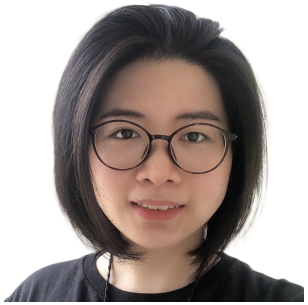
Deep Jump Learning for Off-Policy Evaluation in Continuous Treatment Settings



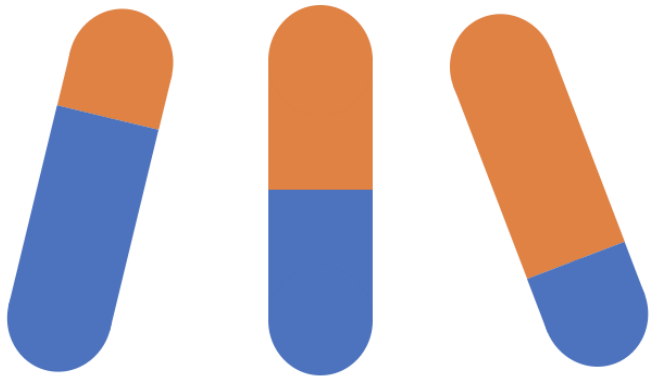

Hengrui Cai[1] Chengchun Shi[2] Wenbin Lu[1] Rui Song[1]

[1]Department of Statistics, North Carolina State University, North Carolina, USA
[2]Department of Statistics, London School of Economics and Political Science, London, UK

NeurIPS 2021

Individualized Decision Making

Personalized dose finding: Developing an individualized dose level for patients to optimize expected clinical outcomes of interest [Medicine];

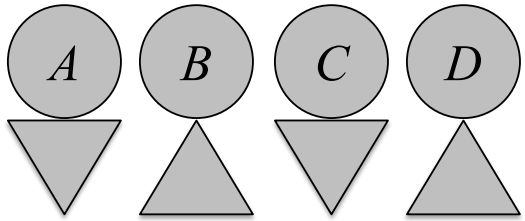


Individualized Decision Making

Dynamic pricing: Offering customized incentives/pricing strategy to increase sales and level of engagement [Economics];

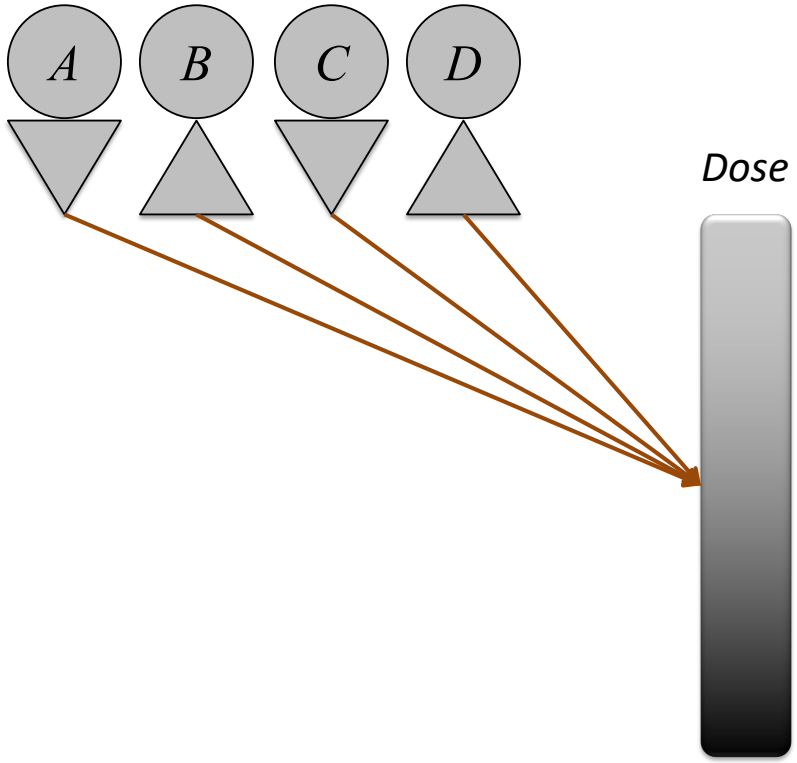
Motivation

Consider a decision making problem in a continuous treatment domain:



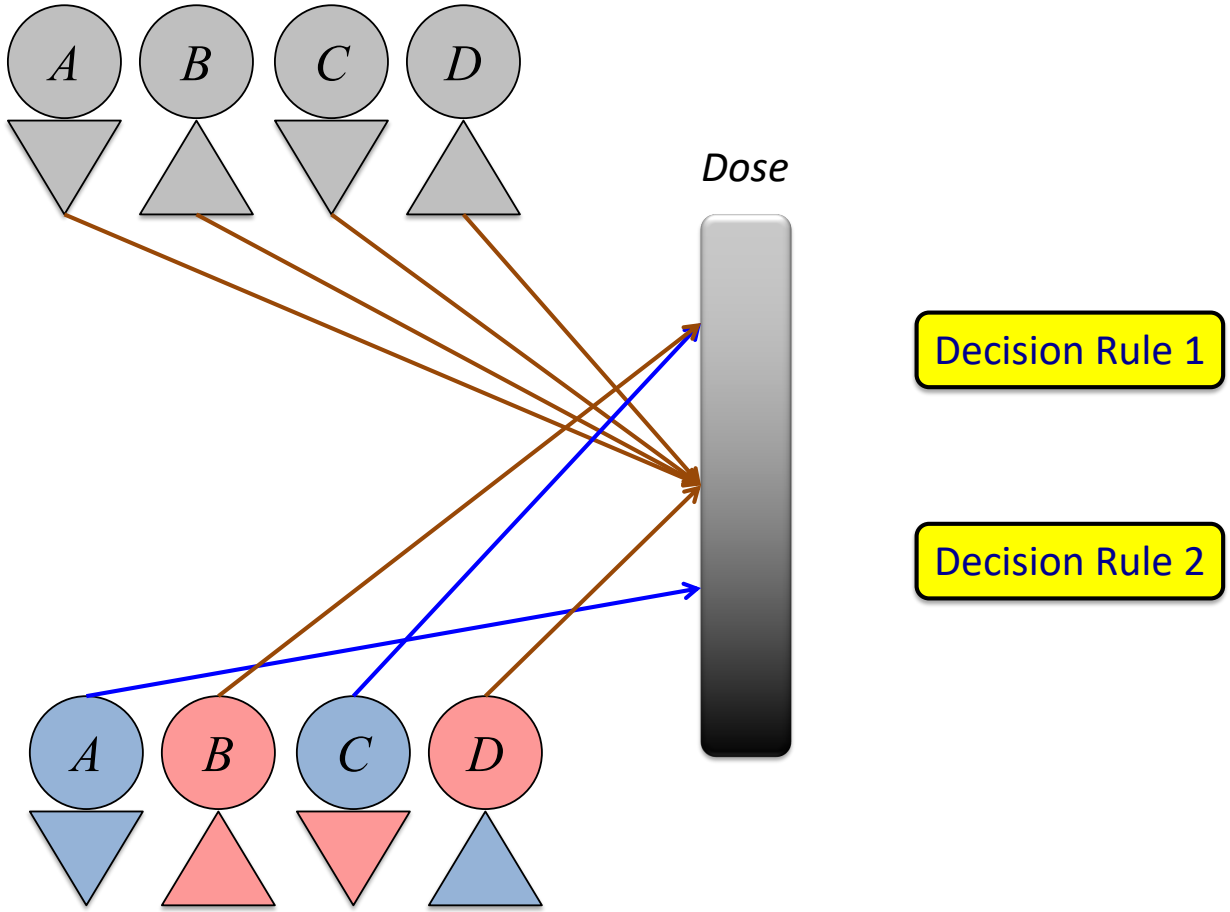
Motivation

Decision 1: a simple decision rule/policy that always assigns individuals to a fixed best treatment option.



