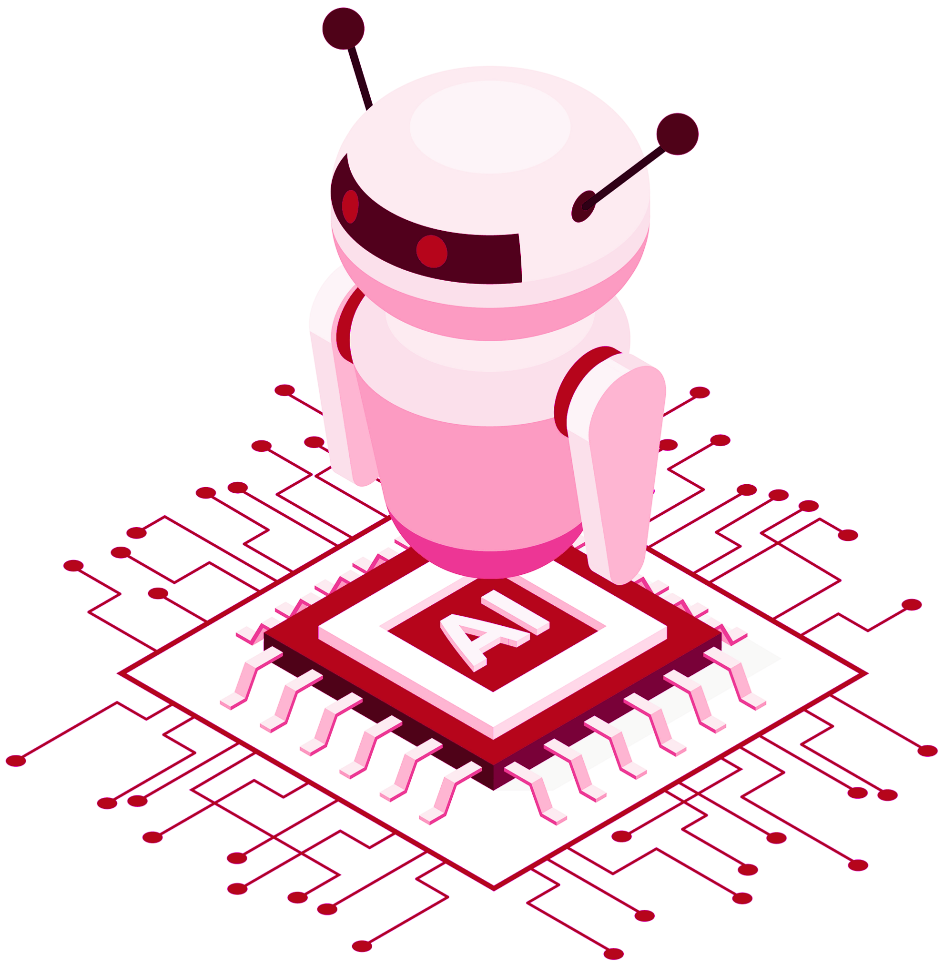

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Homework 6:

Multi-Armed Bandits

By:

Ali Ghasemzade

401106339

Spring 2025

Contents

Grading

The grading will be based on the following criteria, with a total of 105.25 points:

Task	Points
Task 1: Oracle Agent	3.5
Task 2: Random Agent	2
Task 3: Explore-First Agent	5.75
Task 4: UCB Agent	7
Task 5: Epsilon-Greedy Agent	2.25
Task 6: LinUCB Agent	27.5
Task 7: Final Comparison and Analysis	6
Task 8: Final Deep-Dive Questions	41.25
Clarity and Quality of Code	5
Clarity and Quality of Report	5
Bonus 1: Writing your report in Latex	10

1 Task 1: Oracle Agent

- The Oracle uses privileged information to determine the maximum expected reward.
- **TODO:** Compute the oracle reward (2 points).
- Questions:
  - What insight does the oracle reward provide? (0.75 points)
    Answer : The oracle reward shows the upper bound of expected performance if the best arm is always chosen.
  - Why is the oracle considered "cheating"? (0.75 points)
    Answer : It's "cheating" because it uses perfect knowledge of probabilities, which isn't available in real scenarios.

