Model-Based and Model-Free Markov Decision Processes for Small and Large state problems

Umesh Jadhav
OMSCS
Georgia Institute of Technology
Georgia, USA
ujadhav6@gatech.edu

***Abstract*—Reinforcement Learning is a sub-domain of AI and ML where an agent learns to work in a random and dynamic environment without predefined knowledge and an objective of achieving maximum rewards. The agent perceives its actions to be positive when it earns a reward whereas a penalty attributing an inappropriate action. Markov Decision Processes(MDP) is a probabilistic decision making model of dynamic system where the outcomes are random or influenced by the sequential actions of an agent taken over time. In this study, two model based techniques, Value Iteration(VI) and Policy Iteration(PI) along with Q-Learning(QL) as the model free algorithm are evaluated for their performance over Cartpole MDP and Blackjack MDP. Here, Cartpole with a discretized state space of 23000 states represents large state MDP whereas, Blackjack with 290 state space constituting small state MDP. For performance, VI and PI are analyzed for policy convergence leveraging mean state values over various iterations whereas, delta convergence of Q-Learning showcasing the policy convergence. Additionally, the trade-off of exploration and exploitation is studied through cumulative rewards over episodes for various gamma values for the three agents and epsilon values specifically for Q-Learning. Finally, the effect of changes in state space of Cartpole is studied in terms of Wall Clock Time. For Cartpole, it is found that Policy Iteration converges to optimal policy in critically lower iterations than that of Value Iteration and Q-Learning. Model based algorithms illustrate similar cumulative rewards though Q-Learning achieving tremendously higher cumulative rewards. In case of Blackjack, Policy Iteration converges to optimal policy in lower iterations than that of Value Iteration and Q-Learning whereas, the cumulative rewards indicate Q-Learning out-performing the model based algorithms.**

I. INTRODUCTION

Reinforcement learning is a branch of AI and ML where an agent trains to perform its task without any supervision by determining the optimal actions for reaching its goal through rewards and penalties with the main aim of accumulating maximum rewards till the task is accomplished. Markov Decision Processes is decision making model where the environment is ever changing and probabilistic in nature whose outcomes are driven randomly or through the decisions of an agent. There are five main components in Reinforcement learning namely, states, actions, environment, rewards and an agent. Here, the agent performs in an environment represented by a set of states changing over the course of agent's actions. The actions define the probabilistic movement of agent in the states of the environment. While learning process, the agent follows a policy that probabilistically maps the actions to states where the estimated rewards are maximum.

To determine the optimal policy, Value Iteration(VI) and Policy Iteration(PI) algorithms are implemented for training the agent. First, VI works by iteratively optimizing the value of states before converging to a maximum value known as the optimal value function $V(s)$ which provides the expected returns generated by following this policy. Here, the value of state means the expected returns the agent would receive if it follows a policy from the current state till the end. On the other hand, Policy Iteration alternate between policy evaluation and policy improvement. In policy evaluation, the chosen policy is evaluated for its value function $V(s)$ for the current state using Bellman's Equation[1] whereas, policy improvement follows a greedy approach with respect to the current state for determining the action that will maximize the expected return. The policy in case of Policy Iteration is said to be converged when the current policy and the previous policy matches.

In case of Q-Learning, the agent maintains a Q-table of state-action pair and each cell of the table known as Q-value specifies the expected return for that state action pair. Here, the learning rate(α) determines the precedence of new information over the old information ranging between 0 and 1. Similarly, the discount factor(γ) indicate the importance of immediate reward over future rewards. Since the technique follows ϵ-greedy strategy, the trade-off between exploration and exploitation can be seen where exploration indicate performing random actions and exploitation indicating choosing maximum q-valued action to perform the task. The policy converges when the α is decayed over time and each state-action pair is visited.

In this study, Value Iteration(VI), Policy Iteration(PI) and Q-Learning techniques are implemented over Cartpole MDP and Blackjack MDP to study their performance on small state and large state MDPs. These algorithms are analyzed for policy convergence using Mean State value($V(s)$) over various iterations for VI and PI while Q-Learning evaluated through mean delta convergence(change in Q-values) over iterations. Additionally, the exploration and exploitation trade-off is studied by visualizing cumulative rewards over various episodes of the agents run for the three algorithms. Finally, the change in state space for Cartpole is leveraged to showcase its effect on runtime of the algorithms.

II. Problem Description and Hypothesis

Cartpole problem is a MDP problem where the goal is keep an inverted pendulum(pole) upright over the cart. To achieve this, the cart can be moved left and right and each time the pole is upright, a reward of 1 is achieved. The episode is considered over when the pole falls below a certain threshold angle. This problem is continuous state problem as although the goal is achieved, the episode continues with the only termination being the falling of the pole hence, the problem is discretized.

Since the discretization of Cartpole problem leads to formation of large state space, Value Iteration will have lower runtime as that of Policy Iteration since it updates $V(s)$ iteratively as opposed to performing policy exploration and policy improvement in case of Policy Iteration. However, Q-Learning would work relatively faster than both the model based algorithms since it updates q-values and chooses action with highest q-value. Further, Policy Iteration would see a quicker convergence since it iteratively performs policy improvement along with policy exploration as opposed to Value Iteration which updates $V(s)$ iteratively and Q-Learning since it does not have prior knowledge of the environment. As a result, Q-Learning would not be affected with the increased state space; outperforming the model based algorithms for wall clock time in the process. However, given the internal operations, Value Iteration would perform relatively faster than that of Policy Iteration for various state sizes.

