Ailin Chu

Weihao He

Xiao Qin

MSAI 349-Homework #4

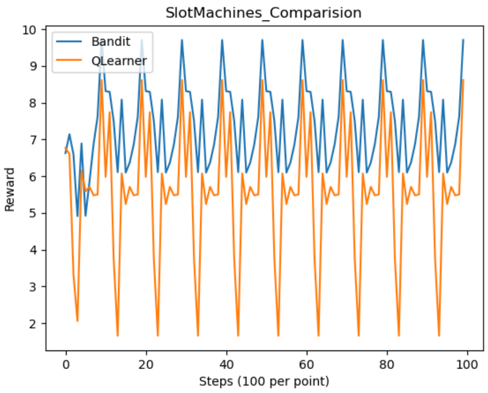
9 December 2024

Reinforcement Learning and Ethics - Homework Report

1 Coding

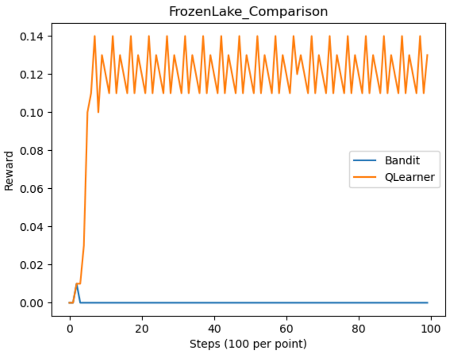
2 Bandits vs. Q-Learning

2.a



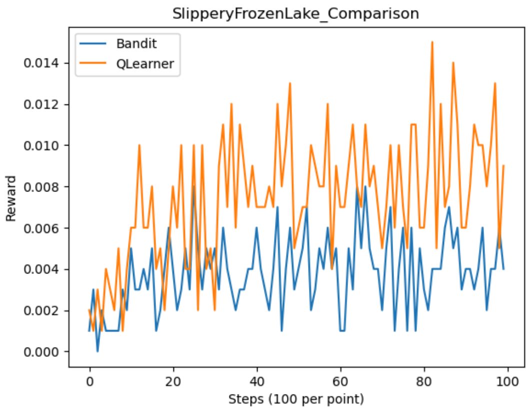
SlotMachines_Comparison.png

The Bandit model consistently achieves higher rewards with stable peaks, while the QLearner exhibits periodic drops to lower reward values, indicating less stability.



FrozenLake_Comparison.png

The QLearner significantly outperforms the Bandit model, achieving a much higher and consistent reward early on, while the Bandit model keeps low reward around zero.



SlipperyFrozenLake_Comparison.png

Notably, the QLearner model achieves higher reward values and shows greater variance compared to the Bandit model across steps.

2.b FrozenLake_Comparison.png. In the FrozenLake plot, Q-Learning consistently receives higher rewards than MultiArmedBandit. This is because Q-Learning is a model-based reinforcement learning algorithm that learns a value function (Q-values) over time. This allows it to make decisions based on accumulated knowledge, which helps it to optimize its actions towards achieving higher rewards, especially in environments with a defined structure like FrozenLake. In contrast, the MultiArmedBandit algorithm lacks this structured exploration and optimization process, as it is more suited for problems where the environment doesn't involve long-term consequences or sequential dependencies, making it less effective in the FrozenLake scenario.

2.c In the FrozenLake environment: MultiArmedBandit performs poorly because it does not account for the sequential nature of the environment and the state-action value learning that Q-Learning utilizes.

Hyperparameter adjustments (such as reducing epsilon to decrease exploration or increasing the learning rate) might slightly improve the MultiArmedBandit performance, but it would still struggle to match Q-Learning. The reason is that Q-Learning uses past experiences to update action-value estimates (Q-values), enabling it to gradually learn the most rewarding actions based on a more sophisticated state-action mapping. In contrast, the MultiArmedBandit is optimized for simpler tasks where actions are independent and not affected by the environment's current state.

Conclusion: No matter how hyperparameters are adjusted, MultiArmedBandit is unlikely to perform as well as Q-Learning in the FrozenLake environment, as the problem's sequential decision-making nature benefits more from Q-Learning's temporal learning approach.

