Learning Guide Unit 7

Site:	University of the People	Printed by:	Ryohei Hayashi
Course:	CS 4408-01 Artificial Intelligence - AY2025-T3	Date:	Thursday, 30 January 2025, 3:44 PM
Book:	Learning Guide Unit 7		

Description

Learning Guide Unit 7

Table of contents

- Overview
- Introduction
- Reading Assignment
- Discussion Assignment
- Programming Assignment
- Learning Journal
- Self-Quiz
- Checklist

Overview

Unit 7: Planning

Topics:

- Planning with Certainty
  - Representing States, Actions, and Goals
  - Forward Planning
  - Regression Planning
  - Planning as a CSP
  - Partial-Order Planning

- Planning Under Uncertainty
  - Preferences and Utility
  - One-Off Decisions
  - Sequential Decisions
  - The Value of Information and Control
  - Decision Processes
  - Value iteration for Markov Decision Process
  - Policy iteration for Markov Decision Process

Learning Objectives:

By the end of this Unit, you will be able to:

1. Examine the concept of a planning system and how it differs from classical search techniques.
2. Apply a decision network to solve a finite sequential decision problem using decision variables and utility.
3. Evaluate the Markov Decision Model on a fully observable environment for modeling sequential decision-making scenarios with probabilistic dynamics and value iteration.

Tasks:

- Read the Learning Guide and Reading Assignments
- Participate in the Discussion Assignment (post, comment, and rate in the Discussion Forum)
- Complete and submit the Development Assignment
- Complete an entry in the Learning Journal
- Take the Self-Quiz

Introduction

In previous units, we examined the process of 'reasoning' as it relates to artificial intelligence. In Unit 5, we looked at reasoning in closed world situations where we have complete knowledge about the state of the world. We learned that we can use propositional logic to 'reason' about worlds with complete knowledge using Abductive or Deductive processes. In Unit 6, we looked at reasoning in an open-world situation where we do not have complete knowledge about the state of the world. In an open world, we typically do not know or cannot know with absolute certainty the truthfulness of a proposition. When such uncertainty exists, we found that we can use measures of probability as a measure of the strength of our 'belief' in a proposition.

Reasoning with both certainty and uncertainty are processes that we use to make decisions based upon our knowledge of the world and its current state to achieve goals. The question is what decisions must be made and how are such goals achieved? The answer to this is through planning.

If you recall, in Unit 3 we learned how we can use search as a way to find solutions to problems. In Unit 4, we introduced the concept of constraints as most problems that we must solve have constraints that limit actions.

In this unit, we begin to put all of these concepts together to create an approach that agents can use to solve problems.

Planning introduces the idea that an agent must be able to reason about the future. In the previous two units, our reasoning processes assumed a point in time. We reasoned about the state of the world either from a position of certainty within a closed world or uncertainty. In both cases, however, the processes were based upon the world at a moment in time with a specific state.

We all know that the world, pretty much any world, is affected by time. Because time is a factor in every world and time continuously marches on, the state of the world will march on or change with it.

In the study of AI, planning is the decision-making process performed by intelligent agents like robots, or computer programs when trying to achieve a goal state. Planning determines a sequence of necessary actions and WHEN those actions are necessary to accomplish the goal.

Planning with Certainty

In the first part of this unit, we will explore planning when there is certainty of the outcome of a particular action, where the state of the world is known and therefore the outcome of any action is also known.

Planning with certainty follows the same set of rules as the reasoning with certainty did when we examined the concepts of propositional logic. Within a closed world where an agent has complete knowledge, the result of a particular action can be known with certainty. In this situation, the deterministic activity or activities required to achieve a goal are known from the perspective that for any state the action (resulting from a decision) required to achieve a goal is known. Planning is simply a process of organizing the actions to achieve intermediary or final states that lead to the goals of the agent.

When there is certainty there are a number of different ways to represent planning. These include Explicit State-space representations, Feature-based representations of actions, and the STRIPS (Stanford Research Institute Problem Solver, one of the first robots built using AI) representation.

