Deep Jump Q-Evaluation for Personalized Decision Making with Continuous Treatments



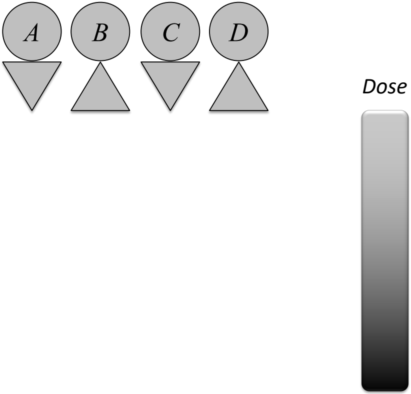

Hengrui Cai[1] Chengchun Shi[2] Wenbin Lu[1] Rui Song[1]

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

ENAR 2021 Spring Meeting

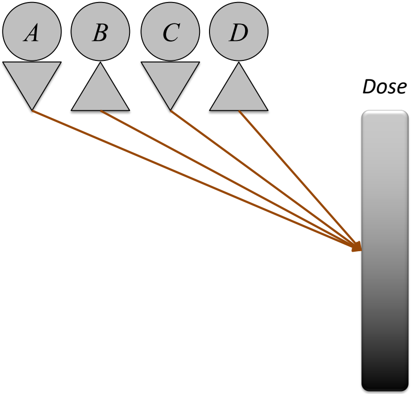
Motivation

Consider a decision making problem to assign individuals with appropriate treatment options in a continuous domain:



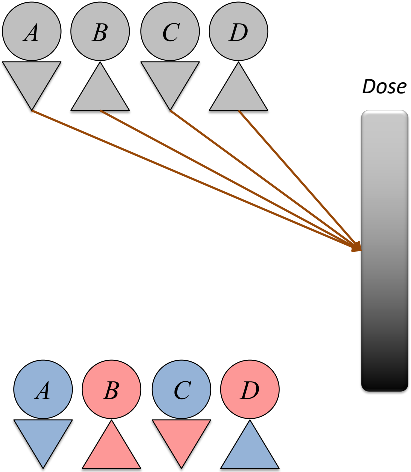
Motivation

An illustration of a simple decision rule that always assigns individuals to a fixed best treatment option:



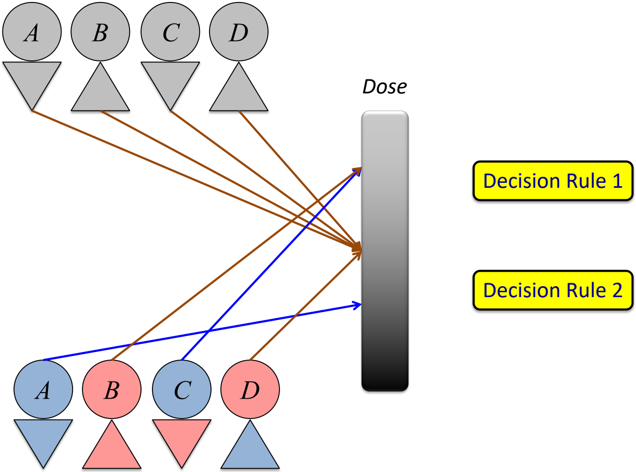
Motivation

Due to individuals' heterogeneity in outcome to different treatment options, there may not exist a unified best treatment.



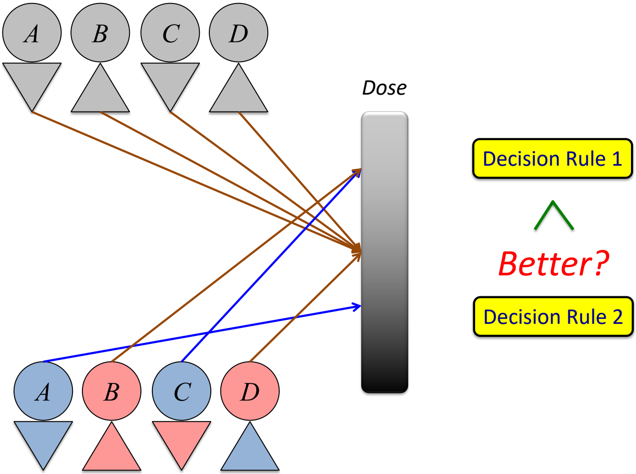
Motivation

An individualized decision rule (IDR) is to assign individuals with a treatment option according to their baseline covariates.



