Boosting algorithms for estimating optimal individualized treatment rules

Duzhe Wang

University of Wisconsin-Madison
Department of Statistics

May 7, 2020

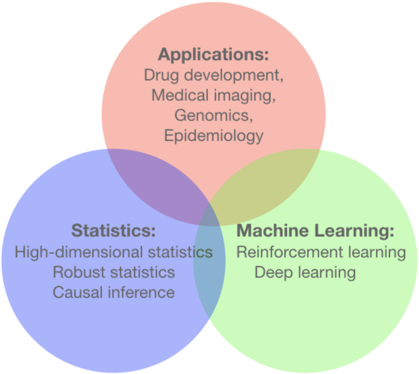
Joint work with Haoda Fu and Po-Ling Loh

About me

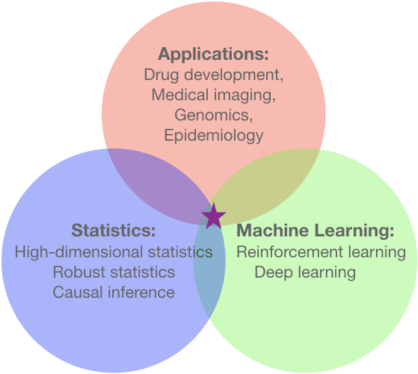
Ph.D. student in statistics
(2015-2020)

MS in mathematics (2013-2015)

Research interests

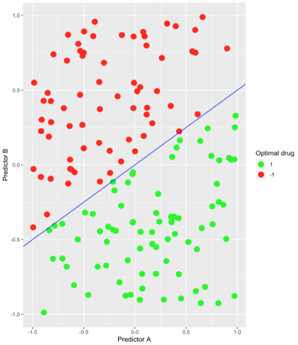


Research interests



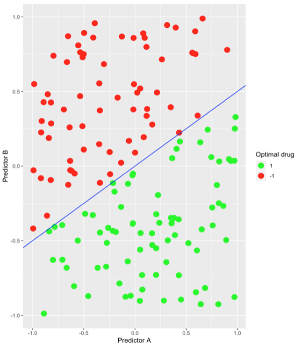
Common thread: develop methods/theory to analyze large-scale/complex real world datasets

Major contribution

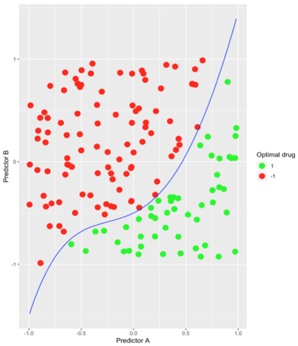


Linear ITR

Major contribution

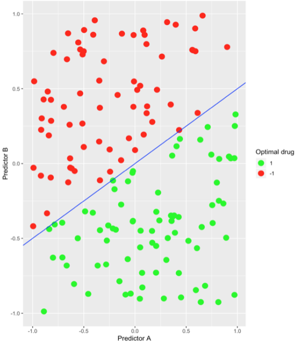


Linear ITR

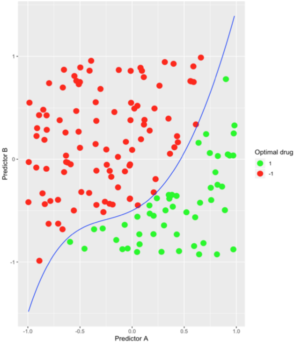


Nonlinear ITR (our focus today)

Major contribution



Linear ITR



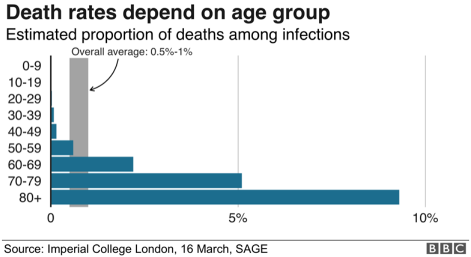
Nonlinear ITR (our focus today)

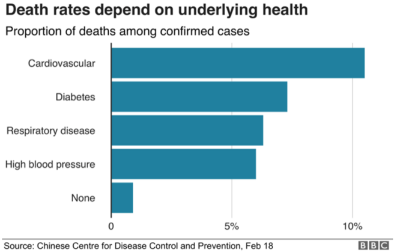
- Provide efficient and accurate estimation of the highly nonlinear and complex optimal ITRs that often arise in practice

Why individualized treatment rules?

