CS-748

Advances in Intelligent and Learning Agents

Responses to Reading Assignment R6

Deepanjan Kundu

We are familiar with the concept of MDPs and the various notions and planning methods used for them. However there exist a number of problems which are partially observed stochastic domain based. A good example would be the case of a robot navigation from intersection to another intersection. The robot has to learn from what he has observed and also estimate its position, though it may have an uncertainty attached to its position. It is possible that the agent has partial knowledge about the states and hence mistakes the current state to be some other one.These cases where the state is not completely known to the system is what is dealt in POMDP. The notion of observation and belief state come into play with this. A belief state can be thought of as a most probable state among the different possible states in the state world and is a function of previous belief states, actions and observations. It can also be viewed as a weighted average of the states in the state world, where the weights represent the belief in a given state.

The paper aims at deriving results for POMDPs similar to MDPs. The concept of value functions and value iteration is considered first. We come across the notion of policy trees, which represent the different possible actions taken from some initial state and observations taken over the next t steps. We evaluate the values of these policy trees for states and then use them for belief states. Value function for states is treated as a vector and optimal t-step value function is estimated over a large policy tree space. Moreover the choice of the policy tree varies over the belief space. Then we consider the value iteration problem, which can be solved similar to MDPs. But the work required is huge and witness algorithm is used to do the task efficiently. But still they stop being efficient for large state and observation sizes.

Next the paper tries to illustrate the numerous properties of policies in the case of POMDPs. Here we consider the tiger problem the one with a tiger behind one of the two door and a reward behind the other. The properties corresponding to finite as well as infinite horizon based policies are discussed. The paper has also focused on the understanding and acquisition of the model describing the world. This can be combined with the learning of computation policy. The author also compares his work with the existing traditional AI planning based work. The comparison focuses

on transition models, observation models, Plan structures etc. A number of related work have also been considered in similar contexts.

The paper is very elaborate and passes over all the important properties and formulations required for POMDPs. Being totally new to POMDPs this paper helped me get a deeper understanding of the topic in both practical as well as theoretical sense. However I found it difficult to hold a grasp of a few topics especially some parts of the witness algorithm and also plan graphs.

PALASH RAJENDRA KALA

This paper is motivated from the practical problems of the operations research, problems like robot navigation, even after giving map of the corridors. The main challenge is to correctly determine the current location of the robot. The main aim of this paper is the overcome the challenge not exactly by finding the exact location, but by approximating and finding the correct
move at this approximate position. It uses the history of the movement of the robot and the rewards that the robot got in the past. For this paper, the location is the state in which the robot is.

It so happens that the state that we are able to observe maps to a limited number of states out of all the states and hence the move selection strategy cannot be stationary. Is it is stationary then on all the observed states that map to the same state, a particular move will be chosen always which is not desirable for the problem.

Problems like the one above can be termed a Partially observed Markov decision Process (POMDPs). They are MDPs in which states cannot be fully observed. Current selection of action depends only on the current real state and hence it is an MDP. So, POMDP can have some more actions which can help it get more information about the state of the current state.

A very good example of POMDP given in the paper is of the tiger problem. In this problem, the agent has two doors and there is a reward behind a door and a tiger behind another door. The agent has to choose one door. But the agent does not know which door has a tiger behind it. So we can formulate it as a POMDP where the agent can take one more action of listening to the voices behind the door. This action's results are probabilistically true. But there is a cost
attached to listening and hence the agent has to balance and create a strategy on when to listen and when to choose a particular door, after how much sure he/she becomes.

The paper starts by explaining value iteration and policy search algorithms that we learnt in our last course. Then it comes to describe what actually is partial observability. The number of observations that an agent can make in POMDP case is lesser than the actual number of states. An POMDP agent can be decomposed

into a state estimator and a policy. State estimator tries to estimate a state from the observations and this state is called belief state.
The policy strategy is based upon what the belief strategy is.

The major algorithm described in this paper to solve a POMDP is the witness algorithm. The witness algorithm is similar to the value iteration algorithm in the outer structure. The inner loop of the witness algorithm includes identifying a witness, checking the witness condition and single step of value iteration at the end. In the tiger algorithm described earlier, this paper explains the solution obtained in the form of wonderful strategy graphs. This graphs are like state machines which asks the agent to listen till
they become sure of the correct door with some probability. The number of listen actions to be performed before selecting a door highly depends on the correctness probability of the listen action. The two examples are given in the paper for correctness probability of listen action of 0.85 and 0.65. These examples help us understand about its importance.

Overall, the paper describes the POMDP in a very nice way by starting at value iteration algorithm and then explaining what actually POMDPs are. After this the paper explains an algorithm to solve POMDPs and finally a real life example of POMDP.