Blackjack problem is a multiplayer MDP comprising of a dealer and a player in which, the goal is to beat the dealer's sum of value of cards with the player's sum of value of cards without exceeding the sum of 21. The game begins with 2 cards each for the player and the dealer where the player face ups both th cards and only gets to see one of the dealer's card. If the player cards add up to exactly 21, the player wins 150% of the bet and the episode automatically ends. Whereas, if the player's cards do not sum up to 21, the player can call hit and deal more cards else, if the player sticks, the dealer turns the second card and if the sum of dealer's card value exceeds that of the player's, the episode ends.

Since Blackjack is not a continuous state MDP, there are definite 290 states for this model. Since, it is a small state MDP, the model based algorithms are expected to show similar convergence than that of Q-Learning. However, given the absence of knowledge such as transition probabilities and reward function, Q-Learning is expected to take longer iterations to converge as that of the model-based algorithms. Also, various gamma values are expected to follow similar trend of convergence for Mean $V(s)$ of model free algorithms.

III. Methodology

The entire experimentation is done using Python programming language with bettermdptools[2] as the core library for implementing all the algorithms. For visualization purposes, matplotlib library is leveraged with an additional usage of pandas and numpy. The experimentation methodology for both the MDP problems follow similar structure and is discussed sequentially. Here, a single seed of 29 is used throughout the experimentation.

A. Gamma(γ) Tuning

To study optimal γ values, Mean V of VI, PI and QL are tested for γ values between 0.1 and 0.99 for 1000 iterations for Cartpole and 200 iterations for Blackjack MDP, extracted from v_track, p_track and q_track for VI, PI and QL respectively.

B. Convergence

The model based algorithms are evaluated for Mean/Max $V(s)$ for $\gamma = 0.9$ whereas, for Q-Learning, the convergence is studied through Delta Q(δQ) with $\gamma = 0.9$ and $\epsilon = 0.1$, extracted from Q-track.

C. Exploration-Exploitation trade-off

The exploration-exploitation trade-off in both the MDP problems is studied by computing the cumulative sum of rewards by testing the policy using test_env() method over ϵ values between 0.1 and 0.9 for 1000 episodes in case of Cartpole and 200 episodes for Blackjack MDP.

D. Wall Clock Time

To study the effect of various state spaces on the working of model based and model free techniques, wall clock time is evaluated iteratively for Cartpole MDP for bin size ranging between 10 and 100 with each model training for 1000 iterations. The limitation of bettermdptools do not allow modifying the states for Blackjack problem, hence it is not implemented in this study.

E. Optimal Policy

The optimal policy visualization allows to interpret the appropriate actions taken by the agent to accomplish the task. This is achieved by heatmap for Blackjack MDP since it is a small state problem however, due to the large state space, Cartpole MDP the optimal policy is showcased through cumulative rewards.

IV. Results

Please note here that, all the time related metrics may vary depending upon the system configurations.

A. Cartpole

1) Gamma(γ) Tuning

   Figure 1 illustrates Mean $V(s)$ of Value Iteration for various gamma values over 1000 iterations. It is seen that all the gamma values showcase proportional mean v though, $\gamma = 0.99$ requires additional iterations to converge to 0. From the graph, it is evident that all the values begin with a higher value while maintaining the momentum for few iterations before falling to the Mean V of 0.

   For Policy Iteration, all the γ values converge to 0 in a lesser iterations than that of Value Iteration, seen in figure 2. Here, $\gamma = 0.99$ takes 30 iterations approximately to converge to optimal policy.

As for Q-Learning, the γ values shows convergence(stable lines) for varied number of iterations visible from figure 3. Here, γ from range 0.1 to 0.5 showcase convergence before 100 iterations however, as gamma exceed 0.5, the iterations to converge rise tremendously.

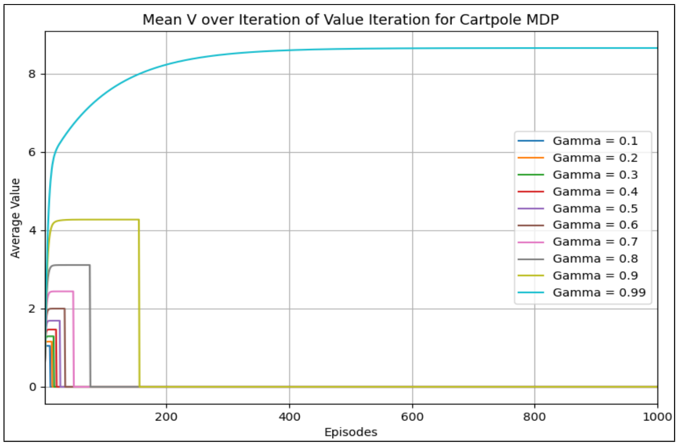

Fig. 1: Mean V of Value Iteration for various Gammas of Cartpole MDP.

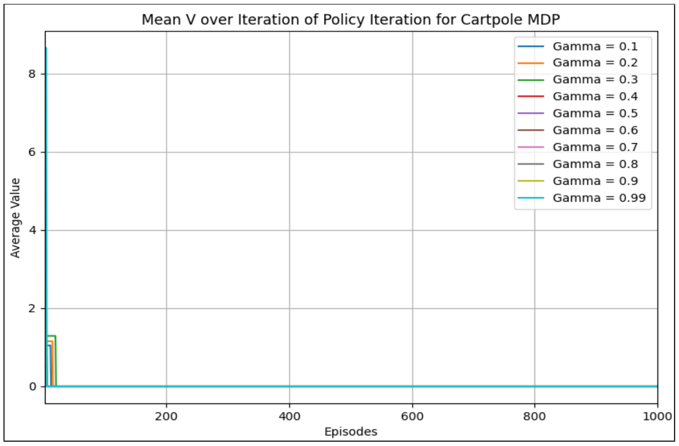

Fig. 2: Mean V of Policy Iteration for various Gammas of Cartpole MDP.

