Precision Medicine: Lecture 08
Q-learning

Michael R. Kosorok,
Nikki L. B. Freeman and Owen E. Leete

Department of Biostatistics
Gillings School of Global Public Health
University of North Carolina at Chapel Hill

Fall, 2021

Outline

Reinforcement learning strategies for clinical trials

Q-learning with observational data

Q-learning with censored data

Backwards outcome weighted learning

Why RL for clinical trials needs extra care

Last time we discussed Q-learning and A-learning in a very general setting and ignored a lot of the challenges that are faced when using these methods to learn optimal treatment regimes from clinical data. Challenges one might face include:

- People are complicated and simple linear Q models may be severely misspecified leading to "bad" estimates of optimal treatment regime(s).
- Patients may be lost to follow-up (censoring).
- Patients may die during the study.
- Decision times can be person specific (e.g. based on disease progression rather than on calendar time)
- Clinical questions may encompass multiple questions at once, e.g. What is the best sequence *and* timing of therapies.

Description of NSCLC

- To consider RL for clinical trials, we will use nonsmall cell lung cancer (NSCLC) as an example.
- Patients with NSCLC typically receive a first-line platinum-based treatment. Approximately 50-60% will receive second-line treatment, usually at the time of disease progression.
- Survival time is the outcome of interest.
- What are the optimal first- and second-line therapies is unknown as is the optimal timing of beginning the second-line therapy.

Our goal

- What kind of trial design should be used to select optimal first- and second-line treatments as well as the optimal time to initiate second-line therapy based on prognostic factors yielding the longest average survival time?
- How can we analyze the data generated from such a trial to answer the primary scientific question?
- Zhao et al. (2011) provides a strategy for answering these questions.

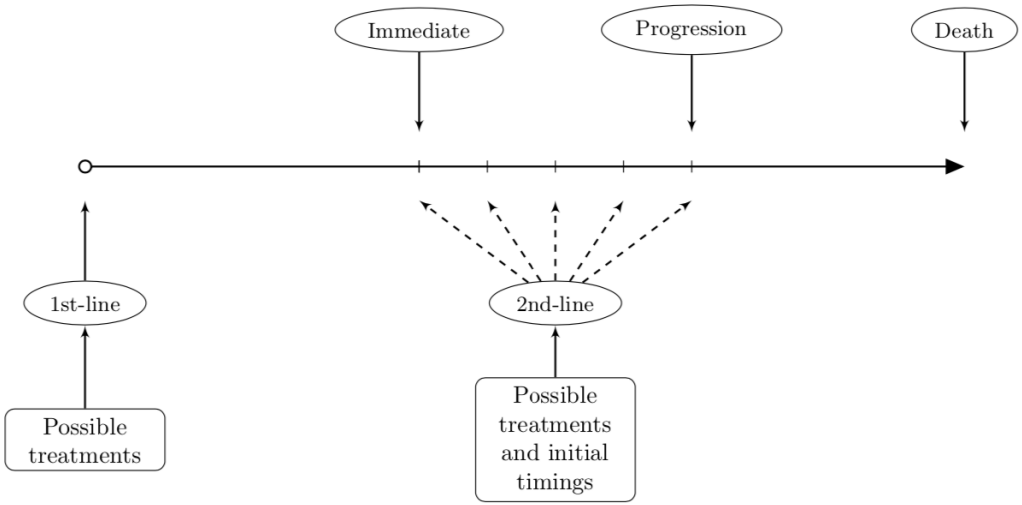


Figure 1: Treatment plan and therapy options for an advanced NSCLC trial (Zhao et al., 2011)

Notation for timing

To describe patient outcomes, we need to describe the important time markers in the study.

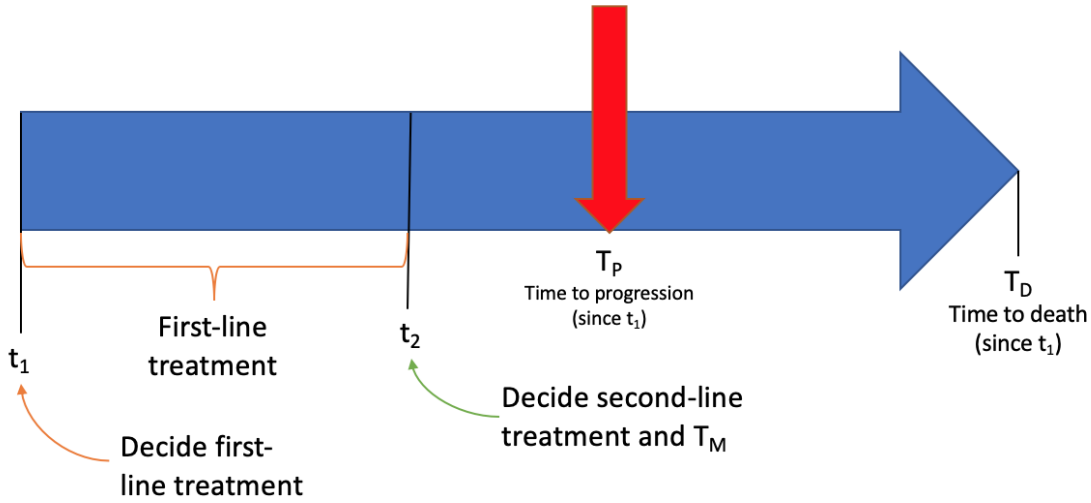


Figure 2: Key time points

Note that

- The actual time to start the second-line therapy is $(t_2 + T_M) \wedge T_P$; assume $T_p \geq t_2$.
- T_D may be truncated at the maximum follow-up time τ.

Censoring is possible, too

- Denote a patient's censoring time by C.
- δ is the censoring indicator

$$\delta = I(T_D \leq C) = \begin{cases} 1 \text{ if death occurs before censoring time} \\ 0 \text{ if censoring occurs before death} \end{cases}$$

- Assume that censoring is independent of both death time and the patient covariates (reasonable when censoring is primarily due to administrative reasons).
- Let

$$T_1 = T_D \wedge t_2, \quad Y_D = I(T_D \wedge C \geq t_2),$$
$$T_2 = (T_D - t_2)I(T_D \geq t_2) = (T_D - t_2)I(T_1 = t_2), \text{ and}$$
$$C_2 = (C - t_2)I(C \geq t_2).$$

- Note that $T_D = T_1 + T_2$, and the total follow-up time $T^0 = T_D \wedge C = T_1 \wedge C + Y_D(T_2 \wedge C_2)$.

Patient covariates

- Patient covariates at the ith decision time are $\mathbf{O}_i = (O_{i1}, \ldots, O_{iq})$ for $i = 1, 2$.
- In first-line therapy, assume that the death time T_1 depends on the covariates $\mathbf{O}_1$ and pssible treatment D_1 according to $[T_1|\mathbf{O}_1, D_1] \sim f_1(\mathbf{O}_1, D_1; \boldsymbol{\alpha}_1)$, where D_1 consists of a finite set of agents d_1.
- If the patient survives long enough to be treated by second-line therapy, we assume $T_P \geq t_2$ and follows another distribution $[T_P|\mathbf{O}_1, D_1] \sim f_2(\mathbf{O}_1, D_1; \boldsymbol{\alpha}_2)$.
- The effects of initial timing of second-line therapy on survival, T_2 given $T_D \geq t_2$, is given by $[T_2|\mathbf{O}_2, D_1, D_2, T_M] \sim f_3(\mathbf{O}_2, D_1, D_2, T_M; \boldsymbol{\alpha}_3)$, where D_2 consists of a finite set of agents d_2 and T_M is a continuous initiation time. Assume that $P(T_D = t_2) = 0$.
- Because of the independence of censoring, conditioning T_2 on $Y_D = 1$ is the same as conditioning on $T_D \geq T_2$.