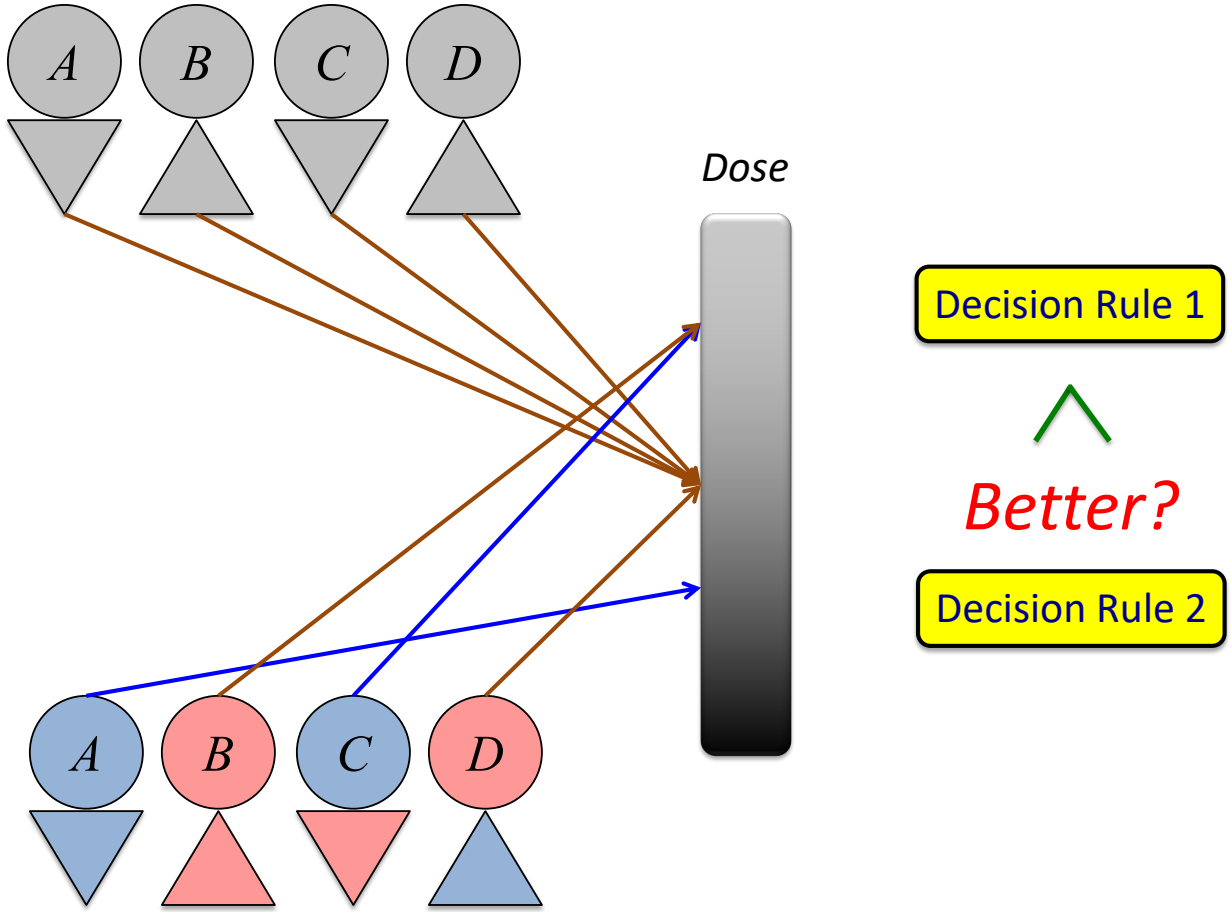
Motivation

Decision 2: an individualized decision rule/policy that assigns individuals with treatments according to their features.