2) Convergence
From figure 4, max V of Value Iteration stabilizes with a value of 5 at approximately 100 iterations whereas, mean V converges at same iterations with the value of 3.2.
In case of Policy Iteration, as evident from figure 5, the Mean as well as max value converges to 0 in 4 iterations approximately. Max V attains a value of 100 whereas, mean V reaches the peak of 10, same as that of Value Iteration. For Q-Learning, lower γ such as 0.1, 0.2 converges faster than that of the higher γ values of 0.9 and 0.99, seen in figure 6. Also, during the early stages of the experiment, high fluctuations are evident specifying the exploration performed by the agent. However, all the γ values converges to 0 with γ=0.99 taking around 400 iterations while γ=0.1 doing so in around 100 iterations.

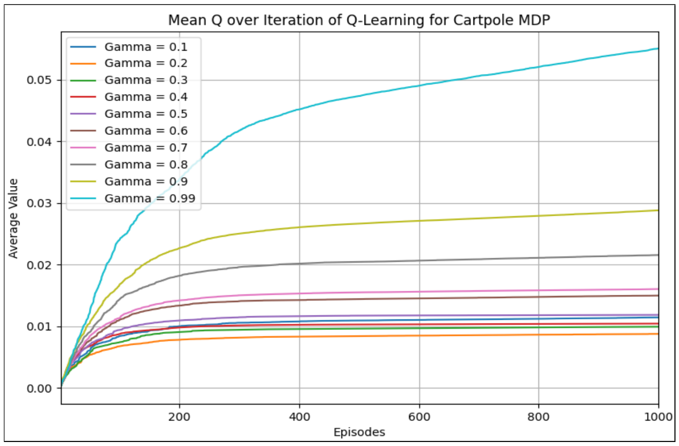

Fig. 3: Mean Q of Q-Learning for various Gammas of Cartpole MDP.

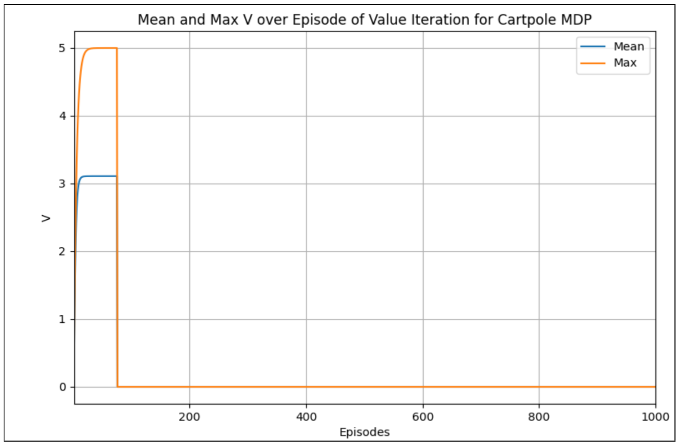

Fig. 4: Mean/Max convergence of Value Iteration of Cartpole MDP.

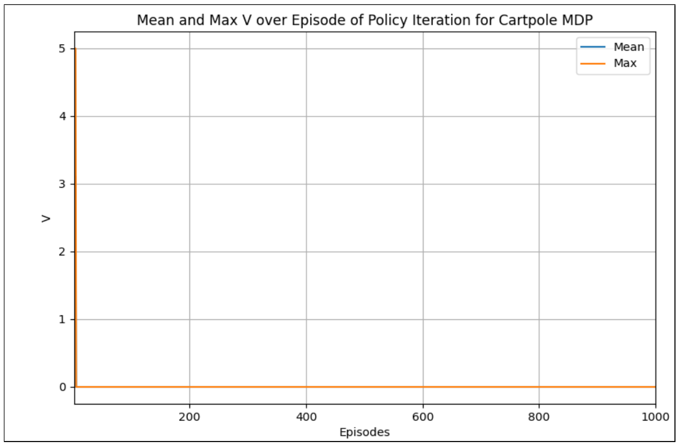

Fig. 5: Mean/Max convergence of Value Iteration of Cartpole MDP.

3) Exploration-Exploitation trade-off
In figure 7, the cumulative rewards for $\epsilon = 0.3$ showcase highest cumulative rewards where the trend stabilizes at

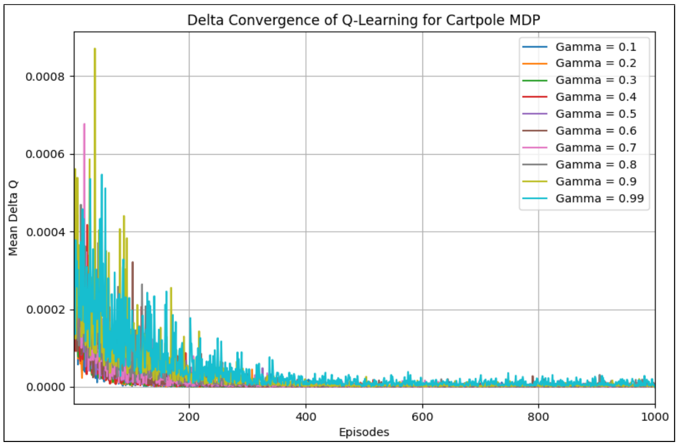

Fig. 6: Delta convergence of Q-Learning for various Gamma of Cartpole MDP.

