WA4 Solutions

Om Khadka, U51801771
Collabs: N/A

Question 1: The Precariousness of RL (25 points)

In the last written assignment you considered a loss function where each point was weighted with a nonzero coefficient $r^{(i)}$. You showed that the optimal solution was a function of the weights each sample was given. In this problem you will express an arbitrary supervised learning dataset (which we use in RL) into a dataset where samples are given weights:

1. Whenever we have a dataset where each sample is unique (e.g. does not appear more than once), we are creating a dataset where each sample is given a uniform weight. Show that when a dataset contains duplicate points we are creating a dataset where samples are not given uniform weights.

2. Now relax the weighting terms so that every weight $r^{(i)} \geq 0$ instead of > 0. Show that this loss function over a dataset of finite samples can be converted into a weighted sum over all possible data points.

3. What happens to terms with weights of 0? Do they impact the solution at all? How do we know that the model will perform well on these points?

Note: this is a **proof** question, meaning you must follow formal proof structure (see the examples of piazza for guidance). There is one exception: you do not have to follow formal proof structure when answering the last part, natural language is fine but I will not accept hand-wavy justification.

For part **1**, I'll show that a dataset with duplicate points is implicitly a weighted dataset, where the weights are non-uniform.

Note: This question references my solution to question 4 of **WA3**.

Assertion	**Explanation**
Consider a dataset D with M unique data points, $\{(x^{(i)}, y^{(i)})\}_{i=1}^{M}$.	Defining the set of all distinct data points.
Let n_i be the frequency of the i-th unique point, so the total dataset size is $N = \sum_{i=1}^{M} n_i$.	A dataset with duplicates can be characterized by the counts n_i of its unique points.
Standard MSE loss over the full dataset: $L(\theta) = \frac{1}{2N} \sum_{j=1}^{N} (y^{(j)} - f_\theta(x^{(j)}))^2$.	The is the real (canonical) unweighted loss, summing over every individual sample.
Regrouping the sum w/ unique points: $L(\theta) = \frac{1}{2N} \sum_{i=1}^{M} n_i (y^{(i)} - f_\theta(x^{(i)}))^2$.	Instead of summing over all N samples, we instead sum over the M unique points, with each point's error term multiploie by its count n_i.
Let $r^{(i)} = n_i$.	
$L(\theta) = \frac{1}{2N} \sum_{i=1}^{M} r^{(i)} (y^{(i)} - f_\theta(x^{(i)}))^2$.	This is identical to the weighted loss function from questions 4, **WA3**, with weights $r^{(i)} = n_i$.
If any $n_i \neq n_j$, then the weights $r^{(i)}$ are not uniform.	By definition, if duplicates exists, then at least ONE $n_i > 1$. $\therefore$ The counts n_i (and also the weights $r^{(i)}$) can't all be equal.

This shows that a dataset with duplicates is equivalent to a weighted dataset, where the weight of a unique point is its frequency of occurrence. Since frequencies are not uniform, the weights are not uniform.

For part 2, we relax the condition from $r^{(i)} > 0$ to $r^{(i)} \geq 0$ and show the loss can be written as a sum over all possible points in the input space.

Assertion	**Explanation**
Let X be the set of all possible input points x.	Defining the domain of the model.
For a given dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ and weights $r^{(i)} \geq 0$, the loss is: $L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} r^{(i)}(y^{(i)} - f_\theta(x^{(i)}))^2$.	This is the given loss function with non-neg weights.
We define a function $p(x, y)$ over $X \times \mathbb{R}$ as: $p(x, y) = \sum_{j \atop (x^{(j)}, y^{(j)}) = (x,y)} r^{(j)}$, and $p(x, y) = 0$ if $(x, y) \notin D$.	$p(x, y)$ aggregates the total weight assigned to all data points with value (x, y). For a point not in the dataset, its weight is zero.
Original loss can be rewritten: $L(\theta) = \frac{1}{2N} \sum_{(x,y) \in X \times R} p(x, y) \cdot (y - f_\theta(x))^2$.	This new sum now iterates over all possible data points (x, y). For any points not in the dataset, $p(x, y) = 0$, so it doesn't contribute to anything. For a point in the dataset, it contributes the sum of its weights $\cdot$ the squared error.
Since the dataset is finite, only finitely many terms in this sum are non-zero.	This ensures the sum is well-defined and computationally tractable, even if X is infinite.