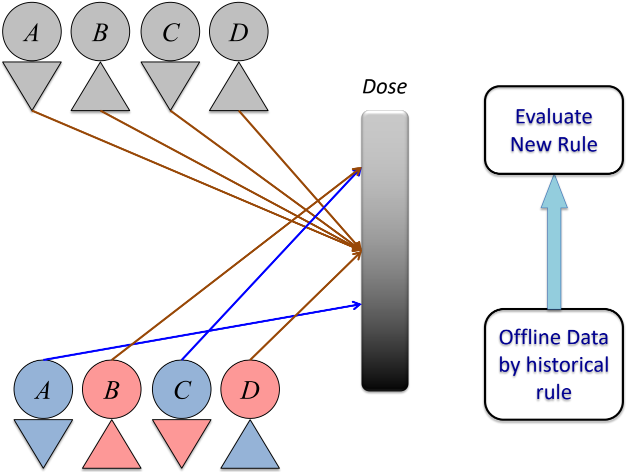
Motivation

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 an IDR online to estimate its mean outcome. Policy evaluation proposes to use the offline data from a different historical rule.



Statistical Framework

- Baseline covariates $X \in \mathcal{X}$ is $p \times 1$ vector;
- Treatment A belongs to a continuous bounded space, say $\mathcal{A} = [0, 1]$;
- Outcome of interest Y, the larger the better by convention;
- Observed Offline Dataset $\{(X_i, A_i, Y_i)\}_{1 \leq i \leq n}$ where n is sample size;
- A decision rule $\pi(\bullet)$ is a deterministic function that maps $\mathcal{X}$ 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.
- Q-function the expected outcome function conditional on the feature-treatment pair: $Q(x, a) = E(Y|X = x, A = a)$.
- Under SUTVA, no unmeasured confounders, and the positivity assumptions, for a decision rule π of interest, its **value** is $V(\pi) = E[Q\{X, \pi(X)\}]$.
- Our goal is to estimate the value $V(\pi)$ based on the observed data.

Statistical Framework

- Baseline covariates $X \in \mathcal{X}$ is $p \times 1$ vector;
- Treatment A belongs to a continuous bounded space, say $\mathcal{A} = [0, 1]$;
- Outcome of interest Y, the larger the better by convention;
- Observed Offline Dataset $\{(X_i, A_i, Y_i)\}_{1 \leq i \leq n}$ where n is sample size;
- A decision rule $\pi(\bullet)$ is a deterministic function that maps $\mathcal{X}$ 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.
- Q-function the expected outcome function conditional on the feature-treatment pair: $Q(x, a) = E(Y|X = x, A = a)$.
- Under SUTVA, no unmeasured confounders, and the positivity assumptions, for a decision rule π of interest, its **value** is $V(\pi) = E[Q\{X, \pi(X)\}]$.
- Our goal is to estimate the value $V(\pi)$ based on the observed data.

Statistical Framework

- Baseline covariates $X \in \mathcal{X}$ is $p \times 1$ vector;
- Treatment A belongs to a continuous bounded space, say $\mathcal{A} = [0, 1]$;
- Outcome of interest Y, the larger the better by convention;
- Observed Offline Dataset $\{(X_i, A_i, Y_i)\}_{1 \le i \le n}$ where n is sample size;
- A decision rule $\pi(\bullet)$ is a deterministic function that maps $\mathcal{X}$ 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.
- Q-function the expected outcome function conditional on the feature-treatment pair: $Q(x, a) = E(Y|X = x, A = a)$.
- Under SUTVA, no unmeasured confounders, and the positivity assumptions, for a decision rule π of interest, its **value** is $V(\pi) = E[Q\{X, \pi(X)\}]$.
- Our goal is to estimate the value $V(\pi)$ based on the observed data.

Statistical Framework

- Baseline covariates $X \in \mathcal{X}$ is $p \times 1$ vector;
- Treatment A belongs to a continuous bounded space, say $\mathcal{A} = [0, 1]$;
- Outcome of interest Y, the larger the better by convention;
- Observed Offline Dataset $\{(X_i, A_i, Y_i)\}_{1 \le i \le n}$ where n is sample size;
- A decision rule $\pi(\bullet)$ is a deterministic function that maps $\mathcal{X}$ 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.
- Q-function the expected outcome function conditional on the feature-treatment pair: $Q(x, a) = E(Y|X = x, A = a)$.
- Under SUTVA, no unmeasured confounders, and the positivity assumptions, for a decision rule π of interest, its **value** is $V(\pi) = E[Q\{X, \pi(X)\}]$.
- Our goal is to estimate the value $V(\pi)$ based on the observed data.

Policy Evaluation

- Most of current works on personalized decision making focus on policy optimization not policy evaluation;
- Less attention has been paid to the continuous treatment setting.
- Available methods rely on the use of a kernel function, and suffer from three limitations.
-
  - Kallus & Zhou (2018), Colangelo & Lee (2020):

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

  - Require the mean outcome to be smooth over the treatment space;
  - Sensitive to the choice of the bandwidth parameter;
  - Use a single bandwidth parameter, which may be sub-optimal.