Review of Q-learning

- T decision points, $t = 1, \ldots, T$.
- States S_t, actions A_t, and rewards R_t, $t = 1, \ldots, T$.
- Q-learning assigns values to action-state pairs, and it is learning, based on S_t, how to best choose A_t to maximize the expected sum $\bar{R}_t = r_1 + \cdots + r_T$.
- The Q-function takes the form

$$Q_t(s_t, a_t) = E[R_t + \max_{a_{t+1}} Q_{t+1}(S_{t+1}, a_{t+1}) | S_t = s_t, A_t = a_t] \quad (1)$$

- The Q-learning algorithm attempts to find a policy π that maps states to actions to maximize the sum of the incremental rewards.
- The optimal policy at time t, $\pi_t^*(s_t)$, satisfies $\pi_t^*(s_t) = \text{argmax}_{a_t} Q_t(s_t, a_t)$.

Translation of NSCLC problem into Q-learning framework

- Consider the overall survival time $T_D = T_1 + T_2$ as the total reward.
- The first action is $d_1 \in D_1$ and provides the patient a progression time T_P and T_1.
- The second action consists of a discrete agent D_2 and a continous time T_M provided that $T_D \geq t_2$.
- The incremental reward function for the first stage is $T_1 \sim G_1(o_1, d_1)$.
- The incremental reward in the second stage, T_2, satisfies $T_2 \sim G_2(o_2, d_1, d_2, T_M)$.
- G_1 and G_2 are obtainable from f_1 and f_2 and usually not observable.
- The reward function is $T_D \wedge \tau$ because we cannot observe events beyond the maximum follow-up time τ.

Observed data

- Given that $T_D \wedge C \geq t_2$, we observe $\{O_1, D_1, T_1 \wedge C, O_2, D_2, T_2 \wedge C\}$.
- Under our usual causal assumptions, we can learn the optimal policy in terms of the observed data using functions Q_t, $t = 1, 2, 3$, where index 3 is a mathematical convenience.
- To estimate each Q_t, we denote $Q_t(O_t, D_t; \boldsymbol{\theta}_t)$ as a function of a set of parameters $\boldsymbol{\theta}_t$ and allow the estimators to have different parameter sets for differnet time points t.
- The Q_t are solved backward through time, then $Q_1(\hat{\boldsymbol{\theta}}_1)$ and $Q_2(\hat{\boldsymbol{\theta}}_2)$ are used to derive $\hat{\pi}_1 = \text{argmax}_{d_1}\, Q_1(o_1, d_1; \hat{\boldsymbol{\theta}}_1)$ and $\hat{\pi}_2 = \text{argmax}_{d_1, T_M}\, Q_2(o_2, d_2, T_M; \hat{\boldsymbol{\theta}}_2)$.

Specifying the Q-functions

- For this setting, Zhao et al. (2011) propose using support vector regression (SVR) for fitting the Q-functions.
- The goal of SVR is to find a function $f : \mathbb{R}^m \to \mathbb{R}$ that closely matches the target y_i for the corresponding $\mathbf{x}_i$ for labeled data $(\mathbf{x}_i, y_i)_{i=1}^n$.
- Whereas SVM uses the hinge loss, a popular loss function invovled in SVR is the ϵ-insensitive loss function:

$$L(f(\mathbf{x}_i), y_i) = (|f(\mathbf{x}_i) - y_i| - \epsilon)_+$$

  where $\epsilon > 0$.
- To accommodate interval-censored data, $(\mathbf{x}_i, \ell_i, u_i)_{i=1}^n$ the loss function can be modified as

$$L(f(\mathbf{x}_i), \ell_i, u_i) = \max(\ell_i - \epsilon - f(\mathbf{x}_i), f(\mathbf{x}_i) - u_i - \epsilon)_+,$$

  and this loss function is called ϵ-SVR-C.
- Observe that by setting $u_i = +\infty$, we can construct a right-censored observation.

Loss functions for ϵ-SVR-C

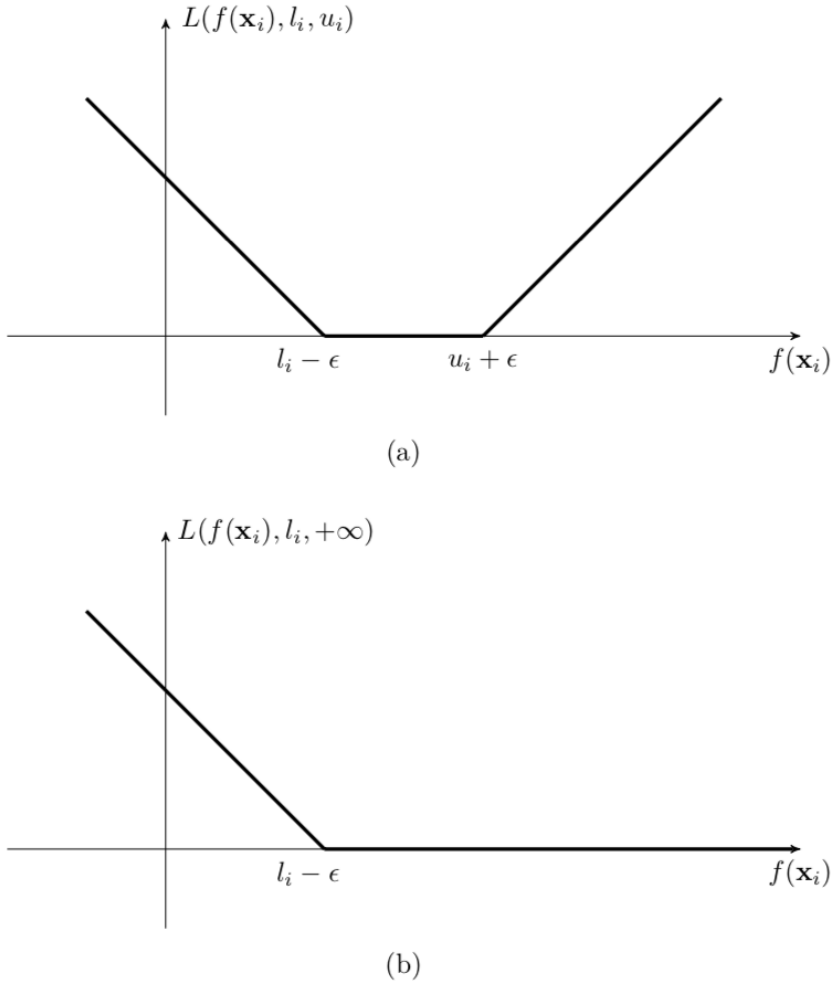


Figure 3: Modified SVR loss functions for interval-censored data (a) and right-censored data (b).

Q-learning framework (cont.)

It can be shown that the ϵ-SVR-C optimization problem is equivalent to solving

$$\min_{\lambda,\lambda'} \frac{1}{2}(\lambda - \lambda')^{\intercal} K(x_i, x_j)(\lambda - \lambda') - \sum_{i \in L}(\ell_i - \epsilon)\lambda'_i + \sum_{i \in U}(u_i + \epsilon)\lambda_i$$

$$\text{s.t. } \sum_{i \in L} \lambda' - \sum_{i \in U} \lambda_i = 0,\ 0 \leq \lambda_i,\ \lambda'_i \leq C_E,\ i = 1, \ldots, n,$$

where $K(\cdot, \cdot)$ is a kernel function, C_E is tuned by cross-validation, $L = \{i : \ell_i > -\infty\}$, $U = \{i : u_i < +\infty\}$, and the approximating function at $\mathbf{x}$ is given by $f(\mathbf{x}) = \sum_{i=1}^{n}(\lambda'_i - \lambda_i)K(\mathbf{x}_i, \mathbf{x}) + b$.