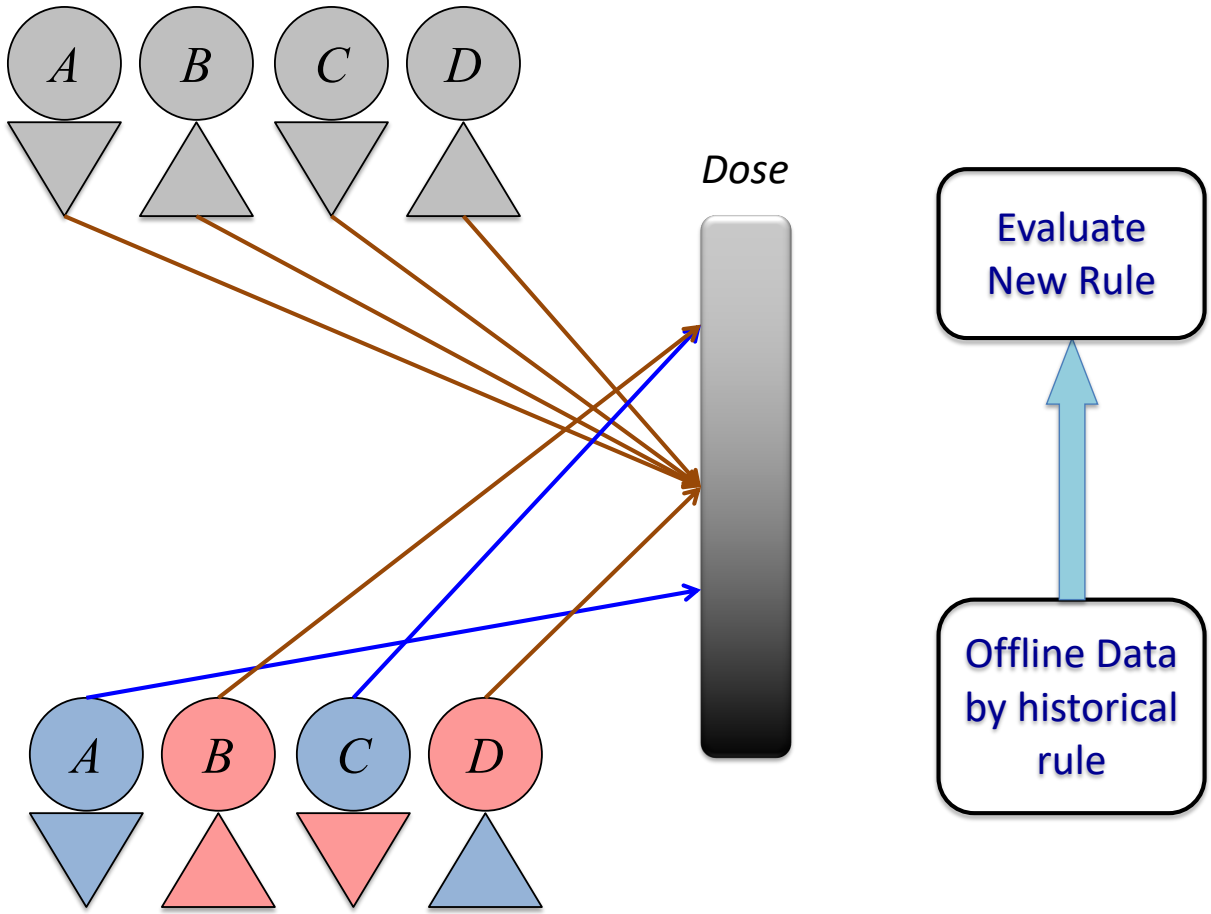
Motivation

Prior to adopting any decision rule in practice, it is crucial to know the impact of implementing such a rule.



Motivation

It is risky to apply a treatment decision rule or policy online to estimate its mean outcome. Policy evaluation proposes to use the offline data from a different historical rule.



Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Statistical Framework

- Offline Data: $O_i = (X_i, A_i, Y_i)$, $i = 1, \cdots, n$;
  - $X_i \in \mathcal{X}$: p-dimensional features.
  - $A_i \in \mathcal{A}$: received continuous treatment. w.l.o.g., set $\mathcal{A} = [0, 1]$.
  - Y_i: outcome of interest, the larger the better.
- A decision rule or policy $\pi(X) : \mathcal{X} \to \mathcal{A}$.
- Propensity score / behavior policy: $b(\bullet|x)$ is the probability density function of A given $X = x$ that generates the observed data.
- Assume stable unit treatment value assumption (SUTVA), no unmeasured confounders, and the positivity.
- Value: $V(\pi) = E[Q\{X, \pi(X)\}]$ with $Q(x, a) = E(Y|X = x, A = a)$.
- **Goal**: estimate the value $V(\pi)$ given any target policy π based on the observed data.

Related Works

- Most of current works on personalized decision making focus on policy optimization not policy evaluation;
  - See e.g.,Chakraborty et al. (2010), Song et al. (2015), Shi et al. (2018).
- Majority of offline policy evaluation methods focus on binary/finite treatment options.
  - See e.g., Wang et al. (2012), Zhang et al. (2012), Chakraborty et al. (2014), Luedtke and Van Der Laan (2016).
- A doubly robust (DR) estimator of $V(\pi)$ for discrete treatments (see e.g., Zhang et al. 2012):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{\mathbb{I}\{A_i=\pi(X_i)\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

where $\mathbb{I}(\bullet)$ denotes the indicator function, $\widehat{Q}$ and $\widehat{b}$ denote some estimators for the Q-function and the propensity score function.

Related Works

- Most of current works on personalized decision making focus on policy optimization not policy evaluation;
  - See e.g.,Chakraborty et al. (2010), Song et al. (2015), Shi et al. (2018).
- Majority of offline policy evaluation methods focus on binary/finite treatment options.
  - See e.g., Wang et al. (2012), Zhang et al. (2012), Chakraborty et al. (2014), Luedtke and Van Der Laan (2016).
- A doubly robust (DR) estimator of $V(\pi)$ for discrete treatments (see e.g., Zhang et al. 2012):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{\mathbb{I}\{A_i=\pi(X_i)\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

where $\mathbb{I}(\bullet)$ denotes the indicator function, $\widehat{Q}$ and $\widehat{b}$ denote some estimators for the Q-function and the propensity score function.

Related Works

- Most of current works on personalized decision making focus on policy optimization not policy evaluation;
  - See e.g.,Chakraborty et al. (2010), Song et al. (2015), Shi et al. (2018).
- Majority of offline policy evaluation methods focus on binary/finite treatment options.
  - See e.g., Wang et al. (2012), Zhang et al. (2012), Chakraborty et al. (2014), Luedtke and Van Der Laan (2016).
- A doubly robust (DR) estimator of $V(\pi)$ for discrete treatments (see e.g., Zhang et al. 2012):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{\mathbb{I}\{A_i=\pi(X_i)\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

where $\mathbb{I}(\bullet)$ denotes the indicator function, $\widehat{Q}$ and $\widehat{b}$ denote some estimators for the Q-function and the propensity score function.

Related Works

- Available methods for continuous treatments rely on the use of a kernel function. A DR estimator of $V(\pi)$ for continuous treatments (see e.g., Kallus and Zhou 2018, Colangelo and Lee 2020):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{K\{\frac{A_i-\pi(X_i)}{h}\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

  where $K(\cdot)$ is a kernel function and h is the kernel bandwidth.

- Limitation 1: Require the mean outcome to be smooth over the treatment space;
  - In dynamic pricing, the expected demand for a product has jump discontinuities as a function of the charged price (den Boer and Keskin 2020).
- Limitation 2: Use a single bandwidth parameter, which may be sub-optimal;
  - when the second-order derivative of the conditional mean function has an abrupt change in the treatment space.

Related Works

- Available methods for continuous treatments rely on the use of a kernel function. A DR estimator of $V(\pi)$ for continuous treatments (see e.g., Kallus and Zhou 2018, Colangelo and Lee 2020):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{K\{\frac{A_i-\pi(X_i)}{h}\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

  where $K(\cdot)$ is a kernel function and h is the kernel bandwidth.
- Limitation 1: Require the mean outcome to be smooth over the treatment space;
  - In dynamic pricing, the expected demand for a product has jump discontinuities as a function of the charged price (den Boer and Keskin 2020).
- Limitation 2: Use a single bandwidth parameter, which may be sub-optimal;
  - when the second-order derivative of the conditional mean function has an abrupt change in the treatment space.

Related Works

- Available methods for continuous treatments rely on the use of a kernel function. A DR estimator of $V(\pi)$ for continuous treatments (see e.g., Kallus and Zhou 2018, Colangelo and Lee 2020):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{K\{\frac{A_i-\pi(X_i)}{h}\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

  where $K(\cdot)$ is a kernel function and h is the kernel bandwidth.
- Limitation 1: Require the mean outcome to be smooth over the treatment space;
  - In dynamic pricing, the expected demand for a product has jump discontinuities as a function of the charged price (den Boer and Keskin 2020).
- Limitation 2: Use a single bandwidth parameter, which may be sub-optimal;
  - when the second-order derivative of the conditional mean function has an abrupt change in the treatment space.