Why individualized treatment rules?

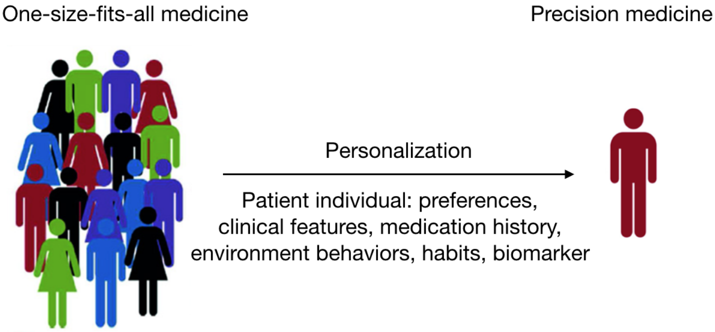
A motivating example:



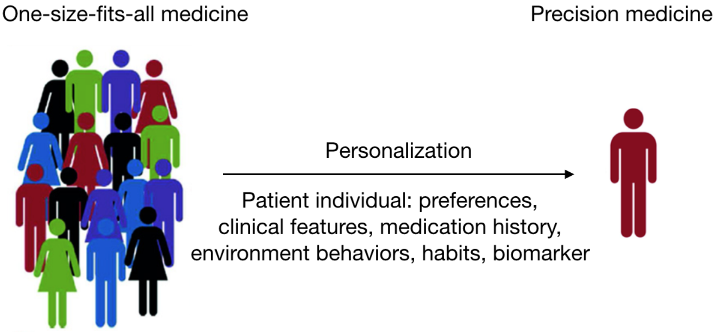


- COVID-19 patients are a very heterogeneous population
- No specific antiviral drug has been proven effective
- COVID-19 presents an opportunity for precision medicine to play expanded role in care

Key questions

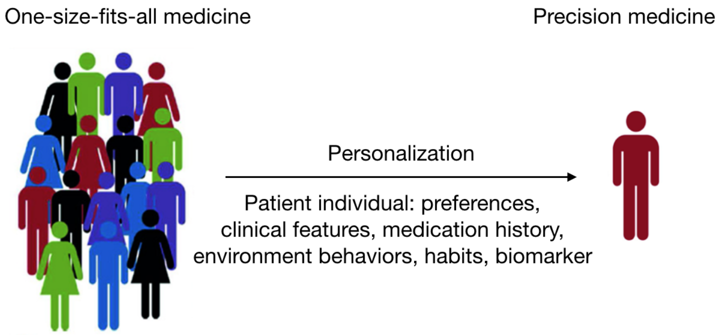


Key questions



- Business question: how do we build individualized treatment recommendation systems?

Key questions



- Business question: how do we build individualized treatment recommendation systems?
- **Statistical question**: how do we estimate optimal individualized treatment rules?

Outline of the remaining talk

1. Background
    - Problem setup
    - Indirect learning
    - Direct learning
2. Proposed methods
    - Proposed method I
    - Proposed method II
    - Proposed method III
3. Simulation and real data analysis
4. Summary

Problem setup

- $\{(X_i, A_i, Y_i), 1 \le i \le n\}$: i.i.d. observations of (X, A, Y)
  - $X \subset \mathcal{X} \subset \mathbb{R}^p$: the vector of patient prognostic variable
  - $A \subset \mathcal{A} = \{-1, +1\}$: the choice of treatment given
  - $Y \subset \mathbb{R}$: the patient clinical outcome (with larger being better)
  - Assume $\pi_a(x) = P(A = a|X = x) > 0$
- Individualized treatment rule:

$$\mathcal{D} : \mathcal{X} \to \{-1, +1\}$$

  - e.g., $\mathcal{D}(x) = 1$, $\mathcal{D}(x) = \text{sign}(x^T 1)$
- **Goal**: find $\mathcal{D}^*(x)$ maximizing the conditional expected outcome

$$\mathcal{D}^*(x) = \underset{a \in \mathcal{A}}{\text{argmax}} \quad Q(x, a) := E(Y|x, a)$$

