Causal Inference and Predicting Counterfactuals II

David Sontag

Acknowledgement: several slides adapted from Fredrik Johansson and Michael Oberst
Reminder of 9/26 lecture: Causal effects

- Potential outcomes under treatment and control, $Y(1), Y(0)$
- Covariates and treatment, $X, T$

- Conditional average treatment effect (CATE)
  
  \[
  \text{CATE}(X) = \mathbb{E}[Y(1) - Y(0) \mid X]
  \]
Today: Sequential decision making

► A policy \( \pi \) assigns treatments to patients
   (typically depending on their medical history/state)

► Single time-point example:
   "Treat if effect is positive"
   For a patient with medical history \( x \), \( \pi(x) = \mathbb{I}[CATE(x) > 0] \)

► Many clinical decisions are made in sequence
  ► Choices early may rule out actions later
  ► Can we optimize the policy by which actions are made?
Example: Sepsis management

- **Sepsis** is a complication of an infection which can lead to massive organ failure and death.
- One of the leading causes of death in the ICU.
- Primary way to treat is to resolve the *infection*, e.g. with antibiotics.
- Other symptoms need *management*: breathing difficulties, low blood pressure, …
Just one action? Easy!

1. Should the patient be put on mechanical ventilation?

With a single action & outcome, suffices to directly reason about potential outcomes – reduce to what we know from 9/26 lecture.
Example: Sepsis management

2. Should the patient be sedated?
(To alleviate discomfort due to mech. ventilation)
Example: Sepsis management

3. Should we artificially raise blood pressure?

(Which may have dropped due to sedation)
Example: Sepsis management

Septic patient with breathing difficulties

- Mechanical ventilation?
  - No
  - Yes
- Sedation?
  - No
  - Yes
- Vasopressors?
  - No
  - Yes

Observed decisions & response

Exponentially many potential outcomes – no longer can reason about them directly
No prob, we’ll use reinforcement learning

► AlphaStar
► AlphaGo
► DQN Atari
► Open AI Five
Reinforcement learning

► Maximize reward!

Figure by Tim Wheeler, tim.hibal.org
Decision processes

► An agent repeatedly, at times $t$ takes actions $A_t$ to receive rewards $R_t$ from an environment, the state $S_t$ of which is (partially) observed.
Decision process: Mechanical ventilation

\[ \begin{align*}
R_t &= R_{t \text{vitals}} + R_{t \text{vent off}} + R_{t \text{vent on}} \\
S_0 &\rightarrow A_0 \\
S_1, R_1 &\rightarrow A_1 \\
S_2, R_2 &\rightarrow A_2 \\
&\rightarrow R_T
\end{align*} \]
Decision process: Mechanical ventilation

- **State** $S_t$ includes demographics, physiological measurements, ventilator settings, level of consciousness, dosage of sedatives, time to ventilation, number of intubations
Decision process: Mechanical ventilation

- **Actions** $A_t$ include intubation and extubation, as well as administration and dosages of sedatives.
Decision processes

► A decision process specifies how states $S_t$, actions $A_t$, and rewards $R_t$ are distributed: $p(S_0, ..., S_T, A_0, ..., A_T, R_0, ..., R_T)$

► The agent interacts with the environment according to a behavior policy $\mu = p(A_t \mid \cdots)$

* The ... depends on the type of agent
Markov Decision Processes

- Markov decision processes (MDPs) are a special case

- **Markov transitions:** \( p(S_t | S_0, ..., S_{t-1}, A_0, ..., A_{t-1}) = p(S_t | S_{t-1}, A_{t-1}) \)

- **Markov reward function:** \( p(R_t | S_0, ..., S_{t-1}, A_0, ..., A_{t-1}) = p(R_t | S_{t-1}, A_{t-1}) \)

- **Markov action policy** \( \mu = p(A_t | S_0, ..., S_t, A_0, ..., A_{t-1}) = p(A_t | S_t) \)
Markov assumption

State transitions, actions and reward depend only on most recent state-action pair
States are independent: $p(S_t | S_{t-1}, A_{t-1}) = p(S_t)$

Equivalent to **single-step case**: potential outcomes!

* The term “contextual bandits” has connotations of efficient exploration, which is not addressed here
Contextual bandits & potential outcomes

Think of each state $S_i$ as an i.i.d. patient, the actions $A_i$ as the treatment group indicators and $R_i$ as the outcomes.
Goal of RL

Like previously with causal effect estimation, we are interested in the effects of actions $A_t$ on future rewards.
Maximize expected cumulative reward

The goal of most RL algorithms is to maximize the expected cumulative reward—the value $V_π$ of its policy $π$.