DEPEN MORWANI

The paper talks about solving MDPs when we do not have full information about the state of the world, i.e. about solving POMDPs(Partially Observable MDPs)
A POMDP along with (S,A,T,R) also consists of a set of observations that the agent can experience and an observation function, which gives, for each action and resulting state, a probability distribution over possible observations. In a POMDP, an agent doesn't know the true state of the world. Instead, there is a concept of belief state that summarizes its previous experience. In this case, belief state is a probability distributions over states of the world.
These encode all information from the past and thus we can say POMDP is a Markov process for belief states.
Here also, The aim is to maximize expected long-term reward. This can be done by obtaining the policy which needs to be followed for each possible belief state. This means plotting all possible policies for different belief states and picking the highest one at each point. It has been proved that the shape so obtained is convex. After this, the paper describes an algorithm(Witness algorithm) for finding the optimal policy by estimating the policy
trees. However, experimental results suggest that even the witness algorithm becomes impractical for problems of modest size ($|S| > 15$ and $|\Omega| > 15$). Now, The authors are exploring the use of function-approximation methods for representing value functions and the use of simulation in order to concentrate the approximations on the frequently visited parts of the belief space.
Another area that is not addressed in this paper is the acquisition of a world model. One approach is to extend techniques for learning hidden Markov models to learn POMDP models. Then, algorithms of the type described in this paper can be used to the learned models. Another approach is to combine the learning of the model with the computation of the policy. This approach has the potential significant advantage of being able to learn a
model that is complex enough to support optimal (or good) behavior without making irrelevant distinctions.
Now, I will conclude with saying that the paper explores a nice approach for solving POMDPs and it is complete with respect to practical and theoretical tests. I really did like the concept of belief state, i.e , not the true state, but an accumulation of all the history information represented in terms of distribution of world states. Also, I think, in this case, reward shouldn't be just related to our final goal, but it should also

include how much information regarding the world we are getting out from taking the particular step.

Rohit Kumar

The paper talks about partial observable stochastic domains. In many situations like robot navigation, the agent is unable to determine its current state with complete reliability. It only achieves partial information about the state it is currently in and based on that it has to decide the proper action to maximize its expected future reward.

The naive approach to this for the agent can be to map the most recent observation of the state directly into an action without remembering past consequences for the same. This approach leads to performing the same action for every state whose observation is same. Other approaches which use current observation only (e.g. adding randomness to observation) can't be truly effective ever and there is a necessity to use memory from previous actions and observations to add clarity of the states.

The formulation of POMDPs build upon MDPs in which where states are clearly known. Apart from the states, actions, transition function and reward function, POMDPs consist of a finite set of observations that the agent can experience from the environment and a probability distribution over the observation space for each action and resulting state pair. The agent's goal remains the same i.e. to maximize the discounted future reward.

The paper decomposes the problem of controlling a POMDP into two parts. It introduces a new entity called belief state which summarizes the previous experiences of the agent. A new component called state estimator is responsible of updating it based on the last action, current observation and previous belief state. The policy which generates actions is modified to take this belief state as input rather than the original state of the environment. The belief state can be formulated in many ways. In the paper it is described as a probability distribution over the original states of the world. The formulation/encoding ensures that they comprise of a sufficient statistic of the past history. If the belief state is provided, no additional data about its past actions or observations would provide any further information which implies that the process executed over belief states is Markovian in nature.

The probability distribution for belief state is computed using Bayes theorem and basic probability concepts using previous probability distribution (belief state). For

calculating the optimal policy, the new state space is made of belief states (which is a continuous space), the action set remains the same, a new transition function is defined which considers transition from one belief state to other for a particular action. The reward function is constructed from the original reward function using the probability distribution of belief state. This belief MDP is made such that an optimal policy along with the correct state estimator (belief state) will give optimal behavior.

Solving a general continuous space MDP is difficult in general, but the optimal value function for the belief MDP has some special properties which has been exploited in the paper. For calculating the value function, value iteration is used over belief space. It is approximated by the optimal t-step discounted value function which is represented by a policy tree. The top node determines the first action to be taken and eventually depending upon the observation, a proper branch of the tree is chosen which determines the next action and so on. Several optimizations over this have been suggested, like the witness algorithm. Finally the paper talks about how to interpret the policy obtained, both for finite and infinite horizons.

Overall the paper provides a good start to the world of partial observability and suggests a lot of ideas which can be taken further to built upon. There can be a lot of extensions e.g. using hidden Markov models etc. The approaches focus both on performing actions taken to gain information from the environment as well as to fetch rewards.

ADIT KABRA

The paper is aimed at giving a good planning approach to Partially observable stochastic domains using the concepts from operations research. from the POMDP perspective, optimal performance involves something akin to a “value of information” calculation, only more complex; the agent chooses between actions based on the amount of information they provide, the amount of reward they produce, and how they change the state of the world.

There is an initial discussion about MDPs and planning and control in MDPs. Then a disussion on partial observability and value functions for POMDPs. The sections on POMDPs was too long, so I will just state the conclusions I could understand from them.

The POMDP model provides a firm foundation for work on planning under uncertainty in action and observation. It gives a uniform treatment of action to gain information and action to change the world. Although they are derived through the domain of continuous belief spaces, elegant finite-state controllers may sometimes be constructed using algorithms such as the witness algorithm.

Even the witness algorithm becomes impractical for problems of modest size. This paper explores the the use of function-approximation methods for representing value functions and the use of simulation in order to concentrate the approximations on the frequently visited parts of the belief space. The results were very good and a very good solution to an 89-state, 16- observation instance of a hallway navigation problem was found. These techniques can be extended to get good solutions to large problems.

Another area that is not addressed in this paper is the acquisition of a world model. One approach is to extend techniques for learning hidden Markov models [50,60] to learn POMDP models. Then, we could apply algorithms of the type described in this paper to the learned models. Another approach is to combine the learning of the model with the computation of the policy. This approach has the potential significant advantage of being able to learn a model that is complex enough to support optimal (or good) behavior without making irrelevant distinctions

Harish Koilada

Introduction

The reinforcement learning setting which we considered till now assumes that the agent in question has perfect perceptual abilities. In practice most of the time, the agent is able to sense the state it is in with uncertainty. These domain of problems are known as Partially Observable Markov Decision
Problems (POMDPs). An action taken by agent must reflect both change of it's current state as well as information gain regarding the current state.
The paper discusses the MDP framework using both finite and infinite horizon discounted problem.
In the case of infinite horizon, the optimal policy is a stationary policy (does not change with time). Where as the finite horizon optimal policy is a non-stationary policy. A variation of the value iteration algorithm in case of infinite horizon is used for calculating the finite horizon optimal policy.