The trial

- Consider therapies A_i, $i = 1, \ldots, 4$, where A_1 and A_2 are first-line treatments and A_3 and A_4 are second-line.
- N patients are recruited into the trial. At enrollment patients are fairly randomized between A_1 and A_2.
- Each patient is followed through to completion of first-line treatment, given that the patient is not dead or lost to follow up.
- For concreteness, fix the duration $t_2 - t_1$ at 2.8 months.
- At the end of the first-line treatment, randomize patients between agents A_3 and A_4 and a randomized target initiation time T_M over the interval $[0, 2]$ (in months).
- Initiate the second-line therapy at $T_M \wedge (T_P - t_2)$.

The analysis

1. Each of n observed individuals will have an initial (time $= t_1$) attributes $\mathbf{x}_1 = (o_1, d_1)$ and index $y_1 = (T_1 \wedge C, \delta_1 = I\{T_1 \leq C\})$; and, if $Y_D = 1$, they will also have time t_2 observations $\mathbf{x}_2 = (o_2, d_2, T_M)$ and $y_2 = (T_2 \wedge C_2, \delta_2 = I\{T_2 \leq C_2\})$.
2. Initialization: Let Q_3 be a function equal to zero.
3. For those with $Y_D = 1$, perform right-censored regression using ϵ-SVR-C of $(T_2 \wedge C_2, \delta_2)$ on (o_2, d_2) to obtain $\hat{Q}_2$.
4. Q_1 is fitted with ϵ-SVR-C through the following equation:

$$Q_1(o_1, d_1; \boldsymbol{\theta}_1) = T_1 + I(T_1 = t_2) \max_{d_2, T_M} Q_2(o_2, d_2, T_M; \hat{\boldsymbol{\theta}}_2) + error$$

$$= T_1 + I(T_1 = t_2)\hat{T}_2 + error$$

   where $T_1 + I(T_1 = 2)\hat{T}_2$ is assessed through the censored observation $(\tilde{X}, \tilde{\delta})$, with $\tilde{X} = \hat{T}_1 \wedge C + Y_D \hat{T}_2$.
5. For the SVR computations, use the Gaussian kernel and a grid to tune the parameters via cross-validation.
6. Given $Q_1(\hat{\boldsymbol{\theta}}_1)$ and $Q_2(\hat{\boldsymbol{\theta}}_2)$, compute the optimal policies $\hat{\pi}_1$ and $\hat{\pi}_2$.

Discussion

- Zhao et al. (2011) propose a clinical study design for NSCLC. This design provides insight on how Q-learning can be applied to a realistic setting.
- While we initially studied Q-learning with linear Q-functions, the NSCLC example demonstrates the complexity of the relationships between the variables and outcomes involved in precision medicine and thus why semiparametric and nonparametric methods may be considered.
- We did not discuss how to calculate sample sizes for a study such as this. How to appropriately size clinical trials designed for precision medicine is an active area of research.

Outline

Motivation

- Due to the complexity and cost of running sequentially randomized trials, it makes sense to first explore potential regimes with observational data.
- Exploring DTRs with observational data first can narrow down the regimes that are explored in a confirmatory study where individuals are randomized to one of a small number of regimes.
- Using Q-learning for observational data requires care, particularly in terms of how causal adjustments are made and in making inferences.
- Moodie, Chakraborty, and Kramer (2012) explore some of these issues.

Notation

- We observed n iid patients with trajectory $(C_1, O_1, A_1, C_2, O_2, A_2, Y)$, where
  - C_j and O_j, $j = 1, 2$, are covariates measured prior to treatment at the beginning of the j interval,
  - Y is observed at the end of the second interval, and
  - A_j, $j = 1, 2$, is the treatment assigned at the jth interval after observing O_j.
- O_j are tailoring (or prescriptive) variables; C_j are confounders.
- Summarize the history at each interval as $H_1 = (C_1, O_1)$ and $H_2 = (C_1, O_1, A_1, C_2, O_2)$.
- $A_j \in \{-1, 1\}$, and receipt of A_j depends on C_j.
- A two-interval *DTR* consists of two decision rules, (d_1, d_2), with $d_j \equiv d_j(H_j) \in \mathcal{A}_j$, where $\mathcal{A}_j$ is the set of possible treatments at the jth interval.

Basic Q-learning with linear models

For two stages, we have the Q-functions:

$$Q_2(H_2, A_2) = E[Y|H_2, A_2]$$
$$Q_1(H_1, A_1) = E[\max_{a_2} Q_2(H_2, a_2)|H_1, A_1].$$

If we knew the true multivariate distribution of the data, the optimal DTR could be found using backwards induction so that

$$d_j(h_j) = \operatorname{argmax}_{a_j} Q_j(h_j, a_j), \quad j = 1, 2.$$

In reality, the Q-functions are not known, so we estimate them from the data. A simple approach is to assume that the Q-functions are linear:

$$Q_j(H_j, A_j; \beta_j, \psi_j) = \beta_j^\mathsf{T} H_{j0} + (\psi_j^\mathsf{T} H_{j1})A_j, \quad j = 1, 2,$$

where H_{j0} are variables that have a predictive effect on the outcome that is not modified by the treatment and H_{j1} are components of the history that do interact with the treatment.

Q-learning with linear models (algorithm)

By now, we are familiar with the Q-learning algorithm. We re-state it here in the notation of Moodie, Chakraborty, and Kramer (2012).

1. Interval 2 parameter estimation: Using regression, find

$$(\hat{\beta}_2, \hat{\psi}_2) = \underset{\beta_2, \psi_2}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - Q_2(H_{2i}, A_{2i}; \beta_2, \psi_2))^2$$

2. Interval 2 optimal rule: By substitution,

$$\hat{d}_2 = \underset{a_2}{\text{argmax}}\, Q_2(h_2, a_2; \hat{\beta}_2, \hat{\psi}_2).$$

3. Interval 1 pseudo-outcome: By substitution,

$$\hat{Y}_{1i} = \max_{a_2} Q_2(H_{2i}, a_2; \hat{\beta}_2, \hat{\psi}_2), \quad i = 1, \ldots, n.$$

Q-learning with linear models (algorithm cont.)

4. Interval 1 parameter estimation: Using regression, find

$$(\hat{\beta}_1, \hat{\psi}_1) = \underset{\beta_1, \psi_1}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_{1i} - Q_1(H_{1i}, A_{1i}; \beta_1, \psi_1))^2.$$

5. Interval 1 optimal rule: By substitution,

$$\hat{d}_1(h_1) = \underset{a_1}{\text{argmax}}\, Q_1(h_1, a_1; \hat{\beta}_1, \hat{\psi}_1).$$

The estimated DTR using Q-learning is given by $(\hat{d}_1, \hat{d}_2)$.