$\therefore$ The loss over a finite, weighted dataset is equivalent to a weighted sum over all possible points, where the weight function, $p(x, y)$ is non-zero only on the dataset points.

For part 3, when weights are allowed to be zero, the nature of the optimization will change. The consequences and its inherent implications, particularly for Reinforcement, is shown below.

Assertion	**Explanation**
Points with weights $r^{(i)}$ contribute nothing to the loss function.	In $L(\theta) = \frac{1}{2N}\sum_{i=1}^{N} r^{(i)}(y^{(i)} - f_\theta(x^{(i)}))^2$, when $r^{(i)} = 0$, the entire term for that data point is zero, regardless of the model's error.
They have no direct influence on the optimal solution θ^*.	In the normal equation $\Phi^T R\Phi\theta = \Phi^T Ry$, rows corresponding to zero weights are multiplied by 0 in the diagnoal matrix R, effectively removing them from solving for θ^*.
The model provides no guarantees about performance on 0-weight points.	The optimization process is insensitive to these points. The solution is determined entirely by points with $r^{(i)} > 0$. Performance on 0-weight points depends on inductive bias and correlation with trained points.
In RL, this then produces an issue: **states with 0 weight are unexplored, and their value estimates are untrained and are potentially arbitrary**.	If a policy never visits a state-action pair, it recieves 0 weight in the value function update. The estimate value for that state is not grounded in experience and could be inaccurate.

Basically, a zero weight acts as a flag for the model to just ignore that particular data point during training. I guess you could say that this is the precariousness of this hypothetical. There is an absolute distinction in this framework between what is trained on (points with $r^{(i)} > 0$) and what is not (points with $r^{(i)} = 0$). The model's performance on the latter is not a matter of bad generalization, but of no generalization constraint at all. This basically results in critical failure, as the value function for unvisited states remains at its potentially random initial initialization, causing poorly informed policies that realistically will never learn to explore effectively. The model's performance on these point, thus, is not just unverified, but moreso literally unconstrained by the learning objective.

Question 2: Reward Function Flavors (25 points)

In lecture, we talked about MDPs that are formulated with a reward function $R(s)$ (i.e. the reward only depends on the current state). However, sometimes MDPs are formulated with a reward function $R(s, a)$ (i.e. a reward function that depends on the action taken), or even $R(s, a, s')$ (i.e. a reward function that depends on the action taken and the way the action is resolved). In this problem, you will show that even though someone may choose one flavor of reward function over another, they are actually identical:

1. Write the bellman equation that uses $R(s, a)$ and write the bellman equation that uses $R(s, a, s')$
2. Show how an MDP with reward function $R(s, a, s')$ can be converted into a different MDP with reward $R(s, a)$ such that optimal policies in the new MDP correspond exactly to optimal policies in the original MDP.
3. Show how an MDP with reward function $R(s, a)$ can be convered into a different MDP with reward $R(s)$ such that optimal policies in the new MDP correspond exactly to optimal policies in the original MDP.

Note: this is a **proof** question, meaning you must follow formal proof structure (see the examples of piazza for guidance).

I will show the equivalence between MDPs formulated with different reward function flavors by constructing transformations that preserve optimal policies.

Firstly, I will show the Bellman equations (**part 1**):

- $$V^*(s) = \max_{a \in A}[R(s, a) + \gamma \sum_{s' \in S} T(s'|s, a)V^*(s')]$$

  Rewards function $R(s, a)$: This equation states that the optimal value of a state is the maximum over actions of the immediate reward plus the discounted expected future value.

- $$V*(s) = \max_{a \in A}[\sum_{s' \in S} T(s'|s, a)(R(s, a, s') + \gamma V^*(s'))]$$

  Reward function $R(s, a, s')$: On this one, the expectation over next states is taken inside the immediate reward term since the reward depends on the transition outcome.

I will now transform an MDP $M = (S, A, T, R_{sas'}, \gamma)$ with reward $R(s, a, s')$ into a new MDP $M' = (S, A, T, R_{sa}, \gamma)$ with reward $R(s, a)$ (**part 2**).

