Veda Sunkara
Submission for Completion of Sc.M. Requirements for Brown University Master's in CS
Spring 2020

Causal Inference for Planning in Reinforcement Learning: Part ½

1. Abstract

We sought to develop environment-understanding and planning methodology for two reinforcement learning (RL) environments: GridWorld and Taxi[1] using causal inference. In this paper, we propose an algorithm for learning non-confounded relationships between objects in the environment, generating actionable rules and a structural causal model (SCM) from these observations, and planning using these rule sets. We show that we are able to successfully generate rules representative of the true, non-confounded relationships between objects and rewards in the environment, and use these rules to optimize agent behavior in these domains. In future work, we hope to scale these techniques to real-world datasets and assess the efficacy of SCM-based predictions at mitigating bias when compared to standard correlatory models.

2. Background and Introduction

We began our research project seeking to assess the efficacy of methodologies for mitigating bias in machine learning models. Motivated by the devastating effects of the COMPAS recidivism model[2], we read about, implemented, and assessed the efficacy of methods for mitigating protected class bias. What we found is outlined in our report from the Fall semester (link), but in summary we found ourselves dissatisfied with the limitations of these approaches as they failed to address the root of the biases in the first place.

Consider the adult ICU dataset[3]. This dataset includes fourteen covariates including some protected class covariates such as gender (for the purposes of this dataset, limited to male/female), and aims to predict annual income (a binary indicator for whether an individual earns more than 50K, annually). A simple DNN can be trained on this dataset to predict annual income. When comparing these predicted income distributions across gender, it can be seen that the model predicts lower incomes for women than men, on average. This makes sense, because if one compares the proportion of men who make < 50K to the proportion of women who make <

---

[1] "Download PDF - arXiv." https://arxiv.org/pdf/cs/9905014. Accessed 5 May. 2020.
[2] "How We Analyzed the COMPAS Recidivism Algorithm ...." 23 May. 2016, https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm. Accessed 4 May. 2020.
[3] "Adult Data Set - UCI Machine Learning Repository." 1 May. 1996, https://archive.ics.uci.edu/ml/datasets/adult. Accessed 4 May. 2020.

50K in the training data, the proportion of women is significantly higher. Namely, if the distribution of outcomes is skewed in the training data across protected class categories, it is reasonable that the respective model predictions would be skewed as well.

To attempt to mitigate this bias, we implemented a paper by Abusitta et. al.[4] that used a generative adversarial network (GAN) to generate synthetic data to supplement the dataset and attempt to even the distributions across genders. What we found was, even with a dataset that was 85% synthetic, we were still not able to completely even out the distributions. This is because, while somewhat effective at evening distributions, this method does not address the root of the issue causing the disparity between these distributions. Suppose we had some understanding of the inherent biases and structures that *cause* the income disparity across genders - could we make more equitable predictions?

Beginning with Judea Pearl and Dana Mackenzie's *The Book of Why*[5] we sought to understand 1. what causality is in both a mathematical and philosophical manner and 2. how it can be used to correct for false associations in correlatory data that perpetuate systemic bias. When thinking about how models can be made 'fair' in the context in which most data, especially that which is implicitly or explicitly influenced by systemic biases, in some way perpetuates that bias requires reconsidering how and why we make predictions based on correlations. Additionally, it requires reassessing the nature of making predictions using purely observable, correlatory data. An application of causal methods in the space of fairness would allow us to (a) more transparently represent how bias propagates through a system (b) evaluate deviations in outcomes on the basis of protected class status on the individual level, and (c) construct fairness definitions with respect to the notion of equity, not just equality.