Most estimators of DTR parameters are non-regular

- Most estimators of DTR parameters, including those found by Q-learning and G-estimation, are non-regular. This is due to the non-differentiability of the pseudo-outcome,

$$\hat{Y}_{1i} = \hat{\beta}_2^{\intercal} H_{20,i} + |\hat{\psi}_2^{\intercal} H_{21,i}|$$

  with respect to the estimate of ψ_2.
- The estimator $\hat{\psi}_1$ is a function of $\hat{Y}_{1i}$ and so it is also a non-smooth function of $\hat{\psi}_2$. The asymptotic distribution of $\hat{\psi}_1$ does not converge uniformly over the space of parameters $\psi = (\psi_1, \psi_2)$.
- In fact, $\sqrt{n}(\hat{\psi} - \psi)$ is normal when $P[H_2 : \psi_2^{\intercal} H_{21} = 0] = 0$ and is non-normal when $P[H_2 : \psi_2^{\intercal} H_{21} = 0] > 0$.
- Non-regularity of $\hat{\psi}_1$ leads to estimation bias and poor coverage of Wald type CIs. Even the usual bootstrap CIs can perform badly.

Soft-thresholding can improve performance

- One approach for reducing the problems of bias and/or incorrect coverage is soft-thresholding.
- Soft-thresholding shrinks the problematic term in the pseudo-outcome term towards zero.
- The soft-thresholding approach to Q-learning replaces the pseudo-outcome of the basic (hard-max) Q-learning algorithm with

$$\hat{Y}_{1i}^{\mathsf{ST}} = \hat{\beta}_2^{\mathsf{T}} H_{20,i} + |\hat{\psi}_2^{\mathsf{T}} H_{21,i}| \cdot \left(1 - \frac{\lambda_i}{|\hat{\psi}_2^{\mathsf{T}} H_{21,i}|^2}\right)^+, \quad i = 1, \ldots, n,$$

  where $\lambda_i > 0$ is a tuning parameter associated with ith subject.
- A data-driven choice of λ_i is the approximate empirical Bayes estimator, $\lambda_i = 3H_{21,i}^{\mathsf{T}} \hat{\Sigma}_2 H_{21,i}/n$, $i = 1, \ldots, n$, where $\hat{\Sigma}_2/n$ is the estimated covariance matrix of $\hat{\psi}_2$.

Q-learning for observational data

- To estimate treatment effects from a non-randomized study, propensity scores are often used. Recall that propensity scores are defined as

$$\pi(x) = P(A = 1|X = x).$$

- Approaches that use propensity scores to estimate treatment effects include:
  - regression adjustment on the propensity score,
  - inverse probabilty of treatment weights (IPTW), and
  - regression estimation on a propensity-score matched subsample.
- How different confounder-adjustment methods implemented with Q-learning performs relative to basic Q-learning (no adjustment) is the focus of the remainder of this section.

The adjustment methods

Let $PS_1 = P(A_1 = 1|C_1)$ and $PS_2 = P(A_2 = 1|C_1, C_2)$. The four adjustment methods adapted to Q-learning by Moodie, Chakraborty, and Kramer (2012) are:

1. Linear adjustment: Include covariates as linear terms in the Q-function with $H_1 = (C_1, O_1)$, $H_2(C_1, C_2, O_1, A_1, O_2)$
2. Adjustment on the propensity score: Include the PS as a linear term in the Q-function with $H_1 = (PS_1, O_1)$, $H_2 = (PS_2, O_1, A_1, O_2)$.
3. Include quintiles of PS_j as covariates in the jth interval Q-function.
4. IPTW with H_1 and H_2 as defined in basic Q-learning.

The simulation set-up

- One continuous confounder, C_j, at each interval where

$$C_1 \sim \mathcal{N}(0, 1), \text{ and}$$
$$C_2 \sim \mathcal{N}(\eta_0 + \eta_1 C_1, 1),$$

  for $\eta_0 = -0.5$, $\eta_1 = 0.5$.

- Treatment assignment depends on the value of the confounding variable: for $j = 1, 2$

$$P[A_j = 1 | C_j] = 1 - P[A_j = -1 | C_j] = \text{expit}(\xi_1 O_1 + \xi_2 A_1).$$

  For randomly allocated treatment, $\xi_0 = \xi_1 = 0$; for confounded treatment, $\xi_0 = -0.8$ and $\xi_1 = 1.25$.

Simulation set-up (cont.)

- The binary covariates that interact with the treatment to produce a personalized rule are generated via

$$P[O_1 = 1] = P[O_1 = -1] = \frac{1}{2}$$
$$P[O_2 = 1|O_1, A_1] = 1 - P[O_2 = -1|O_1, A_1]$$
$$= \text{expit}(\delta_1 O_1 + \delta_2 A_1).$$

- Let $\mu = E[Y|C_1, O_1, A_1, C_2, O_2, A_2]$, and $\epsilon \sim \mathcal{N}(0, 1)$ be the error term. Then $Y = \mu + \epsilon$ with

$$\mu = \gamma_0 + \gamma_1 C_1 + \gamma_2 O_1 + \gamma_3 A_1 + \gamma_4 O_1 A_1 +$$
$$\gamma_5 C_2 + \gamma_6 A_2 + \gamma_7 O_2 A_2 + \gamma_8 A_1 A_2.$$

  Confounding can be removed from the data generation seting by setting $\eta_1 = 0$ or $\gamma_1 = \gamma_5 = 0$.

Simulation set-up (cont.)

- Non-regularity in interval 1 parameter estimators is a consequence of non-uniqueness in the optimal treatment at the second interval for some non-zero fraction of the population:

$$p = P[\gamma_6 + \gamma_7 O_2 + \gamma_8 A_1 = 0] > 0,$$

where p describes the degree of non-regularity.

- Non-regularity also occurs when $\gamma_6 + \gamma_7 O_2 + \gamma_8 A_1$ is near 0. Because the standardized effect size is

$$\phi = \frac{E[\gamma_6 + \gamma_7 O_2 + \gamma_8 A_1]}{\text{var}[\gamma_6 + \gamma_7 O_2 + \gamma_8 A_1]},$$

it is also considered.

Simulation set-up (cont.)

Parameters were chosen to consider two regular settings and two non-regular settings.

- First regular setting: Large treatment effect so that $p = 0$ and $\phi = 0.3451$.
- Second regular setting: Small treatment effect so that $\phi = 0.1/0$ and first-interval estimators exhibit bias due to the near-non-regularity.
- First non-regular setting: Designed so that $p = 0.5$ and $\phi = 1$.
- Second non-regular setting: Designed so that $p = 1$ and $\phi = 0/0$.

In the random treatment settings, $\gamma_1 = \gamma_5 = 0$; in the confounded treatment settings $\gamma_1 = \gamma_5 = 1$.