**Return:** $G_t = \sum_{s=t}^{T} R_s$

**Value:** $V_π = \mathbb{E}_{A_t \sim π}[G_0]$

The expectation is taken with respect to scenarios acted out according to the learned policy $π$. 

Sum of future rewards

Expected sum of rewards under policy $π$
Example

Let’s say that we have data from a policy $\mu$

- Patient 1: $a_1 = 1$, $R_1^1$, $a_2 = 0$, $R_2^1$, $a_3 = 1$, $R_3^1$
- Patient 2: $a_1 = 0$, $R_1^2$, $a_2 = 1$, $R_2^2$, $a_3 = 1$, $R_3^2$
- Patient 3: $a_1 = 0$, $R_1^3$, $a_2 = 0$, $R_2^3$, $a_3 = 0$, $R_3^3$

Value

$V_\mu \approx \frac{1}{n} \sum_{i=1}^{n} G^n$

Return

$G^1 = R_1^1 + R_2^1 + R_3^1$
$G^2 = R_1^2 + R_2^2 + R_3^2$
$G^3 = R_1^3 + R_2^3 + R_3^3$
1. Decision processes

2. **Reinforcement learning**

3. Learning from batch (off-policy) data

4. Reinforcement learning in healthcare
**Paradigms**

<table>
<thead>
<tr>
<th>Model-based RL</th>
<th>Value-based RL</th>
<th>Policy-based RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transitions $p(S_t \mid S_{t-1}, A_{t-1})$</td>
<td>Value/return $p(G_t \mid S_t, A_t)$</td>
<td>Policy $p(A_t \mid S_t)$</td>
</tr>
<tr>
<td>G-computation</td>
<td>Q-learning G-estimation</td>
<td>REINFORCE</td>
</tr>
<tr>
<td>MDP estimation</td>
<td></td>
<td>Marginal structural models</td>
</tr>
</tbody>
</table>

*We focus on off-policy RL here*
Paradigms*

Model-based RL

- Transitions
  \[ p(S_t | S_{t-1}, A_{t-1}) \]
- G-computation
- MDP estimation

Value-based RL

- Value/return
  \[ p(G_t | S_t, A_t) \]
- Q-learning
- G-estimation

Policy-based RL

- Policy
  \[ p(A_t | S_t) \]
- REINFORCE
- Marginal structural models

*We focus on off-policy RL here
Q-learning

Q-learning is a value-based reinforcement learning method

The value of a state-action pair \((s, a)\) is

\[
Q_\pi(s, a) := \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]
\]

(the expectation is over future states and rewards, for future actions taken according to \(\pi\))

*Mathematical tool more than anything*
Q-learning

- Instead of directly optimizing over $\pi$, Q-learning optimizes over functions $Q(s, a)$. $\pi$ is assumed to be the deterministic policy
  $$\pi(s) = \arg\max_a Q(s, a)$$

- The best $Q$ is the best **state-action value** function
  $$Q^*(s, a) = \max_{\pi} Q_\pi(s, a)$$
Bellman equation

► For the optimal Q-function $Q^*$, “Bellman optimality” holds

$$Q^*(s, a) = \mathbb{E}_\pi \left[ R_t + \gamma \max_{a'} Q^*(S_{t+1}, a') \mid S_t = s, A_t = a \right]$$

State-action value  Immediate reward  Future (discounted) rewards*

► Look for functions with this property!
Q-learning (from last Thursday, 10/10)

Algorithm 3 Q-learning

\[
Q_0(s, a) \leftarrow 0 \text{ for all } s \in S, a \in A \\
\text{for } k = 1 \ldots N \text{ do } \\
\quad \text{Collect sample } (s, a, s', \hat{r}) \text{ by playing with a policy induced from } Q_k \text{ (we will discuss choices for this policy)} \\
\quad \hat{Q}(s, a) \leftarrow \hat{r} + \gamma \max_{a' \in A} Q(s', a') \\
\quad Q_{k+1}(s, a) \leftarrow (1 - \alpha)Q_k(s, a) + \alpha \hat{Q}(s, a) \\
\text{end for} 
\]

► Fitted Q-learning

► If \( s \) is not discrete, we cannot maintain a table for \( Q(s, a) \)

► Instead, we may represent \( Q(s, a) \) by a function \( Q_\theta \)
Q-learning (from last Thursday, 10/10)

Unfortunately, we do not have direct access to the reward function \( r \) or the transition probabilities \( P(s'|s,a) \). However, it turns out that we can use empirical transitions and rewards seen during play to form approximations of the \( Q \) function.

