Improving upon Iterative Approaches for Solving Fully-Specified POMDPs

Benjamin Kern, Andrew Forney

Introduction

The **Markov Decision Process (MDP)** formulation simulates generalized, sequential decision making.

Partially Observable MDPs (POMDPs) are a further generalized form of the MDP in which the exact state of the system is not known by an agent acting within the environment.

Traditional approaches for solving POMDPs require iterative learning processes that converge slowly.

We propose a novel approach that yields an accelerated learning rate compared to traditional methods that utilizes algebraic linearization rather than convergent iteration.

The Algorithm

POMDPs are comprised of a finite set of states that an agent can be in at any given time step, a finite set of actions to choose from at each state, and a finite set of observations that allow an agent to receive information about the environment.

One can define a policy $\boldsymbol{\pi}$ as a discrete mapping between each possible state and a chosen action to be performed at that state. An omniscient agent that knows exactly which state it is in will be able to use a policy to traverse the finite state transitions. In POMDPs, the agent does not have this luxury, and it must rely on *belief states*. The belief state $\boldsymbol{b}$ of an agent is a measure of how much it believes it is in each of the possible states.

The value of a single state as estimated by a policy used as a heuristic $\boldsymbol{\pi_h}$ is given by the expected total reward an agent will receive by following only the heuristic policy. $\boldsymbol{V^{\pi}}$ is defined as a function a state that returns the expected value of being in the state and proceeding to follow $\boldsymbol{\pi}$ for all actions. Because these agents potentially run infinite time steps, each simulated future time step is discounted by a factor $\boldsymbol{\gamma}$ in order to prevent the function from returning an infinite value.

$$V^{\pi}\left(s^{(t)}\right) = R\left(s^{(t)}, a = \pi(s^{(t)})\right) + \gamma \sum_{s^{(t+1)} \epsilon \mathcal{S}} P\left(s^{(t)}, a = \pi(s^{(t)}), s^{(t+1)}\right) V^{\pi}\left(s^{(t+1)}\right)$$

In traditional methods, an update rule is employed to iteratively converge upon an optimal value estimation function $\boldsymbol{V^{\pi *}}$. For example, in the *Q-Learning* method, the heuristic policy is updated on each iteration of this convergent process based on which action maximizes value estimation function based on the current state.

Once this optimal evaluation function $\boldsymbol{V^{\pi}}$ is obtained, defining the optimal action $\boldsymbol{a*}$ of the agent any given timestep is as simple as figuring out which action maximizes $\boldsymbol{V^{\pi}}$ when applied to the agent's belief state $\boldsymbol{b}$.

$$a^* = argmax_{a \epsilon \mathcal{A}} \sum_{s \epsilon \mathcal{S}} V^{\pi}(s|a) * b[s]$$

If viewed as an equation of all states simultaneously rather than just looking at one state at a time, a simple linearization can be applied that allows for algebraically solving the value estimation function for a given heuristic policy in one calculation – rather than having to perform value iteration for each different state repeatedly until converging within some bound.

$$V = R + \gamma P V \longrightarrow V = \frac{R}{1 - \gamma P}$$

The Tiger/Door Problem

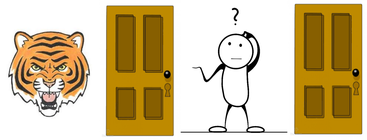


Illustration of the Tiger/Door Problem. Shown is the **Tiger-Left** state. Choosing to open the left door in this state will result in a reward of **-100**, while opening the right door in this state will result in a result of **+10**. If the agent chooses to *listen* instead, it will receive a reward of **-1**.

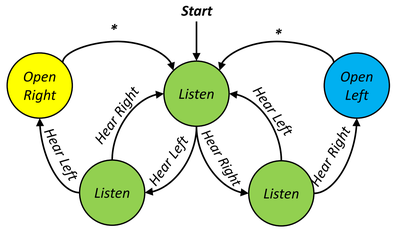
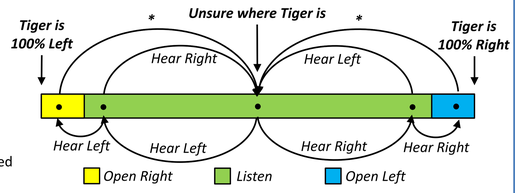


Diagram showing the flow chart of an agent following the solved POMDP's heuristic function. Each node holds the necessary optimal **action** to be performed upon arrival at the node. The transitions between nodes represent an **observation** received by the agent after performing the action in the node.

