{(2A_i - 1) \neq \text{sign}(f(X_i))\}$$



X is a patient who was untreated and event-free *or* treated and had the event

O is a patient who was treated and event-free *or* untreated and had the event

Image source: http://mlwiki.org/index.php/Overfitting

Avoid overfitting by adding penalties

$$\min_f \frac{1}{n}\sum_{i=1}^{n} \frac{-D_i}{P\{d(X_i)|X,\hat{\gamma}\}}\phi\{(2A_i-1)f(X_i))\} + \lambda_n\|f\|^2. \qquad (3)$$

- $\|f\|$ is some norm for f, and λ_n controls the severity of the penalty on the functions.
- A linear decision rule: $f(X) = X^T\beta + \beta_0$, with $\|f\|$ as the Euclidean norm of β.
- **Estimated treatment rule**:

$$\hat{d}_n(X) = \text{sign}(\hat{f}_n(X)),$$

where $\hat{f}_n$ is the solution to (3).

More Efficient form of Outcome Weighted Learning

- Residual weighted learning: use residuals (after subtracting main effects) instead of the original outcomes as the weights.
- Efficient augmentation and relaxation learning: use an improved estimator of $E[D(d)]$
    - Doubly robust augmented inverse probability weighted estimator: model both the propensity score and the outcome
    - Consistent if either the propensity score or the expected outcome conditional on treatment and covariates is consistently estimated
    - Outcome weighted learning is a special case.

Zhou et al. (*JASA* 2017)

Summary on Direct Optimization Approach

- Direct optimization: conceptual appeal / robustness
- How to implement, e.g. surrogate loss function, form of penalties for variable selection, depends on the context
- Disadvantage of direct optimization relative to Q-learning: more difficult to interpret the final output
  - Does not give magnitude of treatment effect
  - Need to add additional constraints if want to derive resource-constrained rule
  - Rule is a "black box": does not characterize contributions of variables to treatment effect or treatment rule

Estimating optimal treatment decision rule

- Q-learning (Regression modeling)
- Direct optimization
- **Super Learning**

What is Super-Learning?

- Suppose want to estimate the regression $E[D|A, X]$ "as well as possible"
  - e.g., minimize mean-squared error (MSE)
  - MSE performance can be related to the performance of the estimated optimal treatment rule that treats if and only if $E[D|A=1, X] < E[D|A=0, X]$

- How could we do this?
  1. Linear regression
  2. Maybe add some interactions
  3. Maybe add a Lasso penalty on the coefficients
  4. Or some other penalty
  5. If X lower dimensional, maybe run kernel regression or nearest neighbors
  6. What about Random Forests?
  7. Or neural networks?

Given all of these options, what should we do?

- One option is to just pick one *a priori*
- This strategy can never do better than the oracle selector, i.e. the best choice of any one algorithm
- Unlikely you will perform as well as the oracle selector

Objective 1

Perform as well as the oracle selector.

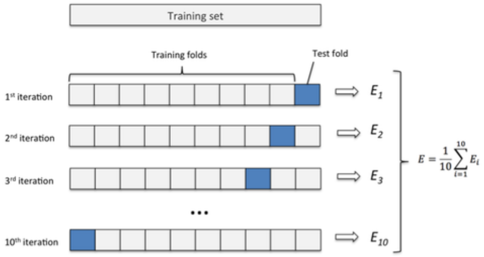
- You've probably seen an algorithm attaining Objective 1 before, though you may not have been aware of its optimality properties

Objective 2

Outperform the oracle selector.

Objective 1: Matching the Oracle

- Could use V-fold cross-validation to select the rule minimizing MSE:



- In what sense?

$$\Big(\text{CV-MSE of Selector}\Big) \leq 1.1 \times \Big(\text{CV-MSE of Oracle}\Big) + C\frac{\log(\#\ \text{Alg})}{n}$$

image source: https://sebastianraschka.com

Objective 1: An Illustration

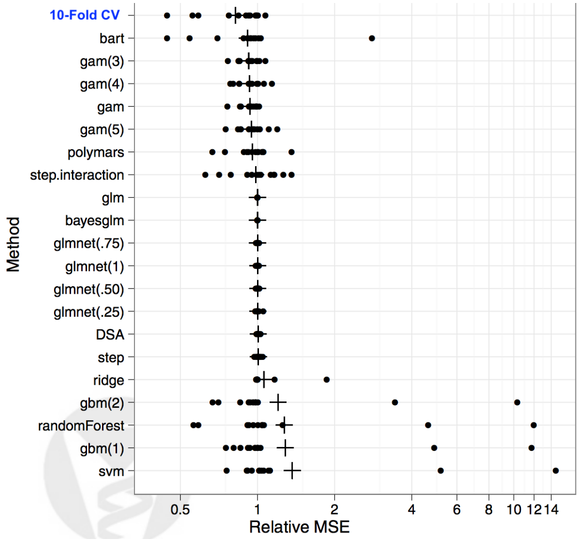


image source: Polley & van der Laan (2010)

Objective 2: A Better Oracle

- So far, we've argued that we can do as well as the best candidate in our library
- Can we do better?