Partial observability

In case of partial observability, one approach is to take the same action for all similar perceived states. This can be further improved using a probability distribution of actions for all similar looking states. For the optimal policy in such a case we need to use past observations and actions to distinguish between these states.
A POMDP is an MDP in which the agent makes an observation of the environment based on the action taken and the resultant state due to the action where Ω is a finite set of observations the agent can experience of it's world and $O: S \times A \rightarrow \Pi(\Omega)$ is the observation function which maps for each action and resultant state, the probability distribution over possible observations.
An agent interacting with a POMDP makes observations and takes actions. There are two components in the agent, the State Estimator and a policy. It also keeps an internal belief state that summarizes the previous experience. The state estimator updates the belief state based on last action, current observation and previous belief state. The belief state is a probability distribution over the states of the world. If the current belief state is properly computed, no additional amount of past observations and actions can improve the reward received. Hence the process over belief states is Markov. The policy now must map the current belief state into action. Now the optimal policy is a solution of the continuous space belief MDP. The states in belief MDP are the set of belief states possible.
The transition function and reward function must also be calculated appropriately using these belief states. The optimal solution for belief MDP coupled with the correct state estimator will lead to optimal behavior for the original POMDP. Solving

a general continuous space MDP is very difficult but the belief MDP has special properties that make it easier to be solved.Value functions for POMDPs
Similar to discrete MDPs we need to find the optimal value function, but an approximation of it since the state space is continuous, and find the greedy policy with respect to this optimal value
function. The approach used is value iteration for t-step discounted value function over the belief space. Finally the paper presents the witness algorithm which uses policy trees to find the optimal policy.

stochastic domains" covers the framework and difficulties of partially observable and stochatic domains. Partially observable stochastic domain considers the problems where state is partially identifiable or non-identifiable by the agent. In such domain, we are restricted to consider the probability of being in different states as an information to the agent rather than exact knowledge of state in which agent is. Author explains frameworks for both completely observable and partially observable environments known as MDP and POMDP. POMDP is extension of MDP only, with certain complexities added. Basic intuition and principles remain same. Additional difficulties and tackles has been discussed at the end.

MDP are discribed as a tuple of states,actions,transition function and reward function. In any state of the agent, it is aware of transitions for a particular actions and corresponding rewards. For any policy, we can find values of states, q-values of the state-action pairs and improve the policy iteratively. The process of improving policies and finding optimal policy iteratively using state values is called "value-iteration algorithm". Policy returned by value iteration is proved to be optimal under markovian constraints.

POMDP is discribed as a tuple of states,actions,transition function, reward function, observation set, observation function. Observation function gives probability distribution over set of possible observations after which the action taken leads to given state. Goal of the agent is same as MDP, "maximize the expected discounted reward". POMDP can be divided into two parts, "belief state" and "state estimater". Belief state can have several representation but the choice author has used is probability distribution over set of states. State estimator is responsible for update belief state based on observations. Belief state computation is simple conditional probablity based. Transition function and reward function are now modified to accomodate distribution of probabilities over rather than state only. But the change is simple generalization of MDP only. Policy and reward are now chosen based on belied state not simple state. Reward is expected value of a belied state.

Value function is first calculation for a particular state in belief state first and then expected value over states in belief state is the value function of belief state. Value function of s state in belief state is calculated by an equation similar to MDP but extended for belief states. But the problem is, at any time, there can be number of policy trees under the root of current belief state node. So we have to find the set of

policy trees which are optimal. But these seem to form convex function which are piecewise in discounted value space. The values at the boundaries of such regions make it easier to choose policies but in the middle, it is harder to choose between policies. There are several methods to compute value function in higher dimesional spaces efficiently which are based on prunning and parsimonious representation of value function.

There are several problems which are not properly tackled by POMDP. Like the tiger problem which poses a problem that no matter what state the agent is in, the action have same probablity of getting choosen. The agent won't be able to choose any of these actions. Further the finite horizon problem dictates that after certain number of actions, the POMDP model will reset belief state vector to initial value and it will be like starting from scratch again. So there are no conclusive evidence to learn optimal policy beyond the finite horizon of steps.
Another problem is that of infinite horizon problem where the precision of the algorithm used to calculate the situation-action mappings can no longer discern any difference between the vectors values for succeeding intervals.

Plan graph is another data structure to store information rather than use belief state. Plan graph work as a finite state controller. But the problem is that the size can shoot up tremendously with decrease in observation reliablility.

Since POMDP is much complex framework, complexity of algorithms increases depending upon observation reliability and belief set configuration. But the framework itself is very well structured and presented in this paper very nicely. There has been no mention of state estimator design and its updates. Will it be domain specific or can be a very generic one?

In the examples given, robot is assumed to have unreliable sensors which makes the problem partially observable. However, if sensory inputs are perfect and its design or environment does not allow for fully observable states, how will be quantize the unreliability in observations?

Ankith M.S.

Authors begin this paper by taking the example of robot navigation in a large office building to explain the partially observable states and how a robot should take actions during uncertainty, should we ignore the uncertainty and take action or purpose of choosing action should be gathering information. Why not both?. In POMDP there is no distinction is drawn between actions taken to change state and action to gain information because performing two different operations may delay the process end goal. So in POMDP, actions are chosen based on the amount of information they provide, the amount of reward they produce and how they change the state of the world.