Results: Randomly assigned treatment

Figure 4: Table 1 from Moodie, Chakraborty, and Kramer (2012): Q-learning adjustment methods for $n = 250$; bias, Monte Carlo variance, MSE, and coverage of 95% bootstrap CIs

Relatively small bias in randomized treatment setting

Pretty good coverage

	Hard-max				Soft thresholding			
Adjustment method	Bias	MC var	MSE	Cover	Bias	MC var	MSE	Cover
Regular setting (large effect)								
None	0.0001	0.0076	0.0076	92.7	0.0028	0.0077	0.0077	95.2
Linear	0.0001	0.0076	0.0076	94.5	0.0028	0.0077	0.0077	95.4
PS (linear)	0.0000	0.0076	0.0076	94.2	0.0027	0.0076	0.0076	95.4
PS (quintiles)	0.0004	0.0077	0.0077	94.5	0.0029	0.0077	0.0077	95.4
IPW	−0.0001	0.0076	0.0076	93.9	−0.0011	0.0079	0.0078	95.5
Regular setting (small effect)								
None	0.0071	0.0055	0.0056	96.6	0.0058	0.0045	0.0046	95.2
Linear	0.0070	0.0056	0.0056	96.5	0.0057	0.0046	0.0046	95.3
PS (linear)	0.0069	0.0055	0.0056	96.5	0.0057	0.0046	0.0046	95.7
PS (quintiles)	0.0072	0.0055	0.0055	96.9	0.0057	0.0046	0.0046	95.6
IPW	0.0070	0.0055	0.0056	96.4	0.0058	0.0046	0.0046	95.3
Non-regular (moderate) setting								
None	−0.0420	0.0071	0.0088	89.2	−0.0202	0.0067	0.0071	92.8
Linear	−0.0423	0.0071	0.0086	89.3	−0.0204	0.0067	0.0071	92.7
PS (linear)	−0.0421	0.0071	0.0088	89.2	−0.0203	0.0067	0.0071	92.8
PS (quintiles)	−0.0425	0.0071	0.0089	89.5	−0.0206	0.0068	0.0071	93.1
IPW	−0.0420	0.0071	0.0088	89.5	−0.0202	0.0067	0.0071	93.1
Non-regular (severe) setting								
None	−0.0015	0.0053	0.0053	95.7	−0.0005	0.0045	0.0045	94.5
Linear	−0.0011	0.0054	0.0054	95.7	−0.0003	0.0045	0.0045	94.7
PS (linear)	−0.0011	0.0054	0.0054	95.7	−0.0002	0.0045	0.0045	94.6
PS (quintiles)	−0.0007	0.0054	0.0054	95.8	−0.0003	0.0045	0.0046	94.8
IPW	−0.0013	0.0054	0.0054	95.4	−0.0002	0.0045	0.0045	94.1

Results: Confounded treatment assignment

Figure 5: Table 2 from Moodie, Chakraborty, and Kramer (2012): Q-learning adjustment methods; bias, Monte Carlo variance, MSE, and coverage of 95% bootstrap CIs

In the presence of confounded treatment assignment, only linear adjustment yielded good performance

	$n = 250$				$n = 500$			
Adjustment method	Bias	MC var	MSE	Cover	Bias	MC var	MSE	Cover
Regular setting (large effect)								
None	−0.7201	0.0256	0.5441	0.2	−0.7194	0.0151	0.5325	0.0
Linear	−0.0027	0.0116	0.0116	95.6	0.0010	0.0065	0.0065	95.0
PS (linear)	−0.2534	0.0233	0.0875	64.0	−0.2594	0.0134	0.0809	35.5
PS (quintiles)	−0.3151	0.0213	0.1206	42.6	−0.3226	0.0121	0.1162	14.0
IPW	−0.4304	0.0189	0.2042	7.6	−0.4213	0.0109	0.1884	0.3
Regular setting (small effect)								
None	−0.4365	0.0436	0.2341	38.0	−0.4427	0.0226	0.2186	12.2
Linear	0.0009	0.0095	0.0095	96.1	0.0036	0.0044	0.0044	95.9
PS (linear)	−0.2537	0.0207	0.0851	62.3	−0.2493	0.0105	0.0727	29.4
PS (quintiles)	−0.2892	0.0191	0.1028	48.2	−0.2872	0.0097	0.0923	17.2
IPW	−0.2544	0.0265	0.0912	59.0	−0.2563	0.0174	0.0831	44.9
Non-regular (moderate) setting								
None	−0.9667	0.0337	0.9681	0.0	−0.9712	0.0184	0.9616	0.0
Linear	−0.0233	0.0121	0.0127	93.9	−0.0093	0.0058	0.0058	93.9
PS (linear)	−0.2988	0.0259	0.1152	47.8	−0.2704	0.0125	0.0856	25.2
PS (quintiles)	−0.3653	0.0237	0.1571	28.5	−0.3378	0.0111	0.1252	8.4
IPW	−0.5768	0.0251	0.3578	1.2	−0.5362	0.0128	0.3003	0.0
Non-regular (severe) setting								
None	−0.4490	0.0380	0.2395	33.9	−0.4513	0.0226	0.2264	12.0
Linear	−0.0025	0.0090	0.0090	96.4	−0.0026	0.0045	0.0045	94.6
PS (linear)	−0.2517	0.0191	0.0824	60.7	−0.2555	0.0104	0.0757	28.3
PS (quintiles)	−0.2900	0.0169	0.1011	46.5	−0.2942	0.0097	0.0963	14.8
IPW	−0.2606	0.0301	0.0980	66.5	−0.2640	0.0171	0.0867	42.8

Soft-thresholding results are presented for all settings except the first; the hard-max results are shown for the regular setting with a large effect.

Additional simulation

Three other simulations were considered:

- Mis-specification of the propensity score model,
- Binary confounders, C_1 and C_2, where both confounders predict Y,
- Binary confounders, C_1 and C_2, where C_2 predicts Y, but C_1 does not.

Results: Additional simulations

Figure 6: Table 3 from Moodie, Chakraborty, and Kramer (2012): Q-learning adjustment methods; bias, Monte Carlo variance, MSE, and coverage of 95% bootstrap CIs

	Hard-max				Soft thresholding			
Adjustment method	Bias	MC var	MSE	Cover	Bias	MC var	MSE	Cover
Non-regular (severe) setting, random treatment								
None	0.0015	0.0220	0.0220	96.9	0.0032	0.0185	0.0185	96.1
Linear	−0.0014	0.0094	0.0094	95.8	−0.0020	0.0086	0.0086	94.8
PS (linear)	−0.0018	0.0138	0.0138	97.4	−0.0012	0.0129	0.0129	97.2
IPW	−0.0001	0.0136	0.0136	96.9	0.0012	0.0115	0.0115	96.6
Non-regular (severe) setting, C_j binary								
None	−0.3947	0.0265	0.1823	29.2	−0.3782	0.0278	0.1708	33.3
Linear	0.0010	0.0099	0.0099	96.2	0.0048	0.0092	0.0093	95.1
PS (linear)	−0.2739	0.0197	0.0947	51.0	−0.2692	0.0186	0.0911	50.2
IPW	−0.2145	0.0213	0.0673	63.5	−0.2026	0.0163	0.0573	62.6
Non-regular (severe) setting, C_j binary								
None	−0.0839	0.0157	0.0227	90.0	−0.0774	0.0171	0.0231	90.8
Linear	−0.0029	0.0101	0.0101	95.6	0.0006	0.0092	0.0092	94.8
PS (linear)	−0.0043	0.0129	0.0129	95.8	−0.0006	0.0118	0.0118	94.7
IPW	−0.0505	0.0150	0.0176	92.5	−0.0465	0.0122	0.0144	92.7

In the first two scenarios, C_1 and C_2 predict Y, while in the third, C_2 predicts Y but C_1 does not.

Relatively low bias and good coverage in the randomized setting

Higher bias and poor coverage with confounding; Only linear adjustment yields good performance

Inclusion of the PS as a covariate acts as a rescaled version of C_2, so less bias and better coverage

Discussion

- Standard approaches to causal adjustment can lead to serious bias and poor coverage, so care must be taken in the observational study setting.
- When the dimension of the confounding variables permits, this study suggests that they should be included as covariates in the models for the Q-functions.
- When the relationship between the potential confounders and the outcome are poorly understood, more flexible models may be necessary.
- On a pragmatic note, Moodie, Chakraborty, and Kramer (2012) provide a blueprint for simulating data for precision medicine that can be used for methods development and trial design.