Suppose that we have some approximation to \( Q^\ast \), \( Q \). Now, when we observe a sample \((s, a, s', \hat{r})\) representing a state transition from \( s \) to \( s' \) after action \( a \) was played with instantaneous reward \( r \), we can use it to update our estimate of \( Q^\ast \). Let the empirical \( \hat{Q} \) for this step, which uses the current estimate \( Q_k \), be:

\[
\hat{Q}(s, a) = \hat{r} + \gamma \max_{a' \in A} Q(s', a')
\]

Now, we use the following update rule to change our current estimate of \( Q \):

\[
Q_{k+1}(s, a) \leftarrow (1 - \alpha)Q_k(s, a) + \alpha \hat{Q}(s, a)
\]

We can now use this update step as inspiration for the Q-learning algorithm below in Algorithm 3.

This algorithm is different from the methods described above: it actually plays the game repeatedly to learn a good agent. This is typical in RL algorithms – many settings are too complicated to learn a closed form solution so instead we rely on sampling the environment repeatedly. Note that this algorithm requires a parameter \( \alpha \) that could impact convergence rates.

**Algorithm 3 Q-learning**

\[
Q_0(s, a) \leftarrow 0 \text{ for all } s \in S, a \in A
\]

for \( k = 1 \ldots N \) do

Collect sample \((s, a, s', \hat{r})\) by playing with a policy induced from \( Q_k \) (we will discuss choices for this policy)

\[
\hat{Q}(s, a) \leftarrow \hat{r} + \gamma \max_{a' \in A} Q(s', a')
\]

\[
Q_{k+1}(s, a) \leftarrow (1 - \alpha)Q_k(s, a) + \alpha \hat{Q}(s, a)
\]

end for

Again, once we find a \( Q \) function we can use Equation (4) to build a policy.

A good choice in practice could be something like (where \( c \) is a constant):

\[
\alpha_k(s, a) = \frac{c}{c + k}
\]

If only single time/action, fitted Q-learning is identical to covariate adjustment

**Fitted Q-learning**

- If \( s \) is not discrete, we cannot maintain a table for \( Q(s, a) \)

- Instead, we may represent \( Q(s, a) \) by a function \( Q_\theta \)
1. Decision processes

2. Reinforcement learning paradigms

3. Learning from batch (off-policy) data

4. Reinforcement learning in healthcare
Off-policy learning

► Trajectories \((s_1, a_1, r_1), \ldots, (s_T, a_T, r_T)\), of states \(s_t\), actions \(a_t\), and rewards \(r_t\) observed in e.g. medical record

► Actions are drawn according to a behavior policy \(\mu\), but we want to know the value of a new policy \(\pi\)

► Learning policies from this data is at least as hard as estimating treatment effects from observational data
Assumptions for (off-policy) RL

**Sufficient conditions for identifying value function**

<table>
<thead>
<tr>
<th>Single-step case</th>
<th>Sequential case</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strong ignorability:</strong></td>
<td><strong>Sequential randomization:</strong></td>
</tr>
<tr>
<td>$Y(0), Y(1) \perp T \mid X$</td>
<td>$G(...) \perp A_t \mid \bar{S}<em>t, \bar{A}</em>{t-1}$</td>
</tr>
<tr>
<td>“No hidden confounders”</td>
<td>“Reward indep. of policy given history”</td>
</tr>
<tr>
<td><strong>Overlap:</strong></td>
<td><strong>Positivity:</strong></td>
</tr>
<tr>
<td>$\forall x, t: p(T = t \mid X = x) &gt; 0$</td>
<td>$\forall a, t: p(A_t = a \mid \bar{S}<em>t, \bar{A}</em>{t-1}) &gt; 0$</td>
</tr>
<tr>
<td>“All actions possible”</td>
<td>“All actions possible at all times”</td>
</tr>
</tbody>
</table>
The problem of overlap shows up all over deep RL

“Our results demonstrate that the performance of a state of the art deep actor-critic algorithm, DDPG (Lillicrap et al., 2015), deteriorates rapidly when the data is uncorrelated... These results suggest that off-policy deep reinforcement learning algorithms are ineffective when learning truly off-policy.”