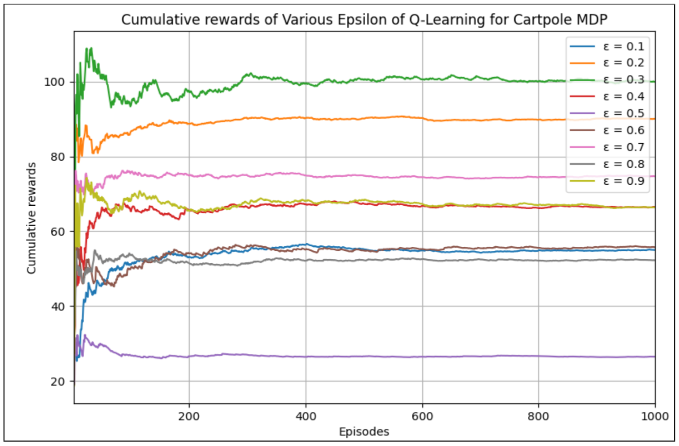

Fig. 7: Cumulative rewards of Q-Learning for various Epsilons of Cartpole MDP.

around 400 iterations. On the other hand, $\epsilon = 0.1$ quickly rises to its highest cumulative reward of 58 before stabilizing for the further iterations. Here, the smaller ϵ values showcase fewer spikes as compared to that of the larger ϵ values.

4) Wall Clock Time
From figure 8, it is evident that Policy Iteration takes the highest runtime over all state sizes from the beginning, reaching the runtime of 547 seconds for 100 position bins. Whereas, Value Iteration showcases a linear growth in runtime, starting 39 seconds for default(10) position bins and ending at 400 seconds for the final state. However, a constant momentum in the trend is seen in case of Q-Learning irrespective of the number of position bins of the cartpole where, the runtime ranges between 6 to 10 seconds throughout the experimentation.

5) Optimal Policy
From figure 9, it is clear that the optimal policy followed by both the model-based algorithms is identical. Here the cumulative rewards initially showcase high fluctuations indicating the exploration of the agent before rising after 50 iterations. In this case the cumulative rewards reach the maximum height of 9.37 at 600 iterations before stabilizing thereafter. However, the Q- Learning agent accumulates higher rewards than that of the model-based algorithms, reaching a maximum reward value of 70 seen in figure 10. As seen in case of model-based algorithms, initial fluctuations are visible in case of Q Learning indicating the initial exploration of the agent.

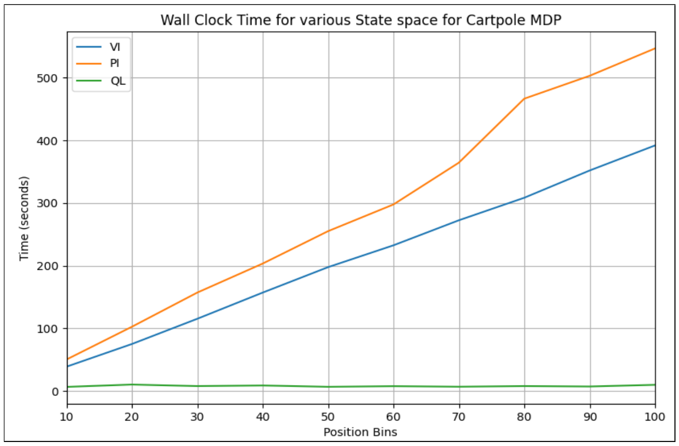

Fig. 8: Wall Clock Time for various position bins of Cartpole MDP.

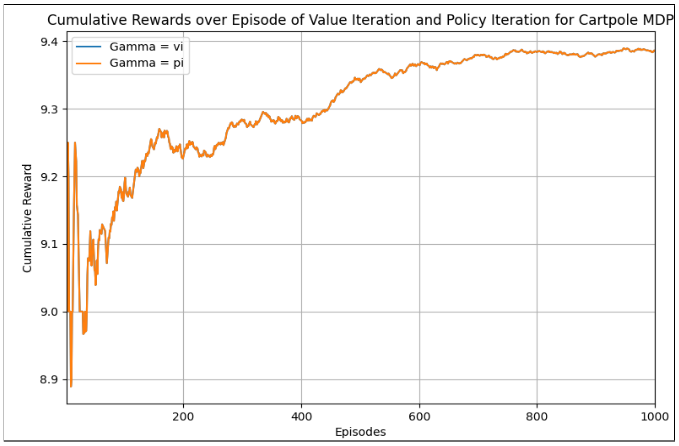

Fig. 9: Cumulative rewards for Value Iteration and Policy Iteration of Cartpole MDP.

B. Blackjack

1) Gamma Tuning
For Gamma tuning of Value Iteration from figure 11, all the γ values shows an instant increase in the Mean V value, indicating that the algorithm is learning optimal state-action values, reaching the heights of 0.10. Further, higher γ values tends to converge relatively slower than that of the lower values. For instance, $\gamma = 0.99$ showcase a higher dip in the value and the lowest level plateau at 0.08 before

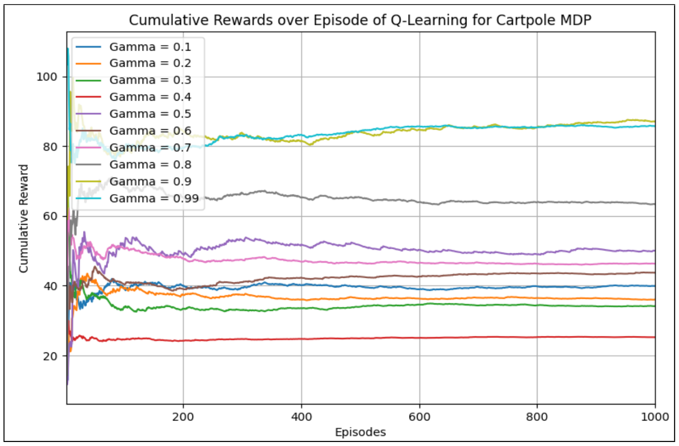

Fig. 10: Cumulative rewards for Q Learning of Cartpole MDP.

converging at 12th iteration whereas, $\gamma = 0.1$ stays at the peak and converges at 10th iteration.