Indirect learning

Generic method

1. Assume $Q(x, 1)$ and $Q(x, -1)$ are in some specified functional space $\mathcal{F}$
2. Estimate $Q(x, 1)$ and $Q(x, -1)$
3. Estimated optimal ITR:

$$\widehat{\mathcal{D}}(x) = \operatorname{sign}\left(\widehat{Q}(x, 1) - \widehat{Q}(x, -1)\right)$$

Examples of indirect learning

- Q-learning:
  1. $$Q(x,1) = \alpha_1 + \beta_1^T x, \quad Q(x,-1) = \alpha_{-1} + \beta_{-1}^T x$$
  2. $$\left(\hat{\alpha}_1, \hat{\beta}_1^T\right) = \underset{\alpha_1, \beta_1}{\operatorname{argmin}} \sum_{i:A_i=1} \left(Y_i - \alpha_1 - \beta_1^T X_i\right)^2$$
     $$\left(\hat{\alpha}_{-1}, \hat{\beta}_{-1}^T\right) = \underset{\alpha_{-1}, \beta_{-1}}{\operatorname{argmin}} \sum_{i:A_i=-1} \left(Y_i - \alpha_{-1} - \beta_{-1}^T X_i\right)^2$$
  3. $$\widehat{\mathcal{D}}(x) = \operatorname{sign}\left(\hat{\alpha}_1 - \hat{\alpha}_{-1} + \left(\hat{\beta}_1^T - \hat{\beta}_{-1}^T\right) x\right)$$

Examples of indirect learning

- Q-learning:
  1. $$Q(x,1) = \alpha_1 + \beta_1^T x, \quad Q(x,-1) = \alpha_{-1} + \beta_{-1}^T x$$
  2. $$\left(\hat{\alpha}_1, \hat{\beta}_1^T\right) = \operatorname*{argmin}_{\alpha_1,\beta_1} \sum_{i:A_i=1} \left(Y_i - \alpha_1 - \beta_1^T X_i\right)^2$$
     $$\left(\hat{\alpha}_{-1}, \hat{\beta}_{-1}^T\right) = \operatorname*{argmin}_{\alpha_{-1},\beta_{-1}} \sum_{i:A_i=-1} \left(Y_i - \alpha_{-1} - \beta_{-1}^T X_i\right)^2$$
  3. $$\widehat{\mathcal{D}}(x) = \operatorname{sign}\left(\hat{\alpha}_1 - \hat{\alpha}_{-1} + \left(\hat{\beta}_1^T - \hat{\beta}_{-1}^T\right) x\right)$$
- ℓ_1-PLS (Qian and Murphy, '11):
  1. $$Q(X,A) = (1, X^T, A, AX^T)\theta$$
  2. $$\hat{\theta} = \operatorname*{argmin}_{\theta\in\mathbb{R}^{2p+2}} \sum_{i=1}^{n} \left\{Y_i - \left(1, X_i^T, A_i, A_i X_i^T\right)\theta\right\}^2 + \lambda\|\theta\|_1$$
  3. $$\widehat{\mathcal{D}}(x) = \operatorname{sign}\left((0,0,2,2x^T)\hat{\theta}\right)$$

Direct learning

Generic method

1. Note $\mathcal{D}^*(x) = \mathrm{sign}(f^*(x))$. Assume $f^*(x) \in \mathcal{F}$
2. Estimate $f^*(x)$: $\hat{f}(x) = \underset{f \in \mathcal{F}}{\mathrm{argmin}} \sum_{i=1}^{n} L(X_i, A_i, Y_i, f(X_i))$
3. Estimated optimal ITR:

$$\widehat{\mathcal{D}}(x) = \mathrm{sign}(\hat{f}(x))$$

Direct learning

Generic method

1. Note $\mathcal{D}^*(x) = \mathrm{sign}(f^*(x))$. Assume $f^*(x) \in \mathcal{F}$
2. Estimate $f^*(x)$: $\hat{f}(x) = \underset{f \in \mathcal{F}}{\mathrm{argmin}} \sum_{i=1}^{n} L(X_i, A_i, Y_i, f(X_i))$
3. Estimated optimal ITR:

$$\widehat{\mathcal{D}}(x) = \mathrm{sign}(\hat{f}(x))$$

Proposition

$$f^*(x) = \underset{g}{\mathrm{argmin}}\ E\left\{\frac{1}{\pi_A(X)}(2YA - g(X))^2\right\},$$

and

$$f^* = \underset{g}{\mathrm{argmin}}\quad E\left\{Y\frac{\phi(Ag(X))}{\pi_A(X)}\right\},$$

where $\phi(x) = (1 - x)_+$ is the hinge loss.