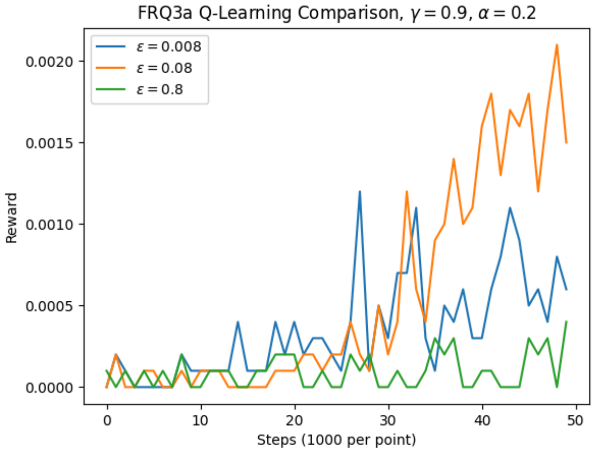
2.d SlotMachines_Comparison.png. In the SlotMachines plot, MultiArmedBandit outperforms Q-Learning across the steps, showing a seasonal reward pattern. This occurs because the MultiArmedBandit algorithm is designed specifically for problems like slot machines, where each "arm" (action) can be pulled independently and has a static probability distribution for rewards. Since there are no long-term state-action dependencies, Q-Learning may struggle to adapt to the relatively simpler environment of slot machines. It might spend unnecessary time exploring non-optimal actions before exploiting them, which leads to suboptimal performance compared to MultiArmedBandit, which is optimized for such bandit-like tasks.

2.e In the SlotMachines environment: To make Q-Learning perform better in the SlotMachines environment, hyperparameters such as epsilon (exploration rate) and alpha (learning rate) could be adjusted. A higher epsilon could lead to more exploration, but this may still not be sufficient to match the simpler, more straightforward performance of the MultiArmedBandit. Also, Q-Learning requires a more complex value iteration process that does not align well with the bandit problem’s simplicity.

Conclusion: Even with optimal hyperparameter adjustments, Q-Learning is unlikely to outperform MultiArmedBandit in this environment. The MultiArmedBandit algorithm is inherently better suited to environments where actions are independent and there is no need for complex state-action value learning. Thus, it is not just a matter of hyperparameters but a mismatch of the algorithm with the environment.

3 Exploration vs. Exploitation

3.a The result plot is as follows:



This code is implementing a **comparison of Q-Learning agents** trained with different values of the exploration parameter on the **FrozenLake-v1** environment, a classic reinforcement learning problem. For each epsilon value (0.008, 0.08, 0.8), it runs multiple trials(10), trains the agents for 50000 steps, and tracks average rewards over time. It then plots the results, showing how exploration affects learning progress and reward collection, and saves the plot as an image file.

3.b

From the plot:

Observation of Trends:

(1) epsilon = 0.008(blue): Shows steady improvement in reward but slower overall growth compared to 0.08

(2) 0.08 (orange) : Achieves the highest reward and grows consistently, suggesting effective exploration and learning.

(3) 0.8 (green): Remains low in rewards with minimal improvement, likely due to excessive exploration and insufficient exploitation.

Best Value of epsilon:

The best epsilon appears to be epsilon = 0.08, as it balances exploration (finding better actions) and exploitation (using learned optimal actions).

Explanation:

(1) Low epsilon (0.008): Too exploitative, limiting the discovery of better strategies.
(2) High epsilon (0.8): Too exploratory, preventing the agent from stabilizing on optimal actions.
(3) Moderate epsilon(0.08): Provides a balance, enabling efficient exploration early in training and effective exploitation later.

3.c

(1) 0.008: The trend would likely continue to improve slowly and eventually converge, as the low exploration rate allows steady but gradual learning.
(2) 0.08: The agent would converge faster and stabilize at a high reward, maintaining its advantage as it balances exploration and exploitation.
(3) 0.8: The agent's performance may remain inconsistent or converge very slowly, as excessive exploration prevents efficient learning of an optimal policy.

3.d The danger of trying 30 or 300 different epsilon values is overfitting to the specific environment. If the chosen epsilon performs well only in the training environment, the agent may fail to generalize to a new domain with different dynamics or challenges. This is because the hyperparameters may be overly relied on some features of the original environment, which would result in poor generalization in other contexts.

4 Tic-Tac-Toe

4.a State: The state of the environment can be represented by a 3x3 board, along with the indication of whose turn it is. A state includes:

Which squares are occupied by X, which squares are occupied by O, and which squares are empty.

Whether it's the agent's turn to place action.

Action: An action consists of choosing one of the empty squares on the board and placing the player's marker (X or O) there.

At the beginning, all 9 squares are empty, so the set of possible actions is to choose any of the 9 squares.

After some moves have been made, the state of the game may have anywhere from 0 up to 8 moves already placed. The available actions are then limited to those squares that remain empty.

Certain states will have no further actions if the game has ended (one player has won/the board is full).

