CS7641 - Assignment 4
Markov Decision Processes

Ashish Panchal
OMSCS Student - College of Computing
Georgia Institute of Technology
apanchal33@gatech.edu

***Abstract*—This Assignment explores Markov Decision Processes in the form of two different systematically chosen environments, of different natures, and use of planning methods, 2 models based along with 1 model-free, to solve such processes. The size and representation of the problem along with near/far-sightedness of the problem solver drastically impact the solution along with the method of balancing exploration and exploitation and convergence criteria**

***Index Terms*—Markov Decision Processes, Value iteration, Policy Iteration, Q learning, Model-based, Model-free planning.**

I. Introduction

Many of the processes in our daily lives are sequential in nature, where one event follows another event before it leads to a final outcome, such as making a coffee, going to the office, or even global warming like weather phenomena. While classical supervised techniques try to map direct cause to effects, there could be many different stages and associated actions that yield a particular kind of result. A subset of such processes, in which the next state is assumed to be only dependent on the current state is called Markov decision process MDP. This assignment aims to explore 2 such processes of different natures to build a better understanding of methods that can optimally solve them.

A. Environments: overview

The major differences between the two environments are their size and structure.

1) Frozen Lake FL: A stochastic Grid world is chosen to analyze a smaller MDP, of shape 4x4. the environment is made up of 3 kinds of tiles, 1 slippery ice, with a 1/3rd chance of going elsewhere than the intended direction of movement, a hole that results in high negative rewards and termination of the game, and a goal state that yields positive rewards. This resembles real-life scenarios where a part of the situation is in our control and others are stochastic, like going to office in a car where every turn and speed could be changed as intended, but red lights and traffic out of our hands, and policy can be made to find the best way possible to reach office based on one's priorities.

2) Mountain car MC: Another stochastic model aims to drive a car on the top of a mountain from the valley, which only yields time-dependent negative rewards unless the top is reached. The statespace is continuous in nature, unlike the FL, does not reflect an event, but rather an observatory feature of a situation, reflecting finer attributes of controlling a process, such as what is the correct degree to turn the steering wheel in order to make a collision-free left turn. These two MDPs reflect different views of similar problems and therefore are interesting to study.

B. Motivation : Algorithms

The following sections aim to comprehensively study 3 different ways to solve the above problem. 2 Model-based methods, where the transition probabilities from one state to other associated with particular actions are known. and 1 Model Free method, which assumes no such knowledge.

1) Value iteration VI: Focus on iteratively updating Bellman's equation of Value associated with different states by traversing the MDP using a transition matrix and collecting associated rewards, until convergence is reached.

2) Policy iteration PI: Although similar to VI, PI focuses on achieving the optimal policy, by dividing the process of update into two parts policy improvement, which resembles an iteration of PI and policy evaluation, to derive the best policy out of the updated Value function.

3) model Free : Q learning: This method explores the environments and maintains a state-action value function, updated using Bellman's optimality function. asymptotically yielding a similar value function as VI.

The analysis in this assignment focuses on behavioral aspect and functionality of algorithms in different circumstances and does not cover the quantification of methods behind the processes.

II. MDP Frozen Lake: FL

A. *Value iterations: VI*

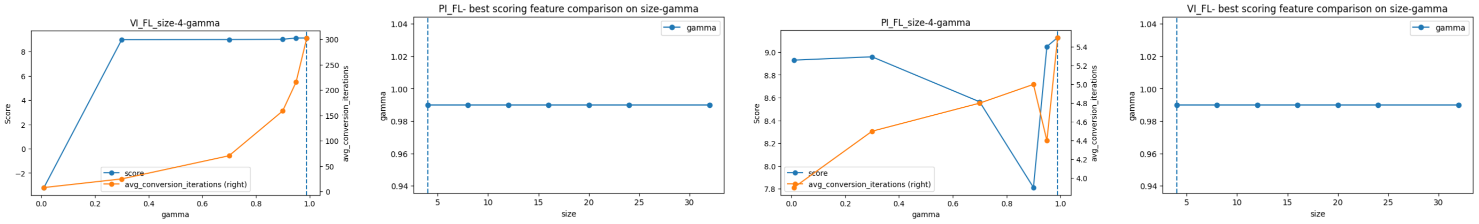
This section covers the Analysis and overview of the experiments conducted on a Frozen lake. A smaller map of size 4x4 was studied in addition to the exploration of map sizes 8x8,16x16,20x20,24x24,32x32. Each experiment was conducted on 10 different variants of the maps, accounting for stochasticity and chance observations. Two model-based methods, Value iteration VI, Policy Iteration PI and one model-free method, Q learning QL, were used to solve the MDP, the implementation details and analysis are covered below. The holes-to-sliding tiles ratio is reduced to 0.1 for clarity of the study and computational feasibility.