Outline

Motivation

- When estimating DTRs for cancer applications, the number of treatments is often flexible and the data are frequently subject to censoring
- Finding an optimal policy for survival data poses many challenges
  - The method needs to incorporate information accrued over time into the decision rule
  - The method needs to avoid treatments which appear optimal in the short term but may lead to poor final outcome in the long run
  - The data are subject to censoring since some of the patients may be lost to follow-up and the final outcome of those who reached the end of the study alive is unknown
  - The number of decision points (i.e., treatments) and the timing of these decision points can be different for different patients

Data and notation

- T decision points.
- Let S_t be the (random) state of the patient at stage $t \in \{1, \ldots, T+1\}$ and let $\mathbf{S}_t = \{S_1, \ldots, S_t\}$ be the vector of all states up to and including stage t
- Similarly, let A_t be the action chosen in stage t, and let $\mathbf{A}_t = \{A_1, \ldots, A_t\}$ be the vector of all actions up to and including stage t
- Let the random reward be denoted $R_t = r(\mathbf{S}_t, \mathbf{A}_t, S_{t+1})$, where r is an (unknown) time-dependent deterministic function of all states up to stage $t+1$ and all past actions up to stage t
- A trajectory is defined as a realization of $(\mathbf{S}_{T+1}, \mathbf{A}_T, \mathbf{R}_T)$

Value Function

- A policy, π, is a sequence of deterministic decision rules, $\{\pi_1, \ldots, \pi_T\}$, where for every pair (s_t, a_{t-1}), the output of the t^{th} decision rule, $\pi_t(s_t, a_{t-1})$, is an action
- The Bellman equation characterizes the optimal policy π^* as one that satisfies the following recursive relation

$$\pi_t^*(\mathbf{s}_t, \mathbf{a}_{t-1}) = \underset{a_t}{\text{argmax}}\, E[R_t + V_{t+1}^*(\mathbf{S}_{t+1}, \mathbf{A}_t) \,|\, \mathbf{S}_t, \mathbf{A}_t]$$

- Where the value function

$$V_{t+1}(\mathbf{s}_{t+1}, \mathbf{a}_t) = E_{\pi^*}\left[\sum_{i=t+1}^{T} R_i \,\middle|\, \mathbf{S}_{t+1} = \mathbf{s}_{t+1}, \mathbf{A}_t = \mathbf{a}_{t+1}\right]$$

is the expected cumulative sum of rewards from stage $t+1$ to stage T, where the history up to stage $t+1$ is given by $\{\mathbf{s}_{t+1}, \mathbf{a}_t\}$, and when using the optimal policy π^* thereafter.

Q-functions

- Define the optimal time-dependent Q-function

$$Q_t^*(\mathbf{s}_t, \mathbf{a}_t) = E[R_t + V_{t+1}^*(\mathbf{S}_{t+1}, \mathbf{A}_t) \,|\, \mathbf{S}_t = \mathbf{s}_t, \mathbf{A}_t = \mathbf{a}_t]$$

- Note that $V_t^*(\mathbf{s}_t, \mathbf{a}_{t-1}) = \max_{a_t} Q_t^*(\mathbf{s}_t, \mathbf{a}_t)$, and thus

$$Q_t^*(\mathbf{s}_t, \mathbf{a}_t) = E[R_t + \max_{a_{t+1}} Q_{t+1}^*(\mathbf{S}_{t+1}, \mathbf{A}_{t+1}) \,|\, \mathbf{S}_{t+1} = \mathbf{s}_{t+1}, \mathbf{A}_t = \mathbf{a}_t]$$

- To estimate the optimal policy, one first estimates the Q-functions backward through time $t = T, T-1, \ldots, 1$ and obtains a sequence of estimators $\{\widehat{Q}_T, \ldots, \widehat{Q}_1\}$. The estimated policy is given by

$$\widehat{\pi}_t^*(\mathbf{s}_t, \mathbf{a}_{t-1}) = \operatorname*{argmax}_{a_t} \widehat{Q}_t(\mathbf{s}_t, \mathbf{a}_{t-1}, a_t)$$

Two main challenges

- Our goal is to develop a Q-learning algorithm that can handle a flexible number of stages and that takes into account the censored nature of the observations
- Estimation of the Q-functions is done recursively, starting from the last stage backward
  - When the number of stages is flexible, it is not clear how to perform the base step of the recursion
- Due to censoring, some of the trajectories may be incomplete
  - Even when the number of stages is fixed, the known number of stages for a censored trajectory may be less than the number of stages in the multistage problem
  - The reward is not known for the stage at which censoring occurs

Preliminaries

- Let T be the maximal number of decision time-points for a given multistage time-dependent decision problem
  - Note that the number of stages for different observations can be different
- For each $t = 1, \ldots, T$, the state S_t is the pair $S_t = (Z_t, R_{t-1})$,
  - Z_t is either a vector of covariates describing the state of the patient at beginning of stage t or $Z_t = \varnothing$
  - $Z_t = \varnothing$ indicates that a failure event happened during the t^{th} stage which has therefore reached a terminal state
  - R_{t-1} is the length of the interval between decision time-points $t-1$ and t, where we denote $R_0 \equiv 0$
- In our context it is useful to think of the sum $\sum_{j=1}^{t} R_j$ as the total survival time up to and including stage t

Censoring

- Let A_t be an action chosen at decision time t, where A_t takes its values in a finite discrete space $\mathcal{A}$
- Let C be a censoring variable and let $S_C(x) = P(C \geq x)$ be its survival function
- We assume that censoring is independent of both covariates and failure time
- We assume that C takes its values in the segment $[0, \tau]$ where $\tau < \infty$ and that $S_C(\tau) > K_{\text{min}} > 0$
- Let δ_t be an indicator with $\delta_t = 1$ if no censoring event happened before the $(t+1)^{\text{th}}$ decision time-point
  - Note that $\delta_{t-1} = 0 \Rightarrow \delta_t = 0$

Complications from censoring

- The inclusion of failure times in the model affects the trajectory structure
- Usually, a trajectory is defined as a $(2T + 1)$-length sequence $\{S_1, A_1, S_2, \ldots, A_T, S_{T+1}\}$
- However, if a failure event occurs before decision time-point T, the trajectory will not be of full length
- Denote by $\overline{T}$ the (random) number of stages for the individual ($\overline{T} \leq T$)
- Assume that a censoring event occurred during stage t.
  - This means that $\delta_{t-1} = 1$ while $\delta_t = 0$ and $C < \sum_{i=1}^{t} R_i$
  - The observed trajectories have the following structure: $\{S_1, A_1, S_2, \ldots, A_t\}$ and C is also observed

Distribution of observed trajectories

- Assume that n trajectories are sampled at random according to a fixed distribution denoted by P_0
- The distribution P_0 is composed of
  - The unknown distribution of each S_t conditional on $(\mathbf{S}_{t-1}, \mathbf{A}_{t-1})$ (denoted by $\{f_1, \ldots, f_T\}$)
  - An exploration policy $\mathbf{p} = \{p_1, \ldots, p_T\}$ that generates the actions where $p_t(a|\mathbf{S}_t, \mathbf{A}_{t-1})$ is the probability that action a is taken given the history $\{\mathbf{S}_t, \mathbf{A}_{t-1}\}$
- The likelihood (under P_0) of the trajectory $\{s_1, a_1, \ldots, a_t, s_{\bar{t}+1}\}$ is

$$f_1(s_1)p_1(a_1 \mid s_1) \prod_{j=2}^{\bar{t}} \left(f_j(s_j|\mathbf{s}_{j-1}, \mathbf{a}_{j-1})p_j(a_j \mid \mathbf{s}_j, \mathbf{a}_{j-1})\right) f_{\bar{t}}(s_{\bar{t}+1}|\mathbf{s}_{\bar{t}}, \mathbf{a}_{\bar{t}})$$