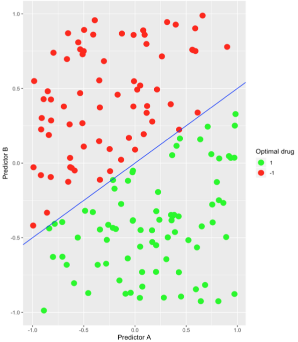
Examples of direct learning

- D-learning (Qi et al. '19):
  1. $$f^*(x) = \alpha^* + (\beta^*)^T x$$
  2. $$\left(\hat{\alpha}, \hat{\beta}^T\right) = \underset{\alpha,\beta}{\text{argmin}} \sum_{i=1}^{n} \frac{1}{\pi_{A_i}(X_i)} \left(2Y_iA_i - \alpha - \beta^T X_i\right)^2$$
  3. $$\widehat{\mathcal{D}}(x) = \operatorname{sign}(\hat{\alpha} + \hat{\beta}^T x)$$

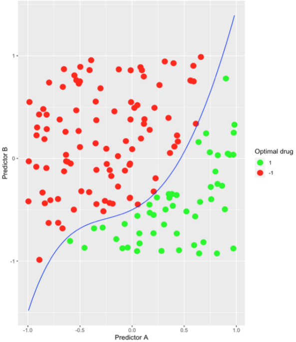
Examples of direct learning

- D-learning (Qi et al. '19):
  1. $$f^*(x) = \alpha^* + (\beta^*)^T x$$
  2. $$\left(\hat{\alpha}, \hat{\beta}^T\right) = \underset{\alpha,\beta}{\text{argmin}} \sum_{i=1}^{n} \frac{1}{\pi_{A_i}(X_i)} \left(2Y_i A_i - \alpha - \beta^T X_i\right)^2$$
  3. $$\widehat{\mathcal{D}}(x) = \mathrm{sign}(\hat{\alpha} + \hat{\beta}^T x)$$
- Outcome weighted learning (Zhao et al. '12):
  1. $$f^*(x) = \alpha^* + (\beta^*)^T x$$
  2. $$\left(\hat{\alpha}, \hat{\beta}\right) = \underset{\alpha,\beta}{\text{argmin}} \quad \frac{1}{n} \sum_{i=1}^{n} \frac{Y_i}{\pi_{A_i}(X_i)} \left(1 - A_i\left(\alpha + \beta^T X_i\right)\right)_+ + \lambda \|\beta\|^2$$
  3. $$\widehat{\mathcal{D}}(x) = \mathrm{sign}(\hat{\alpha} + \hat{\beta}^T x)$$

Our motivation

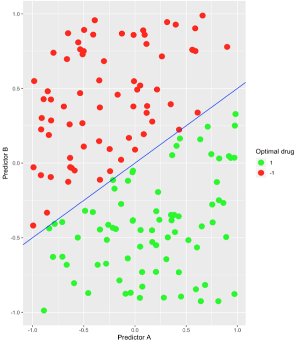


Linear ITR

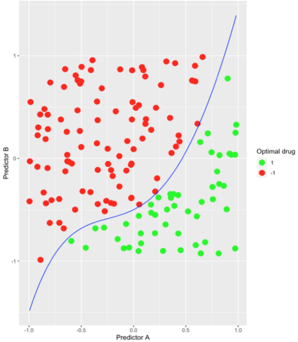


Nonlinear ITR (our focus today)

Our motivation



Linear ITR



Nonlinear ITR (our focus today)

- **Motivation**: how can we use indirect and direct learning frameworks to accurately estimate highly nonlinear optimal ITRs?