They before explaining the POMDP, by giving the brief introduction on MDP as this will serve as a basis for POMDP. MDP is a model which interacts with the world by taking the state as input from them and responds with actions. In MDP, there is uncertainty about agent's current state. Author restricted their algorithms to finite state and action space. In order to measure the performance of the model, authors used expectation of the discounted reward in an infinite horizon.

Authors also explained about stationary and non-stationary policies. In the finite-horizon model, typically non-stationary policies are implemented. They explained on Value function. Explained to get the value function given the policy by solving the set of the linear equation and explained to compute greedy policy based on value function.

In order to compute optimal policy, authors explained value iteration method since it will also serve as the basis for finding policies in the POMDP. In value iteration, policy converges to optimal policy long before value function converges to the optimal value.

Partial observability, when the system is not able to determine the state it is currently in with certainty. The slightly better approach than naive way is to add randomness to the agent's behavior i.e probability distribution over actions. The author also claims "in practice deterministic observation-action mappings are prone to get trapped in deterministic loops". The actor also claims in order to behave effectively in an uncertainty, it is necessary to use a memory of previous actions and observations to aid in the disambiguation of the states of the world.

Authors described the POMDP framework by extending MDP framework. Added the finite set of observations the agent can experience of its world, observation function. Agent makes an observation based on the action and resulting state.

Authors approached POMDP by dividing the problem into parts, state estimator- which outputs belief state b summarizing the previous observation and policy estimator- chooses an action based on belief state. A belief state is a probability distribution over State. The state estimator must compute a new belief state, b 0, given an old belief state b, an action a, and an observation o. The optimal policy is found based on belief state.

In order to optimal policy, optimal value function is required. So in POMDP, policy trees are used to approximate value function. Authors gave the geometric interpretation of value function evaluated using policy tree. In two world states, belief state is the line. In three world states, the belief space can be seen as the triangle and value function associated with a single policy is a plane space. The author proposed many algorithms to get optimal value, some are value-iteration-just like in MDP and more complexity, in order to tackle complexity introduced the Withness algorithm by generating elements directly instead of constructing the superset and pruning. In a practical scenario, it runs faster. There are alternatives approaches also, one of them the one-pass algorithm works by identifying linear regions of the value function one at a time. Another approach is to find Q functions instead of the complete value functions.

The author took the simple example on tiger behind two closed doors to illustrate the properties of POMDP policies. They explained how policies behave in finite horizons especially how policies change their decision in time steps and in infinite discounted horizons, the optimal situation-action mapping for large t looks much the same as the optimal situation-action mapping for t-1, unlike in finite horizon case. Authors are not aware whether the value iteration algorithm mentioned in the paper will solve all POMDPs with finite transient optimal policies in finite time.

One of the drawbacks of POMDP is maintaining the belief state and if state space is large, computation will be expensive. To overcome this problem, they suggested plan graphs which do not use any explicit representation of the belief state. Plan graph is a finite-state machine, the current node is a sufficient representation of the current belief.

Authors before concluding the paper they explained how earlier work of AI-assisted in the assumption they made in POMDP model in terms of knowledge representation, transition model, objection function and so on. Authors concluded,

their current work explores the use of function approximation methods for representing value functions and simulation to concentrate the approximations on the frequently visited parts of the belief space. They also claim that their work gave good solution where witness algorithm was impractical to use.

Eeshan Malhotra

In a conventional MDP, complete information is available about which state the system exists in at any given point of time. A POMDP generalizes this to the case where only a probability distribution over all the states is available, indicating the likelihood of being in each state. Two factors reduce the uncertainty here - first, states are not completely unobserved. Certain factors impacting the state may be observed by the agent. These change the probability of being in a given state. Second, the agent stores some information about its previous states (also known only probabilistically), and the actions taken. This also reduces the uncertainty, and gives a better estimate of the probable current state.

Initially, it seems that the agent may want to sometimes take actions that further its goal (achieve higher reward), and at other times, actions that provide more information about the environment to better estimate the current state, and therefore, in the future, take a line of actions more specifically suited to the state(s). However, in the formulation that is proposed, both of these objectives are achieved together, and there isn't a discrete dichotomy, but rather a spectrum between the two.

The framework for a POMDP is similar to that for an MDP, except that on taking an action a from a state s, the environment does not return a distinct next state s', but rather an 'observation'. An observation function is provided that gives a probability distribution over possible observations for the probability of making observation o given that we took action, a and ended up in state s'. Instead of maintaining a variable for current state, we maintain a 'belief state', that is a probability distribution over all states, indicating our belief probability of being in each state. It is clear that given the current belief state, and action, and an observation, we can calculate the subsequent belief state easily (though not trivially).

Analogous to an MDP, where we associate a value function with a given starting state and a policy to be followed, in a POMDP, we want to associate a value with a given belief state, and a policy tree to be followed. It is essential to introduce policy trees because with a finite horizon, the optimal set of actions may not be static. That is, it is possible (and likely) that the optimal policy is non-stationary.

The paper describes an algorithm for computing the Q-value (an altered definition from the one we're used to, to allow for non-stationary policies, and limited horizon),

and thereby also computing the optimal action to take at any given point of time. Note the since we're considering finite horizons, Q is not just a function of the current belief state and an intended action, but also of the number of time steps till the horizon, i.e. Q(a, t, b). Enumerating all policy trees and selecting the best is clearly an intractable approach. The paper describes an algorithm known as the Witness algorithm, which runs in polynomial time, given certain constraints. While these constraints are certainly not trivial, in practice, these are often found to hold. (A polynomial time algorithm for a general case will imply NP = RP.) The witness algorithm employs linear programming on a minimal set of policy trees, with a reduced belief state space to search for an optimal.