2 Task 2: Random Agent (RndAg)

- This agent selects actions uniformly at random.
- **TODO:** Choose a random action (1 point).
- Questions:
  - Why is its reward lower and highly variable? (0.25 points)
    Answer : Because it picks arms uniformly without exploiting better arms, its average reward is around the mean of all arms and is highly variable.
  - How could the agent be improved without learning? (0.75 points)
    Answer : we can choose bias selection toward arms that are known or suspected (from external info) to be better, or cycle systematically through arms instead of pure random.

3 Task 3: Explore-First Agent (ExpFstAg)

- The agent explores randomly for a fixed number of steps and then exploits the best arm.
- **TODOs:**
  - Update Q-value (3 points)
  - Choose action based on exploration versus exploitation (1 point)
- Questions:
  - Why might a short exploration phase lead to fluctuations? (0.75 points)
    Answer : During the early exploration, random pulls cause large reward swings before the agent settles on best arm.
  - What are the trade-offs of using a fixed exploration phase? (1 point)
    Answer : A fixed exploration window can be too short (missing the best arm) or too long (wasting steps on suboptimal arms).

4 Task 4: UCB Agent (UCB_Ag)

- Uses an exploration bonus to balance learning.
- **TODOs:**
  - Update Q-value (3 points)
  - Compute the exploration bonus (4 points)

5 Task 5: Epsilon-Greedy Agent (EpsGdAg)

- Selects the best-known action with probability $1 - \varepsilon$ and a random action with probability ε.
- **TODO:** Choose a random action based on ε (1 point)
- Questions:
  - Why does a high ε result in lower immediate rewards? (0.5 points)
    Answer : With a high ϵ, the agent often chooses random arms instead of exploiting the best-known arm, reducing immediate reward.
  - What benefits might decaying ε over time offer? (0.75 points)
    Answer : Decaying ϵ over time allows sufficient exploration early on but increases exploitation later, improving long-term performance.

6 Task 6: LinUCB Agent (Contextual Bandits)

- Leverages contextual features using a linear model.
- **TODOs:**
  - Compute UCB for an arm given context (7 points)
  - Update parameters A and b for an arm (7 points)
  - Compute UCB estimates for all arms (7 points)
  - Choose the arm with the highest UCB (4 points)
- Questions:
  - How does LinUCB leverage context to outperform classical methods? (1.25 points)
    Answer : LinUCB uses context features to estimate rewards with a linear model. This helps more generalization from the past expriences in similar contexts and often outperform classical methods that ignore contextual information.
  - What role does the α parameter play in the exploration bonus? (1.25 points)
    Answer : The α parameter scales the exploration bonus. A higher α increases the bonus and encourages more exploration of uncertain arms while lower α make the exploitation better.

7 Task 7: Final Comparison and Analysis

- **Comparison:** UCB vs. Explore-First agents.
  - Under what conditions might an explore-first strategy outperform UCB? (1.25 points)
    Answer : In very short time horizons or rapidly changing environments, a simple fixed exploration phase might outperform UCB if it quickly samples enough to pick the best arm.
  - How do design choices affect short-term vs. long-term performance? (1.25 points)
    Answer : Algorithms with strong exploration can take longer to exploit but often reach better long-term performance, while simpler strategies can yield faster early rewards but they risk missing the true best arm.
- Impact of extending the exploration phase (e.g., 20 vs. 5 steps). (1.5 points total)
  Answer : It overcomes the UCB and find the best arm more quickly and stabilizes at a higher reward sooner than UCB.
- Discussion on why ExpFstAg might sometimes outperform UCB in practice. (2 points)
  Answer : ExpFstAg sometimes wins in practice because it quickly commits to the best option after a short period of exploration. In settings with limited time or steps, its aggressive shift to exploitation can capture rewards faster, while UCB's cautious, ongoing exploration holds it back

early on.

8 Task 8: Final Deep-Dive Questions

- **Finite-Horizon Regret and Asymptotic Guarantees** (4 points)

  Many algorithms (e.g., UCB) are analyzed using asymptotic (long-term) regret bounds. In a finite-horizon scenario (say, 500–1000 steps), explain intuitively why an algorithm that is asymptotically optimal may still yield poor performance. What trade-offs arise between aggressive early exploration and cautious long-term learning? Deep Dive: Discuss how the exploration bonus, tuned for asymptotic behavior, might delay exploitation in finite time, leading to high early regret despite eventual convergence.