B. *Reward Function R*

For faster training, R was changed, r=10 for reaching the goal, -15 for getting into a hole and -0.04 as a time penalty. As the function penalized more for getting into holes, the environment presses to generate a cautious behavior, while ensuring less game length. This may still create a policy optimal for the original environment. The number of steps was limited to size*size*5, to ensure enough exploration to reach the goal, while limiting getting stuck. [1]

C. *Value iterations: VI*

1) ***Convergence Criteria:*** :, Value iteration is a monotonically improving model, as the Bellman equation follows contraction mapping behaviour, the estimated average state-value V can only improve towards a version closer to optimality, i.e ensuring highest total returns. Convergence was assumed, when for any state maximum difference between the two subsequent V was less than



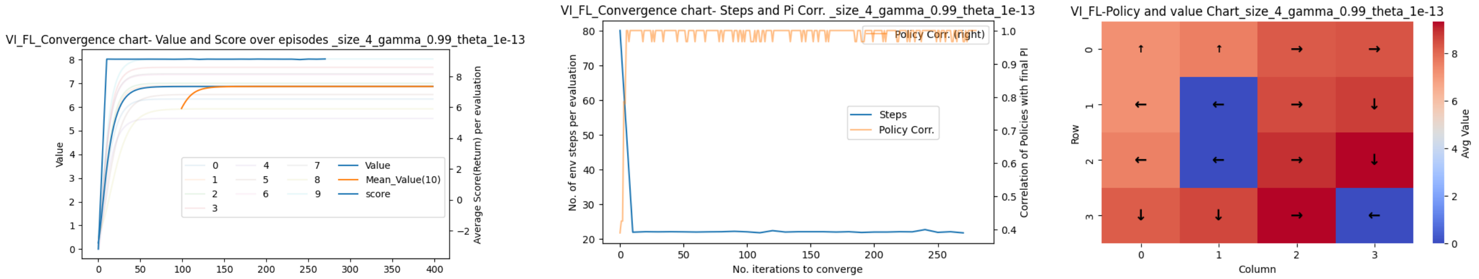
(a) γ vs Value and CI : VI

(b) Best performing γ vs size

(c) γ vs Value and CI : VI

(d) Best performing γ vs size

Fig. 1: Frozen lake VI :performance comparison over γ and size

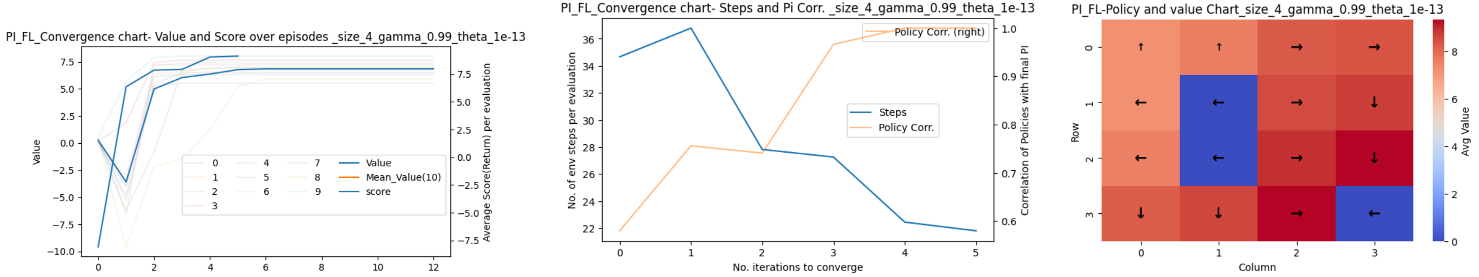


(a) Convergence: Value and score

(b) Convergence: Steps and Policy Corr.

(c) Policy heatmap

Fig. 2: Frozen lake: Convergence charts for Value iteration and Policy value heatmap



(a) Convergence: Value and score

(b) Convergence: Steps and Policy Corr.

(c) Policy heatmap

Fig. 3: Frozen lake: Convergence charts for policy iteration and Policy value heatmap,(right bottom corner blue cell is goal, middle two are holes)

thresholds θ. The number of iterations was limited to n_iter x Multiplier, where the latter reduces the number of operations to increase Q_track and P_track lengths.

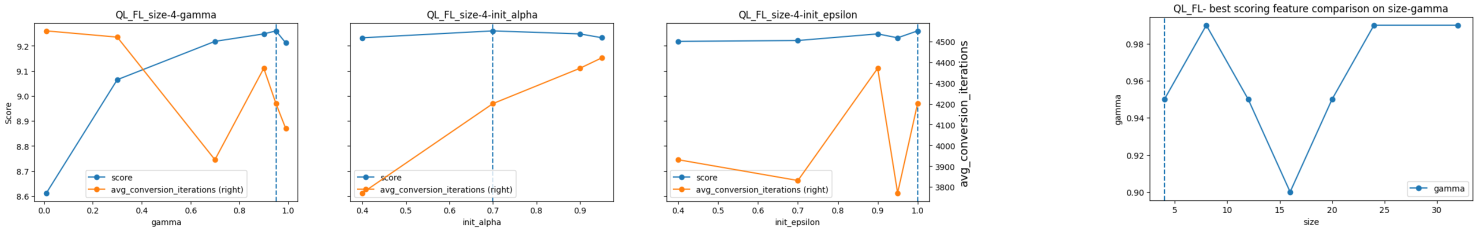
2) ***Exploration Exploitation analysis:*** : *Fig. (1a)* ,This tradeoff is controlled by 3 factors, Q and V initialization with zero, Discount factor γ, and convergence threshold θ, of which γ is explored in depth. For the purposes of our analysis θ is kept constant at 1e-13, keep it smaller enables any intermediate value studies. Using 0 initialization results in optimism, in -vly time-dependent rewarding environments results it in exploration of 0 V state-action pairs. this results in faster exploration, compared to random initialization. The best model, with Avg Return R_ 9.13 at γ 0.99 was found, over 10 maps, converged on average at 302 iterations Reaching the goal in 21 avg steps. The convergence iteration count IC, is highest at large γ i.e 0.99. As this result in larger changes in V(s), therefore longer for δ V(s) to reach θ. These large changes propagate deeper in network, and therefore faster amendment of V(s) once a positive reward is reached and therefore better policy even at less experiences is created. Larger γ could be identified as a generalization of the impact of one state to a large number of states, to perform generally well even with small iterations to create a good enough policy.