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

– Wikipedia (2018)

Objective 2: A Better Oracle

- How can we hope to outperform the best candidate?
- Could consider all linear combinations of candidate algorithms:

$$\widehat{E}[D|A,X] = \sum_{i=1}^{\#\text{ Alg}} \alpha_i \widehat{E}_i[D|A,X],$$

  where α_i is a real number and $\widehat{E}_i[D|A,X]$ are candidate estimates
- Issue with this choice of combination is that it may be unstable (candidate estimates will be highly correlated)
  - To stabilize regression, restrict α to be a convex combination
- General combination approaches called stacking in the literature
- Weighted sums known as ensemble averaging
- Using convex combination known as super-learning

Objective 2: A Better Oracle

- In the remainder, we refer to the oracle selector as the selector that returns the best convex combination of estimators, rather than the best estimator
- Have the same oracle inequality as before:

$$\Big(\text{CV-MSE of Selector}\Big) \leq 1.1 \times \Big(\text{CV-MSE of Oracle}\Big) + C\frac{\log n}{n}$$

- Do at least as well as the best candidate algorithm
  - Only exception is if one can *a priori* correctly specify a parametric model, in which case perform slightly worse

Better Oracle: An Illustration

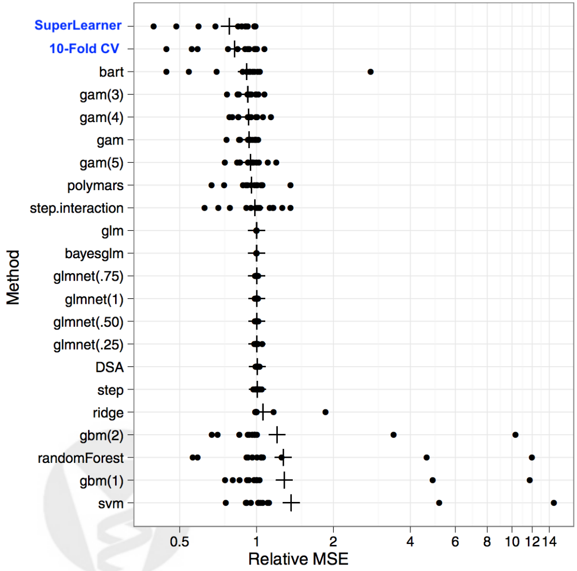


image source: Polley & van der Laan (2010)

SuperLearner to estimate the contrast function

- Can directly estimate the contrast function $\Delta(X) = E[D|A=0,X] - E[D|A=1,X]$ using SuperLearner
  - Allows us to focus exclusively on estimating how X influences the treatment effect
  - In a linear model, this would correspond to estimating the interaction term without needing to estimate the main effect
  - Can make it much easier to estimate Δ
- The approach involves defining a pseudo-outcome Y:

$$Y = \frac{1-2A}{P(A|X,\hat{\gamma})} D$$

  and regressing this pseudo-outcome against X only (not A)
- A more efficient approach uses pseudo-outcome

$$\frac{1-2A}{P(A|X,\hat{\gamma})}\left(D - \hat{E}[D|A,X]\right) + \hat{E}[D|A=0,X] - \hat{E}[D|A=1,X],$$

  where $\hat{E}[D|A,X]$ is an estimate of $E[D|A,X]$

Luedtke & van der Laan (*Int J Biostat*, 2016)

SuperLearner Summary

- Advantages:
  - Can give optimal estimates of $E[D|A, X]$ by optimally selecting from a user-specified collection of modeling approaches, which in turn provides gurantees about the quality of the treatment rule[1]
  - Estimated magnitude of effect for a stratum X can be computed
  - Also can directly estimate the contrast function or perform direct optimization using the SuperLearner framework[2]
- Disadvantage:
  - Because SuperLearner allows for very flexible regression models, the models may be difficult to interpret

[1] Qian & Murphy (*AoS*, 2011)
[2] Luedtke & van der Laan (*Int J Biostat*, 2016)