Seeking a concrete context to assess this technique, as well as a springboard for how it could be applied to vastly complex issues of systemic bias, we focused on the idea of confounding. Put simply, confounding variables skew the observed relationship between two factors in an environment. Pearl often conveys this issue in the context of medicine and treatment: Let's say we have data on two variables: X, the patient's prescribed treatment, and Y, the patient's state of wellness. We observe correlation between X and Y. By the Principle of Common Cause, this correlation implies that X causes Y, Y causes X, or there exists some other variable Z (for example, a patient's income) that is the common cause of X and Y. We can say that the relationship between X and Y is *confounded* by Z[6]. It is necessary to control for confounding variables when assessing a causal relationship, but often confounders are unobservable and therefore must be inferred.

---

[4] "Generative Adversarial Networks for Mitigating Biases in ...." 23 May. 2019, https://arxiv.org/abs/1905.09972. Accessed 5 May. 2020.

[5] "The Book of Why: The New Science of Cause and Effect ...." http://bayes.cs.ucla.edu/WHY/why-ch1.pdf. Accessed 4 May. 2020.

[6] "Causal inference in statistics - Project Euclid." https://projecteuclid.org/euclid.ssu/1255440554. Accessed 4 May. 2020.

Is there a way, then, to learn what these confounding factors are from observation? Can we learn the true relationship between objects and rewards in an environment, purely based on observation, in spite of these confounding factors? Prior work on this question has been discussed in our previous writeup, and includes Lu et. al.'s "Deconfounding Reinforcement Learning in Observational Settings"[7], which served as a primer for deconfounding reinforcement learning in observational settings.

Through carefully constructed RL environments, we demonstrate the efficacy of using causal inference to learn the relationships between objects and rewards in an environment. We also demonstrate the success of using these relationships to plan in more general environments than those presented in training. From these rules, we can also construct an SCM for a given environment. It is our hope that these techniques for learning rules about an environment, constructing an SCM, and planning for a particular task can be scaled to more complex, real-world domains and help mitigate some of the biases demonstrated by purely correlatory models.

3. Methodology: GridWorld

We began our experiments with a standard GridWorld domain, in which we are trying to train an agent to go from some starting point to a goal with a high reward. There may be other elements in this GridWorld with different rewards, such as lava with a low reward. We consider the agent on the goal or on the lava to be absorbing states, thus terminating a run in the environment. We chose this domain for its simplicity, legibility, and for the easy addition of confounders (described below).

Let goal and lava be denoted by a symbol (i.e. star, circle, etc.) in their respective positions. Let us also now introduce color into this GridWorld, such that all positions with a given symbol have the same color. These colors are randomized after every episode, and do not affect reward. Our intent is to train an agent to deconfound its internal representation of the environment from observation. In this case, color is a confounder of the relationship between the latent space representations, Z, the observations of the GridWorld, X, and the reward function R. So, we want the agent to learn that there is a relationship between shape and reward and the color is spurious. We can consider learning these relationships as a baseline for learning a SCM for our environment.

We can extend this idea further by introducing more and more confounders. Consider a GridWorld in which each position in the grid is represented by an n-bit vector (we will refer to n as the 'feature dimension'). The bit at position 0 represents whether or not the agent exists at that location, the bit at position 1 represents whether or not the goal exists at that location, and the bit

[7] "Deconfounding Reinforcement Learning in Observational ...." 26 Dec. 2018, https://arxiv.org/abs/1812.10576. Accessed 4 May. 2020.

at position 2 represents whether or not the lava exists at that location. If bits 0 and 1 are set to 1, we receive the goal reward. If bits 0 and 2 are set to 1, we receive the lava reward. The remaining *n-3* values have no effect on the reward (we can think of them like the color factor in the previous example). We present an algorithm for learning the salient relationships between each position in the grid representation and reward. If successful, irrespective on the number of confounders, we should be able to learn that only the first bit is related to the rest and the remainder are spurious.