Related Works

- Available methods for continuous treatments rely on the use of a kernel function. A DR estimator of $V(\pi)$ for continuous treatments (see e.g., Kallus and Zhou 2018, Colangelo and Lee 2020):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{K\{\frac{A_i-\pi(X_i)}{h}\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

  where $K(\cdot)$ is a kernel function and h is the kernel bandwidth.
- Limitation 1: Require the mean outcome to be smooth over the treatment space;
  - In dynamic pricing, the expected demand for a product has jump discontinuities as a function of the charged price (den Boer and Keskin 2020).
- Limitation 2: Use a single bandwidth parameter, which may be sub-optimal;
  - when the second-order derivative of the conditional mean function has an abrupt change in the treatment space.

Related Works

- Available methods for continuous treatments rely on the use of a kernel function. A DR estimator of $V(\pi)$ for continuous treatments (see e.g., Kallus and Zhou 2018, Colangelo and Lee 2020):

$$\frac{1}{n}\sum_{i=1}^{n}\left[\widehat{Q}\{X_i,\pi(X_i)\}+\frac{K\{\frac{A_i-\pi(X_i)}{h}\}}{\widehat{b}(A_i|X_i)}\{Y_i-\widehat{Q}(X_i,A_i)\}\right],$$

  where $K(\cdot)$ is a kernel function and h is the kernel bandwidth.
- Limitation 1: Require the mean outcome to be smooth over the treatment space;
  - In dynamic pricing, the expected demand for a product has jump discontinuities as a function of the charged price (den Boer and Keskin 2020).
- Limitation 2: Use a single bandwidth parameter, which may be sub-optimal;
  - when the second-order derivative of the conditional mean function has an abrupt change in the treatment space.

Summary of Our Work

- Propose deep jump evaluation method for continuous treatments by integrating **multi-scale change point detection**, **deep learning**, and the **doubly-robust** value estimators in discrete domains;
- Our method does not require kernel bandwidth selection, by adaptively discretizing the treatment space using deep discretization;
- Our method has a better convergence rate, allowing the conditional mean outcome to be either a continuous or piecewise function of the treatment.

Summary of Our Work

- Propose deep jump evaluation method for continuous treatments by integrating **multi-scale change point detection**, **deep learning**, and the **doubly-robust** value estimators in discrete domains;
- Our method does not require kernel bandwidth selection, by adaptively discretizing the treatment space using deep discretization;
- Our method has a better convergence rate, allowing the conditional mean outcome to be either a continuous or piecewise function of the treatment.

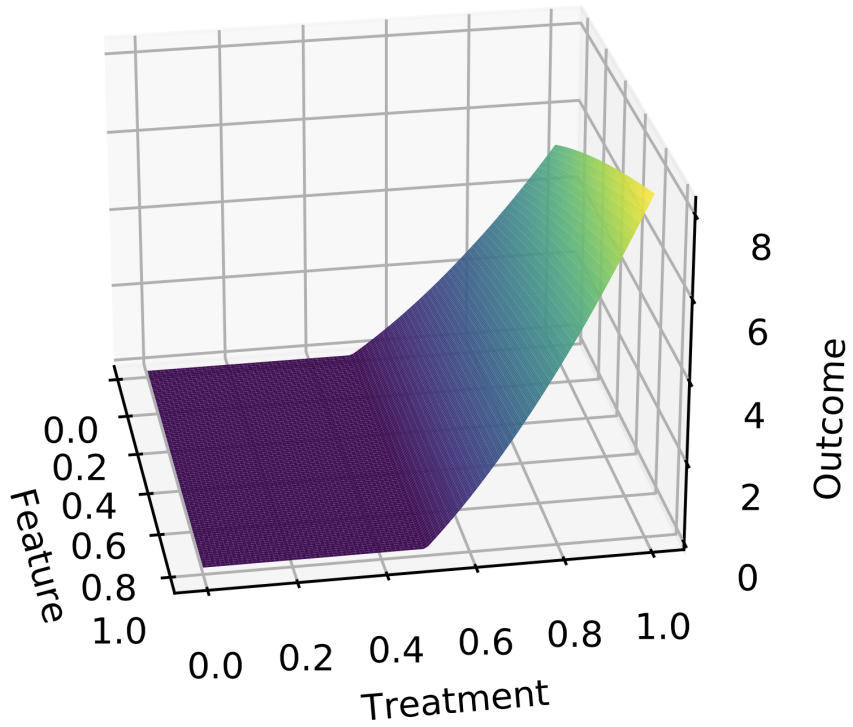
Summary of Our Work

- Propose deep jump evaluation method for continuous treatments by integrating **multi-scale change point detection**, **deep learning**, and the **doubly-robust** value estimators in discrete domains;
- Our method does not require kernel bandwidth selection, by adaptively discretizing the treatment space using deep discretization;
- Our method has a better convergence rate, allowing the conditional mean outcome to be either a continuous or piecewise function of the treatment.

Toy Example

Consider a **smooth** function $Q(x, a) = 10 \max(a^2 - 0.25, 0) \log(x + 2)$ for any $x, a \in [0, 1]$: with different patterns when the treatment belongs to different intervals:

- For $a \in [0, 0.5]$, $Q(x, a)$ is constant as a function of a.
- For $a \in (0.5, 1]$, $Q(x, a)$ depends quadratically in a.



Sub-optimality of Kernel-Based Method in Toy Example

Target policy: $\pi(x) = x$; the value $V(\pi) = V^{(1)}(\pi) + V^{(2)}(\pi)$ where

- $V^{(1)}(\pi) = \mathsf{E}[Q\{X, \pi(X)\}\mathbb{I}\{\pi(X) \leq 0.5\}]$;
- $V^{(2)}(\pi) = \mathsf{E}[Q\{X, \pi(X)\}\mathbb{I}\{\pi(X) > 0.5\}]$.

Bias (SD)	Indicator	Kernel with $h = 0.4$	Kernel with $h = 1$
$V^{(1)}(\pi)$	$\mathbb{I}\{\pi(X) \leq 0.5\}$	0.50 (0.08)	0.40 (0.05)
$V^{(2)}(\pi)$	$\mathbb{I}\{\pi(X) > 0.5\}$	0.16 (0.20)	1.09 (0.09)

Due to the use of a single bandwidth, the kernel-based estimator suffers from either a large bias or a large variance.

- By Theorem 1 of Kallus and Zhou (2018), the leading term of bias:

$$h^2 \frac{\int u^2 K(u) du}{2} \mathsf{E}\left\{ \frac{\partial^2 Q(X, a)}{\partial a^2}\bigg|_{a=\pi(X)} \right\}.$$

Sub-optimality of Kernel-Based Method in Toy Example

Target policy: $\pi(x) = x$; the value $V(\pi) = V^{(1)}(\pi) + V^{(2)}(\pi)$ where

- $V^{(1)}(\pi) = \mathsf{E}[Q\{X, \pi(X)\}\mathbb{I}\{\pi(X) \leq 0.5\}]$;
- $V^{(2)}(\pi) = \mathsf{E}[Q\{X, \pi(X)\}\mathbb{I}\{\pi(X) > 0.5\}]$.