The approach of using a belief state seems natural and sound, but is definitely not the only possible approach. It is possible to select actions without explicitly representing the current state as a probability distribution. While the paper describes this approach of plan graphs briefly, it is not clear how this would work to discover the optimal action for each state.

In some sense, a POMDP is a continuum generalization of MDPs: A 'state' essentially is distinctly defined if taking an action results in the same outcome, irrespective of other factors. In this sense, a belief state is also a 'state'. However, even with a finite number of actual states, the number of belief states (probability distributions) that can exists is an infinite spectrum. With this in mind, it may be possible to explore the idea of mixture models to find optimal actions in a POMDP, where the uncertainty in state information is transferred to the uncertainty in the transition function, and an infinite number of belief states are represented as mixtures of unitary states, for which the optimal actions have already been discovered.

Ashish Ramteke

Authors suggest that we are not much interested in deterministic problem where we can plan sequence of actions that needs to be taken , rather we want a mapping from situation to action that specify the agents behavior. Adding further they say the goal is not to have full policy what is needed is method to developing partial policies and conditional plans for completely observable domains. Authors explicitly states at the very beginning a weakness of their method is that they require the states to be represented enumerative rather than composition representation such as Bayes nets or probabilistic operator description.

To start with they gave a brief overview of MDP, explaining action optimality in terms of finite horizon and infinite discounted. They later describe Stationary policy and non-stationary policy and way to calculate value function for that given policy. They then touch upon calculating optimal policy with value iteration algorithm with just a basic introduction of how algorithm works.

Partial Observability: In case of MDPs we have seen a way to calculate optimal policy based on the current state but what if the agent is not able to determine its state correctly. Authors suggest that performing the same action at every state that looks the same hardly result in any promising result, instead better result can be obtained by adding randomness to the agents behavior, i.e. a policy can be mapped from observation to probability distribution over actions. Authors claim that in order fr agents to act effectively in partially observable world they need to use memory of pervious observations and action so as to distinguish states. In case of POMDP the agent makes observation and generates action, it keeps belief state based on which it determines its current state. Belief state summarizes its previous experience. And the state estimator is responsible to update the belief state. As in case of MDP pi is the policy use to for generating action but here its based on belief state. With sufficient information of the past history and initial belief state of the agent the process over belief state act as Markov.

Authors then describe way to calculate belief state which is just a probability distribution over states b(s) degree of belief in some state s is just probability of s given observation, action and previous belief state. Then based on the belief state tries to find optimal policy, i.e. it must map the current belief state into action for determining optimal policy they define transition function and reward function in terms of belief state. Optimal policy can be directly determined based on optimal value function, so authors then describe how value function can be obtained. Based on the action it can be represented as a t- step policy tree, where the first node selects the action that needs to be taken then based on the resulting observation next action is determined. So value function can be written as,

V(s)= R(s,a(p))+gamma x expected value of the future
Where a(p) is action selected at the top of the policy tree.
Now as the agent will never know the exact state its value function is calculated based on belief state as,
V(b)= sum over all state (b(s) X V(s))
Authors claim that its possible that every policy tree represent the optimal strategy at some point in the belief space and they each can contribute to the computation of optimal policy. They further suggest that it is possible to define the same value function defined using a given set of policy tree using a unique minimal subset of given policy trees. The value function can be computed using value iteration even for POMDPs but the problem is in computing minimal subset of policy tree at t step form minimal subset from t-1 step. Now one possible approach would be to do exhaustive enumeration by constructing a large representation for t step and the pruning it. The idea is to take t-1 step policy tree and compute t step policy tree , now the t- step policy tree consist of a root node with an associated action a and each of a t-1 subtree.
They introduce witness algorithm to improve the complexity of the value iteration algorithm , where they suggest that we must avoid exhaustive enumeration instead we should generate the elements of t step value function directly. They further explain witness function through outer loop, inner loop and identifying a witness where in outer loop for each action in t-1 step a set of minimal policy trees are take and combined and then any extraneous policy tree is pruned to get a set of useful t step policy . In inner loop for every action and for some belief state Qt is compared with estimated value of Qt and if they are not equal they our set of policy tree is not perfect. The running time of single pass of value iteration using the witness algorithm is bounded by a polynomial to number of factors like size of state space , action space, number of tree representation in t-1 step and few more.
The authors then goes on to state few more algorithm some of which even perform better than witness like incremental pruning algorithm which runs faster by repeatedly pruning out non useful policy trees during the generation procedure.
Authors then by help of example tries to explain how it work, in case of finite horizon and infinite horizon where in case of finite horizon its better to listen at t=2 so that he can take action with some certainty in next step, while in case of infinite horizon as t increases, the rewards received for the final few steps have less influence on the situation action mappings for earlier time steps and the value function begins to converge. They then introduce the concept of plain graphs where its not require to maintain the belief state and all the information can be maintained in the current sate itself reducing the memory overhead.
To end with the author suggest that the witness algorithm is impractical for problems with large size, to overcome that they are trying to explore the use of function approximation methods to represent value function and use of simulation to

concentrate on approximating to frequently visited belief state. Their implementation have achieved some good results but still not applicable to very large problems.

INDRADYUMNA ROY

This paper is a comprehensive introduction to the problem of planning in partially observable stochastic domains.