Our methodology for learning these relationships is presented in *Algorithm 1*. Key to the success of this approach is the initialization strategy, which skews the distribution of the remaining *n*-1 bits such that the mean rewards for each bit are highly correlated with one of the two types of reward. The graphs presented in **Results** are made with respect to a batch-mode reinforcement-learning method. We collected trajectories using a random agent, average agent (using averages from the same training set), and causal agent to compare average rewards and episode lengths as we increased the number of confounding factors. As shown in our results, only with causal rule generation are we able to successfully determine which factor(s) affect reward, irrespective of the number of confounders.

Algorithm 1

1. Initialization
    a. Generate an *mxm* maze, with feature dimension *n*.
    b. Initialize the goal, lava, and agent to three unique random positions
        i. Let bit 0 represent whether or not the agent exists at a given location
        ii. Let bit 1 represent whether or not the goal exists at a given location
        iii. Let bit 2 represent whether or not the lava exists at a given location
    c. Initialize noisy bits in positions where there exist goal and lava ONLY according to the following specifications (explanation in comments)

```
# to represent color split with a skewed proportion, we will force the first two extraneous bits to demonstrate a strong correlation bias
color_split = random value between (0.9, 1)
base_split = random value between (0, 1)
if (color_split < base_split):
        goal_position[i] = 1
        lava_position[i] = 0
else:
        goal_position[i] = 0
        lava_position[i] = 1
```

```
# initialize the remainder of the bits with an even random distribution
for i in range (5, n-5):
    split = random value between (0, 1)
    if(split < 0.5):
        goal_position[i] = 1
        lava_position[i] = 0
    else:
        goal_position[i] = 0
        lava_position[i] = 1
```

2. Data Collection
    a. for *x* iterations:
        i. Initialize a GridWorld as described in (1)
        ii. for *k* iterations:
            1. Initialize random rewards for goal and lava. Goal is chosen random uniformly from (0, 100) and lava is chosen randomly from (-100, 0).
            2. Randomly wander through the gridworld until termination state.
            3. For each bit set to 1 in the feature vector of the terminal state, record the reward value in a global table.
        iii. Generate a confidence interval for all the reward values found for each of the positions in the feature vector.
    b. Store the confidence intervals generated for each position in the feature vector across all *x* runs.

3. Rule Generation
    a. for each position in the feature vector:
        i. if all confidence intervals generated for feature *i* overlap, generate a rule associating *i* with the mean reward found when *i* is set to 1.

4. Planning
    a. Use standard q-learning, using the rules generated as a reward function.

4. Results

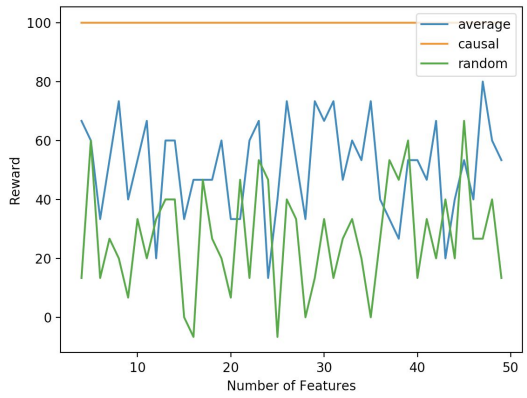
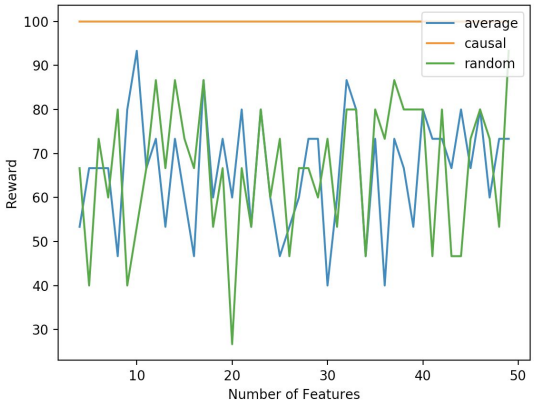
For each of the following test environments, we calculated the average reward and episode length over 30 runs for boards of size 4x4 and 10x10. We performed these experiments for 0 to 46 confounding variables (i.e. ‘noise bits’), which are initialized as described alongside each experiment.