3) ***Impact on Statespace size and exploration behaviour :*** : At small maps, as there are less number of states to explore, even the small enough gamma prioritizing immediate reward performs well, reaching optimal policy, explaining the similar Score at optimality until γ = 0.01 where very small changes false trigger θ. This is further supported by score/Return . With a decrease in γ the IC drastically decreases, this is shared by all env, due to smaller changes in V causing early reach of θ.*Fig. (1b)* , γ 0.99 performed the best across statespace sizes, i.e highest total return, The performance (total return) over the range of γ monotonically decreases in most of the sizes, this is more pronounced at larger sizes. However at larger sizes, very small γ result in a highly exploratory nature, therefore could result in a good policy, unlike med γ rangers which still restrict exploration and early reach of theta. This could be amended by further training and smaller θ.

In general, reasons which results in more exploration result in higher overall returns, controled by conversion. This impact of exploratory behaviour is more pronounced on larger maps. Explaining low change in performance at smaller maps and monotonic decrease in performance over gamma as the size grows and higher performance at very small maps.

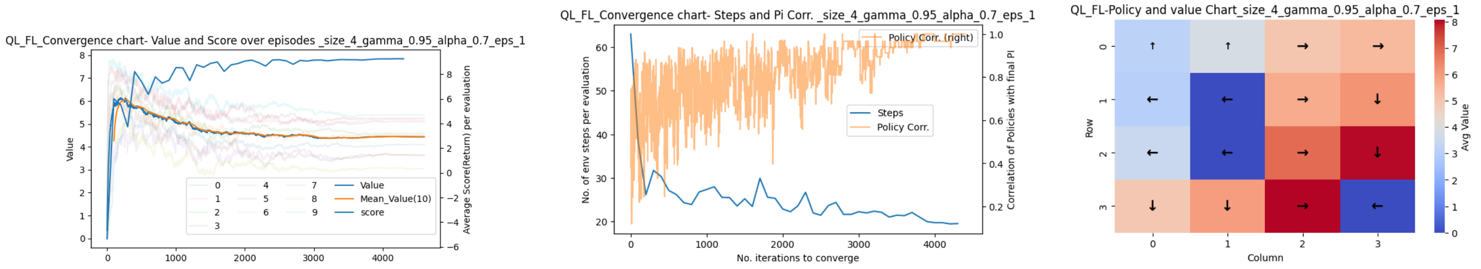
4) ***Convergence Analysis:*** :

Fig. (2a,2b) ,The study was performed using the best model. At early iterations, being all V(s) =0, random actions are taken, which seldom returns in +ve reward, as seen from steps per evaluation convergence plot, with large number of states visited, accumulating time base -ve reward before reaching goal, resulting in low Avg V(s) and score. With further iterations, this exploration results in frequent positive rewards, which is much more pronounced from “Avg score per evaluation”, reaching optimal behavior early on, taking 21 steps to terminate. But the V(s) is not yet converged, working as a confirmatory factor, confirmed from fig(correlation



(a) γ, init α and init ϵ vs Value and CI (b) γ vs size

Fig. 4: Frozen Lake Q learning :performance comparison over γ, init α and init ϵ and size



(a) Convergence: Value and score (b) Convergence: Steps and Policy Corr. (c) Policy heatmap

Fig. 5: Frozen lake: Convergence charts for Q learning and Policy value heatmap

chart), although convergence of V is reached 300, optimal policy π^* is reached at 10 episodes.

Fig. (2b) ,The impact of the reward structure can be seen from the found optimal policy, prioritizing escape from holes, which expects a chance slip into a state which is better.

D. *Policy iterations: PI*

Fig. (3c),PI convergence criteria: Unlike VI, convergence is reached when two subsequent Policies are same, created after 1 policy improvement, which is similar to VI governed by θ and policy evaluation step.

*1) **Exploration Exploitation analysis**:* : *Fig. (1c)*,The same features as VI govern the exploration and exploitation tradeoff, however, PI allows a lower θ, due to its design and following monotonic improvement, it could still converge. For this analysis, θ is still kept at 1e-13, to better study the difference between VI and PI, i.e how can PI improve over VI policy.

A single iteration PI could have multiple VI iterations, therefore cannot be directly compared. The best peak score of 9.12 similar to VI was found at $\gamma = 0.99$, found in 5.5 PI iterations, which intern follows similar properties to VI, i.e 21 steps to reach goal at optimal policy. Similar to VI, IC mostly decreases with gamma. However score does not follow the same pattern as PI, i.e have less variance, R_ ranging between 7.8- 9.13, unlike VI which ranged from -2 to -9.13. this is because each PI runs many VI, therefore even with smaller γ, PI ensures sufficient exploration and updates V(s) to reach a better estimate of optimal V*. This overcomes the issues of smaller updates and early convergence of small θ and smallγ, while retaining the exploratory nature of such γ, as explained for PI at large maps. This also follows the adverse effects at mid γ explained for large maps.

Impact on State-space size: *Fig. (1d)*, Similar to PI, best γ remains 0.99 which follows the same behaviour as well. With increased state space, the impact of VI becomes more prominent and similar and the benefit of γ decreases given θ.