POMDPs are a framework for agent planning under uncertainty. Unlike MDPs where actions are stochastic but states are fully observable, in POMDPs the states are not fully observable. Instead, we have partial observability and the states must be inferred from the observations. Additionally, there may be uncertainty about the outcomes of the actions. Hence there is an aspect of information collection, and we perform tracking in addition to decision making.

The mapping between MDPs and POMDPs can be stated as such that each MDP state is a belief state from the POMDP. Since the belief states are continuous and infinite, planning is P-Space complete. Algorithms usually trade off plan quality for computational speed. Typically an assumption is made that there are a discrete number of states, actions and observations. An observational model relates observations to states.
Tracking in POMDPs is done by maintaining a number of probability distributions over the uncertainties. These are the POMDP parameters, namely belief states updating distribution, observational probabilities and transitional probabilities.

The agent keeps an internal belief state that summarizes its experience. The agent uses a state estimator to update its belief state based on latest action and latest belief state. The belief state is a sufficient statistic, since it satisfies the Markov property.
Therefore, since the POMDP is seen as a continuous space belief MDP as the agent's beliefs are encoded through a continuous belief state, we can solve the belief MDP using the usual algorithms, such as value iteration.
However, some adaptations to the algorithms are needed. Generally, it is very hard to solve continuous space MDPs. DP updates cannot be carried out since there are uncountably many belief states. We cannot enumerate all equations of the value function.
However, some special properties are exploited to simplify the problem using policy trees and piecewise linear and convexity properties.Subsequently, we can find

approximation algorithms to construct the optimal t-step discounted value functions over the value space using value iteration.

The paper uses the tiger problem, to demonstrate the POMDP planning methodology. There are two states corresponding to two doors, one on the left and one on the right . Behind one door is a tiger and behind the other is a treasure. The goal of the agent is to open the door with the treasure behind it. There are three possible actions, opening the left or the right door, or listening. Listening is the information gathering action that does not open a door, but provides a noisy observation about the position of the tiger. Rewards of the problem are constituted such that the agent wants to open the door with the treasure.To increase the chances of opening the correct door, at each time step, the agent must receive an independent observation and update its beliefs regarding the position of the tiger. The observations, which are the results of the information gathering actions, are noisy. Thus, the tiger problem serves as an effective framework for studying the various aspects of POMDPs, including uncertainties in observations as well as uncertainties in results of actions.

BHAMBARKAR GANESH BHAGWAN

This paper applies techniques from operations research to the problem of choosing optimal actions in partially observable and stochastic domains. The paper starts off by introducing the Markov Decision Process (MDP), and then introduces us to POMDPs.

Partially observable Markov decision process:
In POMDPs, the agent is not reliably able to determine the state it is currently in. The agent only has access to observations. The agent needs to be able to remember the previous observations and actions to be able to determine future actions to be taken. The agent's goal, as in MDPs, is to maximize its future reward.

A partially observable Markov decision process can be described as a tuple <S, A, T, R,
Ω, O>, where
• S, A, T , and R describe a Markov decision process;
• Ω is a finite set of observations the agent can experience of its world; and
• $O : S \times A \rightarrow \Pi(\Omega)$ is the observation function, which gives, for each action and resulting state, a probability distribution over possible observations

The authors decompose the problem into two parts: belief state and state estimator. The belief state summarizes the previous experience of the agent. In this paper, for the belief state, they use distributions over the states of the world. The authors claim that this information is enough to track all the observations in the past. No additional information more than this from the past will provide more information about the current state of the world. The state estimator takes an action, observation and current belief state and outputs changed belief state.

Policy trees:
Policy tree is a representation of the POMDP which stores actions at nodes and observations at the edges. The top node determines the action to be taken, then depending on the observation, the edge corresponding to that observation is followed.

The authors introduces an innovative algorithm called withness algorithm. I did not quite understand it yet, but it is an improvement over previous algorithms in that it is a polynomial time algorithm in terms of the input and ouputs. The witness algorithms is also faster over a wide range of problem sizes.

Some properties of POMDPs are described in the paper:

1) Finite-horizon policies:
Finite horizon policies are, as I understand, policies in which fixed t number of actions are taken.
2) Infinite horizon policies:
These policies incorporate a discount factor so that as t increases, the rewards received in the later steps have lesser influence on the agent's decisions.
3) Plan graphs:
In many cases, explicit representation of belief states is not required. The policy can be encoded in the form of graphs called plan graphs.

SAMIRAN ROY

A POMDP models an agent decision process in which it is assumed that the system dynamics are determined by an MDP, but the agent cannot directly observe the underlying state.

Key Concepts for formulating and solving the POMDP Problem:

1) Deterministic Policies give suboptimal results:

The paper uses a stochastic policy (mapping from observations to probability distributions over actions). The argument is that randomness effectively allows the agent to sometimes choose different actions in different locations with the same appearance, increasing the probability that it might choose a good action; in practice deterministic observation-action mappings are prone to getting trapped in deterministic
loops. Also, when properly computed, they comprise a sufficient statistic for the past history and initial belief state of the agent: given the agent's current belief state.

2) Observation Function

Apart from the States, Actions, Rewards, Policy of the normal MDP, we have in the pomdp:

$O : S \times A \rightarrow \Pi(\Omega)$ is the observation function, which gives, for each action and resulting state, a probability distribution over possible observations (we write O(s', a, o) for the probability of making observation o given that the agent took action a and landed in state s').