Assertion	**Explanation**
Let $M = (S, A, T, R_{sas'}, \gamma)$ be the original MDP.	Defintion, by given.
Define a new MDP $M' = (S, A, T, R_{sa}, \gamma)$ with $R_{sa}(s, a) = \sum_{s' \in S} T(s'\|s, a) R_{sas'}(s, a, s')$.	Compute the expected immediate reward for taking action a in state s.
In M' : $V'^*(s) = \max_a [R_{sa}(s, a) + \gamma \sum_{s'} T(s'\|s, a) V'^*(s')]$.	Standard Bellman equation for $R(s, a)$.
$V'^*(s) = \max_a [\sum_{s'} T(s'\|s, a) R_{sas'}(s, a, s') + \gamma \sum_{s'} T(s'\|s, a) V'^*(s')]$.	Substibution of R_{sa}, by definition.
$V'^*(s) = \max_a \sum_{s'} T(s'\|s, a) [R_{sas'}(s, a, s') + \gamma V'^*(s')]$.	Factor the expectation over s'.
This matches the Bellman equation for the original MDP M.	In comparison to the 2nd equation from part 1.
Since the Bellman equations are identical, $V'^* = V^*$ an optimal policies are preserved.	Both MDPs have the same optimal value function.

$\therefore$ The transformed MDP M' with $R(s, a)$ has identical optimal policies to the original MDP with $R(s, a, s')$.

Now I will transform an MDP $M = (S, A, T, R_{sa}, \gamma)$ into a new MDP $M' = (S', A, T', R_s, \gamma)$ with reward $R(s)$ (**part 3**).

Assertion	**Explanation**
Let $M = (S, A, T, R_{sa}, \gamma)$ be the original MDP.	Given
Define $S' = S \times A$. New states are (s, a) pairs.	We encoe the action choice into the state.
Keep the same action space A.	Actions remain meaningful in the new state representation.
Define $T'((s', a')\|(s, a), a'') = \begin{cases} T(s'\|s, a) & \text{if } a'' = a' \\ 0 & \text{otherwise} \end{cases}$	The transition first goes to (s', a') deterministically based on chosen action a'.
Define $R_s(s, a) = R_{sa}(s, a)$.	The Reward depends only on the current "state" (s, a) in M'.
Any policy π' in M' correspons to π in M via $\pi(a\|s) = \pi'(a\ (s, a_{\text{prev}}))$.	Policies in M' make the same decision as in M.
The value functions satisfy $V'^*((s, a)) = Q^*(s, a)$ from original MDP.	The new state (s, a) directly represents the Q-val.
Optimal policies in M' correspond exactly to the optimal policies in M.	The optimal Q-vals are preserv through the transformation.

$\therefore$ The transformed MDP M' with $R(s)$ has identical optimal policies to the original MDP with $R(s, a)$.

This proves how all 3 reward functions flavors $(R(s, a, s'), R(s, a), R(s))$ are equivalent in terms of their ability to represent the same set of decision problems with preserved optimal policies.

Question 3: Sum of Discounted Rewards vs. Max Reward (25 points)

In lecture we defined the utility of a trajectory to be some additive combination of the rewards along that trajectory. So far this has taken two forms: additive rewards and discounted rewards. However, what happens if we define the utility of a trajectory as the maximum reward observed in that trajectory? Show that this utility function does not result in stationary preferences between trajectories (i.e. that such an agent may change its preference for the optimal trajectory as a function of time). Is it still possible to define autility function on trajectories such that a policy which maximizes the expected trajectory utility results in optimal behavior?

Note: this is a **proof** question, meaning you must follow formal proof structure (see the examples of piazza for guidance).

We can first show that the maximum reward utility leads to non-stationary preferences.