The following are the reward-function baselines:
causal: only use the experimental average reward values if the confidence intervals for reward values in *train* overlap at all times
average: use all experimental average reward values
random: use 0.0 reward values

We expect the average baseline to perform better in the parallel training environment than in the randomized environment.

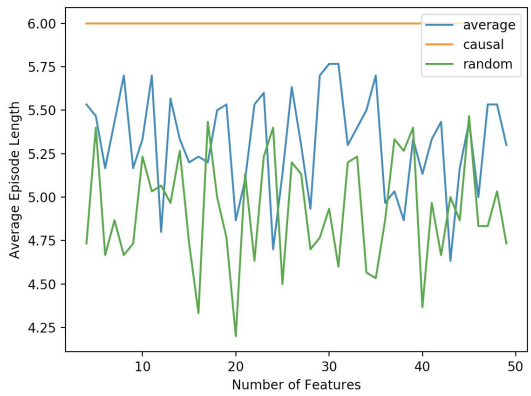
Experiment 1: *Randomized Gridworld* - Each bit is flipped on a random number of times (0 → number of grid positions) in random positions. Below are graphs of average reward across 30 runs, with the goal position initialized to 100 and the lava position initialized to -100.

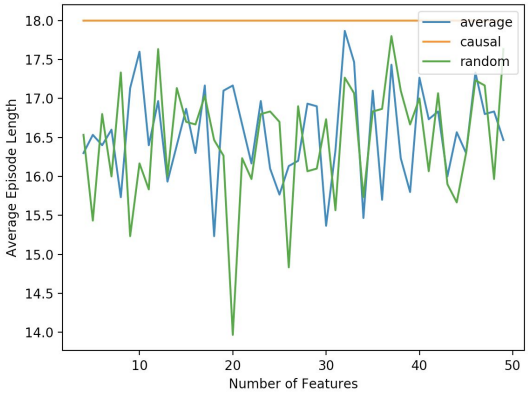
Reward

4x4	**10x10**
	

Average Episode Length

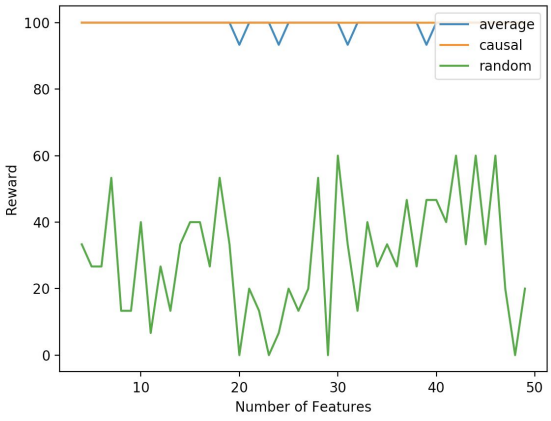
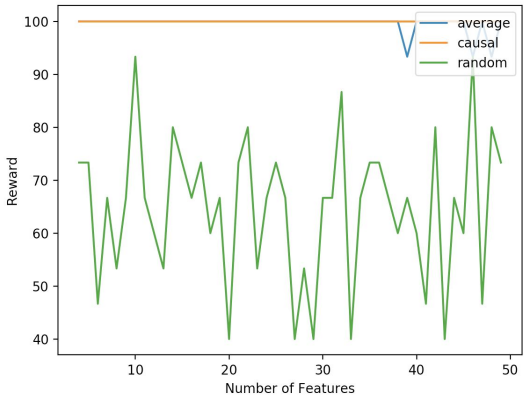
4x4	**10x10**