Proposed method I: nonparametric version of Q-learning

Key ideas

- Additive regression trees: assume

$$Q(x,1) = \sum_{t=1}^{K} b_1^{(t)}(x),$$

and

$$Q(x,-1) = \sum_{t=1}^{K} b_{-1}^{(t)}(x),$$

where $b_1^{(t)}(x)$ and $b_{-1}^{(t)}(x)$ are regression trees
- Use boosting algorithm to estimate regression trees sequentially

XGBoost algorithm

Take $A_i = 1$ group as an example:

- 1st iteration:

Estimation of $b_1^{(1)}$

1. Fit a tree to the training data (X_i, Y_i):

$$\hat{f} = \underset{f}{\operatorname{argmin}} \sum_{i:A_i=1} (Y_i - f(X_i))^2 + J(f),$$

where f is a regression tree, $J(f)$ is the cost complexity of a regression tree,

$$J(f) = \gamma|T| + \frac{1}{2}\lambda\|w\|_2^2$$

2. Shrinkage:

$$\hat{b}_1^{(1)} = \eta\hat{f},$$

where $0 < \eta < 1$

XGBoost algorithm

- t-th iteration:

Estimation of $b_1^{(t)}$

1. Fit a tree to the training data (X_i, Y_i):

$$\hat{f} = \underset{f}{\operatorname{argmin}} \sum_{i:A_i=1} [Y_i - (\hat{Y}_i^{(t-1)} + f(X_i))]^2 + J(f),$$

   where $\hat{Y}_i^{(t-1)} = \sum_{k=1}^{t-1} \hat{b}_1^{(k)}(X_i)$ is the estimated outcome value of X_i after (t-1)-th iteration

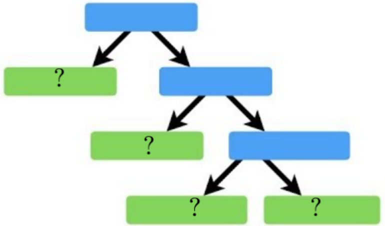
2. Shrinkage:

$$\hat{b}_1^{(t)} = \eta \hat{f}$$

- Output the boosted model:

$$\widehat{Q}(x, 1) = \sum_{t=1}^{K} \hat{b}_1^{(t)}(x)$$

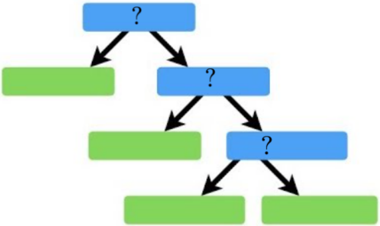
How do we fit a regression tree?



- Decide optimal leaf weights: for a fixed tree structure T, let $I_j = \{i | q(X_i) = j\}$ be the instance set of leaf j. Then

$$w_j^* = \frac{2\sum_{i \in I_j}(Y_i - \hat{Y}_i^{(t-1)})}{2|I_j| + \lambda}$$

How do we fit a regression tree?



- Decide optimal leaf weights: for a fixed tree structure T, let $I_j = \{i | q(X_i) = j\}$ be the instance set of leaf j. Then

$$w_j^* = \frac{2\sum_{i \in I_j}(Y_i - \hat{Y}_i^{(t-1)})}{2|I_j| + \lambda}$$

- Split finding algorithm for estimating tree structure T: Chen and Guestrin, '16

Summary of Algorithm I

Algorithm I (W. and Fu, '20)

Input: data set $\{(X_i, Y_i, A_i)\}_{i=1}^n$, number of iterations K, learning rate η, maximum of tree depth d

1. Train bst.plus1 = XGBoost($\{(X_i, Y_i); A_i = 1\}, K, \eta, d$)
2. Train bst.minus1 = XGBoost($\{(X_i, Y_i); A_i = -1\}, K, \eta, d$)
3. The estimated optimal ITR is

$$\widehat{\mathcal{D}}(x) = \mathrm{sign}(\text{bst.plus1}(x) - \text{bst.minus1}(x))$$

Proposed method II:
nonparametric version of D-learning

Key ideas

- Assume $f^*(x) = \sum_{t=1}^{K} b^{(t)}(x)$ where $b^{(t)}$ are regression trees
- Use boosting algorithm to estimate $b^{(t)}$ sequentially

Key ideas

- Assume $f^*(x) = \sum_{t=1}^{K} b^{(t)}(x)$ where $b^{(t)}$ are regression trees
- Use boosting algorithm to estimate $b^{(t)}$ sequentially
- t-th iteration of XGBoost:

Estimation of $b^{(t)}$

1. Fit a tree to the training data $(X_i, 2Y_iA_i)$:

$$\hat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{1}{\pi_{A_i}(X_i)} \left[2Y_iA_i - \left(\hat{Y}_i^{(t-1)} + f(X_i) \right) \right]^2 + J(f)$$

2. Shrinkage:

$$\hat{b}^{(t)} = \eta \hat{f}$$

Summary of Algorithm II

Algorithm II (W. and Fu, '20)

Input: data set $\{(X_i, A_i, Y_i)\}_{i=1}^n$, number of iterations K, shrinkage parameter η and maximum tree depth d.