*2) **Convergence Analysis**:* :

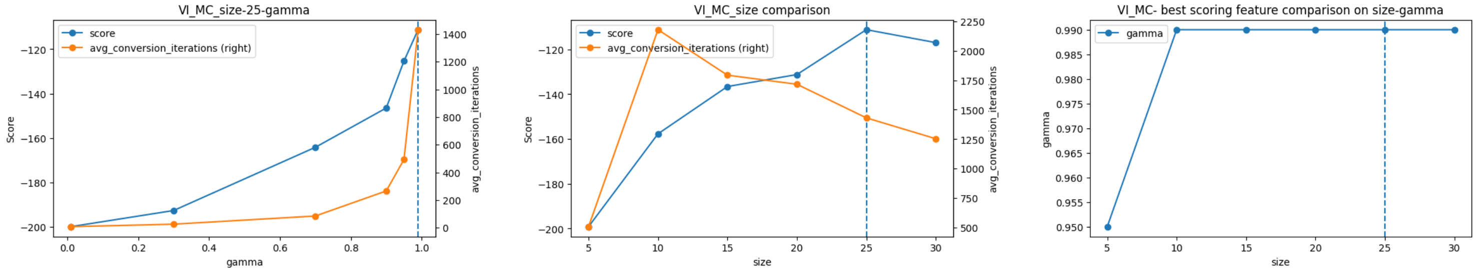
Fig. (3a,3b),PI produces the same Optimal policy π^* as VI, following a similar explanation, following the smaller map. However unlike VI, PI evaluates V(s)' for each policy improvement and changes π^*, this π differed greatly over different iterations. This is also reflected in changes in V(s), unlike VI, which shows an initial decrease, and thereafter follows a pattern similar to PI of monotonic improvement. It is to be noted that even though V(s) decreased on the second iteration, the average score improved. And since PI governs convergence, even if θ is kept large, it might result in faster convergence.

E. *Q learning QL*

*1) **Convergence Criteria**:* : Exploration greatly governs convergence for model-free methods, as it does not have information on state space transition. Therefore for a stricter rule, the maximum difference between V(s) over all the states from subsequent V(s)' over a specified window is used, therefore even if V(s) – V(s)' at time t is ¡ threshold of 0.05, it would optimize for atleast w(window) more iterations. However Q learning follows Bellman's optimality equation and is contractual in nature, asymptotically.

*2) **Exploration Exploitation analysis**:* :

Fig. (4a),In addition to parameters that govern exploration in VI and PI, 2 more parameters control this process, **learning rate** $alpha$ and probability to take random action also governs this behavior $epsilon$, where $alpha$ controls how much the value of an action changes. In particular, we change the initial values of both, where the iterative decrement is governed by a set ratio, 0.5 and 0.9 respectively, and maximum iterations until it reaches minimum values of 0.001 and 0.1 respectively, as defined before. This decrement is necessary to converge given the terminal nature of the environment. The best peak score of 9.25 similar to VI and PI was found at $\gamma = 0.95$, init_ $alpha$ = 0.7 and init_ $epsilon$ = 1, found in 4201 iterations, this is 14x more than VI , i.e 19.5 steps to reach goal at optimal policy. **Learning rate** γ, at high values as attains higher scores similar to VI and PI, and also follow a similar monotonic increase in over its range, however a higher value of γ emphasize long term goals resulting in particular path fixation with only small number of observations, which could reduce exploration, however, this is not prominent due to small state space itself, while it also produces faster convergence, which is reflected in CI which decrease with γ, unlike VI and PI. This ensures to exploration of all possible states with the knowledge of

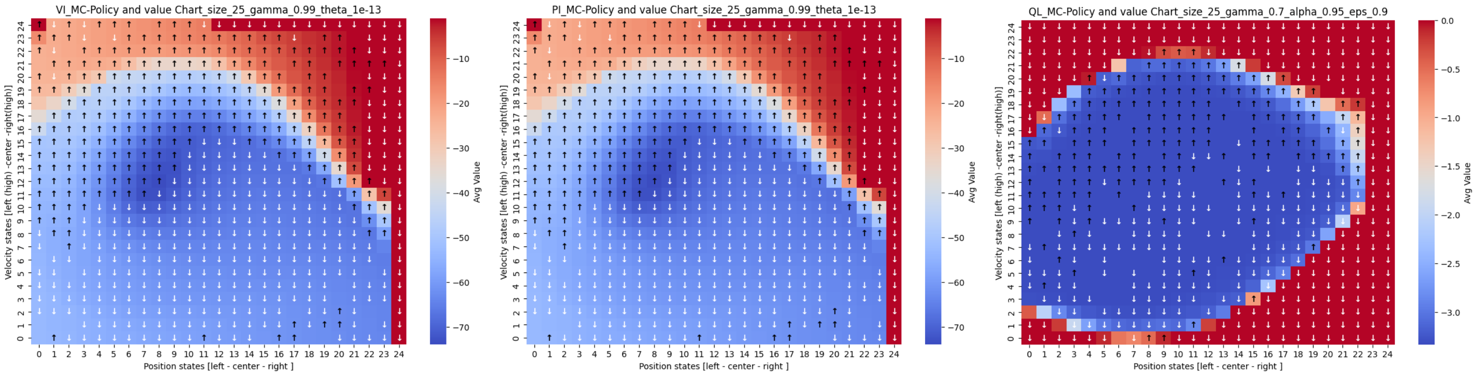


(a) γ vs Value and CI

(b) Score and CI vs size

(c) best performing γ over size

Fig. 6: Mountain car VI :performance comparison over γ and size



(a) Value Itertation

(b) Policy Iteration

(c) Q learning

Fig. 7: Mountain car : Converged Policy -Value heatmap, up arrow is accel. right, down arrow is accel. left