- Examples

Depression Data

- Compare drug therapy ($A = 0$) with drug + behavioral therapy ($A = 1$)
- Five covariates: Age, Gender, HAMABase (pre-treatment total Hamilton Anxiety Rating Scale score), Sleep (sleep disturbance score), Mood (mood cognition score)
- Response: 24-item Hamilton Rating Scale for Depression
- Number of patients: 436

Analyzing Depression Data

- Q-learning: model the depression score using the covariate, the treatment and their interactions

$$D \sim 1 + X + A + XA$$

- Efficient Augmentation and Relaxation Learning: will model both the outcome and the propensity score
  - Logistic loss: $\phi(t) = \log(1 + e^{-t})$
  - Outcome model: $D \sim 1 + X + A + XA$
  - Propensity model: $A \sim X$

Results

- Q-learning: $\hat{d}(X) = I(-0.83 + 0.01 Age - 0.55 Gender + 0.06 HAMABase + 0.01 Sleep - 0.04 Mood < 0)$.
- Efficient Augmentation and Relaxation Learning: $\hat{d}(X) = I(-0.94 + 0.00 Age - 0.33 Gender + 0.05 HAMABase + 0.02 Sleep - 0.01 Mood < 0)$.

Simulation Example

- $X_1, \ldots, X_5 \sim \mathit{Uniform}(-1, 1)$
- $A \sim \{0, 1\}$ w.p. 0.5
- $D \sim 3 + X_1^2 + X_2^2 + (2X_1 + X_3 - 1)A + N(0, 1)$
- The optimal rule: $d^*(x) = I(2x_1 + x_3 < 1)$

Simulation Example

- $X_1, \ldots, X_5 \sim \mathit{Uniform}(-1, 1)$
- $A \sim \{0, 1\}$ w.p. 0.5
- $D \sim 3 + X_1^2 + X_2^2 + (2X_1 + X_3 - 1)A + N(0, 1)$
- The optimal rule: $d^*(x) = I(2x_1 + x_3 < 1)$

Simulation Example

```
set.seed(1111)

n  = 300
p  = 5
X  = matrix(runif(n*p,-1,1),n,p)
A  = rbinom(n,1,0.5)
mX = 3 + X[,1]^2 + X[,2]^2
cX = 2*X[,1]+ X[,3] -1
D  = mX + A*cX + rnorm(n,1)

## optimal rule
dstar  = (cX<0)
> table(dstar)
dstar
FALSE  TRUE
   85   215
```

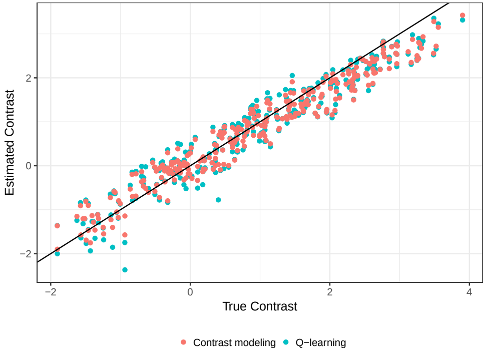
Simulation Example: Q learning (regression modeling)

```
library(SuperLearner)

# candidate algorithms: run "listWrappers()" to see more
SL.library = c("SL.glm","SL.glm.interaction","SL.nnet",
  "SL.cforest","SL.gam","SL.glmnet")

# SuperLearner calls for E[D|A=0,X] and E[D|A=1,X]
SL.out0 = SuperLearner(D[A==0],data.frame(X)[A==0,],
  newX=data.frame(X),SL.library=SL.library,family=gaussian())
SL.out1 = SuperLearner(D[A==1],data.frame(X)[A==1,],
  newX=data.frame(X),SL.library=SL.library,family=gaussian())

# Q estimates
Q0 = SL.out0$SL.predict[,1]
Q1 = SL.out1$SL.predict[,1]

# contrast function as estimated by Q-learning
Q.contrast = Q0-Q1

# Q-learning rule
QTrtRec = as.numeric(Q.contrast>0)
QTrtRec
  0   1
 80 220
```