Likelihood under a given policy

- Define a policy $\pi = \{\pi_1, \ldots, \pi_T\}$ to be a sequence of deterministic decision rules
- Let the distribution $P_{0,\pi}$ denote the distribution of a trajectory for which the policy π is used to generate the actions
- The likelihood (under $P_{0,\pi}$) of the trajectory $\{s_1, a_1, \ldots, a_t, s_{\bar{t}+1}\}$ is

$$f_1(s_1)1_{\pi_1(s_1)=a_1} \prod_{j=2}^{\bar{t}} \left(f_j(s_j|\mathbf{s}_{j-1}, \mathbf{a}_{j-1})1_{\pi_j(\mathbf{s}_j, \mathbf{a}_{j-1})=a_j}\right) f_{\bar{t}}(s_{\bar{t}+1}|\mathbf{s}_{\bar{t}}, \mathbf{a}_{\bar{t}})$$

- This likelihood can be used to measure the value of implementing a policy, but it doesn't yet address different trajectory lengths, or censoring

The auxiliary problem

- To solve the trajectory length and censoring issues, we construct an auxiliary Q-learning model for our original problem using modified trajectories
- The modified trajectories of the construction are of fixed length T, and the modified sum of rewards is less than or equal to τ
- We complete all trajectories to full length in the following way
  - Assume that a failure time occurred at stage $t < T$
  - The trajectory up to S_{t+1} is already defined
  - Write $S'_j = S_j$ for $1 \leq j \leq t+1$ and $A'_j = A_j$ for $1 \leq j \leq t$
  - For all $t+1 < j \leq T+1$ set $S'_j = (\varnothing, 0)$ and for all $t+1 \leq j \leq T$ draw A'_j uniformly from $\mathcal{A}$

The auxiliary problem, cont.

- We also modify trajectories with overall survival time greater than τ in the following way
  - Assume that t is the first index for which $\sum_{i=1}^{t} R_i \geq \tau$
  - For all $j \leq t$, write $S' = S_j$ and $A' = A_j$
  - Write $R' = \tau - \sum_{i=1}^{t-1} R_i$ and assign $Z' \equiv \varnothing$ and thus the modified state $S'_{t+1} = (\varnothing, R'_t)$
  - If $t < T$, then for all $t+1 < j \leq T+1$ set $S'_j = (\varnothing, 0)$ and for all $t+1 \leq j \leq T$ draw A'_j uniformly from $\mathcal{A}$
- The modified trajectory is given by the sequence $\{S'_1, A'_1, S'_2, \ldots, A'_T, S'_{T+1}\}$
- Note that trajectories with fewer than $2T+1$ entries and for which $\sum_{i=1}^{t} R_i \geq \tau$ are modified twice

Distribution of modified trajectories

- The n modified trajectories are distributed according to the fixed distribution P which can be obtained from P_0
- This distribution is composed of the unknown distribution of each S'_t conditional on (S'_{t-1}, A'_{t-1}), denoted by $\{f'_1, \ldots, f'_{T+1}\}$ where

$$f'_t(\mathbf{s}'_t|\mathbf{s}'_{t-1}, \mathbf{a}'_{t-1}) = \begin{cases} f_t((z'_t, r'_t)|\mathbf{s}'_{t-1}, \mathbf{a}'_{t-1}), & \text{if } z'_{t-1} \neq \varnothing, \sum_{i=1}^{t} r'_i < \tau \\ \int_{G_{z'}} f_t((z'_t, r_t)|\mathbf{s}'_{t-1}, \mathbf{a}'_{t-1}) dr_t, & \text{if } z'_{t-1} \neq \varnothing, \sum_{i=1}^{t} r'_i = \tau \\ 1_{\mathbf{s}'_t = \{\varnothing, 0\}} & \text{if } z'_{t-1} = \varnothing \end{cases}$$

and $G_{z'_t} = \{(z_t, r_t) : \sum_{i=1}^{t} r_i \geq \tau\}$

Q-learning with modified trajectories

- The exploration policy $\mathbf{p}'$ agrees with $\mathbf{p}$ on every pair $(\mathbf{S}_t, \mathbf{A}_{t-1})$ for which $Z_t \neq \varnothing$ and draws A_t uniformly from $\mathcal{A}$ whenever $Z_t = \varnothing$
- The modified trajectories are distributed according to the fixed distribution P which can be obtained from P_0
- Using f' and $\mathbf{p}'$ we can now define a likelihood based on the modified trajectories
  - The likelihood is now based on trajectories of the same length
  - The likelihood is not affected by observations after they have been censored
- The standard Q-learning method can now be applied to the data

Q-learning with censored data

- We are looking for a policy $\widehat{\pi}$ that approximates the maximum over all deterministic policies of the following expectation:

$$E_{0,\pi}\left[\left(\sum_{t=1}^{\overline{T}} R_t\right) \wedge \tau\right]$$

- We find this policy in three steps
  - First, we map our problem to the corresponding auxiliary problem
  - Second, we approximate the functions $\{Q_1^*, \ldots, Q_T^*\}$ using backwards recursion and obtain the functions $\{\widehat{Q}_1, \ldots, \widehat{Q}_T\}$
  - Finally, we define $\widehat{\pi}$ by maximizing $\widehat{Q}_t(\mathbf{s}_t, (\mathbf{a}_{t-1}, a_t))$ over all possible actions a_t

Outline

Reinforcement learning strategies for clinical trials

Q-learning with observational data

Q-learning with censored data

Backwards outcome weighted learning

Motivation

- Q-learning targets the optimal Q-function rather than directly targeting the optimal regime
- A postulated class of Q-functions induces a corresponding class of DTRs, namely those representable as the arg max of the postulated Q-functions
  - Linear working models are typically used to approximate the Q-functions
- Q-learning can be inconsistent when the postulated Q-functions are misspecified even in cases where the optimal DTR resides in the class of induced DTRs
- If we can avoid specifying Q-functions we might get better performance

Data and notation

- Consider a multistage decision problem with T stages.
- For $j = 1, \ldots, T$, let $A_j \in \{-1, 1\}$ be the treatment assigned at the j^{th} stage
- X_j is the observation after treatment assignment A_{j-1} but prior to j^{th} stage
- X_{T+1} denotes the covariates measured at the end of stage T.
- We denote the j^{th} reward by R_j and assume it is bounded and has been coded so that larger values correspond to better outcomes.
- Note that R_j can be a part of X_{j+1}. The overall outcome of interest is the total reward $\sum_{j=1}^{T} R_j$

Value function for multiple decisions

- The value associated with a regime $\mathbf{d}$ is

$$V^{\mathbf{d}} = E_{\mathbf{d}}\left[\sum_{j=1}^{T} R_j\right] = \int \sum_{j=1}^{T} R_j dP_{\mathbf{d}}$$

  where $P_{\mathbf{d}}$ is the measure generated under the given regime
- Thus, $V^{\mathbf{d}}$ is the expected long-term benefit if the population were to follow the regime $\mathbf{d}$
- Let P denote the measure generated from the observed data
- $V^{\mathbf{d}}$ can be expressed in terms of P if the data are collected in a SMART

A different approach

- Existing methods model the temporal relationship between the historical information and future rewards (e.g Q-learning)
- Backwards outcome weighted learning (BOWL) examines the data retrospectively
- By investigating the differences between subjects with observed high and low rewards, we can determine what the optimal treatment assignments should be relative to the actual treatments received for different groups of patients
- From this perspective, the optimal DTR estimation can be reformulated as a sequential weighted classification problem