In case of Policy Iteration, all the γ values showcase an overlapping trend however, all of them converging to the optimal policy. However, from figure 12, it is seen that as γ increases, the mean V decreases, establishing an inverse relationship between the two. For instance, $\gamma = 0.1$ reaches mean V of 0.1 whereas, $\gamma = 0.99$ is able to rise to the mean value of 0.08.

On the other hand, from figure 13, the mean Q of Q-Learning conveys that the gamma values take longer iterations to converge to the optimal policy. During the initial iterations, a mix of spikes and troughs can be seen before stabilizing for all the γ values. Here, $\gamma = 0.99$ attains lowest Mean Q value of -0.012 whereas, $\gamma = 0.1$ running at the Mean Q value of -0.012. However, all the γ values converges at approximately 75 iterations.

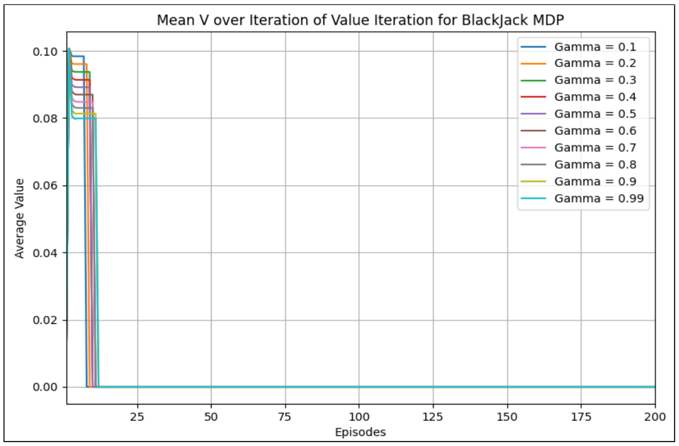

Fig. 11: Mean V of Value Iteration for various Gamma for Blackjack.

2) Convergence

From figure 14, it is evident that the Mean V and Max V lines rises quickly where the former reaches the maximum value of 1.0 while the latter attaining the highest value of 0.95. However, both the lines settle to 0 value after 10 iterations, conveying that Value Iteration algorithm has converged to the optimal policy.

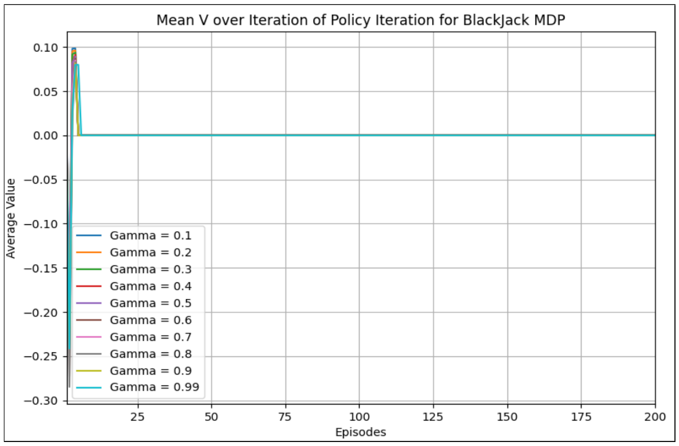

Fig. 12: Mean V of Policy Iteration for various Gamma for Blackjack.

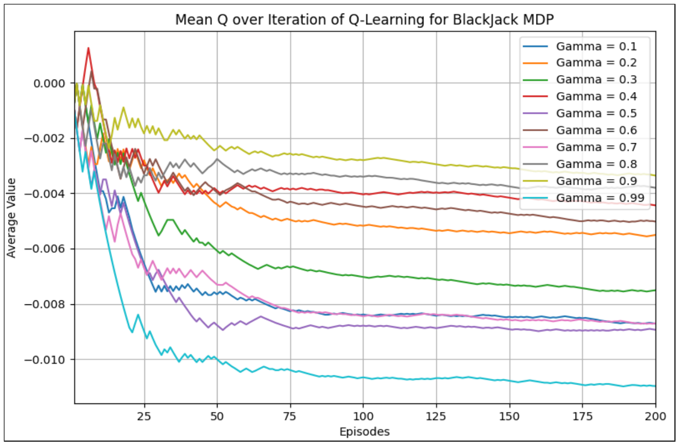

Fig. 13: Mean Q of Q-Learning for various Gamma for Blackjack.

In case of Policy Iteration, figure 15 indicate a different behavior in the initial values of Mean V and Max V lines. Here, the Max V showcase a similar trend as that of Value Iteration, reaching the maximum value of 1.0. However, Mean V starts with negative trend before swiftly rising the neutral value. Finally, both the trends sees a similar convergence point at around 6th iteration.

As for convergence in case of Q-Learning, the trend for all the γ values begins with high mean delta Q values and gradually stabilizing at 0.000 point in figure 16. Here, the higher γ values showcase higher fluctuations such as seen in case of $\gamma = 0.8$ and $\gamma = 0.99$ whereas, γ values of 0.1, 0.2 and 0.3 displaying lower degrees of spikes in the trend. However, as the agent is trained over the iterations, the fluctuations appear to be diminished to a large amount and the convergence can be considered at 100th iteration.