3) Belief State and State Estimator Function

It keeps an internal belief state, b, that summarizes its previous experience (Since it does not know the actual state). State Estimator is responsible for updating the belief state based on the last action, the current observation, and the previous belief state. The reward function ois based on how good the agents estimate of this belief state is. Since it is estimated from actual data, the belief state represents the true occupation probabilities for all states and therefore the reward function represents the true expected reward to the agent.It has been proved that the belief MDP is such that an optimal policy for it, coupled with the correct state estimator, will give rise to optimal behavior (in the discounted infinite-horizon sense) for the original POMDP.

How to solve the POMDP?

Representing a POMDP in the form of a policy tree gives rise to an important property - Each policy tree p induces a value function Vp that is linear in b, and the collection of these functions is is piecewise-linear and convex.

The problem then becomes:

Given a piecewise-linear convex value function and the t-step policy trees from which it was derived, it is straightforward to determine the optimal situation-action mapping for execution on the tth step from the end. The optimal value function can be projected back down onto the belief space, yielding a partition into polyhedral regions. Within each region, there is some single policy tree p such that $b \cdot \alpha_p$ is maximal over the entire region. The optimal action for each belief state in this region is a(p), the action in the root node of policy tree p;

Value iteration and Witness Algorithm is used to then solve the POMDP.

Witness Algorithm Intuition:

A vector at a belief point is constructed by selecting a transformed vector from V for each observation. The Witness Algorithm starts with an arbitrary belief point and generates its vector. It adds this vector to a set and assumes that this set is V' it then goes about trying to prove or disprove whether or not this set truly is V'.

It does that by constructing a vector and looking at the choices made with respect to a future belief strategy, observation by observation to see where a different choice would yield a better value. It defines the region where it is assured that the particular

choice is best. If it can find a belief point where a different strategy would be better, uses this fact to eliminate V vectors.

This simply reduces the search for a point to the region over the current approximation we have.

The paper also illustrates the working of a POMDP, based on the tiger problem

Some Important points Discussed:

1) The optimal t-step value function is always, piecewise-linear and convex. but is not necessarily true for the infinite-horizon discounted value function .It remains convex, but may have infinitely many facets. It can be approximated arbitrarily closelyby a finite-horizon value function for a sufficiently long horizon

2)One drawback of the POMDP approach is that the agent must maintain a belief state and use it to select an optimal action on every step; if the underlying state space is large, then this computation can be expensive. In many cases, it is possible to encode
the policy in a graph that can be used to select actions without any explicit representation of the belief state - In the form of plan graphs. A plan graph is also called a finite-state controller. It uses the minimal possible amount of memory to act optimally in a partially observable environment.

Discussion:

POMDP have been applied to a wide range of problems - Machine Maintanance, Robot Path Planning, Elevator Control, Machine Vision, management of patients with ischemic heart disease, poker. It is impossible to solve many of these problems exactly, but the above methods that approximate solutions are often sufficient for real life applications

A Siddharth

Markov decision processes (MDPs) form the core of reinforcement learning. But, in most situations, the agent does not have the capability to disambiguate between different similar states of the environment. That is, it is not able correctly identify trivially using just the initial state, but needs observation of the environment to help it do so. These observations, unlike the other actions that the agent takes, do not alter the state of the environment, but provides us useful information about the current state, in contrast, the regular actions change the state but provide no additional information about the environment. Moreover, the observations are noisy, and may not help the agent determine the state completely. Hence, the agent maintains a belief over the set of all valid states, denoting the probability of it being in each state.

The authors first begin by covering the necessary background about MDPs, value functions and value iteration, which is required to understand the theory leading to several different algorithms used to solve the planning problem of POMDPs. We are also introduced to the concept of planning a non-stationary policy, in the case of a finite horizon MDP. The authors elaborate, that the optimal policy of a finite horizon MDP, can be non-stationary, i.e., the mapping between state and action may change at different points in time, as the way the agent chooses the last step of its life (to maximize the immediate reward) is way different from the agent chooses its initial actions(to maximize long term reward). However, this distinction does not exist in infinite horizon discounted MDPs, as at each step, the agent still has an infinite number of steps to execute before the end, making the policy stationary. The authors then describe the structure of the optimal value functions and action value functions, and reason as to why they should be piecewise continuous and convex. However, this is true only for finite horizon POMDPs, as the optimal value functions of may not be piecewise continuous, due to it being a combination of infinite policies, and thus need to be approximated.

The authors discuss two algorithms to solve this planning problem for finite horizon POMDPs, the one step value iteration algorithm and the witness algorithm. Both these algorithms compute the t-step policy trees recursively using the (t-1) step

policy trees. The step-value iteration algorithm does this by enumerating all possible t- step policy trees and pruning out the non-useful ones. This algorithm runs in an exponential time on the input and hence is very slow, but this does ensure the discovery of the optimal t-step policy. The witness algorithm on the other hand, approximates the value functions (similar to what is done in value iteration for MDPs). Instead of enumerating each t-step policy tree, different mutations of existing policy trees are evaluated to discover the existence of a witness. If no witness is found, this algorithm terminates. The witness algorithm runs in a time polynomial in input and output and in many cases out performs the one-step iteration algorithm.

The authors end by illustrating an example, explaining the intuition behind how policies are generated/planned, and how it may be possible to contain the policy to an infinite horizon POMDP in a finite state controller, when the policy itself is representable using a finite number of policies.

Deepak Garg

The paper introduces us to the field of POMDPs. A partially observable Markov Decision Process is a generalization of MDPs. In POMDP's, the system model is based on MDP, but there is an uncertainty for the agent about which state it is in. So, a probability distribution is maintained over a set of states based on observation and observation probabilities and the underlying MDP. The author gives the example of robot navigation to show the uncertainty of agent about states. In POMDP's there generally is no difference between actions that changes states and that gain world information. The agent chooses between actions based on amount of information they provide. In POMDP, instead of having accurate current state, the agent predicts its state based on action and resulting state. The agent's goal is to maximize expected discounted reward.