Fujimoto, Meger, Precup, ICML 2019
Assumptions for (off-policy) RL

► Sufficient conditions for identifying value function

**Single-step case**

*Strong ignorability:*
\[ Y(0), Y(1) \perp T \mid X \]
“No hidden confounders”

*Overlap:*
\[ \forall x, t: p(T = t \mid X = x) > 0 \]
“All actions possible”

**Sequential case**

*Sequential randomization:*
\[ G(\ldots) \perp A_t \mid S_t, \overline{A}_{t-1} \]
“Reward indep. of policy given history”

*Positivity:*
\[ \forall a, t: p(A_t = a \mid S_t, \overline{A}_{t-1}) > 0 \]
“All actions possible at all times”
Recap: Learning potential outcomes

Anna
Age = 54
Gender = Female
Race = Asian
Blood pressure = 150/95
WBC count = 6.8*10^9/L
Temperature = 36.7°C
Blood sugar = High

May 15
Medication A
“Control”
$t = 0$

Sep 15
Blood sugar = ?
$Y(0)$

Medication B
“Treated”
$t = 1$

Blood sugar = ?
$Y(1)$
Treating Anna once

We assumed a simple causal graph. This let us identify the causal effect of treatment on outcome from observational data.
Treating Anna over time

► Let’s add a time point…

$$R_t(a) \perp A_t | S_t$$
Treating Anna over time

What influences her state?

Anna’s health status depends on how we treated her

It is likely that if Anna is diabetic, she will remain so

Ignorability

$R_t(a) \perp A_t \mid S_t$
Treating Anna over time

What influences her state?

The outcome at a later time may depend on earlier choices.

The outcome at a later time may depend on an earlier state.

Ignorability

\[ R_t(a) \perp A_t \mid S_t \]
Treating Anna over time

What influences her state?

- If we already tried a treatment, we might not try it again.
- If the last treatment was unsuccessful, it may change our next choice.
- If we know that a patient had a symptom previously, it may affect future decisions.

\[ R_t(a) \perp A_t \mid S_t \]

**Ignorability**
State & ignorability

To have sequential ignorability, we need to remember history!

History $H_2$

\[ A_1 \rightarrow S_1 \rightarrow R_1 \rightarrow A_2 \]

\[ S_2 \rightarrow R_2 \]

Ignorability

$R_t(a) \perp A_t \mid H_t$
Summarizing history

- The difficulty with history is that its size grows with time

- A simple change of the standard MDP is to store the states and actions of a length $k$ window looking backwards

- Another alternative is to learn a summary function that maintains what is relevant for making optimal decisions, e.g., using an RNN
State & ignorability

► We cannot leave out unobserved confounders
What made success possible/easier?

► **Full observability**
  *Everything important to optimal action is observed*

► **Markov** dynamics
  *History is unimportant given recent state(s)*

► Limitless **exploration** & self-play through simulation
  *We can test “any” policy and observe the outcome*

► **Noise-less** state/outcome (for games, specifically)
How do we build trust in RL policies?

► **Goal:** Apply reinforcement learning in high risk settings (e.g., healthcare)

► **Problem:** How to safely evaluate a policy? No simulator, and off-policy evaluation can fail due to
  ► Unobserved confounding
  ► Small sample sizes & lack of overlap
  ► Poorly specified rewards
Building trust in RL policies

► **Goal**: Apply reinforcement learning in high risk settings (e.g., healthcare)

► **Problem**: How to safely evaluate a policy? No simulator, and off-policy evaluation can fail due to
  - Unobserved confounding
  - Small sample sizes & lack of overlap
  - Poorly specified rewards

► Could try to interpret the policy directly, but if not possible, what can we do?

► **Approach**: look at the proposed policy in the context of a specific individual

![Graph showing average reward comparison between different methods]

**Legend**:
- **Obs**: Observed Reward of behavior policy
- **WIS**: Weighted Importance Sampling
- **MB**: Model-Based Rollouts
- **CF**: Counterfactual Rollouts
- **True**: Actual RL reward, not known
Building trust in RL policies

Suppose we are given:
- Markov Decision Process (MDP)
- Policy (e.g., learned using MDP)

Markov Decision Process (MDP)

\[ P(S', R | S, A) \]

- \( S \): Current State
- \( A \): Action
- \( R \): Reward
- \( S' \): Next State

Policy

\[ \pi(A | S) \]
Using counterfactuals to “sanity check”

\[ S: \text{State} \]
\[ A: \text{Action} \]

- Patient has infection
- Drug reaction
- Significant agitation

\[ A_1 \quad \text{Antibiotics} \]
\[ A_2 \quad \text{Mechanical Ventilation} \]
\[ A_3 \quad \text{Sedation} \]

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

If the new policy had been applied to this patient...