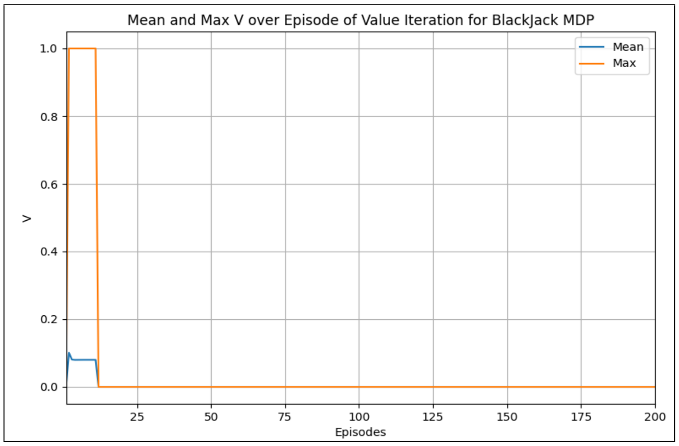

Fig. 14: Mean/Max convergence of Value Iteration of Blackjack.

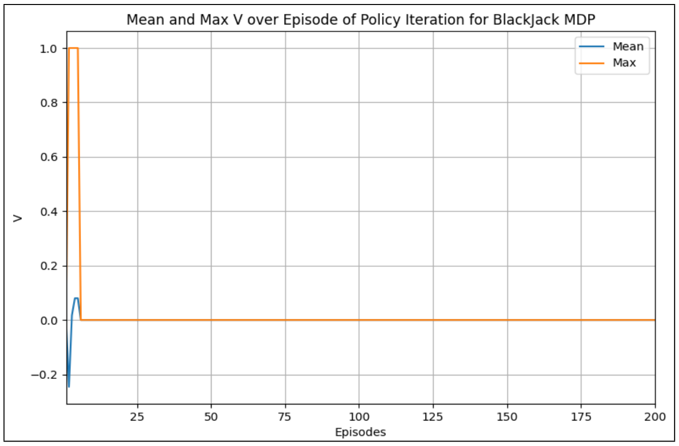

Fig. 15: Mean/Max convergence of Policy Iteration of Blackjack.

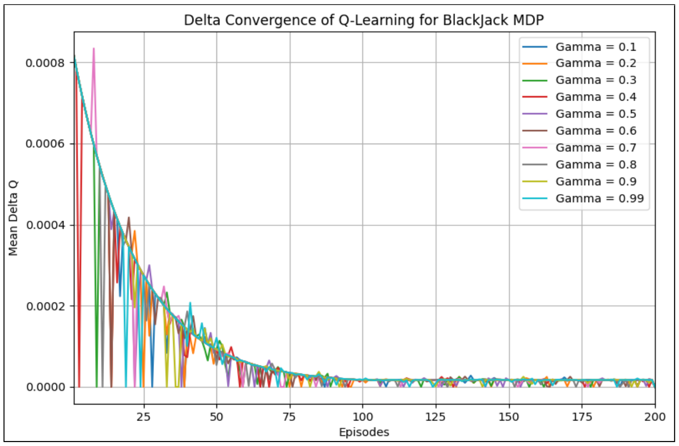

Fig. 16: Delta convergence of Q-Learning of Blackjack.

3) Exploration-Exploitation trade-off

From figure 17, it is seen that higher initially all the ϵ values illustrate higher volatility in rewards indicating the exploration performed by the agent. The lower ϵ values such as 0.1, 0.2 and 0.3 begins with negative cumulative rewards whereas, the higher ϵ values showcasing the opposite behavior. The agent however accumulates negative rewards during the long run although being trained for 200 iterations.

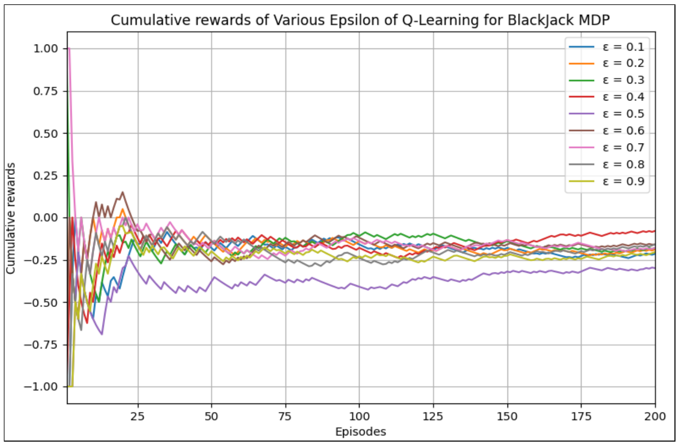

Fig. 17: Cumulative rewards of Q-Learning for various ϵ of Blackjack.

4) Optimal Policy

From figure 18 and figure 19, the optimal policy of Value Iteration and Policy Iteration respectively are found to be identical. In both the cases, the optimal policy suggest for the player to purely perform ”Hit” for hand value below 9 though $V(s)$ is low. However, as the player's hand value increases, the effect of combination of ”hit” and ”stick” is evident as confirmed by greater value of $V(s)$. As the game progresses, the policy once again suggest to perform ”hit” between the hand value of 18 and 24. Finally, exceeding the hand value of 24, the policy is to only perform stick to beat the dealer's hand value.

In case of Q-Learning from figure 20, the maximum $V(s)$ is only 0.5 which conveys the shortcoming of this agent. In figure 18, it is clearly seen that the optimal policy suggest higher number of ”stick” than that of ”hit” with only 3 values attaining maximum $V(s)$ of 0.5. During the initial hand values, a mix of ”stick” and ”hit” can be seen however, as the hand value increases, the game is seen to be dominated by ”stick” actions so much that the last 5 hand values(24 to 29) completely suggest the player to ”stick”.