1. Train bst = XGBoost($\{X_i, 2Y_iA_i\}, K, \eta, d$) with weighted quadratic loss
2. The estimated optimal ITR is

$$\widehat{\mathcal{D}}(x) = \mathrm{sign}(\mathsf{bst}(x))$$

Proposed method III:
nonparametric **refined** version of outcome weighted learning

Key ideas

Fisher consistency theorem (W. and Fu, '20)

Assume $Y = \mu(X) + \delta(X) \times A + \varepsilon$. Then we have

$$\mu = \underset{g}{\operatorname{argmin}} \quad E\left\{\frac{1}{\pi_A(X)}(Y - g(X))^2\right\}.$$

Furthermore, let

$$f^{**} = \underset{f}{\operatorname{argmin}} \quad E\left\{\frac{|Y - \mu(X)|}{\pi_A(X)}\phi\left(Af(X) \times \operatorname{sign}(Y - \mu(X))\right)\right\},$$

where $\phi(x) = \log(1 + e^{-2x})$. Then we have

$$\mathcal{D}^*(x) = \operatorname{sign}(f^{**}(x)).$$

- Assume $f^{**}(x) = \sum_{t=1}^{K} b^{(t)}(x)$ where $b^{(t)}$ are regression trees
- Use boosting algorithm to estimate $b^{(t)}$ sequentially

Key ideas

- Before XGBoost:

Estimation of $\mu(x)$

1. Assume $\mu(x) = \alpha_0 + \alpha^T x$
2. Estimate α_0 and α: $\hat{\alpha}_0, \hat{\alpha} = \underset{\alpha_0, \alpha}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{1}{\pi_{A_i}(X_i)} \left(Y_i - \alpha_0 - \alpha^T X_i\right)^2$
3. Estimate $\mu(x)$: $\hat{\mu}(x) = \hat{\alpha}_0 + \hat{\alpha}^T x$

Key ideas

- Before XGBoost:

Estimation of $\mu(x)$

1. Assume $\mu(x) = \alpha_0 + \alpha^T x$
2. Estimate α_0 and α: $\hat{\alpha}_0, \hat{\alpha} = \underset{\alpha_0, \alpha}{\text{argmin}} \sum_{i=1}^{n} \frac{1}{\pi_{A_i}(X_i)} \left(Y_i - \alpha_0 - \alpha^T X_i\right)^2$
3. Estimate $\mu(x)$: $\hat{\mu}(x) = \hat{\alpha}_0 + \hat{\alpha}^T x$

- t-th iteration of XGBoost:

Estimation of $b^{(t)}$

1. Fit a tree to the training data:

$$\hat{f} = \underset{f}{\text{argmin}} \sum_{i=1}^{n} \frac{|Y_i - \hat{\mu}(X_i)|}{\pi_{A_i}(X_i)} \phi\left(A_i\left(\hat{Y}_i^{(t-1)} + f(X_i)\right) \times \mathrm{sign}\left(Y_i - \hat{\mu}(X_i)\right)\right) + J(f)$$

2. Shrinkage: $\hat{b}^{(t)} = \eta \hat{f}$

Summary of Algorithm III

Algorithm III (W. and Fu, '20)

Input: data set $\{(X_i, A_i, Y_i)\}_{i=1}^n$, number of iterations K, shrinkage parameter η and maximum tree depth d.