- Antimicrobials
- Mechanical ventilation
- Sedation

\[ S: \text{State} \]
\[ A: \text{Action} \]

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

If the new policy **had been** applied to this patient...

- **$A_1$**: Antibiotics
- **$S_0$**: Patient has infection

...Patient has infection

- **$A_1$**: Antibiotics
- **$A_2$**: Mechanical Ventilation
- **$A_3$**: Sedation

...Drug reaction

...Significant agitation

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

*If the new policy had been applied to this patient...*

- **$A_1$**: Antibiotics
  - Patient has infection
  - Infection cleared
- **$S_0$**: Patient has infection
- **$S_1$**: Infection cleared

---

- **$A_1$**: Antibiotics
  - Patient has infection
- **$A_2$**: Mechanical Ventilation
  - Drug reaction
- **$A_3$**: Sedation
  - Significant agitation

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

If the new policy had been applied to this patient...

- $A_1$: Antibiotics
- $S_0$: Patient has infection
- $A_2$: Mechanical Ventilation
- $A_3$: Sedation

Time

- $S_1$: Infection cleared
- $S$: State
- $A$: Action

Model-based rollout not a fair comparison

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

If the new policy had been applied to this patient...

\[ S_0: \text{patient has infection} \rightarrow S_1 \]

\[ A_1: \text{Antibiotics} \rightarrow \text{patient has infection} \]

\[ A_2: \text{Mechanical Ventilation} \rightarrow \text{drug reaction} \]

\[ A_3: \text{Sedation} \rightarrow \text{significant agitation} \]

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

If the new policy had been applied to this patient...

- **$A_1$**: Antibiotics
  - patient has infection
- **$S_0$**: patient has infection
- **$A_2$**: Mechanical Ventilation
  - drug reaction
- **$S_1$**: drug reaction
- **$A_3$**: Sedation
  - significant agitation

Counterfactual influenced by actual outcome

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

If the new policy had been applied to this patient...

- **$A_1$**: Antibiotics
  - Patient has infection

- **$A_2$**: No action
  - Drug reaction

- **$A_3$**: Discharge
  - Patient recovers

- **$S_0$**: State
  - **$A$**: Action

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

If the new policy **had been** applied to this patient...

- **A₁**: Antibiotics
  - **S₀**: Patient has infection
  - **S₁**: Drug reaction
  - **S₂**: Patient recovers

- **A₂**: No action
  - **S₀**: Patient has infection
  - **S₁**: Drug reaction
  - **S₂**: Patient recovers

- **A₃**: Discharge
  - **S₀**: Patient has infection
  - **S₁**: Drug reaction
  - **S₂**: Patient recovers

**Idea:** If the counterfactual trajectory is unreasonable given full context of patient, the model / policy may be flawed

[Balke & Pearl, 1994]
Using counterfactuals to “sanity check”

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decomposition of average <strong>reward</strong> over real episodes, to identify interesting cases</td>
<td><img src="chart.png" alt="" /></td>
</tr>
</tbody>
</table>

- **Example**: Comparison of outcomes (90-day survival) between the observed and counterfactual trajectories, on the test set. Most notably, under the counterfactual it is estimated that only one patient would have died, and most of the patients who died would have lived. However, 14% of patients have no outcomes in the counterfactuals, due to a nuance discussed in Section 7.5.
Using counterfactuals to “sanity check”

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Decomposition of average reward over real episodes, to identify interesting cases.</td>
<td><img src="#" alt="Table" /></td>
</tr>
<tr>
<td>2. Examine counterfactual trajectories under new policy.</td>
<td><img src="#" alt="Table" /></td>
</tr>
<tr>
<td>3. Validate and/or criticize conclusions, using full patient information (e.g., chart review).</td>
<td><img src="#" alt="Table" /></td>
</tr>
</tbody>
</table>