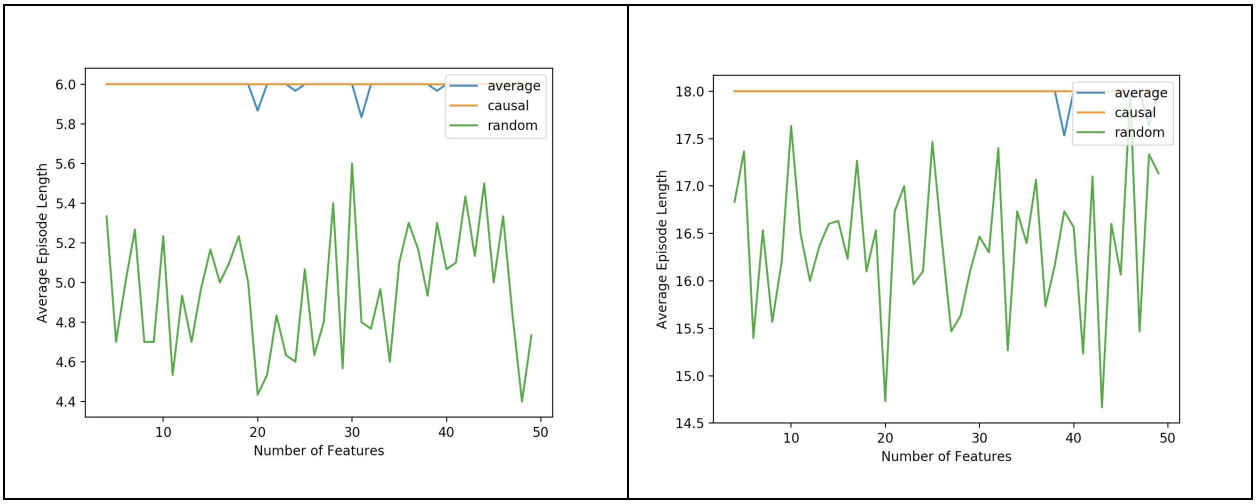
Experiment 2: *Parallel Training Environment Reward* - Each bit is randomly flipped on or off in ONLY the positions with goal or lava (i.e. a reward). Below are graphs of average reward across 30 runs, with the goal position initialized to 100 and the lava position initialized to -100.

Reward

4x4	**10x10**
	

Average Episode Length

4x4	**10x10**



5. Discussion

We have shown that, using our causal inference methodology, we are able to de-confound our internal representation of the factors in the environment that induce various rewards, and are able to use that information to plan perfectly in this environment. We can see that the average baseline performs significantly better in the testing environment that mirrors the training environment, but still does not outperform the causal planner. However, in a randomized testing environment that does not mirror the training scheme, we can see that our causal model significantly outperforms all other baselines. This goes to support our initial goal, which was to generate rules that were not specific to a particular instance of the domain, but rather applied to the domain in general.

6. Future Work

We have shown that using causal inference allows us to effectively learn which factors in a GridWorld cause the agent to achieve reward. We have also shown that knowing this information has allowed for perfect planning in this domain. What remains to be seen is how well this approach scales to more complex domains with more complex factors and outcomes. To demonstrate this, we assessed our causal rule generation technique on the TaxiCab domain.

This domain, with multiple factors, has a much more complex series of actions that are required to achieve reward (as opposed to the strictly translational transition dynamics of the GridWorld). We learned these transition dynamics using causal inference, and generated an outcome structure that allows us to plan and optimize reward. For specific methodology, outcomes, and results, see Part 2 of this report written by Naveen Srinivasan.

Finally, as is rooted in our initial motivation for this research, we would like to understand if/how these methodologies can scale to real world environments to help mitigate bias. This is certainly a lofty goal, and it is difficult to see how these techniques can scale to such

structurally complex issues, but it is our hope that if we can learn to develop causal frameworks for predictions that can better and more equitably represent the state of various domains in the world.

7. Acknowledgements

Thank you to our advisor, Professor Michael Littman, for over a year of mentorship and support; your patience and insights have been invaluable. Thank you to the Computer Science department at Brown for five years of education, community, and growth.