V. Discussion

A. Cartpole

For Cartpole MDP, comparing the effect of discount factor γ within the model based algorithms illustrate the clear working of Value Iteration and Policy Iteration. Here the idea is that the lower γ values stating the lower importance to the future rewards, the convergence should be earlier than that of the higher γ values that is, γ is directly proportional to number of iterations. As Value Iteration iteratively updates $V(s)$ and determines the optimal value function as opposed to Policy Iteration which explores the policy while tuning it iteratively,

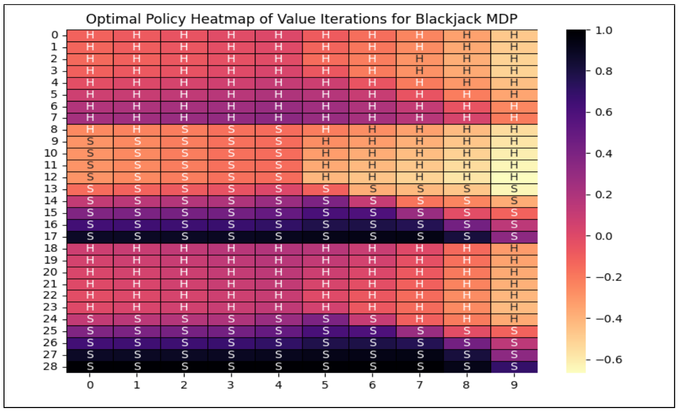

Fig. 18: Optimal Policy Map of Value Iteration of Blackjack.

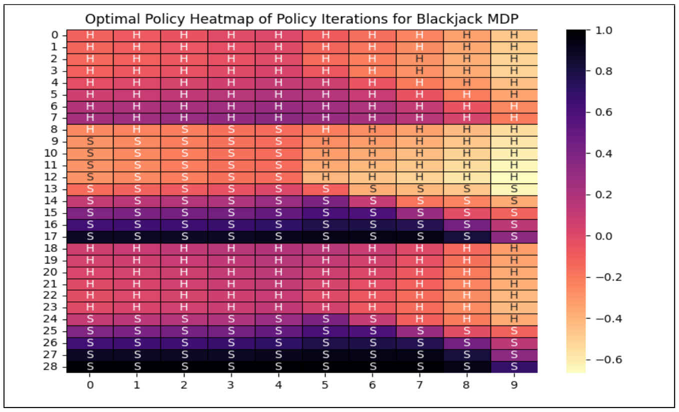

Fig. 19: Optimal Policy Map of Policy Iteration of Blackjack.

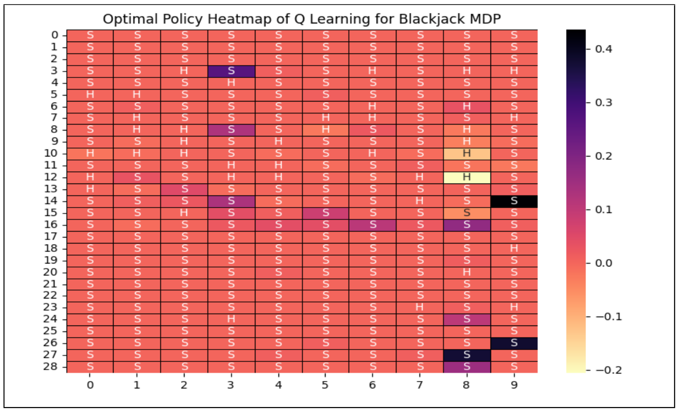

Fig. 20: Optimal Policy Map of Q-Learning of Blackjack.

the former takes longer to converge as that of the latter. This behaviour is spontaneously showcased in figure 1 and figure 2 for Value Iteration and Policy Iteration respectively. An interesting point here is the behaviour of both the algorithms when $\gamma = 0.99$, where Value Iteration exceeds the mark of 1000 iterations for convergence while Policy Iteration does so in approximately 20 iterations. This is due to the fact that large γ values requires changes in distant future rewards take longer to influence current state values, requiring more iterations to propagate these changes through the entire state space.

On the other hand, Q-Learning also looks to be affected by γ values, showcasing the evidence of proportionality between the γ and number of iterations. As seen in case of model-based algorithms, for Q-Learning when $\gamma = 0.99$, the agent converges to the optimal policy, though the number of iterations are greater than that of Policy Iteration. This is due to the fact that the agent determines the Q values of all the state-action pair that is explore the environment and then extracts the pair with highest q values from every state. Also, the continuous state space transformed to the discrete states space enables Policy Iteration to perform better as the environment in terms of transition probabilities is well known which is absent in case of Q-Learning.

Further, the Mean/Max V for model-based algorithms and Mean Delta Q for model-free algorithms from figure 4, figure 5 and figure 6 respectively conveys similar behaviour as discussed. Here, the mean V and max V of Value Iteration showcase longer iterations for convergence due to the inherent behaviour of the algorithm discussed. Similarly, Policy Iteration's mean V and max V converges in 10 iterations highlighting the benefits of policy exploration and policy optimization. Whereas, the Mean delta Q illustrate the drawback of having definite state space with no prior knowledge of the environment, making the model to converge at approximately 400 iterations. Hence, the null hypothesis that Policy Iteration algorithm would converge quicker than that of Value Iteration and Q-Learning is accepted.

Additionally, the effect of epsilon in Q-Learning conveys exploration-exploitation trade-off where lower values indicate low exploration and high exploitation and higher values indicating the vice-versa. In figure 7, the cumulative rewards for $\epsilon = 0.1, 0.2, 0.3$ showcase minor initial fluctuations than that of the ϵ values belonging to 0.7, 0.8 and 0.9. This is because the epsilon controls the randomness of the agent to explore the environment in order to determine the best actions. Once the epsilon decays completely, the agent stops the exploration and uses the learned policy to accomplish the task.