In order to estimate it current states, there are internal belief states that summarizes agents previous experience. The state estimators updates the belief states. The policy now becomes a function of agent's belief states. A belief state is a probability distribution over S and can be computed simple using the given model, ie, the previous belief state, action and observation. The optimal policy of POMDP maps the current belief state set to the actions. The reward function may seem to reward the agent for predicting that it is good states turns out to be a representative of the true expected reward to the agent.

Just like MDPs, we can compute value functions for POMDPs. One way is to use a policy tree to represent agent's non-stationary t-step policy. The optimal t-step value of starting in belief state is the value which is obtained by executing the best policy tree in that belief state. There are low and high entropy states and by paying some "value of information" agent can travel from low to high entropy states. At times, value function of a policy trees are associated with value functions of other policy trees. One can prune the policy trees whose value functions are totally dominated by other policy tree or a combination of trees. The Value function for POMDP can be computed using value iteration but the problem is of representation of V t . One way to resolve is to construct a large representation and then prune, ie, generation and pruning. The author says, this approach works more than needed and then gives witness algorithm for this problem.

To improve performance, generate the elements of V t directly. Witness algorithm says compute Q-value of t-step policy trees with action 'a' at root, for each action 'a'. Witness algorithm is polytime(in number of policy trees in Qt) algorithm for computing Q. These Q-functions are also linear and piecewise and can be represented by group of policy trees. We tend to find minimal set of policy trees for representing Q for each a at depth t. If at any iteration, there exists a belief state for which the one step look-ahead value is different from the estimated value. Such a beliefstate is called the witness of the fact that the estimator function is not yet perfect. For this
witness, a policy tree with action a at root giving best value is found. Repeat till no witness exists. The witness theorem provides a similar reasoning that we use for policy iteration. Witness algorithm is not the only algorithm. Sondik provides one-pass algorithm for the same but it is proven to be slow for problems with small state space. Sondik also proposed two-pass
algorithm which then was given no attention but was eventually found to be faster. There are other methods such as incremental-pruning algorithm, a change in reward function to have efficient implementation. The optimal t-step value function, piecewise-linear and convex, may not be the same for inifinite horizon problems. The optimal infinite-horizon discounted value function can be approximated using finite-horizon value function by taking long horizons.
If the underlying state space is large, then maintaining belief states and using it select optimal action at each step can be expensive. The solution could be the plan graph, where policy is
encode into graphs. The plan graph doesn't require online-representations of belief states, use the current node for current belief. The plan graph can be considered as finite-state controller, which uses least required memory to act in partial information environment.
Thus, POMDP model provides us with an insight of planning under incomplete information environment. POMDP models several real world sequential decision processes. This approach originates from operations research field and have been adopted by AI community. POMDPs have been used for aircraft collision avoidance, assistance for persons with dementia.

SUSHANT SHAMBHARKAR

Review of "Planning and acting in partially observable stochastic domains by L.P.Kaelbling et. al."

Markov decision processes (MDPs) form the core of reinforcement learning. But, in most situations, the agent does not have the capability to disambiguate between different similar states of the environment. That is, it is not able correctly identify trivially using just the initial state, but needs observation of the environment to help it do so. These observations, unlike the other actions that the agent takes, do not alter the state of the environment, but provides us useful information about the current state, in contrast, the regular actions change the state but provide no additional information about the environment. Moreover, the observations are noisy, and may not help the agent determine the state completely. Hence, the agent maintains a belief over the set of all valid states, denoting the probability of it being in each state.

The authors first begin by covering the necessary background about MDPs, value functions and value iteration, which is required to understand the theory leading to several different algorithms used to solve the planning problem of POMDPs. We are also introduced to the concept of planning a non-stationary policy, in the case of a finite horizon MDP. The authors elaborate, that the optimal policy of a finite horizon MDP, can be non-stationary, i.e., the mapping between state and action may change at different points in time, as the way the agent chooses the last step of its life (to maximize the immediate reward) is way different from the agent chooses its initial actions(to maximize long term reward). However, this distinction does not exist in infinite horizon discounted MDPs, as at each step, the agent still has an infinite number of steps to execute before the end, making the policy stationary. The authors then describe the structure of the optimal value functions and action value functions, and reason as to why they should be piecewise continuous and convex. However, this is true only for finite horizon POMDPs, as the optimal value functions of may not be piecewise continuous, due to it being a combination of infinite policies, and thus need to be approximated.

The authors discuss two algorithms to solve this planning problem for finite horizon POMDPs, the one step value iteration algorithm and the witness algorithm. Both these algorithms compute the t-step policy trees recursively using the (t-1) step policy trees. The step-value iteration algorithm does this by enumerating all possible t- step policy trees and pruning out the non-useful ones. This algorithm runs in an exponential time on the input and hence is very slow, but this does ensure the discovery of the optimal t-step policy. The witness algorithm on the other hand, approximates the value functions (similar to what is done in value iteration for MDPs). Instead of enumerating each t-step policy tree, different mutations of existing policy trees are evaluated to discover the existence of a witness. If no witness is found, this algorithm terminates. The witness algorithm runs in a time polynomial in input and output and in many cases out performs the one-step iteration algorithm.

The authors end by illustrating an example, explaining the intuition behind how policies are generated/planned, and how it may be possible to contain the policy to an infinite horizon POMDP in a finite state controller, when the policy itself is representable using a finite number of policies.