Simulation Example: Directly modeling the contrast

```
library(SuperLearner)

# candidate algorithms: run "listWrappers()" to see more
SL.library = c("SL.glm","SL.glm.interaction","SL.nnet",
  "SL.cforest","SL.gam","SL.glmnet")

# Defining a data frame of X
Xdf = data.frame(X)

PA1givenX = predict(glm(A~.,data=Xdf,family=binomial),type="response")

# Use AIPW pseudo-outcome
pseudoOutcome = (1-2*A)*(D - A*Q1 - (1-A)*Q0)/(A*PA1givenX + (1-A)*PA1givenX) + Q0-Q1

# Run SL. Specifying "family=gaussian()" because outcome is continuous
# and this will to minimize mean-squared error
SL.out = SuperLearner(pseudoOutcome,Xdf,SL.library=SL.library,family=gaussian())

# Contrast function estimates
direct.contrast = SL.out$SL.predict[,1]

# Contrast estimation rule
contrastTrtRec = as.numeric(direct.contrast>0)

table(ContrastTrtRec)
contrastTrtRec
  0   1
 70 230
```

Comparison of contrast function estimates



```
library(ggplot2)

# Comparison of contrast function estimates at the /observed/ X's
df = data.frame(X=c(-cX,-cX),val=c(Q.contrast,direct.contrast),
  method=rep(c("Q-learning","Contrast modeling"),each=n))

ggplot(data=df,aes(x=X,y=val,colour=method)) + theme_bw() +
  geom_point() + geom_abline(a=0,b=1) + xlab("True Contrast") +
  ylab("Estimated Contrast") +
  theme(legend.title=element_blank(),legend.position="bottom")
```

Simulation Example: Restricted Rule

- R package: DynTxRegime, methods for Estimating Optimal Dynamic Treatment Regimes, including single decision setup
- For restricted regime:

```
## A doubly robust Augmented Inverse Propensity Weighted Estimator (AIPWE) or Inverse
Propensity Weighted Estimator (IPWE) for population mean outcome is optimized over a
restricted class of regimes. Methods are available for both single-decision-point and multiple-
decision-point regimes. This method requires the rgenoud package.

Usage

optimalSeq(..., moPropen, moMain, moCont, data, response, txName, regimes,
           fSet = NULL, refit = FALSE, iter = 0, verbose = TRUE)
```

Simulation Example: OWL/EARL

- For OWL/EARL:

```
##Estimation of optimal treatment regime using efficient augmentation and relaxation
learning (EARL). The method is limited to single-decision-point scenarios with binary
treatment options.

## by setting moMain and moCont to NULL, the function is to estimate the optimal
treatment regime using outcome weighted learning (OWL).


Usage

earl(..., moPropen, moMain, moCont, data, response, txName, regime,
     iter = 0L, lambdas = 0.5, cvFolds = 0L, surrogate = "hinge",
     guess = NULL, verbose = TRUE)
```

- There is also a function on OWL implementation with more features (R function: owl). See help for details.

Simulation Example: Restricted Rule

```
library(DynTxRegime)

# implementation to estimate the optimal restricted rule
# Data Preparation
 data <- data.frame(X, A, D)
 colnames(data) <- c("x1", "x2", "x3", "x4", "x5","a","D")


# Define the propensity for treatment model and methods.
 moPropen<- buildModelObj(model =  ~ x1 + x2 + x3 + x4 + x5,
                                  solver.method = 'glm',
                                  solver.args = list('family'='binomial'),
                                  predict.method = 'predict.glm',
                                  predict.args = list(type='response'))


# Create modelObj object for main effect component
  moMain <- buildModelObj(model = ~ x1 + x2 + x3 + x4 + x5,
                         solver.method = 'lm')

# Create modelObj object for contrast component
  moCont <- buildModelObj(model = ~ x1 + x2 + x3 + x4 + x5,
                         solver.method = 'lm')
```

Simulation Example: Restricted Rule

```
# treatment regime rules at each decision point.
  regimes <- function(a,b,c,d, e, f, data){
                as.numeric( a + b*data$x1 + c*data$x2 + d*data$x3 + e*data$x4 + f*data$x5 > 0)
    }

# genoud requires some additional information
  c1 <- c(-1,-1,-1,-1,-1,-1)
  c2 <- c( 1, 1, 1, 1, 1, 1)
  Domains <- cbind(c1,c2)
  starts <- c(0,0,0,0,0,0)

#!! A LARGER VALUE FOR POP.SIZE IS RECOMMENDED
#!! THIS VALUE WAS CHOSEN TO MINIMIZE RUN TIME OF EXAMPLES
  pop.size <- 50
```

Simulation Example: Restricted Rule

```
 estAIPWE <- optimalSeq(moPropen = moPropen,
                         moMain = moMain,
                         moCont = moCont,
                         data = data,
                         response = -data$D,
                         txName =  "a",
                         regimes = regimes,
                         iter=0L,pop.size = pop.size, starting.values = starts,
                         Domains = Domains, solution.tolerance = 0.0001)

> regimeCoef(estAIPWE)
           a             b             c             d             e             f
4.506975e-01 -7.614161e-01 -5.267877e-05 -5.334575e-01  3.900155e-03 -1.398331e-01

 AIPWTrtRec<- optTx(estAIPWE)

> table(AIPWTrtRec)
AIPWTrtRec
  0   1
 70 230
```

