Homework 4

1 Off Policy Evaluation And Causal Inference

In class we have discussed Markov decision processes (MDPs), methods for learning MDPs from data, and ways to compute optimal policies from that MDP. However, before we use that policy, we often want to get an estimate of the its performance. In some settings such as games or simulations, you are able to directly implement that policy and directly measure the performance, but in many situations such as health care implementing and evaluating a policy is very expensive and time consuming.

Thus we need methods for evaluating policies without actually implementing them. This task is usually referred to as off-policy evaluation or causal inference. In this problem we will explore different ways of estimating off policy performance and prove some of the properties of those estimators.

Most of the methods we discuss apply to general MDPs, but for the sake of this problem, we will consider MDPs with a single timestep. We consider a universe consisting of states S, actions A, a reward function $R(s,a)$ where s is a state and a is an action. One important factor is that we often only have a subset of a in our dataset. For example, each state s could represent a patient, each action a could represent which drug we prescribe to that patient and $R(s,a)$ be their lifespan after prescribing that drug.

A policy is defined by a function $\pi_i(s,a) = p(a|s,\pi_i)$. In other words, $\pi_i(s,a)$ is the conditional probability of an action given a certain state and a policy.

We are given an observational dataset consisting of $(s, a, R(s,a))$ tuples.

Let $p(s)$ denote the probability density function for the distribution of state s values within that dataset. Let $\pi_0(s,a) = p(a|s)$ within our observational data. π_0 corresponds to the baseline policy present in our observational data. Going back to the patient example, $p(s)$ would be the probability of seeing a particular patient s and $\pi_0(s,a)$ would be the probability of a patient receiving a drug in the observational data.

We are also given a target policy $\pi_1(s,a)$ which gives the conditional probability $p(a|s)$ in our optimal policy that we are hoping to evaluate. One particular note is that even though this is a distribution, many of the policies that we hope to evaluate are deterministic such that given a particular state s_i, $p(a|s_i) = 1$ for a single action and $p(a|s_i) = 0$ for the other actions.

Our goal is to compute the expected value of $R(s,a)$ in the same population as our observational data, but with a policy of π_1 instead of π_0. In other words, we are trying to compute:

$$\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$$

Important Note About Notation And Simplifying Assumptions:

We haven't really covered expected values over multiple variables such as $\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$ in class yet. For the purpose of this question, you may make the simplifying assumption that our states and actions are discrete distributions. This expected value over multiple variables simply indicates that we are taking the expected value over the joint pair (s,a) where s comes from $p(s)$ and a comes from $\pi_1(s,a)$. In other words, you have a $p(s,a)$ term which is the probabilities of observing that pair and we can factorize that probability to p$(s)p(a|s) = p(s)\pi_1(s,a)$. In math notation, this can be written as:

$$\begin{aligned}\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a) &= \sum_{(s,a)} R(s,a)p(s,a) \\ &= \sum_{(s,a)} R(s,a)p(s)p(a|s) \\ &= \sum_{(s,a)} R(s,a)p(s)\pi_1(s,a)\end{aligned}$$

Unfortunately, we cannot estimate this directly as we only have samples created under policy π_0 and not π_1. or this problem, we will be looking at formulas that approximate this value using expectations under π_0 that we can actually estimate.

We will make one additional assumption that each action has a non-zero probability in the observed policy $\pi_0(s,a)$. In other words, for all actions a and states s, $\pi_0(s,a) > 0$.

Regression: The simplest possible estimator is to directly use our learned MDP parameters to estimate our goal. This is usually called the regression estimator. While training our MDP, we learn an estimator $\hat{R}(s,a)$ that estimates $R(s,a)$. We can now directly estimate

$$\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$$

with

$$\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}\hat{R}(s,a)$$

If $\hat{R}(s,a) = R(s,a)$, then this estimator is trivially correct.

We will now consider alternative approaches and explore why you might use one estimator over another.

(a) Importance Sampling

One commonly used estimator is known as the importance sampling estimator. Let $\hat{\pi}_0$ be an estimate of the true π_0. The importance sampling estimator uses that $\hat{\pi}_0$ and has the form:

$$\mathbb{E}_{s\sim p(s),a\sim\pi_0(s,a)}\frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}R(s,a)$$

Please show that if $\hat{\pi}_0 = \pi_0$, then the importance sampling estimator is equal to:

$$\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$$

Note that this estimator only requires us to model π_0 as we have the $R(s,a)$ values for the items in the observational data.

(b) Weighted Importance Sampling

One variant of the importance sampling estimator is known as the weighted importance sampling estimator. The weighted importance sampling estimator has the form:

$$\frac{\mathbb{E}_{s\sim p(s),a\sim\pi_0(s,a)}\frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}R(s,a)}{\mathbb{E}_{s\sim p(s),a\sim\pi_0(s,a)}\frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}}$$

Please show that if $\hat{\pi}_0 = \pi_0$, then the weighted importance sampling estimator is equal to

$$\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$$

(c)

One issue with the weighted importance sampling estimator is that it can be biased in many finite sample situations. In finite samples, we replace the expected value with a sum over the seen values in our observational dataset. Please show that the weighted importance sampling estimator is biased in these situations.