The optimal policy followed by the model-based algorithms is exactly same between the two however, Q Learning gaining higher rewards. This is because the Q learning encourages exploration and can discover optimal actions even where the policy is not completely explored. Whereas, the model-based algorithms require evaluation of the policy first and then improve it.

Finally, the change in state space by tuning the position bins of the environment showcase its effect on the three algorithms. First, Policy Iteration leads the runtime followed by Value Iteration while, Q-Learning is seen to be least affected by the state space. Since Policy Iteration performs Policy Exploration and Policy Improvement per iteration, the time for training the algorithm is inadvertently increased. Whereas, Value Iteration only updates $V(s)$ over the iterations which is not as tedious as that of the tasks followed by Policy Iteration. On the other hand, Q Learning is a model-free algorithm and the ability to balance exploration and exploitation mechanism enables it to train much faster. The Q-Learning agent maintaining the balance of exploration-exploitation covers some of the region of the state space and enhance further working as opposed to covering the entire state space in case of model-based algorithms. Hence, the change in state space of the environment shows no effect on the wall clock time for training the Q-learning agent.

B. Blackjack

In case of Blackjack MDP, the effect of γ on Value Iteration is not as tremendous as that of Carpole MDP since the state space of Blackjack is much smaller. Here, the large γ values do not appear to drastically affect the convergence of Value Iteration due to the very small state space though, the higher γ values converge relatively slower than that of the lower γ values. In case of Policy Iteration, the γ values converges at the same number of iterations, however, the lower γ values reach greater Mean V values since the immediate rewards are prioritized over future rewards. Specifically, for $\gamma = 0.99$, the algorithms are yet able to converge with approximately the same rate as that of the lower values due to the smaller state space as that of Cartpole MDP.

On the other hand, Q Learning showcases a drastic effect of the γ values because the knowledge such as transition probabilities, reward functions are not available as in the case of model-based algorithms. As a result, the updation of Q values also requires more iterations to converge. Interestingly, $\gamma = 0.99$ here showcase higher iterations to converge cementing the fact that high γ values require higher iterations to converge, same as that seen in case of Cartpole MDP. Here, given the state space, both the model-based algorithms outperform the model-free algorithm due to the presence of knowledge in case of former and absence in latter.

As for the comparison between Mean/Max V, for Value Iteration, both metrics begin positively where Max V reaches the maximum value of 1.0 meeting Mean V at 13 iterations. However, Policy Iteration though has Mean V starting below zero, the point of convergence stays below 10 iterations, thus outperforming value iteration by close margin. This behaviour is expected given the internal working of both the model-based algorithms as discussed earlier. On the other hand, the mean delta convergence of Q Learning illustrates the convergence for all the γ values at around 100 iterations. The lack of knowledge of the state space contributes to the drastic performance of Q Learning in this case. However, the initial fluctuations in the values spontaneously indicate the effect of γ on the convergence. Finally, the null hypothesis that model-based algorithms would showcase identical behavior is rejected, as Policy Iteration is seen to converge better than that of Value Iteration by a minor degree. Whereas, the hypothesis that Q-Learning will take longer iterations to converge is accepted.

Regarding exploration-exploitation trade-off, the cumulative rewards for incremental ϵ values illustrate the notion that higher values lead to higher initial fluctuations describing the random exploration of the agent. However, cumulative rewards for all values ϵ gain a negative reward, indicating that the policy converged by the Q-Learning agent is not optimal.

Further, the optimal policy map of Value Iteration and Policy Iteration displays an identical policy taken by their respective agents. The reason for similar policies is the small size of the state space of the problem. Compared with the optimal policy map of Q Learning, a drastic difference is evident where the agent has not taken high-rewarding actions in the specified number of iterations and is outperformed by that of Value Iteration and Policy Iteration. The reason for the failure of Q Learning is again the absence of prior knowledge of the state space as available in the case of the model-based algorithms. This evidence illustrates the importance of transition probabilities and the reward function as knowledge. As a result, the null hypothesis that Q Learning will suffer due to the absence of prior environmental knowledge is accepted.

VI. Conclusion

Evaluating the Model-based and Model-free MDP algorithms over large and small state space reveals the inherent working of these algorithms clearly. In case of Cartpole representing large state MDP, the γ factor had no effect on Policy Iteration while affecting the convergence of Value Iteration tremendously as it reaches value of 1. Whereas, Q Learning though needing higher iterations to converge performed better than Value Iteration. For blackjack as small state MDP, both the model-based algorithms perform relatively well with a minute variation in the convergence, whereas Q Learning converging in higher number of iterations. This cements the fact that as the state space increases, the need of prior environmental knowledge increases harnessing which the model-based algorithms are expected to perform spontaneously. Whereas, the optimal policy map conveys an interesting notion in Cartpole MDP where Q Learning receives tremendously higher cumulative rewards, which means that though the policies have converged, the model-based algorithms are unable to unearth the optimal policy as that of model-free algorithm for large state problems. In case of Blackjack, the model-based algorithms are clearly showcasing the optimal policy followed by them outperforming Q Learning when the state space is small since, Q learning needs to explore the environment before which the model-based have already evaluated the higher rewards generating policy.

Finally, to improve and further deepen the understanding of the model-based and model-free algorithms, they can be subjected to various seeds to take the luck factor out of the equation. Additionally, since this study is about nongrid world MDPs, these algorithms can be evaluated for grid world MDPs such as Frozen Lake. In addition, if resources are available, the effect of $\gamma = 1.0$ can be visualized on the three algorithms discussed here.

References

[1] Bellman, R. A Markovian Decision Process. Journal of Mathematics and Mechanics. 1957.

[2] J. Mansfield, ”bettermdptools” (Version 0.7.2) [Source code]. Available: https://github.com/jlm429/bettermdptools.