**CPSC 4610 – Artificial Intelligence**

**Homework 3**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Seattle University**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_**

**Jakob Balkovec 02/24/2025**

**Q1 (15 points) Perspective**

1. **Copy-paste your definition of ‘Artificial Intelligence’ from EX0**
2. **Modify this definition to reflect what you’ve learned so far this quarter**
3. **Discuss key point(s) of your modification**

**A1**

* **My Definition:** AI is a technology that enables machines to think and perform tasks similarly to  
  humans, though with some margin of error.
* **New Definition:** AI is a field of CS, that focuses on developing algorithms that enable machines to solve “intractable” problems, make decisions, and learn from experience. It includes techniques like search algorithms, CSPs, MDPs, POMDPs, RL, … This allows machines to operate on their own/autonomously and effectively in complex environments.

**Key Points**

I think my new definition is a little more concise and provides some more clarity, compared to my original one. In my original one I mentioned that AI allows machines to “think” like humans, which is a broad and “[anthropomorphic](https://www.oxfordreference.com/display/10.1093/oi/authority.20110803095416390)” description. In my revised version, I provided some more clarity about the fact that AI is about algorithms, heuristics, and models designed to solve specific problems rather than simulating human-thinking.  
  
I also used some of the course topics we covered (Until Feb 25th) in my definition.

* **Search Algorithms**: pathfinding
* **CSPs:** Essential for solving structured problems with constraints
* **MDPs and POMDPs**: Decision-making under certainty
* **RL**: Enables AI to learn optimal actions through trial and error

My last key point would be the recognition of decision making and learning. I highlighted how AI is not just about performing tasks, but also about making decisions and improving over time (RL is a good example). I think this better represents modern AI approaches.

**Q2 (30 points) Reinforcement Learning Bandits**

1. **Why are Optimistic Greedy and epsilon-Greedy approaches considered suboptimal? Explain**
2. **Why do we study Optimistic Greedy and epsilon-Greedy if they are suboptimal? Explain**
3. **Compare Upper Confidence Bound and Thompson sampling approaches**

**A2**

Both Optimistic Greedy and -Greedy are considered suboptimal because they do not balance exploration and exploitation effectively, leading to slower convergence to the optimal action.

* **Optimistic Greedy**
  + Assumes high initial values for action rewards, which encourage exploration early on
  + Once an action is selected and its true reward is revealed, the algorithm shifts to exploitation, making it prone to getting stuck in suboptimal choices if the early estimates are misleading
  + It lacks continuous exploration, meaning it may fail to discover better actions over time
* **-Greedy**
  + It introduces randomness by choosing a non-greedy action with probabilityand the greedy action (best known up to that point) with probability 1-
  + While this ensures exploration, it does so in an inefficient manner since it explores uniformly at random, potentially wasting time on poor choices rather than focusing on promising actions
  + The fixed or decaying schedule still leads to suboptimal performance compared to more advanced strategies

**Why do we study Optimistic Greedy and epsilon-Greedy if they are suboptimal? Explain**

They provide a simple introduction to exploration vs exploitation. They illustrate the core challenge in reinforcement learning: balancing new actions (exploration) vs. using known best actions (exploitation). They serve as baselines for other more optimal algorithms (Thompson Sampling, UCB) to evaluate their effectiveness. In simple environments, where computational resources are limited, these methods are pretty easy to implement and may perform well enough. They also drive the development of more advanced techniques.

**Compare Upper Confidence Bound and Thompson sampling approaches**

The Upper Confidence Bound (UCB) and Thompson Sampling are two strategies used for exploration in reinforcement learning, but they go about it in different ways. UCB explores actions based on an upper confidence bound of expected rewards, which is kind of an optimistic approach. On the other hand, Thompson Sampling takes a more probabilistic route, exploring actions based on a posterior probability distribution of rewards.

UCB uses confidence intervals to pick actions that might be the best, while Thompson Sampling relies on Bayesian inference to handle uncertainty in action values. In terms of complexity, UCB needs to keep track of confidence bounds and do some log-based math, while Thompson Sampling must sample from a posterior distribution, which can be more expensive if the priors are complicated. UCB is more deterministic and focuses on reducing uncertainty, while Thompson Sampling adapts more easily, making it better for non-stationary environments.