Assertion	**Explanation**
Let $U(\tau) = \max_t r_t$ be the utility of trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$.	Given utility function based on maximum reward.
Preferences are stationary iff $\forall t, \tau_A \succeq \tau_B \iff \tau_A[t:] \succeq \tau_B[t:]$.	Defining stationary preferences.
Consider 2 trajectories starting at time $t = 0$: • τ_A is rewards $= [0, 0, 5]$ • τ_B are rewards $= [4, 0, 0]$	An exmaple with different reward distributions.
$U(\tau_A) = 5, U(\tau_B) = 4, \therefore$ Agents prefers τ_A over τ_B on time 0.	By the definition of max utility.
At $t = 1$, remaining trajectories are: • τ'_A's reward $= [0, 5]$ • τ'_B's reward $= [0, 0]$	Examining the next step.
$U(\tau'_A) = 5, U(\tau'_B) = 0, \therefore$ Agent prefers τ'_A over τ'_B at time 1.	Still consistent.
Now we consider the following: • τ_E's rewards $= [0, 3]$ • τ_F's rewars $= [2, 0]$	Show an another exmaple.
$U(\tau_E) = 3, U(\tau_F) = 2$, Prefers τ_E.	On $t = 0$.
Suppose at $t = 0$, the agent must choose path leaing to either τ_E or τ_F.	The desicion point in question.
If the agent takes the path to τ_E, after getting $r = 0$, remaining utility is 3. If the agent takes the path to τ_F, after recieving $r = 2$, remaining utility is 0.	The agent's assessment would change mid-trajectory.
At $t = 0$, moel with prefer the path to τ_E, but after recieving $r = 2$, it would then prefer to have taken τ_F	At $t = 0$, max $= 3 >$ max $= 2$, but then at $t = 1$, as it is now 2 vs. 0. This is the point of preference reversal, as the agent would regret its first choice.

$\therefore$ The maximum reward utility function leas to non-stationary preferences because of agent's assessment of trajectory value changing as the rewards are being observe and removed for future consideration.

Now to prove if we can define the optimal behavior with maxmim reward utility.

Assertion	**Explanation**
Since $U(\tau) = \max_t r_t$ violates the principle of optimality, thus the Bellman's equation fails.	Optimal decisions depend on max rewards already observed, breaking time consistency.
Define augmented state $\tilde{s} = (s, m)$ where $m = \max_{0 \le k < t} r_k$.	This makes the process Markovian, as m tracks the current maximum reward.
New objective: maximize probability that $m_{\text{final}} \ge \theta$ for some threshold θ.	Equivalent to $\mathbb{E}[\mathbf{1}(m_T \ge \theta)]$, where m_T is final max reward.
Define $V(s, m) = \max_a \mathbb{E}\big[\mathbf{1}(r \ge m) + V(s', \max(m, r))\big]$.	This yields a well-defined Bellman equation in the augmented space.
More generally, define $U_\theta(\tau) = \mathbf{1}(\max_t r_t \ge \theta)$.	Then $\pi^* = \arg\max_\pi \mathbb{E}_{\tau \sim \pi}[U_\theta(\tau)]$ is a valid objective.
However, θ must be known or optimized over, leading to a multi-objective problem.	This is computationally harder than standard additive/discounted MDPs.

$\therefore$ While it is *theoretically* possible to define optimal behavior for a maximum reward utility, the reuslting optimization problem would lose the recursive structure of standard MDPs and thusly becomes computationally challenging.

Question 4: Proof that the Bellman Equation is a Contraction Function (25 points)

In lecture we claimed that the bellman equation is a contraction function. Specifically, we said that, for any two vectors of utilities $\vec{u}$ and $\vec{u}'$:

$$||B(\vec{u}) - B(\vec{u}')||_\infty \leq \gamma||\vec{u} - \vec{u}'||_\infty$$

1. Show that, for any functions f and g:

$$|\max_a f(a) - \max_a g(a)| \leq \max_a |f(a) - g(a)|$$

2. Derive an expression for $\left|\Big(B(\vec{u}) - B(\vec{u}')\Big)(s)\right|$ and then apply the result from part 1 to complete the proof that the bellman equation is a contraction function.

Note: this is a **proof** question, meaning you must follow formal proof structure (see the examples of piazza for guidance).

We first go on to proof the assertion shown in part 1.