Bias (SD)	Indicator	Kernel with $h = 0.4$	Kernel with $h = 1$
$V^{(1)}(\pi)$	$\mathbb{I}\{\pi(X) \leq 0.5\}$	0.50 (0.08)	0.40 (0.05)
$V^{(2)}(\pi)$	$\mathbb{I}\{\pi(X) > 0.5\}$	0.16 (0.20)	1.09 (0.09)

Due to the use of a single bandwidth, the kernel-based estimator suffers from either a large bias or a large variance.

- By Theorem 1 of Kallus and Zhou (2018), the leading term of bias:

$$h^2 \frac{\int u^2 K(u) du}{2} \mathsf{E}\left\{ \frac{\partial^2 Q(X, a)}{\partial a^2}\bigg|_{a=\pi(X)} \right\}.$$

Sub-optimality of Kernel-Based Method in Toy Example

Target policy: $\pi(x) = x$; the value $V(\pi) = V^{(1)}(\pi) + V^{(2)}(\pi)$ where

- $V^{(1)}(\pi) = \mathsf{E}[Q\{X, \pi(X)\}\mathbb{I}\{\pi(X) \leq 0.5\}]$;
- $V^{(2)}(\pi) = \mathsf{E}[Q\{X, \pi(X)\}\mathbb{I}\{\pi(X) > 0.5\}]$.

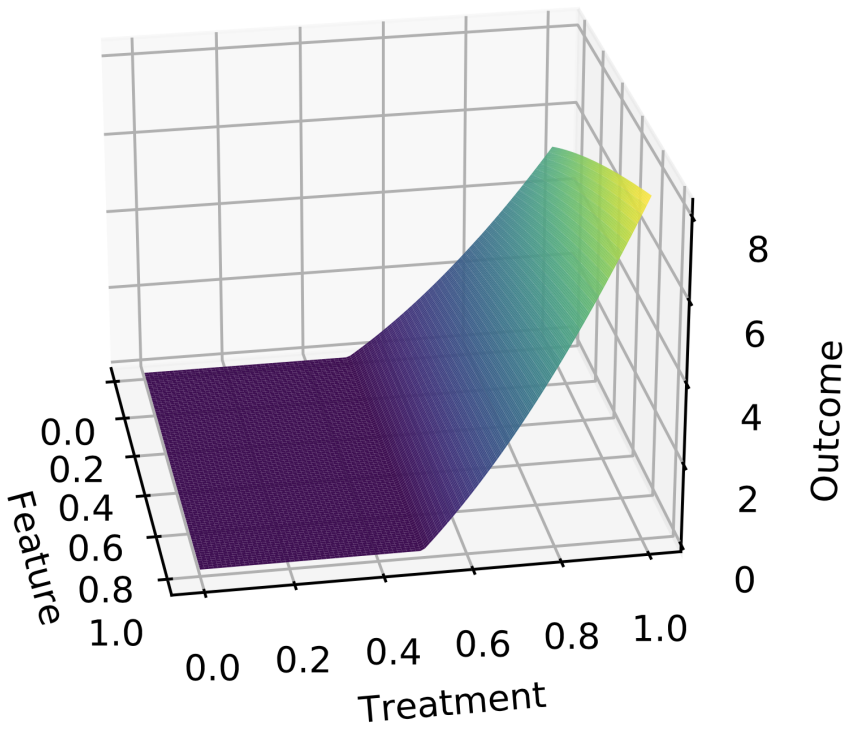
Bias (SD)	Indicator	Kernel with $h = 0.4$	Kernel with $h = 1$
$V^{(1)}(\pi)$	$\mathbb{I}\{\pi(X) \leq 0.5\}$	0.50 (0.08)	0.40 (0.05)
$V^{(2)}(\pi)$	$\mathbb{I}\{\pi(X) > 0.5\}$	0.16 (0.20)	1.09 (0.09)

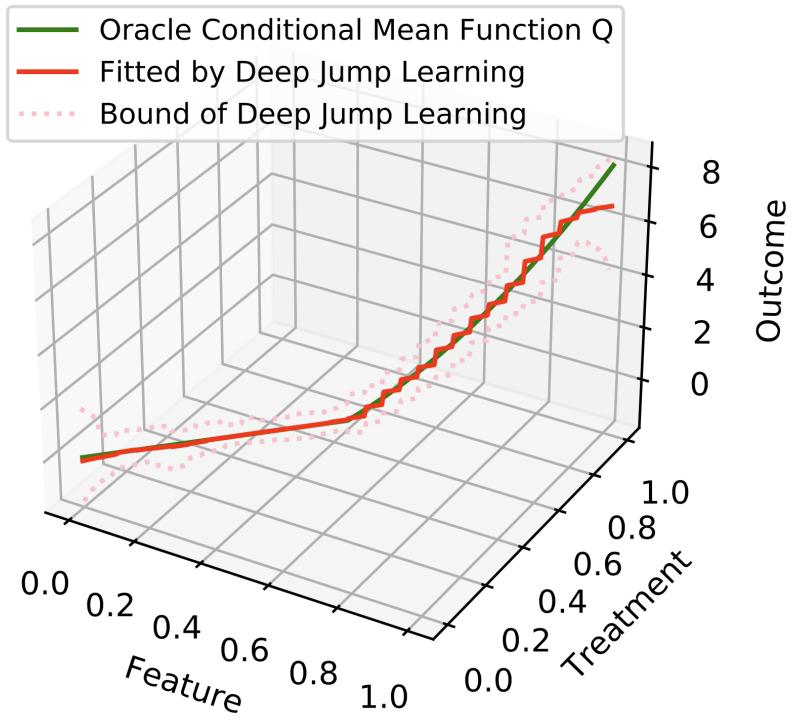
Due to the use of a single bandwidth, the kernel-based estimator suffers from either a large bias or a large variance.

- By Theorem 1 of Kallus and Zhou (2018), the leading term of bias:

$$h^2 \frac{\int u^2 K(u) du}{2} \mathsf{E}\left\{ \frac{\partial^2 Q(X, a)}{\partial a^2}\bigg|_{a=\pi(X)} \right\}.$$