There is a tiger behind one of two doors, and an agent must determine which door it is behind by listening. If the agent chooses to open the door that has the tiger behind it, the agent receives a large negative reward, and otherwise receives a positive reward.

In a regular MDP, this problem is trivial. This is analogous to the agent being able to see through both doors and choosing the one that the tiger is not behind. In the POMDP however, the current state is not visible to the agent. The agent must formulate a policy for optimal cumulative long-term reward.

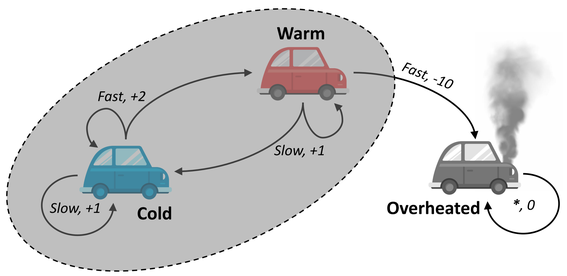


Belief State Evaluation Diagram for the optimal heuristic policy π_h^*. The belief of the agent can be anywhere in between the two states, and this diagram allows for optimal decision making given any possible belief state.

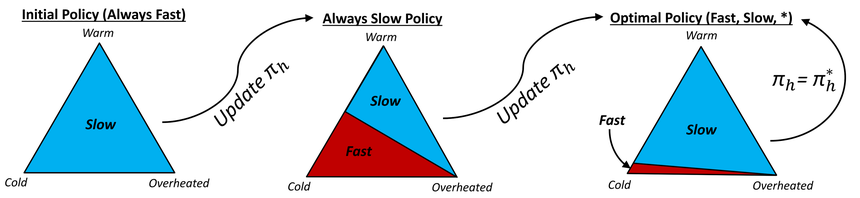
The Overheating Car Problem

An Automated Car needs to get to its destination as quickly as possible without overheating. The Car can be in one of three states, *Cold*, *Warm*, and *Overheated*. However, the agent controlling the car can only observe whether the car is overheated or not.

An optimal car agent knows to go slow until it is estimated that the expected reward of going fast is worth the risk of overheating.



The underlying MDP for the Overheating Car Problem. The greyed-out ellipse denotes that the agent does not know whether it is *Cold* or *Warm* when it in this region.



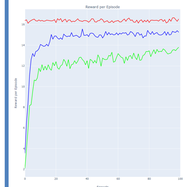
Belief State Evaluation Diagrams for different possible heuristic policies that an agent might have. The final policy is optimal when the update rule no longer changes the heuristic policy. The maximum number of update steps needed to solve this POMDP is 2.

Different values of γ make affect how cautious the agent is when making decisions and result in slightly different belief state evaluations.

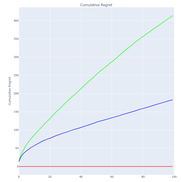
Results

Tiger Example

Reward per Episode

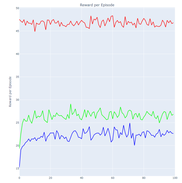


Cumulative Regret

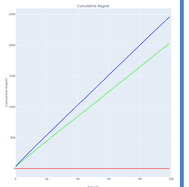


Car Example

Reward per Episode



Cumulative Regret



Linearization method *Q Learning Agent* *Witness Algorithm*

Comparing our method with other methods of solving POMDPs. On the shown POMDPs, the traditional methods either converge much slower than the linearization method, or they fail to reach the optimal solution offered by the linearization method entirely.

Conclusion

POMDPs represent more realistic sequential decision making processes than MDPs. The methods discussed here can be used to solve POMDPs and MDPs at an accelerated rate as compared to traditional methods.

Simulation results support the efficacy of this method on traditional POMDPs. Where traditional iterative approaches to solving POMDPs take many time steps to converge within a certain boundary, our method uses linear algebra to solve for the converged upon result in a single time step, leaving only the need for heuristic policy updates.

Future Work

This method can possibly be used to solve non-specified POMDPs (Where the underlying MDP is completely invisible to the agent and it only receives observations and rewards from its environment) by linearizing based on a transition function that depends on observations rather than the states themselves.

Additionally, since MDPs are essentially just finite state machines with a reward function, an interesting step to make might be experimenting with pushdown automata or other types of automata that are stronger than finite state machines, and to see what types of systems can be simulated with these change.

We look forward to being able to continue this research and exploring these avenues in the future.