Hint: Consider the case where there is only a single data element in your observational dataset.

TA's Remark: This problem asks you to prove that the value of weighted importance sampling formula is not equal to $\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$ in some way when data size is "finite", especially equals to 1.

(d) Doubly Robust

One final commonly used estimator is the doubly robust estimator. The doubly robust estimator has the form:

$$\mathbb{E}_{s\sim p(s),a\sim\pi_0(s,a)}((\mathbb{E}_{a\sim\pi_1(s,a)}\hat{R}(s,a)) + \frac{\pi_1(s,a)}{\hat{\pi}_0(s,a)}(R(s,a) - \hat{R}(s,a)))$$

One advantage of the doubly robust estimator is that it works if either $\hat{\pi}_0 = \pi_0$ or $\hat{R}(s,a) = R(s,a)$

1 Please show that the doubly robust estimator is equal to $\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$ when $\hat{\pi}_0 = \pi_0$.

2 Please show that the doubly robust estimator is equal to $\mathbb{E}_{s\sim p(s),a\sim\pi_1(s,a)}R(s,a)$ when $\hat{R}(s,a) = R(s,a)$.

(e)

We will now consider several situations where you might have a choice between the importance sampling estimator and the regression estimator. Please state whether the importance sampling estimator or the regression estimator would probably work best in each situation and explain why it would work better. In all of these situations, your states s consist of patients, your actions a represent the drugs to give to certain patients and your $R(s,a)$ is the lifespan of the patient after receiving the drug.

1 Drugs are randomly assigned to patients, but the interaction between the drug, patient and lifespan is very complicated.

2 Drugs are assigned to patients in a very complicated manner, but the inter- action between the drug, patient and lifespan is very simple.

2 PCA

In class, we showed that PCA finds the ”variance maximizing” directions onto which to project the data. In this problem, we find another interpretation of PCA.

Suppose we are given a set of points $\{x^{(1)},\ldots,x^{(m)}\}$. Let us assume that we have as usual preprocessed the data to have zero-mean and unit variance in each coordinate. For a given unit-length vector u, let $f_u(x)$ be the projection of point x onto the direction given by u. I.e., if $\mathcal{V} = \{\alpha u : \alpha \in \mathbb{R}\}$, then

$$f_u(x) = \arg\min_{v\in\mathcal{V}} ||x - v||^2$$

Show that the unit-length vector u that minimizes the mean squared error between projected points and original points corresponds to the first principal component for the data. I.e., show that

$$\arg\min_{u:u^Tu=1}\sum_{i=1}^{m}\|x^{(i)} - f_u(x^{(i)})\|_2^2$$

Remark. If we are asked to find a k-dimensional subspace onto which to project the data so as to minimize the sum of squares distance between the original data and their projections, then we should choose the k-dimensional subspace spanned by the first k principal components of the data. This problem shows that this result holds for the case of $k = 1$.

3 Markov decision processes

Consider an MDP with finite state and action spaces, and discount factor $\gamma < 1$. Let B be the Bellman update operator with V a vector of values for each state. I.e., if $V' = B(V)$, then

$$V'(s) = R(s) + \gamma\max_{a\in A}\sum_{s'\in S} P_{sa}(s')V(s')$$

(a)

Prove that, for any two finite-valued vectors V_1, V_2, it holds true that

$$||B(V_1) - B(V_2)||_\infty \leq \gamma||V_1 - V_2||_\infty$$

where

$$||V||_\infty = \max_{s \in S} |V(s)|$$

(This shows that the Bellman update operator is a "γ-contraction" in the max-norm.)

(b)

We say that V is a fixed point of B if $B(V) = V$. Using the fact that the Bellman update operator is a γ-contraction in the max-norm, prove that B has at most one fixed point-i.e., that there is at most one solution to the Bellman equations. You may assume that B has at least one fixed point.

Remark: The result you proved in part(a) implies that value iteration converges geometrically to the optimal value function V^*. That is, after k iterations, the distance between V and V^* is at most γ^k.

4 One-class SVM

Given an unlabeled set of examples $\{x^{(1)}, \ldots, x^{(m)}\}$, the one-class SVM algorithm tries to find a direction w that maximally separates the data from the origin. More precisely, it solves the (primal) optimization problem:

$$\begin{aligned} \min_w \quad & \frac{1}{2} w^\top w \\ \text{s.t.} \quad & w^\top x^{(i)} \geq 1 \quad \text{for all} i = 1, \ldots, m \end{aligned}$$

A new test example x is labeled 1 if $w^\top x \geq 1$, and 0 otherwise.

(a)

The primal optimization problem for the one-class SVM was given above. Write down the corresponding dual optimization problem. Simplify your answer as much as possible. In particular, w should not appear in your answer.

(b)

Can the one-class SVM be kernelized (both in training and testing)? Justify your answer.

(c) (Optional)

Give an SMO-like algorithm to optimize the dual. I.e., give an algorithm that in every optimization step optimizes over the smallest possible subset of variables. Also give in closed-form the update equation for this subset of variables. You should also justify why it is sufficient to consider this many variables at a time in each step.

TA's Remark: You should point out the minimum number of variables for each step's optimization and justify it based on the principles of SMO algorithm.