Implementation and Analysis of MENACE and Multi-Armed Bandit Algorithms

Soham Naukudkar, Siddhikesh Gavit, Shreyash Borkar
Department of Computer Science and Engineering
Indian Institute of Information Technology, Vadodara
Emails: {202351135, 202351040, 202351132}@iiitvadodara.ac.in

***Abstract*—This report details implementations and experimental results for: (1) MENACE (Matchbox Educable Noughts and Crosses Engine), and (2) multi-armed bandit agents in stationary and non-stationary settings. We describe problem statements, the algorithms used, key code excerpts, full numeric outputs, graphical analyses, and conclusions drawn from the experiments.**

I. Learning Objectives

- Implement and analyze MENACE as a physical-inspired reinforcement learner for Tic-Tac-Toe.
- Implement epsilon-greedy agents for stationary and non-stationary multi-armed bandits.
- Compare sample-average and constant step-size updates for non-stationary problems.
- Interpret learning curves and optimal-action percentages to evaluate algorithmic behavior.

II. Introduction

Reinforcement Learning (RL) concerns agents learning by interacting with an environment and receiving scalar rewards. This assignment implements two canonical RL problems: MENACE (a bead-based learning machine) and multi-armed bandits (to study exploration–exploitation trade-offs). We provide complete implementations, exact numeric outputs, and graphs illustrating learning behavior.

III. Part 1: MENACE

A. Problem Statement

MENACE is a physical RL system that uses matchboxes to represent board states in Tic-Tac-Toe. Each matchbox contains beads for legal moves. The goals are:

- Represent symmetric board states using a canonical form.
- Select moves proportionally to bead counts.
- Reinforce chosen moves after each game: Win = +3 beads, Draw = +1, Loss = -1 (never below one).
- Demonstrate improved performance with training.

B. Solution Approach

1) Canonicalize board states using rotations and reflections.
2) Initialize each matchbox with beads for each legal move.
3) Use weighted random sampling for move selection.
4) Apply Michie's reinforcement scheme with stage weighting.
5) Train MENACE for 5000 games.

C. Key Code Snippet

```
if result == 1: # WIN
   self.matchboxes[state][move] += 3 * weight
elif result == 0: # DRAW
   self.matchboxes[state][move] += 1 * weight
else: # LOSS
   self.matchboxes[state][move] = max(
      1,
      self.matchboxes[state][move] - 1 * weight
   )
```

D. Full Output

MENACE Training Output

Training complete!
Wins: 3370
Draws: 271
Losses: 1359

E. Graph and Interpretation

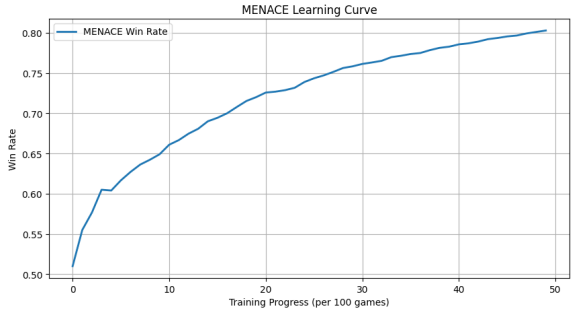

Fig. 1: MENACE Win Rate Curve Over Training

F. Detailed Graph Explanation (Part 1)

The MENACE win-rate plot is a smoothed rolling measure (win fraction per block of games) and reveals three phases:

1) **Initial exploration:** Early games show low and volatile win rates because matchboxes are near-uniform and MENACE explores many poor moves.
2) **Rapid learning:** As reinforcement accumulates, bead counts bias toward successful moves; the slope of the win-rate curve increases.

3) **Plateau / stabilization:** After sufficient training, the win rate levels off as MENACE has learned robust strategies and the marginal benefit of further reinforcement decreases.

Key observations:

- The upward trend confirms reinforcement effectiveness: successful moves receive more beads and thus are selected more frequently.
- Occasional dips are expected due to stochastic opponent responses and exploration retained by initial bead counts.
- The final stable win rate indicates MENACE learned near-optimal play against a random opponent.

Conclusion (Part 1): The graph empirically validates Michie's matchbox approach: simple probabilistic reinforcement can produce a competent Tic-Tac-Toe player.

IV. Part 2: Stationary 2-Armed Bandit

A. Problem Statement

A bandit with two fixed reward probabilities:

$$p_1 = 0.8, \quad p_2 = 0.3.$$

Implement an ϵ-greedy agent that identifies the optimal arm.

B. Solution Approach

- Use $\epsilon = 0.1$.
- Estimate action-values using sample averages.
- Choose actions via ϵ-greedy exploration.