  Answer : Altough UCB is optimal in the long run, its high exploration bonus early on delays exploitation and results in high regret over a small finit horizon. we should Balance aggressive early exploration with cautious long-term learning. Since UCB's bonus tuned for asymptotic behavior, even when the best arm is evident it forces extra sampling and means delaying the exploitation and higher early regret while too little exploration risks missing the optimal arm.

- **Hyperparameter Sensitivity and Exploration–Exploitation Balance** (4.5 points)

  Consider the impact of hyperparameters such as ϵ in ϵ-greedy, the exploration constant in UCB, and the α parameter in LinUCB. Explain intuitively how slight mismatches in these parameters can lead to either under-exploration (missing the best arm) or over-exploration (wasting pulls on suboptimal arms). How would you design a self-adaptive mechanism to balance this trade-off in practice? Deep Dive: Provide insight into the "fragility" of these parameters in finite runs and how a meta-algorithm might monitor performance indicators (e.g., variance in rewards) to adjust its exploration dynamically.

  Answer : Small mismatches in parameters can cause under/over exploration missing the best arm or wasting pulls. a self adaptie mechanism would adjust these based on performance.(for example reward variance) A meta-algorithm could monitor metrics like reward stability or Q-value variance to weak exploration dynamically.

- **Context Incorporation and Overfitting in LinUCB** (4 points)

  LinUCB uses context features to estimate arm rewards, assuming a linear relation. Intuitively, why might this linear assumption hurt performance when the true relationship is complex or when the context is high-dimensional and noisy? Under what conditions can adding context lead to worse performance than classical (context-free) UCB? Deep Dive: Discuss the risk of overfitting to noisy or irrelevant features, the curse of dimensionality, and possible mitigation strategies (e.g., dimensionality reduction or regularization).
  Answer : The linear assumption in LinUCB may estimate rewards bad if the true relationship is nonlinear or the context is high-dimensional and noisy, potentially leading to overfitting. In these cases a contex-free UCB might perform better. When many context features are irrelevant or noisy, the linear model can mislead especially with limited data. Dimensionality reduction or regularization can also helps.

- **Adaptive Strategy Selection** (4.25 points)

  Imagine designing a hybrid bandit agent that can switch between an explore-first strategy and UCB based on observed performance. What signals (e.g., variance of reward estimates, stabilization of Q-values, or sudden drops in reward) might indicate that a switch is warranted? Provide an intuitive justification for how and why such a meta-strategy might outperform either strategy alone in a finite-time setting. Deep Dive: Explain the challenges in detecting when exploration is "enough" and how early exploitation might capture transient improvements even if the long-term guarantee favors UCB.
  Answer : signals like high reward variance, unstable Q-values, or sudden drops in performance can show switching between explore-first and UCB might be beneficial. a meta-strategy can capture early transient improvements while having the long-term optimality. By monitoring performance indicators, the agent can determine when enough exploration has occured. changing the strategies when estimates stabilize can combine the quick gains of explore-first with UCB's robust long-term learning.

- **Non-Stationarity and Forgetting Mechanisms** (4 points)

  In non-stationary environments where reward probabilities drift or change abruptly, standard bandit algorithms struggle because they assume stationarity. Intuitively, explain how and why a "forgetting" or discounting mechanism might improve performance. What challenges arise in choosing the right decay rate, and how might it interact with the exploration bonus? Deep Dive: Describe the delicate balance between retaining useful historical information and quickly adapting to new trends, and the potential for "chasing noise" if the decay is too aggressive.
  Answer : A forgetting mechanism lets agent discount old data to quickly to adapt to changing rewards, but choosing the decay rate is challenging because if it is too fast loses valuable info and if it is too slow hampers adaptation. The decay must balance retaining useful history and reacting to new trends, an aggressive decay can make the agent chase noisr and a conservative decay may not reflect changes and this misguides the exploration bonus.