![Comparison of outcomes (90-day survival) between the observed and counterfactual trajectories, on the test set. Most notably, under the counterfactual it is estimated that only one patient would have died, and most of the patients who died would have lived. However, 14% of patients have no outcomes in the counterfactuals, due to a nuance discussed in Section 7.5.](image)
Simulating counterfactual trajectories

**What we need**

1. Observed trajectories

2. Policy to evaluate
   \[ \pi(A \mid S) \]

3. Model of discrete dynamics, e.g., Markov Decision Process

\[ S: \text{Current State} \]
\[ A: \text{Action} \]
\[ S': \text{Next State} \]
Simulating counterfactual trajectories

What we need

1. Observed trajectories
2. Policy to evaluate $\pi(A \mid S)$
3. Model of discrete dynamics, e.g., Markov Decision Process

$$S' = f(S, A, U_{st})$$

$U_{st} \sim P(U_{s'})$

$S$: Current State  
$A$: Action  
$S'$: Next State

Structural Causal Model (SCM)
Simulating counterfactual trajectories

What we need

1. Observed trajectories
2. Policy to evaluate $\pi(A \mid S)$
3. Model of discrete dynamics, e.g., Markov Decision Process

$S$: Current State
$A$: Action
$S'$: Next State

Form of SCM is an assumption: SCM is not identifiable from data!
Structural Causal Models (SCMs)

Causal Graph (Example)

\[ S \rightarrow S' \]

\[ A \rightarrow S' \]

\[ S, S', A \text{ are R.V.s} \]

Structural Causal Model (SCM)

\[ U_s \rightarrow S \]

\[ U_a \rightarrow A \]

\[ S' = \begin{cases} 1, & U_s \leq p \\ 0, & U_s > p \end{cases} \text{ where } p := \Pr[S' = 1 \mid S, A] \]

\[ U_s \sim \text{Unif}(0, 1) \]

\[ U_{st} \]

Example: \[ U_{st} \sim \text{Unif}(0, 1) \]

U’s are R.V.s \ / \ S, S’, A are functions

Use post-treatment information to reveal exogenous factors
Counterfactuals with SCMs

1. **Infer** the posterior of $U_y$ given $X, Y_a = 1$
2. **Intervene** to set $T = b$
3. **Predict** counterfactual outcome $Y_b$

**Structural Causal Model**

$U_y \sim \text{Unif}(0, 1)$,

$Y_t = \begin{cases} 
1, & U_y \leq p_t \\
0, & U_y > p_t 
\end{cases}$

where $p_t := \Pr[Y = 1 | T = t, X]$

This SCM has the **monotonicity** (Pearl 2000) property, which implies that if $p_b \geq p_a$, then $Y_a = 1 \rightarrow Y_b = 1$
SCMs for Markov Decision Processes

Causal Graph (one step)

Structural Causal Model

Choosing a structural mechanism

What is an appropriate SCM for categorical transitions?

Criteria 1: Want to choose $f_s(S_t, A_t, U)$ and $P(U)$ such that:

$$E_u [ f_s (S_t = s, A_t = a, U) = i ] = p_{i|s,a}$$

Criteria 2: Given unidentifiability of counterfactuals, want to make a “reasonable assumption” analogous to monotonicity (Pearl, 2000)
Counterfactual Stability & Gumbel-Max SCM

Counterfactual Stability
New counterfactual stability condition:

If we observe \( S' = i \) under \( A = a \), then under counterfactual \( A = \tilde{a} \),
\[
\frac{p_j}{p_i} > \frac{\tilde{p}_j}{\tilde{p}_i} \Rightarrow S' \neq j.
\]

Theorem 1 (Oberst, Sontag 2019):
Counterfactual stability implies monotonicity (Pearl, 2000) when \( k = 2 \)

Gumbel-Max SCM
Use the Gumbel-Max trick to sample from a categorical distribution with \( k \) categories:

\[
g_j \sim \text{Gumbel} \\
S' = \arg\max_j \{ \log P(S' = j \mid S, A) + g_j \}
\]

Theorem 2 (Oberst, Sontag 2019):
The Gumbel-Max SCM satisfies the Counterfactual Stability condition
Summary

• Causal inference is a special case of off-policy reinforcement learning
• As a result, off-policy reinforcement learning is subject to the same assumptions:
  • Overlap
  • No unobserved confounding
• We suggested one approach of using introspection to help detect errors
• Much more work needed to get safe & robust algorithms