Learning "What-if" Explanations For Sequential Decision Making

IE 708 Project

Hanish Prashant Dhanwalkar

November 19, 2024

Batch IRL

What is IRL:

- Learning technique that aims to infer the reward function of an agent by observing its behavior in a given environment.
- Unlike traditional reinforcement learning, where the reward function is explicitly defined, Batch IRL learns the reward function from a set of expert demonstrations or trajectories.

Why Batch IRL

- **Online Learning:** Classic IRL algorithms require interactive access to the environment, or full knowledge of the environment's dynamics.
- Limited by the assumption that state dynamics are fully-observable and Markovian. NOT true for domains like medicine, where treatment depends on how patient covariates (tumour, side effects) have evolved over time.

"What-if" Explanations

Incorporating counterfactual reasoning into batch IRL

- Learn a parameterized reward function $R(h_t, a_t)$ that is defined as a weighted sum over potential outcomes for taking action at given history h_t.
- This helps in reasoning out why an expert (eq. doctor) have chosen a particular action and “what” would have happen if any other alternative was taken. These experimentation are not possible in medicine domain.

Example: Cancer Treatment

- Consider the decision making process of assigning a binary action given,
  - Tumour volume (U)
  - Side effects (Z)
- Let $\mathbb{E}[U_{t+1}[a_t]|h_t]$ and $\mathbb{E}[Z_{t+1}[a_t]|h_t]$ be the counterfactual outcomes for the two covariates when action a_t is taken given the history h_t of covariates and previous actions.
- The reward as the weighted sum of these counterfactuals:
  - $R(h_t, a_t) = w_u \, \mathbb{E}[U_{t+1}[a_t]|h_t] + w_z \, \mathbb{E}[Z_{t+1}[a_t]|h_t]$,
- This allows us to model the preferences of experts: e.g. finding that $|w_u| > |w_z|$ indicates that the expert is treating more aggressively, by placing more weight on reducing tumour volume than on minimizing side effects.

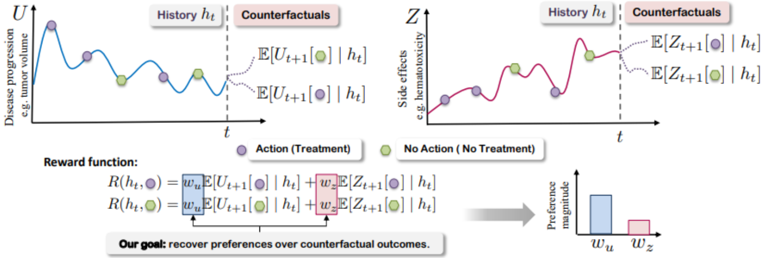

Figure: Explaining decision-making behaviour in terms of preferences over "what if" outcomes. Evolution of tumour volume (U) and side effects (Z) under a binary action.

Problem Formulation

At timestep t, let $X_t \in X$ be the observed patient features and $A_t \in A$ be the action (e.g. treatment) taken. Let x_t and a_t be realizations of these random variables, $h_t = (x_0, a_0, ..., x_{t1}, at1, x_t) = (x_{0:t}, a_{0:t1}) \in H$ be a realization of the history $H_t \in H$ of patient observations and actions.

- A policy $\pi : H \times A \to [0, 1]$, where $\pi(a|h)$ indicates the probability of choosing action $a \in A$ given history $h \in H$ and $\sum_{a \in A} \pi(a|h) = 1$.
- Taking action a_t under history h_t results in observing x_{t+1} and obtaining h_{t+1}. The reward function is $R : H \times A \to \mathbb{R}$ where $R(h, a)$ represents the reward for taking action a A given history $h \in H$.
- The value function of a policy π, $V : H \to \mathbb{R}$ is defined as: $V^\pi(h) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(H_t, A_t)|\pi, H_0 = h]$, where $\gamma \in [0, 1)$ is the discount factor.
- The action-value function $Q : H \times A \to \mathbb{R}$ of a policy is defined as $Q^\pi(h, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(H_t, A_t)|\pi, H_0 = h, A_0 = a]$