Assertion	**Explanation**
Let $a_f \in \arg\max_a f(a)$	Definition: $f(a_f) = \max_a f(a)$
Let $a_g \in \arg\max_a g(a)$	Definition: $g(a_g) = \max_a g(a)$
$f(a_f) \geq f(a_g)$	a_f maximizes f
$g(a_g) \geq g(a_f)$	a_g maximizes g
Case 1: $f(a_f) \geq g(a_g)$	We now consider 2 seperate cases: One where $\max_a f(a) \geq \max_a g(a)$ and one where $\max_a g(a) \geq \max_a f(a)$. This case looks at the former.
$\max_a f(a) - \max_a g(a) = f(a_f) - g(a_g)$	By definition
$f(a_f) - g(a_g) \leq f(a_f) - g(a_f)$	Since $g(a_g) \geq g(a_f)$
$f(a_f) - g(a_f) \leq \lvert f(a_f) - g(a_f)\rvert$	Property of absolute value
$\leq \max_a \lvert f(a) - g(a)\rvert$	Definition of max
Case 2: $g(a_g) \geq f(a_f)$	Now we look at the symmetric case, where $\max_a g(a) \geq \max_a f(a)$
$\max_a g(a) - \max_a f(a) = g(a_g) - f(a_f)$	By definition
$g(a_g) - f(a_f) \leq g(a_g) - f(a_g)$	Since $f(a_f) \geq f(a_g)$
$g(a_g) - f(a_g) \leq \lvert g(a_g) - f(a_g)\rvert$	Property of absolute value
$\leq \max_a \lvert f(a) - g(a)\rvert$	Definition of max
$\lvert \max_a f(a) - \max_a g(a)\rvert \leq \max_a \lvert f(a) - g(a)\rvert$	Combining both cases

Now, let $\overrightarrow{u}, \overrightarrow{u}$ be any 2 utility vectors.

Assertion	**Explanation**
$B(\vec{u})(s) = \max_a [R(s,a) + \gamma \sum_{s'} T(s' \mid s,a)u(s')]$	Defining the Bellman update
$B(\vec{u}')(s) = \max_a [R(s,a) + \gamma \sum_{s'} T(s' \mid s,a)u'(s')]$	Is the same for $\vec{u}'$
Let $f_s(a) = R(s,a) + \gamma \sum_{s'} T(s' \mid s,a)u(s')$	A shorthand
Let $g_s(a) = R(s,a) + \gamma \sum_{s'} T(s' \mid s,a)u'(s')$	Another shorthand
$\lvert B(\vec{u})(s) - B(\vec{u}')(s)\rvert = \lvert \max_a f_s(a) - \max_a g_s(a)\rvert$	Substiution the expression of $B(\vec{u})(s)$ and $B(\vec{u}')(s)$ with the shorthand functions.
$\leq \max_a \lvert f_s(a) - g_s(a)\rvert$	First part of the lemma
$f_s(a) - g_s(a) = \gamma \sum_{s'} T(s' \mid s,a)[u(s') - u'(s')]$	Subtract, and $R(s,a)$ cancels out
$\lvert f_s(a) - g_s(a)\rvert = \gamma \lvert \sum_{s'} T(s' \mid s,a)[u(s') - u'(s')]\rvert$	Factor out $\gamma > 0$
$\leq \gamma \sum_{s'} T(s' \mid s,a)\lvert u(s') - u'(s')\rvert$	Is the triangle inequality, $T \geq 0$
$\leq \gamma \sum_{s'} T(s' \mid s,a) \max_{s''} \lvert u(s'') - u'(s'')\rvert$	$\lvert u(s') - u'(s')\rvert \leq \lVert u - u'\rVert_\infty$
$= \gamma \lVert \vec{u} - \vec{u}'\rVert_\infty \sum_{s'} T(s' \mid s,a)$	Pulling out constant w.r.t. s'
$\sum_{s'} T(s' \mid s,a) = 1$	Transition probabilities add up to 1
$\lvert f_s(a) - g_s(a)\rvert \leq \gamma \lVert \vec{u} - \vec{u}'\rVert_\infty$	Subbing the entire bound of $\gamma \sum_{s'} T(s' \mid s,a)\lvert \vec{u} - \vec{u}'\rvert_\infty$ with a more simplified bound of $\gamma \lvert \vec{u} - \vec{u}'\rvert_\infty$. Works since transition probabilities sum to 1.
$\max_a \lvert f_s(a) - g_s(a)\rvert \leq \gamma \lVert \vec{u} - \vec{u}'\rVert_\infty$	Max over a of the constant bound
$\lvert B(\vec{u})(s) - B(\vec{u}')(s)\rvert \leq \gamma \lVert \vec{u} - \vec{u}'\rVert_\infty$	Transitivity from the earlier steps
This holds for all s	s was arbitrary
$\lVert B(\vec{u}) - B(\vec{u}')\rVert_\infty \leq \gamma \lVert \vec{u} - \vec{u}'\rVert_\infty$	Is the definition of $\lVert \cdot \rVert_\infty$ as max over s

$\therefore$ The Bellman operatore B is a contraction with modulus γ in the ∞-norm.