Backwards outcome weighted learning

- OWL cannot be directly generalized to estimating optimal multi-decision DTRs
  - Optimal DTRs need to be determined for multiple stages and the estimation at stage t depends on the decision for the treatment at future stages
  - Thus we estimate the optimal decision rule at future stages first and work backwards to the current stage
- Suppose that we already possess the optimal regimes at stages $t+1, \ldots, T$ and denote them as $d_{t+1}^*, \ldots, d_T^*$
- Let $H_t = (X_1, A_1, \ldots, A_{j-1}, X_j)$ be the history information available when making the j^{th} decision

Backwards outcome weighted learning

- The optimal decision rule at stage t, $d_t^*(h_t)$ should maximize

$$E\left[\frac{\left(\sum_{j=t}^{T} R_j\right)\prod_{j=t+1}^{T} I(A_j = d_j^*(H_j))}{\prod_{j=t}^{T}\pi_j(A_j, H_j)} I(A_t = d_t(H_t)) \,\middle|\, H_t = h_t\right]$$

- As with regular OWL, we can replace $I(A_t = d_t(H_t))$ with $I(A_t \neq d_t(H_t))$ and minimize the resulting expression
- Further replacing $I(A_t \neq d_t(H_t))$ with a convex surrogate allows us to solve the above optimization with SVM methods
- Consider the standard hinge loss

$$\phi(v) = \max(1 - v, 0)$$

BOWL Algorithm

- The BOWL estimation algorithm is as follows

Step 1: Minimize

$$n^{-1}\sum_{i=1}^{n}\frac{R_{iT}\phi(A_{iT}f_T(H_{iT}))}{\pi_T(A_{iT},H_{iT})}+\lambda_{T,n}\|f_T\|^2$$

with respect to f and let $\hat{f}$ denote the minimizer. the estimated optimal decision rule is $\hat{d}(h)=\text{sign}(\hat{f}(h))$

Step 2: For $t=T-1,T-2,\ldots,1$, we backward sequentially minimize

$$n^{-1}\sum_{i=1}^{n}\frac{\left(\sum_{j=t}^{T}R_{ij}\right)\prod_{j=t+1}^{T}I(A_{ij}=\hat{d}_j(H_{ij}))}{\prod_{j=t}^{T}\pi_j(A_{ij},H_{ij})}\times\phi(A_{it}f_t(H_{it}))+\lambda_{t,n}\|f_t\|^2$$

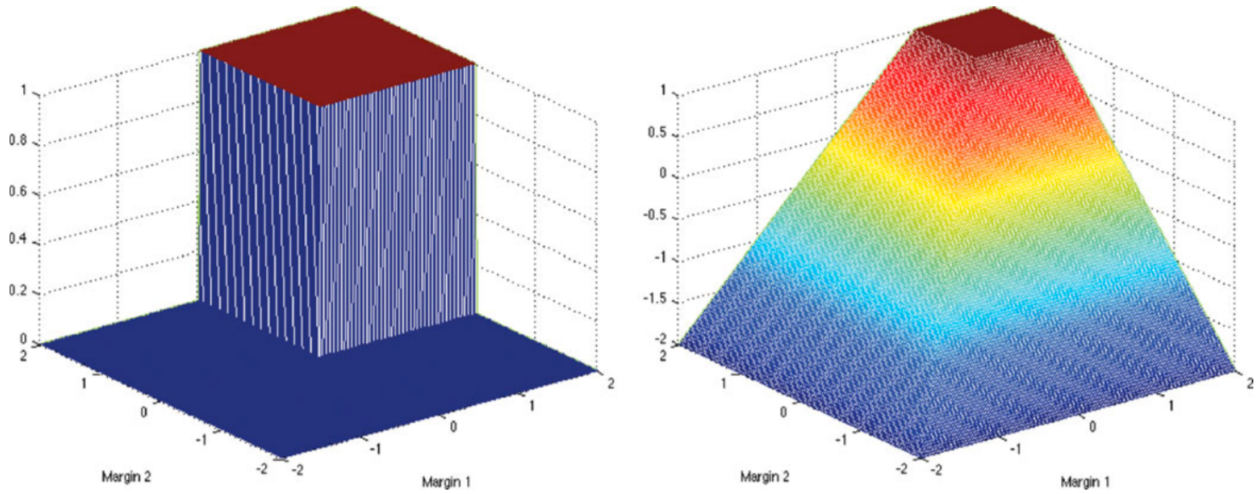
where $\hat{d}_{t+1},\ldots,\hat{d}_T$ are obtained prior to step t

Limitations

- Becaues of the term $\prod_{j=t+1}^{T} I(A_{ij} = \hat{d}_j(H_{ij}))$ the number of subjects actually used in learning optimal decision rules decreases geometrically as t decreases
  - With randomization probability 0.5, the number of nonzero terms is reduced by half at each decision point
- The number of subjects included in the estimation process can be increased using a iterative version of the algorithm (IOWL)
  - Upon obtaining the stage 1 estimated rule $\tilde{d}_1$ using BOWL, reestimate the optimal stage 2 rule $\tilde{d}_2^{new}$ based on the subset of patients whose stage 1 treatment assignments are consistent with $\tilde{d}_1$
  - Continue with the reestimation of the optimal stage 1 rule $\tilde{d}_1^{new}$ using the information of patients consistent with $\tilde{d}_2^{new}$

Sequential outcome weighted learning (SOWL)

- Rather than conducting the estimation in multiple steps, we can directly optimize the value function in a single step
- We describe this method using a two-stage setup but generalization to any number of stages is possible
- Direct maximization of the value function is computationally difficult due to discontinuity of the indicator functions, so we use a concave surrogate for the product of two indicator functions



SOWL Algorithm

- Mimicking the hinge loss in a one-dimensional situation, we consider the following surrogate reward ψ:

$$\psi(Z_1, Z_2) = 1 - \max((1 - Z_1)_+, (1 - Z_2)_+)$$

- The SOWL estimator maximizes

$$n^{-1} \sum_{i=1}^{n} \left[\frac{\left(\sum_{j=1}^{2} R_{ij} \right) \psi(A_{i1} f_1(H_{i1}), A_{i2} f_2(H_{i2}))}{\prod_{j=1}^{2} \pi_j(A_{ij}, H_{ij})} \right] - \lambda_n(\|f_1\| + \|f_2\|)$$

- To extend SOWL to T stages, where $T > 2$, we need to find a concave proxy function for the indicator function. A natural generalization of surrogate reward is $\psi(Z_1, \ldots, Z_T) = 1 - \max((1 - Z_{1-1})_+, \ldots, (1 - Z_{T-1})_+)$

Performance

- For a moderate number of stages ($\sim$2-3), the OWL-based methods show improved empirical performance when compared with regression based methods Q- and A-learning
- IOWL improves on the decision rule from BOWL by taking advantage of an iterative process cycling over the complete dataset
- Simultaneous optimization may cause some numerical instability in SOWL when the number of stages is large
- Q-learning approaches may be more efficient when the number of stages is large, especially if the Q model is correctly specified

Conclusions

- BOWL/SOWL can effectively handle the potentially complex relationships between sequential treatments and prognostic or intermediate variables
- We focused on using SVMs, but any weighted classification algorithm could be used
- The methods could be extended to multiple treatment options using weighted multicategory classification
- We only considered data generated from SMART designs here, but the methodology can be extended for use with observational data