Motivation from Toy Example: Adaptive Discretization

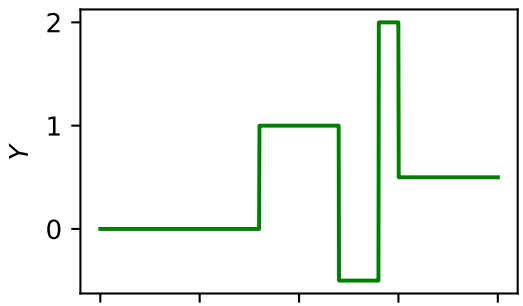


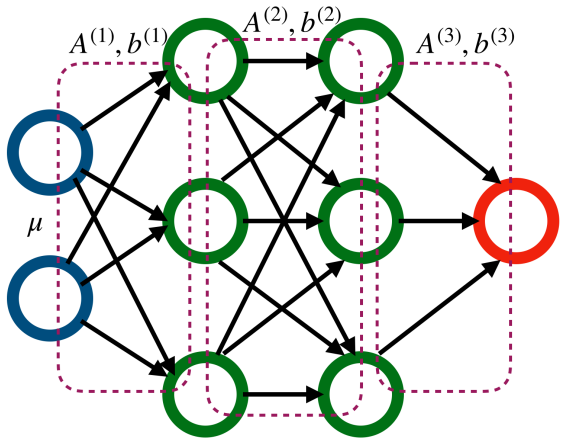


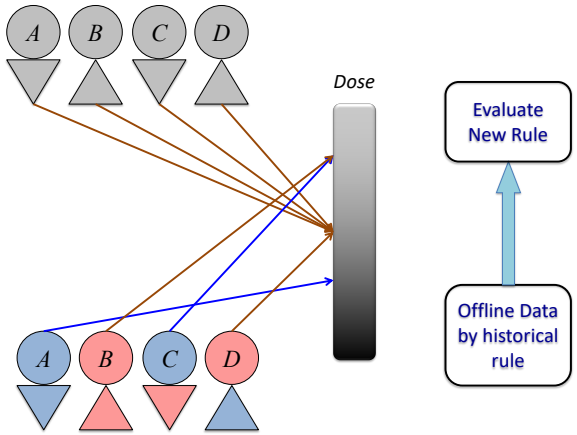
Bias (SD)	Indicator	Deep Jump Learning	Kernel with $h = 0.4$	Kernel with $h = 1$
$V^{(1)}(\pi)$	$\mathbb{I}\{\pi(X) \leq 0.5\}$	0.31 (0.06)	0.50 (0.08)	0.40 (0.05)
$V^{(2)}(\pi)$	$\mathbb{I}\{\pi(X) > 0.5\}$	0.09 (0.19)	0.16 (0.20)	1.09 (0.09)

Deep Jump Evaluation

Deep jump evaluation integrates **multi-scale change point detection**, **deep learning**, and the **doubly-robust** value estimators in discrete domains.







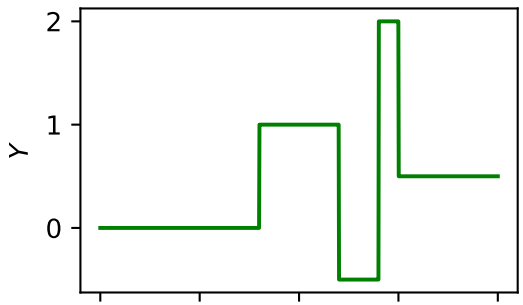
Deep jump evaluation works for both Model I and Model II:

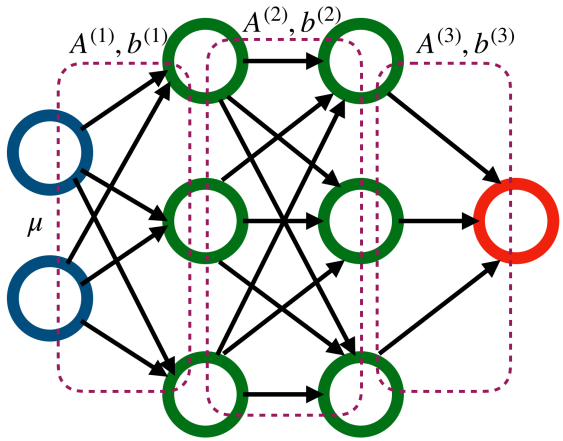
Model I: Piecewise function: $Q(x,a) = \sum_{\mathcal{I}\in\mathcal{D}_0}\{q_{\mathcal{I},0}(x)\mathbb{I}(a\in\mathcal{I})\}$, for some partition $\mathcal{D}_0$ of $[0,1]$ and a collection of functions $\{q_{\mathcal{I},0}\}_{\mathcal{I}\in\mathcal{D}_0}$.

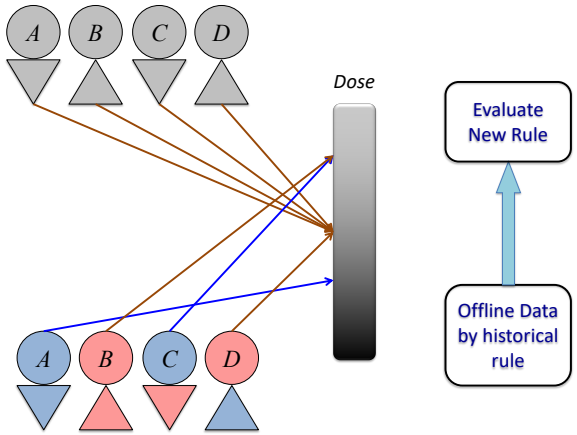
Model II: Continuous function: Q is a continuous function of a and x.

Deep Jump Evaluation

Deep jump evaluation integrates **multi-scale change point detection**, **deep learning**, and the **doubly-robust** value estimators in discrete domains.







Deep jump evaluation works for both Model I and Model II:

Model I: Piecewise function: $Q(x,a) = \sum_{\mathcal{I}\in\mathcal{D}_0} \{q_{\mathcal{I},0}(x)\mathbb{I}(a \in \mathcal{I})\}$, for some partition $\mathcal{D}_0$ of $[0,1]$ and a collection of functions $\{q_{\mathcal{I},0}\}_{\mathcal{I}\in\mathcal{D}_0}$.

Model II: Continuous function: Q is a continuous function of a and x.

Step 1: Deep Discretization

- Divide the treatment space $\mathcal{A}$ into m disjoint initial intervals $[0, 1/m), [1/m, 2/m), \ldots, [(m-1)/m, 1]$.
- Define $\mathcal{B}(m)$ as the set of candidate discretizations $\mathcal{D}$ so each interval $\mathcal{I} \in \mathcal{D}$ corresponds to a union of some of the m initial intervals.
- Each discretization $\mathcal{D} \in \mathcal{B}(m)$ is associated with a set of functions $\{q_{\mathcal{I}}\}_{\mathcal{I}\in\mathcal{D}}$, which **depend on** features, but not on the treatment.
- Model these $q_{\mathcal{I}}$ in some function class of deep neural networks $\mathcal{Q}_{\mathcal{I}}$, to capture the complex dependence between the outcome and features.
- **Estimate Discretization by**:

$$\left(\widehat{\mathcal{D}}, \{\widehat{q}_{\mathcal{I}} : \mathcal{I} \in \widehat{\mathcal{D}}\}\right) = \underset{(\mathcal{D}\in\mathcal{B}(m), \{q_{\mathcal{I}}\in\mathcal{Q}_{\mathcal{I}}:\mathcal{I}\in\mathcal{D}\})}{\arg\min} \left(\sum_{\mathcal{I}\in\mathcal{D}} \left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(A_i \in \mathcal{I}) \{Y_i - q_{\mathcal{I}}(X_i)\}^2 \right] + \gamma_n |\mathcal{D}| \right),$$

for some regularization parameter γ_n.