Problem Formulation

- **Batch IRL**: consider a linear reward function $R(h_t, a_t) = w \cdot \phi(h_t, a_t)$ where $||w||_1 \leq 1$
- π_E is attempting optimise some unknown reward function $R^*(h_t, a_t) = w^* \cdot \phi(h_t, a_t)$ where w^* are the 'true' reward weights.
- The value of policy π can be re-written as:

$$\mathbb{E}[V^\pi(H_0) = w \cdot \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(H_t, A_t)|\pi]$$

- The feature expectation of π, defined as the expected discounted cumulative feature vector obtained when choosing actions according to π is

$$\mu^\pi = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(H_t, A_t)|\pi] \in \mathbb{R}^d$$

such that: $\mathbb{E}[V^\pi(H_0)] = w \cdot \mu^\pi$

Problem Formulation

- Our aim is to recover the expert weights W^* as well as find a policy π that is close to the policy of the expert π_E
- max-margin IRL approach and measure the similarity between the feature expectations of the expert's policy and the feature expectations of a candidate policy using $||\mu^{\pi_E} - \mu^{\pi}||_2$
- In this batch IRL setting, we do not have knowledge of transition dynamics and we cannot sample more trajectories from the environment.

Problem Formulation

- **Counterfactual reasoning:** To explain the expert's behaviour in terms of their trade-off associated with "what if" outcomes
- define the feature map $\phi(h_t, a_t)$ part of the reward $R(h_t, a_t) = w \cdot \phi(h_t, a_t)$
- Let $Y[a]$ be potential potential outcome, either factual or counterfactual, for treatment $a \in A$. Using Dataset D, learn feature map $\phi(h_t, a_t)$ such that $\phi(h_t, a_t) = \mathbb{E}[Y_{t+1}[a_t]|h_t$.
- The potential outcomes for the other actions are the counterfactual ones and they allow us to understand what would happen to the patient if they receive a different treatment.
- Consider the model for estimating counterfactuals as a black box such that the feature map ϕ represents the effect of taking action a_t for history h_t.

$$R(h_t, a_t) = w \cdot \phi(h_t, a_t) = w \cdot \mathbb{E}[Y_{t+1}[a_t]|h_t$$

Batch IRL using Conterfactuals

- Max-margin IRL starts with an initial random policy π and iteratively performs the following steps to recover the expert policy and its reward weighs:
  1. estimate feature expectations μ^{π} of candidate policy π,
  2. compute new reward weights w,
  3. find new candidate policy π that is optimal for reward function R
- This approach finds a policy $\tilde{\pi}$ that satisfies $||\mu^{\pi_E} - \mu^{\pi}||_2 < \epsilon$ such that $\tilde{\pi}$ has an expected value function close the expert policy.
- The expert feature expectations can be estimated empirically from the dataset D using:

$$\mu^{\pi_E} = \frac{1}{N} \sum_{i=0}^{N} \sum_{t=0}^{T^i} \gamma^t \phi(h_t^i, a_t^i)$$

Batch IRL using Conterfactuals

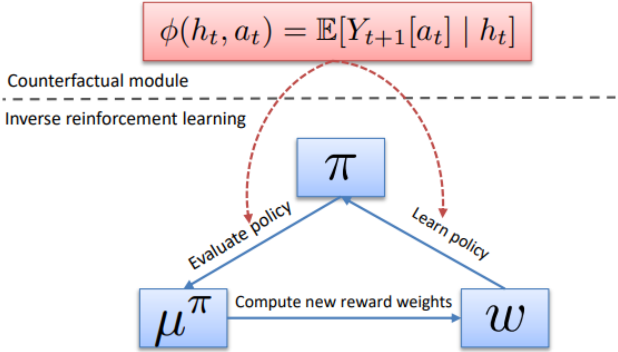

Figure: CIRL. Counterfactuals are used to define $\phi(h, a)$, to estimate feature expectations μ^π of candidate policy π in batch setting and to learn optimal policy for reward weights w.