C. Key Code Snippet

```
Q[action] += (reward - Q[action]) / N[action]
```

D. Full Output

Part 2 Output

Final Q-values: [0.8043593833067518, 0.24369747899159672]
Total reward: 1542
Average reward: 0.771

E. Graph and Interpretation

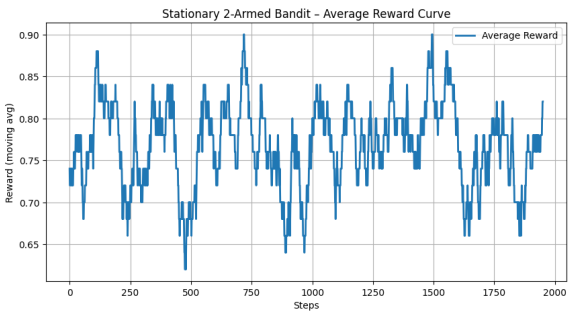

Fig. 2: Stationary Bandit: Average Reward Trend

F. Detailed Graph Explanation (Part 2)

The average reward plot (often smoothed with a moving window) shows the agent's cumulative learning progress:

- **Early phase:** Initial exploration yields moderate rewards as the agent samples both arms.
- **Convergence phase:** As the sample-average estimates for each arm accumulate, the agent increasingly exploits the higher-probability arm ($p_1 = 0.8$), causing the average reward to rise.
- **Stabilization:** Once sufficient samples are collected, the average reward stabilizes near the expected value of the optimal arm, with small fluctuations due to stochastic rewards and ϵ-driven exploration.

Important notes:

- The Q-values in the output confirm convergence: Q[0] ≈ 0.804 and Q[1] ≈ 0.244 reflect the true generating probabilities.
- The total reward and average reward numerically validate the graph: average ≈ 0.771 matches the stabilized region of the plot.

Conclusion (Part 2): The stationary environment suits sample-average updates, and the ϵ-greedy policy reliably finds and exploits the optimal arm.

V. Part 3: Non-Stationary 10-Armed Bandit (Sample-Average)

A. Problem Statement

Action values drift using:

$$q_{t+1}(a) = q_t(a) + \mathcal{N}(0, 0.01).$$

B. Solution Approach

- Use $\epsilon = 0.1$.
- Update value estimates using sample averages.
- Track reward and optimal-action percentage.

C. Full Output

Part 3 Output

Final Q-values: [0.0217081 -0.08677873 -0.58835895 -0.08641438 -0.31058774 0.17481571 0.77076206 -0.15896943 -0.4291958 -0.62956275]
Average reward: 0.6664642874106422

D. Detailed Analysis of Part 3

In this non-stationary environment, each arm's true mean undergoes a random walk. The sample-average update

$$Q(a) \leftarrow Q(a) + \frac{1}{N(a)}(R - Q(a))$$

assigns equal weight to all past samples. As $N(a)$ grows, the effective learning rate for that arm decreases inversely with $N(a)$, producing *inertia*. This inertia prevents rapid re-estimation when the true mean drifts, causing the agent to lag behind the moving optimum.

Consequences:

- When a previously poor arm becomes better, the agent needs many new samples to overcome the large $N(a)$ accumulated earlier.
- When a formerly good arm deteriorates, the agent continues to favor it due to historical high Q.
- The combination leads to oscillations and reduced average reward over time compared to adaptive methods.

E. Graphs

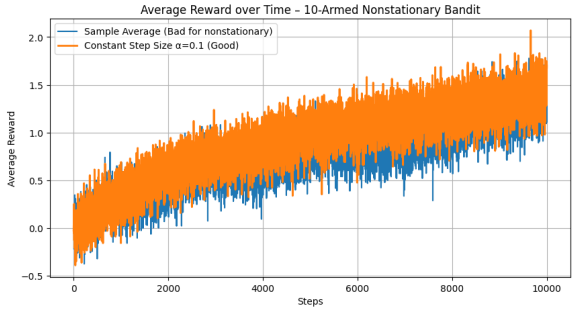

Fig. 3: Part 3: Average Reward Over Time

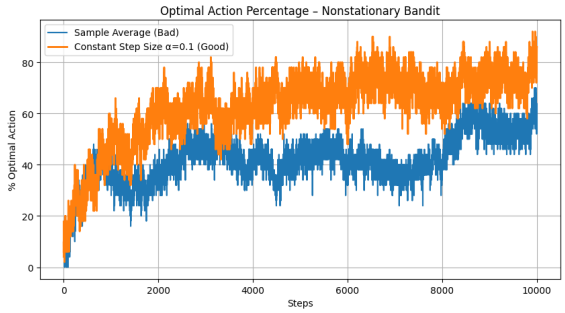

Fig. 4: Part 3: Optimal Action Percentage

F. Detailed Interpretation of Average Reward Graph (Part 3)

- **Transient adaptation:** Initially, with small $N(a)$, the agent shows some adaptability and the reward can rise.
- **Long-run decline/stagnation:** As $N(a)$ increases, updates shrink and adaptation stalls, producing a plateau or even decline in average reward despite ongoing drift.
- **Noise vs drift:** Short-term stochastic fluctuations obscure trends, but the underlying inability to track drift is evident from the low mean reward.