the state transition matrix. It is interesting to note that Q_learning maintains a lower variance on score ranging (8.6 – 9.25) than VI and PI, this is because of higher and longer exploration and a more robust convergence criteria. **Alpha.** For a smaller map, a higher initial learning rate results in larger changes in V(s), which could result in large CI, therefore CI linearly increases with init_α. Given a small map, the impact on the score at optima is not significant, as similar policies could be identified, but it is hypothesized that with even smaller alpha at larger maps, the CI is very high, as the algorithm would find it hard to converge given the number of states. **Epsilon**, A higher init_ϵ results in more random actions initially and therefore higher exploration, however similar to α, as the state space is small this is not very prominent, and ϵ sufficiently explores the state space. The impact on CI is not clearly increasing, because low ϵ results in more random actions later causing larger CI, similarly a larger value would result in more random action initially, when it could have converged, resulting in larger CI.

3) *Convergence Analysis*: :

Fig. (5a,5b,5c) , QL converged to the same π* as PI and VI, however has a different V*, which is the result of the difference in the method of exploration and convergence criteria, it is hypothesized that with sufficient training, the same V* could be achieved. This is particularly not an issue for a smaller map, as observed for VI as well, optimal π* was achieved earlier than V*. Stochasticity plays a major role in convergence. From fig(), the Avg V' increases following a similar argument as VI, with the occurrence of paths leading to goal, however with 0 initialized Q table, the model is overly optimistic for unexplored state-action pair, but by the design of env. each action results in a negative reward, which updates the Q(s,a) when explored, resulting in a followed downward trend of V'. It can be confirmed from the average score per evaluation, which still increases, as a result of better policies and understanding of environment. Therefore V' can still be used to find convergence but is not a direct proxy for total Return.

The exploratory nature of QL along with 0 initialized Q table is visible in larger changes in policy correlation at over iterations, with more similar policies towards convergence and less variance intimating of good enough approximation of the environment.

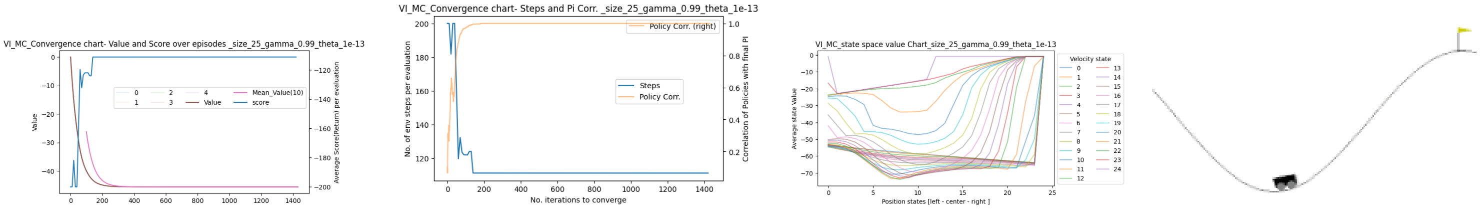
III. MDP Mountain car: MC

MC is a continuous state space environment with 2 Dimensions, x-axis position of the car varying, between -1.2 to 0.2 and the velocity of the car, -0.07 to 0.07. and three actions, accelerate left, no acceleration and accel. right. A -1 reward is received every timestep, until flag post on the mountain. Given the best possible Return achieved is -110 in 110 steps, a maximum of 2000 steps were used to generate a transition matrix, training and evaluations to ensure enough exploration and limiting getting stuck. [2]

A. *State space discretization and mapping*

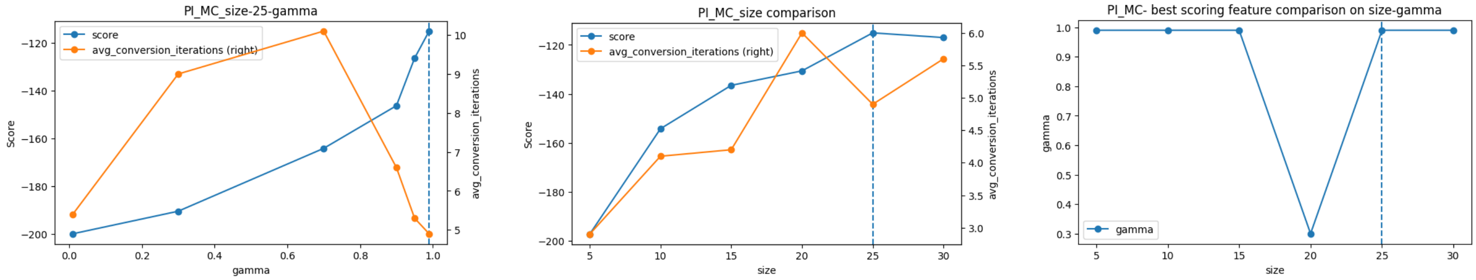
For simplicity and comparison with Frozen Lake, both states are equally divided into equal parts. As a study for a larger problem, a 25x25 division is done. (Highlow)/no. of state_per_axis is used to create an average bin-width and then used to bin the entire state space. Each state bin is mapped to an index, which are used as state proxy. A state space transition matrix is created using 2000 max step per episode limitation, over 500 episodes using randomly initialized states from the state index matrix. Each observation was mapped to the respective state index using the bin mapper function. This mapping function was used during q-learning and evaluation. For exploration, of different state space sizes, a higher space of 30x30 was used to discretize.