Step 1: Deep Discretization

- Divide the treatment space $\mathcal{A}$ into m disjoint initial intervals $[0, 1/m), [1/m, 2/m), \ldots, [(m-1)/m, 1]$.
- Define $\mathcal{B}(m)$ as the set of candidate discretizations $\mathcal{D}$ so each interval $\mathcal{I} \in \mathcal{D}$ corresponds to a union of some of the m initial intervals.
- Each discretization $\mathcal{D} \in \mathcal{B}(m)$ is associated with a set of functions $\{q_{\mathcal{I}}\}_{\mathcal{I} \in \mathcal{D}}$, which **depend on** features, but not on the treatment.
- Model these $q_{\mathcal{I}}$ in some function class of deep neural networks $\mathcal{Q}_{\mathcal{I}}$, to capture the complex dependence between the outcome and features.
- **Estimate Discretization by**:

$$\left(\widehat{\mathcal{D}}, \{\widehat{q}_{\mathcal{I}} : \mathcal{I} \in \widehat{\mathcal{D}}\}\right) = \underset{(\mathcal{D} \in \mathcal{B}(m), \{q_{\mathcal{I}} \in \mathcal{Q}_{\mathcal{I}} : \mathcal{I} \in \mathcal{D}\})}{\arg\min} \left(\sum_{\mathcal{I} \in \mathcal{D}} \left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(A_i \in \mathcal{I}) \{Y_i - q_{\mathcal{I}}(X_i)\}^2 \right] + \gamma_n |\mathcal{D}| \right),$$

for some regularization parameter γ_n.

Step 1: Deep Discretization

- Divide the treatment space $\mathcal{A}$ into m disjoint initial intervals $[0, 1/m), [1/m, 2/m), \ldots, [(m-1)/m, 1]$.
- Define $\mathcal{B}(m)$ as the set of candidate discretizations $\mathcal{D}$ so each interval $\mathcal{I} \in \mathcal{D}$ corresponds to a union of some of the m initial intervals.
- Each discretization $\mathcal{D} \in \mathcal{B}(m)$ is associated with a set of functions $\{q_{\mathcal{I}}\}_{\mathcal{I} \in \mathcal{D}}$, which **depend on** features, but not on the treatment.
- Model these $q_{\mathcal{I}}$ in some function class of deep neural networks $\mathcal{Q}_{\mathcal{I}}$, to capture the complex dependence between the outcome and features.
- **Estimate Discretization by**:

$$\left(\widehat{\mathcal{D}}, \{\widehat{q}_{\mathcal{I}} : \mathcal{I} \in \widehat{\mathcal{D}}\}\right) = \underset{(\mathcal{D} \in \mathcal{B}(m), \{q_{\mathcal{I}} \in \mathcal{Q}_{\mathcal{I}} : \mathcal{I} \in \mathcal{D}\})}{\arg\min} \left(\sum_{\mathcal{I} \in \mathcal{D}} \left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(A_i \in \mathcal{I}) \{Y_i - q_{\mathcal{I}}(X_i)\}^2 \right] + \gamma_n |\mathcal{D}| \right),$$

for some regularization parameter γ_n.

Step 1: Deep Discretization

- Divide the treatment space $\mathcal{A}$ into m disjoint initial intervals $[0, 1/m), [1/m, 2/m), \ldots, [(m-1)/m, 1]$.
- Define $\mathcal{B}(m)$ as the set of candidate discretizations $\mathcal{D}$ so each interval $\mathcal{I} \in \mathcal{D}$ corresponds to a union of some of the m initial intervals.
- Each discretization $\mathcal{D} \in \mathcal{B}(m)$ is associated with a set of functions $\{q_{\mathcal{I}}\}_{\mathcal{I}\in\mathcal{D}}$, which **depend on** features, but not on the treatment.
- Model these $q_{\mathcal{I}}$ in some function class of deep neural networks $\mathcal{Q}_{\mathcal{I}}$, to capture the complex dependence between the outcome and features.
- **Estimate Discretization by**:

$$\left(\widehat{\mathcal{D}}, \{\widehat{q}_{\mathcal{I}} : \mathcal{I} \in \widehat{\mathcal{D}}\}\right) = \underset{(\mathcal{D}\in\mathcal{B}(m), \{q_{\mathcal{I}}\in\mathcal{Q}_{\mathcal{I}}:\mathcal{I}\in\mathcal{D}\})}{\arg\min} \left(\sum_{\mathcal{I}\in\mathcal{D}} \left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(A_i \in \mathcal{I}) \{Y_i - q_{\mathcal{I}}(X_i)\}^2 \right] + \gamma_n |\mathcal{D}| \right),$$

for some regularization parameter γ_n.

Step 2: Policy Evaluation

Doubly Robust Estimator under Deep Jump Evaluation

Given $\widehat{\mathcal{D}}$ and $\{\widehat{q}_{\mathcal{I}} : \mathcal{I} \in \widehat{\mathcal{D}}\}$, the value for any decision rule of interest π is

$$\widehat{V}(\pi) = \frac{1}{n} \sum_{\mathcal{I} \in \widehat{\mathcal{D}}} \sum_{i=1}^{n} \left(\mathbb{I}\{\pi(X_i) \in \mathcal{I}\} \left[\frac{\mathbb{I}(A_i \in \mathcal{I})}{\widehat{b}_{\mathcal{I}}(X_i)} \{Y_i - \widehat{q}_{\mathcal{I}}(X_i)\} + \widehat{q}_{\mathcal{I}}(X_i) \right] \right),$$

where $\widehat{b}_{\mathcal{I}}(x)$ is some estimator of the generalized propensity score function $\Pr(A \in \mathcal{I} | X = x)$.

The complete algorithm consists of:

- Data Splitting: use different subsets of data samples to estimate the discretization and to construct the value estimator.
- Deep Discretization: apply pruned exact linear time method (Killick et al., 2012) in multi-scale change point detection.
- Cross-fitting: to remove the bias induced by overfitting.

Step 2: Policy Evaluation

Doubly Robust Estimator under Deep Jump Evaluation

Given $\widehat{\mathcal{D}}$ and $\{\widehat{q}_{\mathcal{I}} : \mathcal{I} \in \widehat{\mathcal{D}}\}$, the value for any decision rule of interest π is

$$\widehat{V}(\pi) = \frac{1}{n} \sum_{\mathcal{I} \in \widehat{\mathcal{D}}} \sum_{i=1}^{n} \left(\mathbb{I}\{\pi(X_i) \in \mathcal{I}\} \left[\frac{\mathbb{I}(A_i \in \mathcal{I})}{\widehat{b}_{\mathcal{I}}(X_i)} \{Y_i - \widehat{q}_{\mathcal{I}}(X_i)\} + \widehat{q}_{\mathcal{I}}(X_i) \right] \right),$$

where $\widehat{b}_{\mathcal{I}}(x)$ is some estimator of the generalized propensity score function $\Pr(A \in \mathcal{I} | X = x)$.

The complete algorithm consists of:

- Data Splitting: use different subsets of data samples to estimate the discretization and to construct the value estimator.
- Deep Discretization: apply pruned exact linear time method (Killick et al., 2012) in multi-scale change point detection.
- Cross-fitting: to remove the bias induced by overfitting.