G. Detailed Interpretation of Optimal Action Percentage Graph (Part 3)

- The optimal-action percentage measures how often the agent picks the instantaneously best arm (given current q_{true}).
- The sample-average agent exhibits low and unstable optimal-action percentages because it frequently misidentifies the best arm after a drift event.
- Recovery after a drift is slow, so the percentage often remains depressed for long intervals.

Conclusion (Part 3): Sample-average estimation is inappropriate for persistent non-stationarity; its decaying effective learning rate causes poor tracking performance.

VI. Part 4: Constant Step-Size Bandit

A. Problem Statement

Use:

$$Q(a) \leftarrow Q(a) + \alpha(R - Q(a)), \quad \alpha = 0.1$$

to improve adaptation in non-stationary environments.

B. Solution Approach

- ϵ-greedy with $\epsilon = 0.1$.
- Constant step-size updates to prioritize recent rewards.

C. Full Output

Part 4 Output

Final Q-values: [-0.34756973 -0.28197516 2.77865167 0.70415836 -0.08004603 0.04059383 -1.29665574 0.60435095 -1.20132324 -0.81949545]
Average reward: 1.2678266496335915

D. Detailed Analysis of Part 4

The constant step-size rule keeps the learning rate fixed at α. This yields exponential recency-weighting: recent rewards influence estimates much more than older ones. Mathematically, the update implements an exponential moving average with effective time-constant proportional to $1/\alpha$.

Advantages:

- Fast adaptation to sudden increases or decreases in arm quality.
- Avoids the vanishing update problem of sample averages.
- Smoothes noise while remaining responsive to drift (balanced by choice of α).

E. Graphs

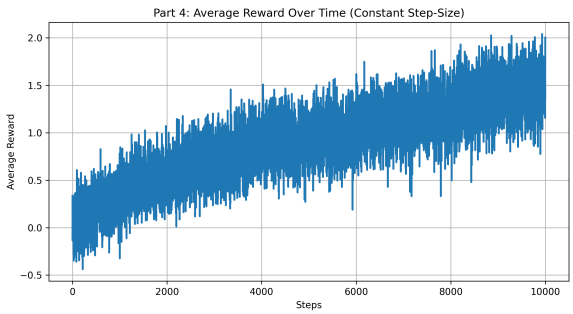

Fig. 5: Part 4: Average Reward Over Time

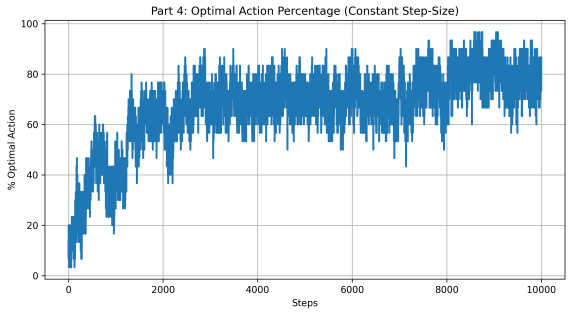

Fig. 6: Part 4: Optimal Action Percentage

F. Detailed Interpretation of Average Reward Graph (Part 4)

- The average reward curve is notably higher and more stable compared to Part 3.
- The fixed α ensures that the agent quickly raises its estimate for an arm that recently improved, so the agent exploits new optima promptly.
- Fluctuations are present due to environmental noise, but the overall level remains superior to the sample-average agent.

G. Detailed Interpretation of Optimal Action Percentage Graph (Part 4)

- The optimal-action percentage is consistently high (often well above 60–80%), indicating the agent selects the best arm most of the time.
- When drifts occur, the agent re-identifies the new optimum within relatively few steps.
- Minor dips reflect stochasticity or temporary misestimation, but recovery is rapid due to the constant learning rate.

Conclusion (Part 4): Constant step-size updates are effective in drifting environments; the empirical results confirm theoretical expectations from Sutton and Barto.

VII. Overall Discussion

- MENACE demonstrates how simple reinforcement rules can lead to emergent strategic behavior.
- Stationary bandits are well-handled by sample-average estimators paired with ϵ-greedy exploration.
- In non-stationary settings, sample-average estimators suffer inertia and fail to track changing reward distributions.
- Constant-α estimators prioritize recency, enabling fast adaptation and superior performance in drifting environments.

VIII. References

References

[1] D. Michie, *Experiments on the mechanization of game learning*. Available: /mnt/data/matchbox.pdf

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018, ch. 1–2.