In practice, UCB works well when rewards follow predictable patterns like Gaussian distributions, but Thompson Sampling tends to do better in real-world situations thanks to its natural exploration. The downside to UCB is that it might be too eager in favoring uncertain actions early on, while Thompson Sampling needs a prior, which might not always be easy to define

**Q3 (30 points) MDPs**

1. **If MAB (Multi-arm Bandits) are similar to MDPs, why not just use MDPs?**
   1. **Explain**
   2. **Give an example verifying the more appropriate use of MABs**
   3. **Would it be more accurate to correlate POMDPs to MABs? Explain**
2. **RL (Reinforcement Learning) has been called a MDP with an unknown model**
   1. **Is this description accurate? Explain**
   2. **Is this description sufficient? Explain why or why not**

**A3**

**If MAB (Multi-arm Bandits) are similar to MDPs, why not just use MDPs?**

Well, while both MDPs and MABs involve decision-making under certainty, they differ in complexity and structure. MDPs consider state transitions and long-term planning, whereas MABs focus on maximizing immediate reward without considering state transitions.  
  
Using both MDPs for MAB problems would introduce this unnecessary complexity, since MDPs require modeling the full transition dynamics and state space. This is unnecessary for problems where the actions we take don’t affect future states.

**Example: Online Ads**

In advertisement, a company X wants to choose the best ad to display on a website to maximize sales/click-through rates. Each ad (arm of a MAB) has an unknown probability of being clicked (reward). There is not need to track transitions between different ”states” since each click event is independent of the previous ones. This fits the MAB model because the problem focuses on choosing the best ad over time without considering the sequential dependencies.

**POMDPs vs MABs**

Yes, but not really. A POMDP extends an MDP by introducing partial observations and a belief state. This means that the agent does not have full knowledge of the environment’s state. In a MAB problem, the agent does not fully know the reward distributions of each arm and must estimate them over time. However, POMDPs still involve state transitions, which MABs don’t.

While MABs share the partial observability aspect of POMDPs, they lack the state-dependent transitions, making them simpler.

**RL (Reinforcement Learning) has been called a MDP with an unknown model**

**Is it accurate?**

It’s somewhat accurate. RL is pretty much an MDP where the transition dynamics and reward function are unknown and must be learned through interaction. In standard MDPs, the transition probabilities and rewards are typically fully known, allowing for planning-based solutions like Value Iteration or Policy Iteration. In RL, since these components are unknown, the agent must use exploration and learning algorithms to approximate an optimal policy.

**Is it sufficient?**

No, it’s too simple in my opinion. While RL is modeled as an MDP with an unknown model, this perspective doesn’t fully capture all RL settings, such as:

* + **Model-Free vs Model-Based RL**
    - Model-Free RL doesn’t attempt to learn the transition model explicitly, instead it directly learns a policy or value function
    - Model-Based RL, on the other hand, tries to approximate the transition model and reward function
  + **POMDPs in RL**
    - In real world applications, the world is hardly ever “fully observable”. This makes the problem more suitable for a POMDP rather than a fully observable MDP.
  + **Exploration vs Exploitation**
    - RL also deals with exploration strategies, such as -greedy, UCB and Thompson Sampling, which are not explicitly considered standard MDP formulations (Bellman)

**Q4 (25 points) Ethics**

**Many consider *bias* unavoidable in Machine Learning**

1. **Define bias**
2. **How can bias be countered?**
3. **Is bias a problem in (PO)MDPs? Explain**
   1. **If so, how**
   2. **If not, why not**
4. **Is bias a problem in Reinforcement Learning? Explain**
   1. **If so, how**
   2. **If not, why not**

**A4**

**Bias**

Bias refers to systematic errors in a model’s predictions due to incorrect assumptions, skewed or even bad training data, or the design of the algorithm itself. It can lead to unfair of inaccurate outcomes.

**How to counter bias**

* We can ensure the training data covers all relevant groups fairly.
* We can use debiasing techniques and enforce fairness constraints (e.g. of hired males and females must be within a certain %).
* We can continuously check for biased outcomes and retrain models if needed
* We can also combine human judgement with AI decisions when necessary

**Is bias a problem in (PO)MDPs?**

In MDPs/POMDPs, if the reward function or state transitions are biased, the learned policy may favor certain actions unfairly. In POMPDs, bias can also come from partial observability, leading to poor decision-making

**Is bias a problem in Reinforcement Learning?**

It can be, yes. If an agent explores some actions more than others, it may develop a skewed policy. If the rewards are designed unfairly or based on biased data, the learned policy will essentially reflect that. On the other hand, if we make sure that RL is properly deigned with diverse data, unbiased rewards and sufficient exploration, bias can be minimized.