Transition matrix was created for state axis size of 5,10,15,20,25 and 30. Since size just changes the granularity of the performance among different state space sizes is feasible, unlike FL. Additionally, the initial stochasticity is governed more by the environment, the number of instances for training convergence was limited to 5 and evaluation to 50 for VI and PI and 20 for QL.



(a) Convergence: Value and score (b) Convergence: Steps and Policy Corr. (c) State Space Value (d) Mountain car visual

Fig. 8: Mountain car :Convergence charts for Value iteration and state value chart



(a) γ vs Value and CI (b) score and CI vs size (c) best performing gamma over size

Fig. 9: Mountain car PI :performance comparison over $\gamma$and size

B. *Value iterations: VI*

VI Convergence criteria : The same convergence criteria is used as FL, however maximum iterations are limited to 5000.

*1) **Exploration Exploitation analysis**:* : *Fig. (6a)*,Evaluations were performed with max steps limited to 200, as governed by γ. It follows FL similarly in terms of score changes, high and low at respective γ. This is also governed by the nature of the environment, there are no immediate positive rewards, therefore the net negative returns can only be controlled by the long-term outlook. However a long-term outlook also required a larger number of exploratory steps to converge with also explains a monotonic increase in CI. The best peak score of -111.2, close to the environment. Optima was achieved in 1431 iterations, at gamma 0.99.

*2) **Convergence Analysis**:* : *Fig. (8a,8b)*,On the best model, initial V(s) are recorded as high 0 values, this is the result of 0 V initialization, which contrasts with the -ve rewarding nature of the environment. This 0 initial V, initially results in P' resulting in almost no perpetual movements in the valley collecting negative rewards. However, this results in the exploration of upper edges which have 0 initial value, ¿ -ve explored yet, and eventually, ability to reach the flag post. This exploration results in an overall decrease in values, associating -ve rewards for most states, but states with a higher probability to terminate have higher values. Identification of which results in policies ensuring higher scores which is inversely proportional to steps, as can be seen.

*3) **Converged policy analysis**:* :

Fig. (7a),(8c,8d), shows the impact of game physics on value of the states. In the middle, at **valley**, being static becomes most hurtful, and a high rightwards velocity v' is rewarded, and high left is not very rewarding. This results in a policy, If slightly right with lower v, accelerate towards left and do it further, resulting in higher downward speed when coming down from left slope and henceforth accelerate to right to reach the goal. If already at high right speed, accelerate Right, If already at high left v' – accelerate left. On **left slope**, Higher right v' are rewarding, 0 v is okay because it can accelerate right. Resulting in a policy unless at high speed towards left, accelerate right. On **right slope** even slightly lower right v has a high value. Resulting in a policy following the same kind of behavior as valley unless sufficient rightwards v', except when already at goal, even accelerating left has high value. Additionally, the value of different speeds loosely maps to the shape of the terrain over x-axis.

*4) **Statespace size comparison : exploration and convergence**:* :

*Fig. (6b,6c)*Supporting the argument above, 0.99 remains the highest V* γ across state space for the problem. From a convergence point of view, scores are convergence increase with the size of discretized state space, this is because it captures the transactions adequately well, and reduces the artificially created stochasticity, resulting in better policies. However given a particular convergence criteria, this may result in inadequate exploration of very large spaces, therefore lower scores, as seen for size =30. Additionally stochasticity at smaller sizes also required more CI to reduce the overall effect. Explaining the decrease in CI at convergence except at 5, which false triggers the θ.

C. *Policy iterations: PI*

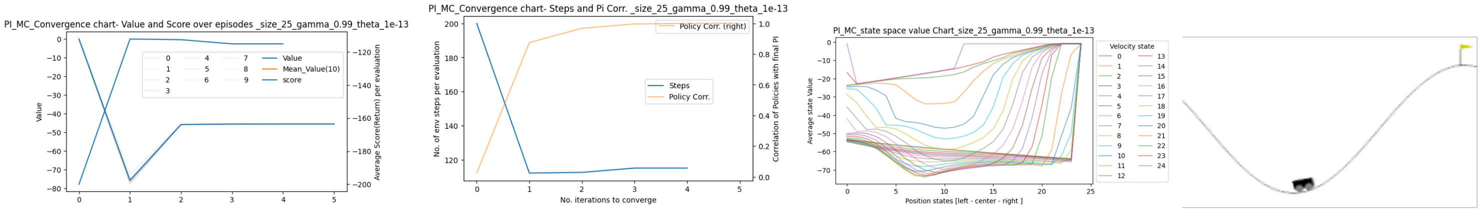
PI Convergence criteria: same convergence criteria is used as FL, however number of maximum iterations is limited to 10000.

*1) **Exploration Exploitation analysis**:* : *Fig. (9a)*,From the score POV, the γ behaves similarly to VI in MC and also PI in FL, ie score is higher at high γ. And CI decreases with γ from 0.7, because, higher γ results in faster convergences aligning with games nature. lower γ results in higher explorations and therefore less frequent corrections, which could result in earlier conversions given the criteria. However it is hypothesized, a more iterations could increase performance at lower γ. The best peak score of -115.1, close to env. Optima, was achived in 4.9 PI iterations at γ 0.99.

*2) **Convergence analysis (comparison with FL)**:* : The same policy as PI was achieved,*Fig. (10a,10b)*, Where the score almost reaches the maxima after a single iteration. Although given the larger statesize from FL, there are gradual changes in policy observed, from the policy correlation chart. The convergence of V' follows a similar behavior as in FL.