Counterfactual μ - Learning

First action a is taken randomly and for $t \geq 1, A_t \sim \pi(\cdot|H_t)$.
This can be re-written as:

$$\mu^{\pi}(h,a) = \phi(h,a) + \mathbb{E}_{h',a' \sim \pi(\cdot|h')}[\sum_{t=0}^{\infty} \gamma^t \phi(h_t, a_t)|\pi, H_1 = h', A_1 = a']$$
$$= \phi(h,a) + \phi\, \mathbb{E}_{h',a' \sim \pi(\cdot|h')}[Q^{\pi}(h', a')]$$

where h' is the next history.

- Existing methods for estimating feature expectations fall into two extremes:
  (1) model-based (online)IRL approaches learn a model of the world and then use the model as a simulator to obtain on-policy roll-outs
  (2) batch IRL approaches use Q-learning for off-policy evaluation (Lee et al., 2019), and can only be used to evaluate policies similar to the expert policy and require warm start.

Counterfactual μ - Learning

- The paper proposes counterfactual μ-learning, a novel method for estimating feature expectations that uses these counterfactuals as part of temporal difference learning with 1-step bootstrapping.
- This approach falls in-between (1) and (2) and allows us to estimate feature expectations for any candidate policy π in the batch IRL setting.
- The counterfactual μ-learning algorithm learns the μ-values for policy π iteratively by updating the current estimates of the μ-values with the feature map plus the μ-values obtained by following policy π in the new counterfactual history $h' = (h, a, \mathbb{E}[Y[a]|h])$

$$\hat{\mu}^{\pi} \leftarrow \hat{\mu}^{\pi}(h, a) + \alpha(\phi(h, a) + \gamma\, \mathbb{E}_{a' \sim \pi(\cdot|h')}[\hat{\mu}^{\pi}(h', a')] - \hat{\mu}^{\pi}(h', a')$$

where α is the learning rate.

Batch, Max-Margin CIRL

Algorithm 1 (Batch, Max-Margin) CIRL

Input: Batch dataset $\mathcal{D}$, max iterations n, convergence threshold ϵ,
feature map $\phi(h_t, a_t) = \mathbb{E}[Y_{t+1}[a_t] | h_t]$
$\mu^{\pi_E} \leftarrow$ compute π_E's feature expectations (Equation 3)
$w_0 \leftarrow$ random initial reward weights, $\pi_0 \leftarrow$ compute optimal policy for $R_0 = w_0 \cdot \phi$
$\mu^{\pi_0} \leftarrow$ compute π_0's feature expectations (counterfactual μ-learning)
$\Pi = \{\pi_0\}, \Delta = \{\mu^{\pi_0}\}, \bar{\mu}_0 = \mu^{\pi_0}$
for $k = 1$ to n **do**
  $w_k = \mu^{\pi_E} - \bar{\mu}_{k-1}$, $\pi_k \leftarrow$ compute optimal policy for $R_k = w_k \cdot \phi$
  $\mu^{\pi_k} \leftarrow$ compute π_k's feature expectations (counterfacual μ-learning)
  $\Pi = \Pi \cup \{\pi_k\}, \Delta = \Delta \cup \{\mu^{\pi_k}\}$
  Orthogonally project μ^{π_E} onto line through $\bar{\mu}_{k-1}, \mu^{\pi_k}$:
  $$\bar{\mu}_k = \frac{(\mu^{\pi_k} - \bar{\mu}_{k-1})^T (\mu^{\pi_E} - \bar{\mu}_{k-1})}{(\mu^{\pi_k} - \bar{\mu}_{k-1})^T (\mu^{\pi_k} - \bar{\mu}_{k-1})} (\mu^{\pi_k} - \bar{\mu}_{k-1}) + \bar{\mu}_{k-1}, \qquad t = \|\mu^{\pi_E} - \bar{\mu}_k\|_2$$
  if $t < \epsilon$ **then** break
end for
$K = \arg\min_{k:\mu^{\pi_k} \in \Delta} \|\mu^{\pi_E} - \mu^{\pi_k}\|_2$, $\tilde{R}(h, a) = w_K \cdot \phi(h, a)$
Output: $\tilde{R}$, Δ, Π

Figure: Psedo Code for CIRL

Thank You