Policy Evaluation

- Most of current works on personalized decision making focus on policy optimization not policy evaluation;
- Less attention has been paid to the continuous treatment setting.
- Available methods rely on the use of a kernel function, and suffer from three limitations.
-
  - Kallus & Zhou (2018), Colangelo & Lee (2020):

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

  - Require the mean outcome to be smooth over the treatment space;
  - Sensitive to the choice of the bandwidth parameter;
  - Use a single bandwidth parameter, which may be sub-optimal.

Policy Evaluation

- Most of current works on personalized decision making focus on policy optimization not policy evaluation;
- Less attention has been paid to the continuous treatment setting.
- Available methods rely on the use of a kernel function, and suffer from three limitations.
-
  - Kallus & Zhou (2018), Colangelo & Lee (2020):

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

  - Require the mean outcome to be smooth over the treatment space;
  - Sensitive to the choice of the bandwidth parameter;
  - Use a single bandwidth parameter, which may be sub-optimal.

Policy Evaluation

- Most of current works on personalized decision making focus on policy optimization not policy evaluation;
- Less attention has been paid to the continuous treatment setting.
- Available methods rely on the use of a kernel function, and suffer from three limitations.
- - Kallus & Zhou (2018), Colangelo & Lee (2020):

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

  - Require the mean outcome to be smooth over the treatment space;
  - Sensitive to the choice of the bandwidth parameter;
  - Use a single bandwidth parameter, which may be sub-optimal.

Toy Example: Adaptive Discretization

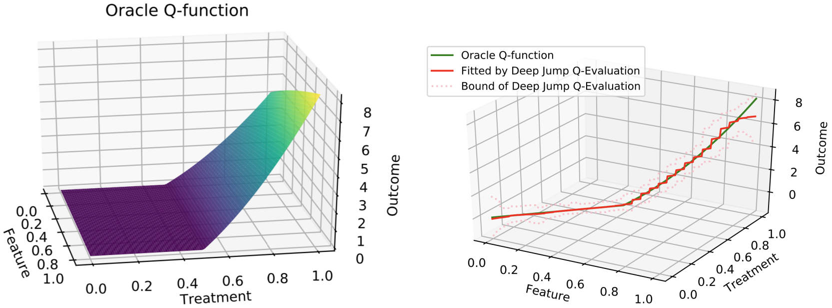


Figure 1: Left panel: the oracle Q-function on the feature-treatment space for the toy example. Right panel: the green curve presents the oracle Q-function $Q\{x, \pi(x)\}$ under decision rule $\pi(x) = x$ in the toy example; and the red curve is the fitted mean value by the deep jump Q-evaluation and the pink dash line corresponds to the 95% confidence bound.

Deep Jump Q-Evaluation

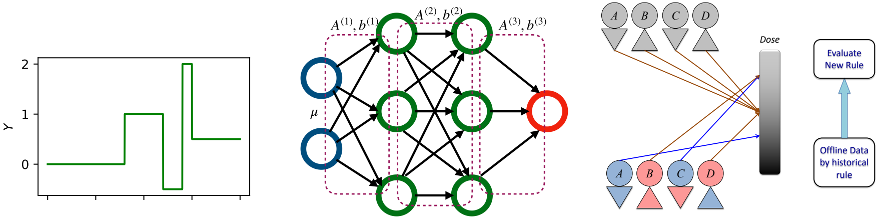


Figure 2: Left: example of piece-wise constant function. Middle: illustration of a deep neural network. Right: demonstration of policy evaluation.

Model 1: 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 2: Continuous function: Q is a continuous function of a and x.

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**:

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

for some regularization parameter γ_n.

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**:

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

for some regularization parameter γ_n.

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**:

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

for some regularization parameter γ_n.

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**:

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

for some regularization parameter γ_n.

Deep Jump Q-Evaluation

Doubly Robust Estimator under Deep Jump Q-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.

Deep Jump Q-Evaluation

Doubly Robust Estimator under Deep Jump Q-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.

Deep Jump Q-Evaluation

Doubly Robust Estimator under Deep Jump Q-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)**, **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 Q-evaluation and two kernel-based methods for the Warfarin data.

Methods	Bias	Standard deviation	Mean squared error
Deep Jump Q-Evaluation	0.259	0.416	0.240
Kallus & Zhou (2018)	0.662	0.742	0.989
Colangelo & Lee (2020)	0.442	1.164	1.550

Contribution

Our deep jump Q-evaluation method for continuous treatments:

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

Contribution

Our deep jump Q-evaluation method for continuous treatments:

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

Contribution

Our deep jump Q-evaluation method for continuous treatments:

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

Thank You!