Additionally, even though the size of the statespace is 40x larger to FL, the policy iteration converges in similar CI. Therefore PI could be environment agnostic.

*3) **Statespace size comparison : exploration and convergence**:* *Fig. (9b,9c)*Given the convergence criteria, and the problem, larger



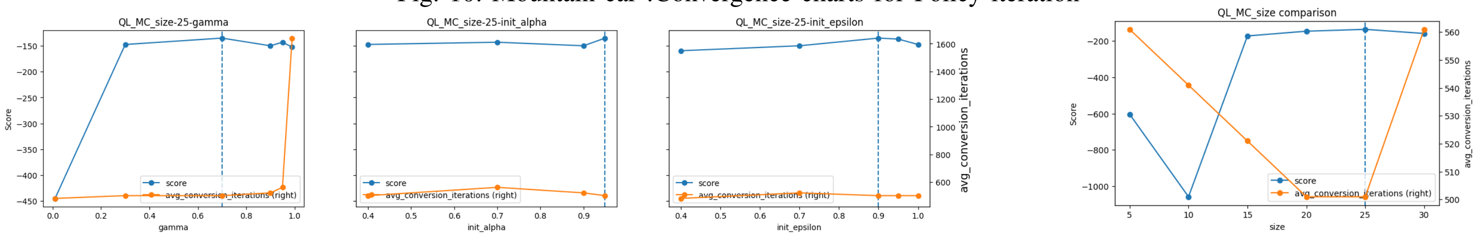
(a) Convergence: Value and score MC-PI

(b) Convergence: Steps and Policy Corr.

(c) State Space Value

(d) Mountain car visual

Fig. 10: Mountain car :Convergence charts for Policy iteration



(a) γ, initα and initϵ vs Value and CI

(b) score and CI vs size

Fig. 11: Mountain car QL :performance comparison over γ, initα and initϵ and size

γ converges to higher values except at 20, which converges best at 0.3, this could be because it balances between highly exploratory nature of low gamma, and a good enough approximation of statespace, with the criteria given, it could find a solution, while decreasing the adverse effects of mid γ mentioned before.

Additionally, as explained for PI, performance increases with higher sizes granularity, however unlike, VI, higher size could induce a different policy even though the values are similar, and therefore for convergence, may require more iterations.

D. Q learning QL

Convergence criteria: is similar to QL of FL, however, 50000 max iterations were allowed to converge, the threshold is set to 0.01, and 10 instances of each size were tested based on predefined random seeds.

*1) **Exploration Exploitation analysis**:* : *Fig. (11a)*,The best peak score of -135 was found at γ = 0.7 , init_ $alpha$ = 0.95 and init_ $epsilon$ = 0.9, found in 501. Which is comparatively worse than the VI and PI models, as hypothesized for model-free learning for larger statespace. γ, *discount factor* governs the near and far-sightedness of the model, from the observed space, lower γ from 0.3 till 0.99 behaves similarly, a behavior explained for FL. However, due to large statespace, at very high γ the changes in the Q(s,a) are large and many, which results in a slower convergence as seen, It is possible that convergence criteria were liberal, to get sufficient amount of training iterations. α - *learning Rate*. Due to the larger state space, there is little impact of α over convergence, which allows larger α to make comparatively bigger changes to reach near optima and results in comparatively fewer iterations. ϵ*-decay factor* has less impact on score and CI, given the convergence criteria, that it is able to generate a significant number of random examples regardless of random exploration. This is largely due to the stochastic nature of the environment, defined by game physics. It has a stronger effect on the end result, therefore many policies could result in optimal behavior diminishing the impact of α and ϵ.

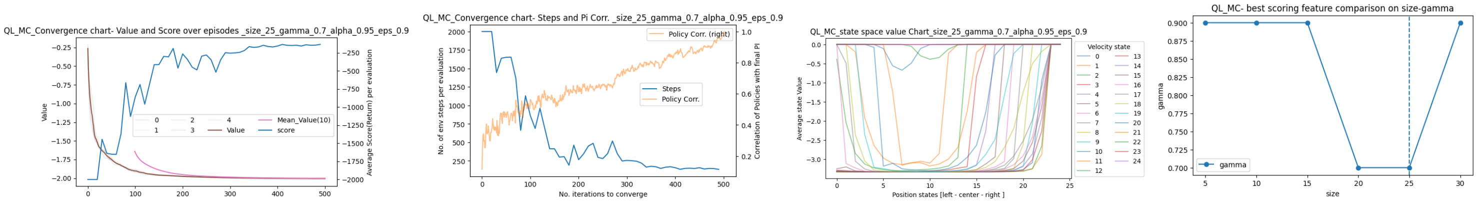
*2) **Convergence analysis**:* :

Fig. (12b),*(12a)*,The convergence of the Value function follows similar to VI and PI, due to zero initialization across 10 instances. Additionally, with the increase in iterations, the score improves as well, as expected, due to higher possibilities of landing at the flagpost. Unlike VI and PI, there is a drastic change in the policies from the final converged policy, however unlike FL, the variance remains the same, this is possibly due to larger spacestates giving multiple optimal policies.

*3) **Converged policy analysis**:* :

Fig. (7c),*Fig. (12c)*,The optimal policy is not entirely same as VI and PI but is similar in states that have a higher likelihood of occurrence. Since the model was not able to observe all the possible states, many of the unexplored states have an optimistic view, 0 value. This might show an adverse effect of 0 initialization, that is the model might result in getting stuck when observing an unknown state when exploiting. Additionally, a major bias is created for left acceleration, which occurs in a serial order of processing, having a 0 value and 1st in order, it would always be marked optimal until explored. The Value chart of Velocity over position has a higher correlation with the shape of the game state.