Simulation Example: EARL

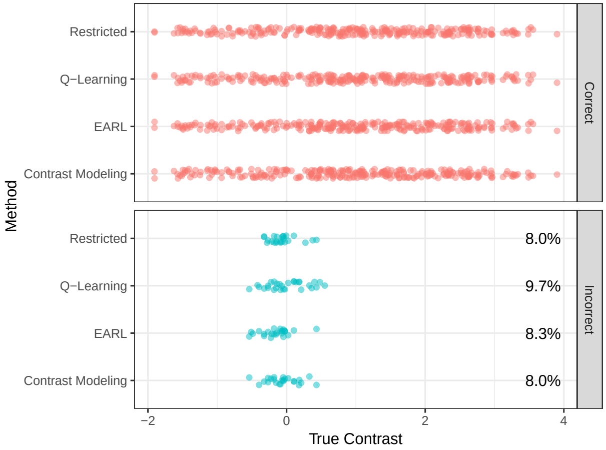
```
library(DynTxRegime)

# Data Preparation
 data <- data.frame(X, A, D)
 colnames(data) <- c("x1", "x2", "x3", "x4", "x5","a","D")


# Define the propensity for treatment model and methods.
 moPropen<- buildModelObj(model =  ~ x1 + x2 + x3 + x4 + x5,
                                  solver.method = 'glm',
                                  solver.args = list('family'='binomial'),
                                  predict.method = 'predict.glm',
                                  predict.args = list(type='response'))


# Create modelObj object for main effect component
  moMain <- buildModelObj(model = ~ x1 + x2 + x3 + x4 + x5,
                          solver.method = 'lm')

# Create modelObj object for contrast component
  moCont <- buildModelObj(model = ~ x1 + x2 + x3 + x4 + x5,
                          solver.method = 'lm')
```

Simulation Example: EARL

```
 earlRes <- earl(moPropen = moPropen, moMain = moMain,
                   moCont = moCont,
                   data = data, response = -data$D, txName = "a", surrogate = 'logit',
                   regime = ~ x1 + x2 + x3 + x4 + x5, lambdas=2^seq(-5,5,1), cvFolds = 5)

> regimeCoef(earlRes)
[1]  0.39663271 -0.59853084 -0.14985610 -0.34259186  0.00478191 -0.02726041


EARLTrtRec <-  optTx(earlRes)$optimalTx

EARLTrtRec <- (EARLTrtRec + 1)/2  ## change coding from (-1,1) to (0,1)

> table(EARLTrtRec)
EARLTrtRec
  0   1
 64 236
```

Simulation Example: Performance Comparison

- Compare predictions to those of optimal rule



Percentages indicate percent discrepancy with true optimal rule in our data set.

- Can further validate performance on an independent data set.

Summary

- Active research area.
- Regression modeling: easy to implement; model may be misspecified.
- Direct optimization: more robust.
- SuperLearner provides a means to learn from the data which method best estimates the optimal treatment rule for the given setting

Extra slides

Simulation Example: OWL

```
library(DynTxRegime)

# Data Preparation
 data <- data.frame(X, A, D)
 colnames(data) <- c("x1", "x2", "x3", "x4", "x5","a","D")


# Define the propensity for treatment model and methods.
 moPropen<- buildModelObj(model =  ~ x1 + x2 + x3 + x4 + x5,
                                   solver.method = 'glm',
                                   solver.args = list('family'='binomial'),
                                   predict.method = 'predict.glm',
                                   predict.args = list(type='response'))
```

Simulation Example: OWL

```
owlRes <-  earl(moPropen = moPropen, moMain = NULL, moCont = NULL,
                  data = data, response = -data$D, txName = "a", surrogate = 'logit',
                  regime = ~ x1 + x2 + x3 + x4 + x5, lambdas=2^seq(-5,5,1), cvFolds = 5)


> regimeCoef(owlRes)
[1]  0.42115454 -0.65789664 -0.25178980 -0.33182440 -0.09571889 -0.03276892

 OWLTrtRec <-  optTx(owlRes)$optimalTx

 OWLTrtRec <- (OWLTrtRec + 1)/2  ## change coding from (-1,1) to (0,1)


> table(OWLTrtRec)
OWLTrtRec
  0   1
 67 233
```