4.b Reward: The reward for winning the game could be +1. This positive reward encourages the agent to find sequences of moves that lead to winning outcomes. Losing the game should result in a reward of -1 to discourage actions that lead to defeat. This negative reward helps the agent avoid strategies that increase the likelihood of losing. A draw should have a reward of 0, as it is a neutral outcome that is neither good nor bad. This prevents over-penalizing ties while still emphasizing the importance of aiming for a win. Non-terminal actions during gameplay should also have a reward of 0. By not providing immediate feedback for intermediate moves, the agent is encouraged to focus on long-term strategies rather than short-sighted actions.

4.c We can introduce a small negative reward for each move the game continues without ending, for example, -0.1 per move. This penalty encourages the agent to achieve a winning state in fewer moves.

5 Fair ML in the Real World

5.a PPV(positive predictive value)=TP/(TP+FP); FPR(false positive rate)=FP/(FP+TN). In a loan decision scenario, a high FPR system means that the system is prone to grant you the decision but you might not be able to repay. A high FNR(false negative rate) system means that the system is prone to deny your request but you might actually be able to repay. Choosing a high FPR system is better than the high FNR system because this system has a higher possibility to grant you a loan decision and lend you money.

5.b PPV(positive predictive value); NPV(negative predictive value)
As a loan applicant, I would prefer high PPV over high NPV. A high PPV system ensures that when I am approved for a loan, the decision is reliable, fair, and based on sound

criteria. However, a system with high NPV may provide a more accurate prediction in denying an applicant who is actually not capable of repaying, which is not relevant to me.

5.c Buolamwini and Gebru make the following recommendations regarding accountability and transparency of machine learning (ML) systems

a) Inclusive Benchmark Datasets: Advocate for increasing phenotypic and demographic representation in training and benchmark datasets to ensure ML systems are evaluated fairly across diverse populations.
b) Subgroup Accuracy Reporting: Emphasize the need for rigorous reporting on algorithmic performance across demographic and phenotypic subgroups.
c) Intersectional Auditing:Highlight the importance of intersectional analysis, which considers overlapping characteristics (e.g. race and gender) to uncover compounded biases that might be hidden in single-attribute evaluations.
d) Active Reduction of Performance Gaps: Recommend that developers actively work to close performance gaps where they arise, ensuring accountability in model improvement and deployment.
e) Transparency in Data Composition: Define transparency as providing clear information about the demographic and phenotypic composition of training and benchmark datasets, enabling stakeholders to understand the context of model evaluations.
f) Extending Beyond Technical Reports: Suggest that accountability and transparency should go beyond just performance metrics and include mechanisms for consent and redress to address potential harms caused by ML systems, though this was not the primary focus of their work.

A disparity in PPV across demographic subgroups signals unequal reliability of positive predictions for different groups.Reporting PPV for subgroups aligns with Buolamwini and Gebru’s recommendation for subgroup accuracy reporting, enabling stakeholders to identify and address biases in prediction reliability.

A high FPR for specific subgroups indicates that individuals from these groups are more likely to be falsely classified as positives. For example, in facial recognition, certain demographics might be disproportionately misidentified. Reporting FPR for subgroups is critical for detecting and addressing over-prediction biases, ensuring fairer treatment across populations.

5.d The analysis conducted by Buolamwini and Gebru is intersectional because it examines algorithmic performance across multiple demographic attributes simultaneously, rather than considering each attribute (e.g., race, gender) in isolation. Specifically, their study evaluates how gender and skin tone together influence the performance of commercial facial analysis systems.

The findings of the intersectional analysis revealed significant disparities in classification accuracy across different gender and skin tone combinations. The highest accuracy was observed for light-skinned men, while the lowest accuracy was for dark-skinned women. Error rates further illustrated this compounding bias, with dark-skinned women being misclassified at rates as high as 34.7%, compared to just 0.8% for light-skinned men. These results highlight the inequitable behavior of commercial facial analysis systems, which disproportionately disadvantage certain demographic groups, particularly those at the intersection of multiple marginalized identities such as gender and skin tone.

5.e In this context, “confounded” refers to a situation where an observed relationship between variables (e.g., gender, skin tone, and classification accuracy) is influenced or distorted by an extraneous factor. Here, the authors are addressing whether the disparities in classification accuracy they observed could be falsely attributed to differences in sensor quality, such as poor image resolution, inconsistent lighting, or pose variations, rather than genuine algorithmic biases.

By stating that their findings are not confounded by sensor quality, the authors mean that the observed disparities in performance across gender and skin tone groups cannot be explained away by differences in image quality or data inconsistencies. Instead, these disparities reflect inherent biases in the algorithm or data used to train it.