1. Estimate the common effect μ.
2. Train bst = XGBoost($\{X_i, \mathrm{sign}(Y_i - \hat{\mu}(X_i))A_i\}, K, \eta, d$) with weighted deviance loss
3. Output the estimated optimal ITR:

$$\widehat{\mathcal{D}}(x) = \mathrm{sign}\left(\mathsf{bst}(x)\right)$$

Comparison of three algorithms

	Nonparametric	Indirect learning	Direct learning	Regression	Classification
Algorithm I	✓	✓		✓	
Algorithm II	✓		✓	✓	
Algorithm III	✓		✓		✓

Simulation and real data analysis

Performance measures

For a data set $\{(X_i, A_i, Y_i), 1 \le i \le n\}$,

- Misclassification rate:

$$\frac{1}{n}\sum_{i=1}^{n} I(\mathcal{D}^*(X_i) \neq \mathcal{D}(X_i))$$

- Value function:

$$V(\mathcal{D}) = E^{\mathcal{D}}(Y) = E\left\{Y \frac{I(A = \mathcal{D}(X))}{\pi_A(X)}\right\}$$

$$\widehat{V}(\mathcal{D}) = \frac{\frac{1}{n}\sum_{i=1}^{n} \frac{Y_i}{\pi_{A_i}(X_i)} I(\mathcal{D}(X_i) = A_i)}{\frac{1}{n}\sum_{i=1}^{n} \frac{I(\mathcal{D}(X_i)=A_i)}{\pi_{A_i}(X_i)}}$$

Simulation settings

- Generate each component of $X_i \in \mathbb{R}^{10}$ independently from $U(-1,1)$
- Generate A_i from $\{-1,1\}$ with $P(A_i = -1) = P(A_i = 1) = 0.5$
- Generate Y_i from the model

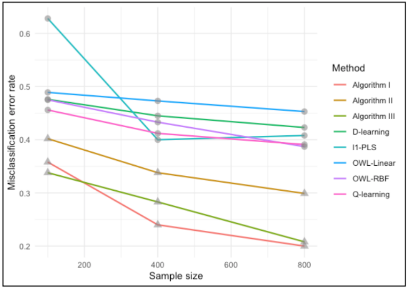
$$Y_i = 1 + 2X_{1i} + X_{2i} + 0.5X_{3i} + \delta(X_i) \times A_i + \varepsilon_i,$$

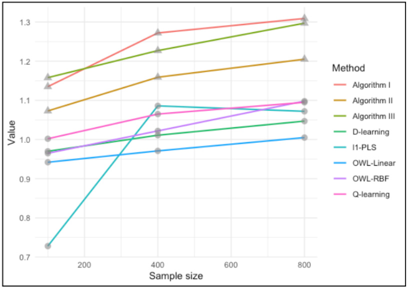
  where $\varepsilon_i \sim N(0,1)$. X_{1i}, X_{2i} and X_{3i} are the first, second and third components of X_i

- Polynomial-type optimal ITR:

$$\delta(X_i) = 0.2 + X_{1i}^2 + X_{2i}^2 - X_{3i}^2 - X_{4i}^2$$

Simulation results





- Algorithm I vs. Q-learning/ℓ_1-PLS: Algorithm I wins
- Algorithm II vs. D-learning: Algorithm II wins
- Algorithm III vs. OWL-Linear/OWL-RBF: Algorithm III wins
- Overall, Algorithm I and Algorithm III outperform Algorithm II

Diabetes data analysis

- The data was collected from a randomized, double-blind, parallel-group Phase III trial (Charbonnel, Matthews et al., '04)
- Compare drug efficacy of gliclazide and pioglitazone
- Among 1247 patients, 624 patients received gliclazide and 623 received pioglitazone
- 21 pretreatment covariates, e.g., BMI and blood pressure
- Primary efficacy endpoint: change of HbA1c level during 52 weeks
- Perform a 10-fold cross validation to obtain the predicted optimal treatment for each patient

Diabetes data analysis

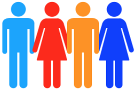
- The data was collected from a randomized, double-blind, parallel-group Phase III trial (Charbonnel, Matthews et al., '04)
- Compare drug efficacy of gliclazide and pioglitazone
- Among 1247 patients, 624 patients received gliclazide and 623 received pioglitazone
- 21 pretreatment covariates, e.g., BMI and blood pressure
- Primary efficacy endpoint: change of HbA1c level during 52 weeks
- Perform a 10-fold cross validation to obtain the predicted optimal treatment for each patient
- Estimated value results:

Method	Algorithm I	Algorithm II	Algorithm III	Q-learning	l1-PLS	D-learning	OWL-Linear	OWL-RBF
Estimated value	1.447	1.422	**1.448**	1.369	1.428	1.416	1.360	1.363

Diabetes data analysis

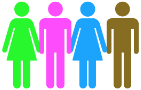
- Hypothesis testing:
  - Welch's t-test

μ_1 : average reduction of HbA1c for Group 1 $\quad$ μ_2 : average reduction of HbA1c for Group 2



$H_0 : \mu_1 = \mu_2$

$H_A : \mu_1 > \mu_2$



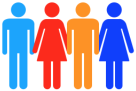
Group 1: patients whose assigned treatments were same with the estimated optimal ones

Group 2: remaining patients

Diabetes data analysis

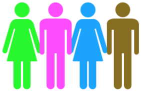
- Hypothesis testing:
  - Welch's t-test

μ_1 : average reduction of HbA1c for Group 1 $\quad$ μ_2 : average reduction of HbA1c for Group 2



$$H_0 : \mu_1 = \mu_2$$

$$H_A : \mu_1 > \mu_2$$



Group 1: patients whose assigned treatments were same with the estimated optimal ones

Group 2: remaining patients

- Results:

Method	Algorithm I	Algorithm II	Algorithm III	Q-learning	l1-PLS	D-learning	OWL-Linear	OWL-RBF
Proportion of significant p-values	**0.71**	0.37	0.69	0	0.44	0.29	0	0.04
Median of p-values	0.022	0.082	0.022	0.500	0.060	0.095	0.637	0.584

Significant: p-value<0.05

Summary

Takeaway points:

- Modelled the conditional mean of clinical outcome and the decision rule via additive regression trees
- Applied boosting technique to estimate each single tree sequentially
- Our approaches are very useful when the underlying optimal ITR is highly nonlinear and complex

Summary

Takeaway points:

- Modelled the conditional mean of clinical outcome and the decision rule via additive regression trees
- Applied boosting technique to estimate each single tree sequentially
- Our approaches are very useful when the underlying optimal ITR is highly nonlinear and complex
- Statistical aspects of ITR are well established. **But making ITR a reality needs collaboration with doctors, engineers, regulators, and enterprise leaders. Together we can save lives**

Reference

- D. Wang and H. Fu (2020). Boosting algorithms for estimating optimal individualized treatment rules. arXiv:2002.00079

Reference

- D. Wang and H. Fu (2020). Boosting algorithms for estimating optimal individualized treatment rules. arXiv:2002.00079

Thank you and stay safe!

Value function

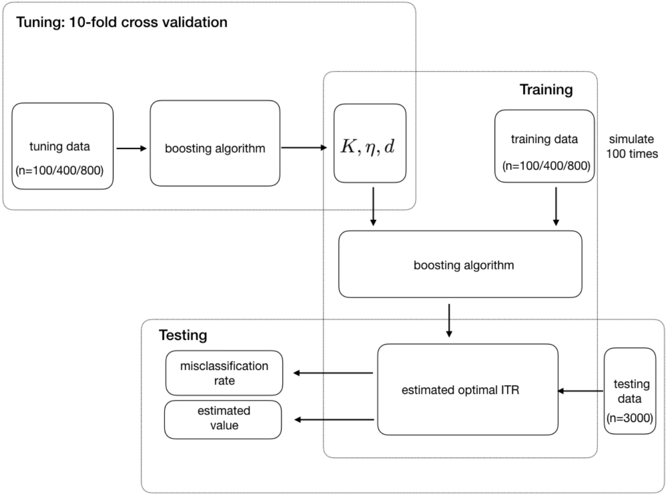
- Value function of ITR $\mathcal{D}$:

$$V(\mathcal{D}) = \mathbb{E}^{\mathcal{D}}(Y) = \int Y dP^{\mathcal{D}} = \int P \frac{dP^{\mathcal{D}}}{dP} dP = E\left[Y \frac{I(A = \mathcal{D}(X))}{\pi_A(X)} \right]$$

- Optimal ITR satisfies

$$\mathcal{D}^* = \underset{\mathcal{D}}{\operatorname{argmax}} \quad V(\mathcal{D})$$

Simulation pipeline



Real data analysis pipeline