*4) **Statespace size comparison : exploration and convergence**:* *Fig. (12d)*,*(11b)*,At sizes 20 and 25 a gamma of 0.7 procures a higher score compared to lower and higher state spaces of 0.9. It is to be noted that none of the sizes found optima at 0.99 as for VI and PI. which a higher gamma could reduce stochasticity at smaller sizes, however, it is hypothesized that due to small size, the difference in performances would not be significant following



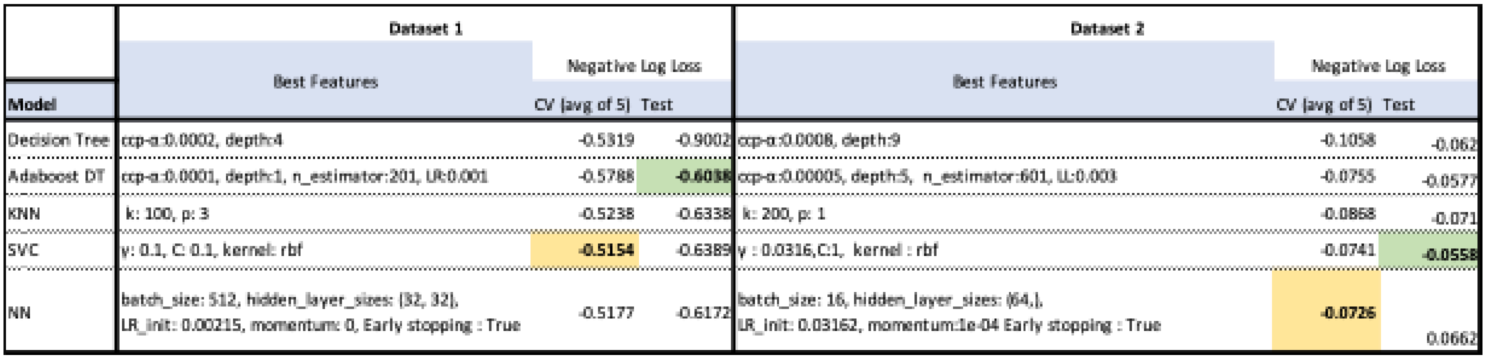
(a) Convergence: Value and score MC-VI

(b) Convergence: Steps and Policy Corr.

(c) State Space Value

(d) Best γ over size

Fig. 12: Mountain car :Convergence charts for Q learning and best γ over size comparison



	Dataset 1			Dataset 2		
	Best Features	Negative Log Loss		Best Features	Negative Log Loss	
Model		CV (avg of 5)	Test		CV (avg of 5)	Test
Decision Tree	ccp-α:0.0002, depth:4	-0.5319	-0.9002	ccp-α:0.0008, depth:9	-0.1058	-0.062
Adaboost DT	ccp-α:0.0001, depth:1, n_estimator:201, LR:0.001	-0.5788	**-0.6038**	ccp-α:0.00005, depth:5, n_estimator:601, LL:0.003	-0.0755	-0.0577
KNN	k: 100, p: 3	-0.5238	-0.6338	k: 200, p: 1	-0.0868	-0.071
SVC	γ: 0.1, C: 0.1, kernel: rbf	**-0.5154**	-0.6389	γ : 0.0316,C:1, kernel : rbf	-0.0741	**-0.0558**
NN	batch_size: 512, hidden_layer_sizes: (32, 32), LR_init: 0.00215, momentum: 0, Early stopping : True	-0.5177	-0.6172	batch_size: 16, hidden_layer_sizes: (64,), LR_init: 0.03162, momentum:1e-04 Early stopping : True	**-0.0726**	0.0662

Fig. 13: Frozen Lake and Mountain car performance comparison for best models — far and near-sighted

our observation from FL, additionally, a higher γ could potentially limit exploration in larger size env. Which would be required to limit possible optimal policies. Discounting along with size affects the number of iterations required to converge, the observed bound is very small. The performance of larger statespaces as hypothesized is sufficient enough to capture the dynamics of the game and reduce stochasticity and therefore perform similarly, while smaller sizes suffer. Although slightly, size captures the tradeoff of limiting search space and capturing environment dynamics, size 25 performs better than 30 as well as from 20 and 15. As mentioned before α and ϵ have limited roles in this particular environment.

IV. Conclusion, comparison and comments

The study explores two different-sized MDP environments. In general, environments that are inherently smaller in statespace are easily solved even with suboptimal methods, because of the limited total number of optimal policies and easier exhaustion of possible options. Long and nearsightedness is not very relevant in such cases, however, it is particularly important in large environments, very nearsightedness can result in sub-par policies or may fail to converge to optima, and having farsightedness could result in a very large number of iterations before convergence could be achieved.*Table. (13)*, Additionally in the second case exploration and exploitation become far more important factors,to feasibly solve the problem, and required a balance to achieve it using methods to explore state-action space stochastically and using them well to converge. Lastly, changing the representational granularity of the statespace vs having a large problem seems like problems that might share qualities but are not entirely same.

References

[1] Frozen Lake, Gym, www.gymlibrary.dev/environments/toy_text/frozen_lake/

[2] Mountain car, Gym, www.gymlibrary.dev/environments/classic control/mountain car