- **Exploration Bonus Calibration in UCB** (3.75 points)

  The UCB algorithm adds a bonus term that decreases with the number of times an arm is pulled. Intuitively, why might a "conservative" (i.e., high) bonus slow down learning—even if it guarantees asymptotic optimality? Under what circumstances might a less conservative bonus be beneficial, and what risks does it carry? Deep Dive: Analyze how a high bonus may force the algorithm to

continue sampling even when an arm's estimated reward is clearly suboptimal, thereby delaying convergence. Conversely, discuss the risk of prematurely discarding an arm if the bonus is too low.
Answer : A high bonus forces contiued sampling of arms even when the optimal arm is found so this delays the convergence, and a less conservative bonus might accelerate learning in clear-cut cases, this risks permaturely dismissing promising arms. The bonus must balance the need for theoritical guarantees with preactical efficiency. If it is too high it make the exploration too long and if it is low the early misestimation can cause the best arm to be overlooked.

- **Exploration Phase Duration in Explore-First Strategies** (4 points)

  In the Explore-First agent (ExpFstAg), how does the choice of a fixed exploration period (e.g., 5 vs. 20 steps) affect the regret and performance variability? Provide a scenario in which a short exploration phase might yield unexpectedly high regret, and another scenario where a longer phase might delay exploitation unnecessarily. Deep Dive: Discuss how the "optimal" exploration duration can depend heavily on the underlying reward distribution's variance and the gap between the best and other arms, and why a one-size-fits-all approach may not work in practice.
  Answer : A short exploration phase might miss the best arm and results in high regret, while a long phase delays exploitation reduces short-term rewards. The optimal duration depends on reward variance and arm gaps. In the environments with small differences between arms, too short exploration phase can lock in a suboptimal arm but in the cases that best arm is clear, extending exploration is unnecessary and increases regret by delaying exploitation.

- **Bayesian vs. Frequentist Approaches in MAB** (4 points)

  Compare the intuition behind Bayesian approaches (such as Thompson Sampling) to frequentist methods (like UCB) in handling uncertainty. Under what conditions might the Bayesian approach yield superior practical performance, and how do the underlying assumptions about prior knowledge influence the exploration–exploitation balance? Deep Dive: Explore the benefits of incorporating prior beliefs and the risk of bias if the prior is mis-specified, as well as how Bayesian updating naturally adjusts the exploration bonus as more data is collected.
  Answer : Bayesian methods leverage prior knowledge and update beliefs, this can cause acceleration in learning if the priors are good, while frequentist methods rely just on the data, and when the prior is accurate the Bayesian performs better. However, mis-specified priors can bias Bayesian method and the bayesian framework naturally adjusts exploration as more data is gathered, whereas UCB maintains a fixed exploration bonus based on count of each observation.

- **Impact of Skewed Reward Distributions** (3.75 points)

  In environments where one arm is significantly better (skewed probabilities), explain intuitively why agents like UCB or ExpFstAg might still struggle to consistently identify and exploit that arm. What role does variance play in these algorithms, and how might the skew exacerbate errors in reward estimation? Deep Dive: Discuss how the variability of rare but high rewards can mislead the agent's estimates and cause prolonged exploration of suboptimal arms.
  Answer : In skewed environments, high variance can cause UCB or ExpFstAg to misjudge arms beacuse of underestimating the best arm due to early noise and leading inconsistent exploitation. Rarely high rewards might be missed or misinterpreted and causes the algorithms to allocate pulls inefficiently. The variability can mask the true performance gap so this makes it harder to consistently exploit the best arm.

- **Designing for High-Dimensional, Sparse Contexts** (5 points)

In contextual bandits where the context is high-dimensional but only a few features are informative, what are the intuitive challenges that arise in using a linear model like LinUCB? How might techniques such as feature selection, regularization, or non-linear function approximation help, and what are the trade-offs involved? Deep Dive: Provide insights into the risks of overfitting versus underfitting, the increased variance in estimates from high-dimensional spaces, and the potential computational costs versus performance gains when moving from a simple linear model to a more complex one.

Answer : In high dimensional contexts with few informative features, LinUCB's linear model risks overfitting noise so techniques like feature selection, regularization or non-linear approximations can improve performance but this addds complexity and computational cost. The trade-off is betwen model simplicty and capturing complex relationships. Overfitting increases variance and hurts performance while aggresive dimensionality reduction might discard useful signals.

At the end some questions were in the notebook but not in the latex template so I answered all the questions in my notebook too. thank you for paying attention.