Step 2: Policy Evaluation

Doubly Robust Estimator under Deep Jump Evaluation

Given $\widehat{\mathcal{D}}$ and $\{\widehat{q}_{\mathcal{I}} : \mathcal{I} \in \widehat{\mathcal{D}}\}$, the value for any decision rule of interest π is

$$\widehat{V}(\pi) = \frac{1}{n} \sum_{\mathcal{I} \in \widehat{\mathcal{D}}} \sum_{i=1}^{n} \left(\mathbb{I}\{\pi(X_i) \in \mathcal{I}\} \left[\frac{\mathbb{I}(A_i \in \mathcal{I})}{\widehat{b}_{\mathcal{I}}(X_i)} \{Y_i - \widehat{q}_{\mathcal{I}}(X_i)\} + \widehat{q}_{\mathcal{I}}(X_i) \right] \right),$$

where $\widehat{b}_{\mathcal{I}}(x)$ is some estimator of the generalized propensity score function $\Pr(A \in \mathcal{I} | X = x)$.

The complete algorithm consists of:

- Data Splitting: use different subsets of data samples to estimate the discretization and to construct the value estimator.
- Deep Discretization: apply pruned exact linear time method (Killick et al., 2012) in multi-scale change point detection.
- Cross-fitting: to remove the bias induced by overfitting.

Convergence Rates

Theorem 1 (under Model 1 (Piecewise Function))

Suppose m *is proportional to* n *and* $\{\gamma_n\}_{n\in\mathbb{N}}$ *satisfies* $\gamma_n \to 0$ *and* $\gamma_n \gg n^{-\epsilon}$ *for some* $\epsilon > -2\beta/(2\beta+p)$ *for* β*-smoothness. Then, there exist some classes of deep neural networks such that for any decision rule* π*,*

$$\widehat{V}(\pi) = V(\pi) + O_p\{n^{-2\beta/(2\beta+p)} \log^8 n\} + O_p(n^{-1/2}).$$

Theorem 2 (under Model 2 (Continuous Function))

Suppose m *is proportional to* n *and* γ_n *is proportional to* $\max\{n^{-3/5}, n^{-2\beta/(2\beta+p)} \log^9 n\}$*. Then for any decision rule* π*,*

$$\widehat{V}(\pi) - V(\pi) = O_p(n^{-1/5}) + O_p\{n^{-2\beta/(6\beta+3p)} \log^3 n\}.$$

Real Data Analysis: Warfarin Dosing

- $p = 81$ baseline covariates X.
- Continuous Treatment A: the dose of Warfarin, converted into $[0, 1]$.
- Outcome of interest Y: is defined as the absolute distance between the international normalized ratio (INR, a measurement of the time it takes for the blood to clot) after the treatment and the ideal value 2.5, i.e, $Y = -|\text{INR} - 2.5|$.
- The goal is to evaluate the value function under a decision rule of interest offline, based on the Warfarin dataset.

Real Data Analysis: Warfarin Dosing

- $p = 81$ baseline covariates X.
- Continuous Treatment A: the dose of Warfarin, converted into $[0, 1]$.
- Outcome of interest Y: is defined as the absolute distance between the international normalized ratio (INR, a measurement of the time it takes for the blood to clot) after the treatment and the ideal value 2.5, i.e, $Y = -|\text{INR} - 2.5|$.
- The goal is to evaluate the value function under a decision rule of interest offline, based on the Warfarin dataset.

Real Data Analysis: Warfarin Dosing

- $p = 81$ baseline covariates X.
- Continuous Treatment A: the dose of Warfarin, converted into $[0, 1]$.
- Outcome of interest Y: is defined as the absolute distance between the international normalized ratio (INR, a measurement of the time it takes for the blood to clot) after the treatment and the ideal value 2.5, i.e, $Y = -|\text{INR} - 2.5|$.
- The goal is to evaluate the value function under a decision rule of interest offline, based on the Warfarin dataset.

Real Data Analysis: Warfarin Dosing

- $p = 81$ baseline covariates X.
- Continuous Treatment A: the dose of Warfarin, converted into $[0, 1]$.
- Outcome of interest Y: is defined as the absolute distance between the international normalized ratio (INR, a measurement of the time it takes for the blood to clot) after the treatment and the ideal value 2.5, i.e, $Y = -|\text{INR} - 2.5|$.
- The goal is to evaluate the value function under a decision rule of interest offline, based on the Warfarin dataset.

Implementation and Results

- Decision rule of interest: **the optimal decision rule** $\pi^{\star}(X)$;
- Benchmarks (kernel-based methods): **Kallus & Zhou (2018)**, **SLOPE (Su et al. 2020)**, **Colangelo & Lee (2020)**.

Table 1: The bias, the standard deviation, and the mean squared error of the estimated values under the optimal decision rule via the proposed deep jump Evaluation and two kernel-based methods for the Warfarin data.

Methods	Bias	Standard deviation	Mean squared error
Deep Jump Evaluation	0.259	0.416	0.240
SLOPE (Su et al. 2020)	0.611	0.755	0.943
Kallus & Zhou (2018)	0.662	0.742	0.989
Colangelo & Lee (2020)	0.442	1.164	1.550

Thank You!

Chakraborty, B., Laber, E. B. & Zhao, Y.-Q. (2014), 'Inference about the expected performance of a data-driven dynamic treatment regime', *Clinical Trials* **11**(4), 408–417.

Chakraborty, B., Murphy, S. & Strecher, V. (2010), 'Inference for non-regular parameters in optimal dynamic treatment regimes', *Stat. Methods Med. Res.* **19**(3), 317–343.

Colangelo, K. & Lee, Y.-Y. (2020), 'Double debiased machine learning nonparametric inference with continuous treatments', *arXiv preprint arXiv:2004.03036* .

den Boer, A. V. & Keskin, N. B. (2020), 'Discontinuous demand functions: estimation and pricing', *Management Science* .

Kallus, N. & Zhou, A. (2018), 'Policy evaluation and optimization with continuous treatments', *arXiv preprint arXiv:1802.06037* .

Luedtke, A. R. & Van Der Laan, M. J. (2016), 'Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy', *Annals of statistics* **44**(2), 713.

Shi, C., Fan, A., Song, R. & Lu, W. (2018), 'High-dimensional a-learning for optimal dynamic treatment regimes', *Annals of statistics* **46**(3), 925.

Song, R., Wang, W., Zeng, D. & Kosorok, M. R. (2015), 'Penalized q-learning for dynamic treatment regimens', *Statistica Sinica* **25**(3), 901.

Su, Y., Srinath, P. & Krishnamurthy, A. (2020), Adaptive estimator selection for off-policy evaluation, *in* 'International Conference on Machine Learning', PMLR, pp. 9196–9205.

Wang, L., Rotnitzky, A., Lin, X., Millikan, R. E. & Thall, P. F. (2012), 'Evaluation of viable dynamic treatment regimes in a sequentially randomized trial of advanced prostate cancer', *Journal of the American Statistical Association* **107**(498), 493–508.

Zhang, B., Tsiatis, A. A., Laber, E. B. & Davidian, M. (2012), 'A robust method for estimating optimal treatment regimes', *Biometrics* **68**, 1010–1018.