In the explicit state-space representation, actions are represented in terms of a graph. Consider an agent designed to play tic-tac-toe. We learned in Unit 3 how problems can be solved by searching a graph. In the case of tic-tac-toe, we must generate a graph-based upon the current state of the game and then search the graph top find a move that will achieve the goal of winning the game.

Of course, after each move, the starting state of the game changes as the other player has the opportunity to make a move. The agent must then generate a new graph-based upon the current state of the game.

In the game of tic-tac-toe, every starting state of the game has a move that can be determined and the state of the game after the move is known. For this reason, the game represents planning with certainty for any player move.

The explicit state-space representation suggests that every possible state and action in that state is represented. This might be possible with a simple problem such as playing tic-tac-toe, but many problems are complex and the sheer number of states makes this approach unworkable in many situations.

The feature-based representation represents the problem as a series of rules that specify the precondition for an action and the effect of the action. This approach is very different from the explicit approach in that the state of everything doesn't need to be addressed. The feature approach provides a shorthand that addresses only actions that are available (precondition is met) and what the result of the action will be. The feature representation then needs two elements. First is a set of preconditions for actions and second are rules (causal and frame) that define the impact of the actions.

We can clearly see the link between propositional logic and planning with certainty in the structure of these rules.

Finally, the STRIPS representation is a simple representation approach that has some but not all of the capabilities of the feature-based representation. STRIPS is based upon defining preconditions and effects of actions. Its key limitation is the fact that it cannot directly model or represent conditional effects.

Each of these models provides the ability to model or define preconditions and actions and to determine the effect of an action, all of which are important to accomplishing a goal. Of course, they simply are the means to define actions and outcomes. Planning is a process of putting actions together to achieve a goal. In some cases, a goal can be achieved with a single action. In many cases, many actions must be performed to complete a goal. Developing these lists of actions requires a planning strategy. Our text explores a number of these planning strategies including forward planning, regression planning, planning as a constraint satisfaction problem (CSP), and partial-order planning.

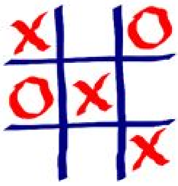
Forward Planning

Forward planning is simply the process of generating and searching the state space for a solution (the goal). Any of the search strategies and optimization techniques that fit the problem can be used. Forward planning takes into account the idea that as each new action is selected, the set of states may change which means that the next action to be performed must be identified by searching the state space again for the next action that leads to the goal.

Regression Planning

In regression planning, the state space graph is replaced with a graph of goals. Each goal represents a set of assignments to some set of features. We have discussed how feature-based approaches are more efficient than state-based approaches to problem-solving because in a feature approach we are not trying to represent every state but rather a set of features or rules.

You may recall that we looked at some problems with variables that could take only two values: True or False. For example, a light switch can be either ON or OFF state. In contrast, we have also seen variables that could take any value. In the case of a problem such as tic-tac-toe, we have a state space with 9 player positions and each position can have only 3 states (position is blank, position contains an X, position contains an O).



This means that there 93 possible states or 729 states, however, if we had variables based upon the set of integers or real numbers, the number of states could potentially be infinite. As such, using a set of features or assignments of values to the variables that represent a goal or end state from an action makes searching for a solution much easier.

Further, as we learned from the various optimization and heuristic techniques, we can make this searching process more efficient. By pruning some sets of values that might be assigned to variables or by adding some randomization into how values are assigned to variables can improve the likelihood of finding a solution faster.

Planning as a CSP

We talked about constraint satisfaction problems (CSP) in Unit 4 and we looked at a number of different approaches to solving a CSP. In many cases, a goal is merely a solution that meets the set of constraints.

The CSP planner essentially searches for an action that leads to a goal using the CSP techniques that we learned in Unit 4. You will notice that in each of these planning approaches we are searching for a solution to get to a state that leads to a goal and that the next actions must always take into account the current state of the problem and search for a solution that moves closer to the goal.

Planning with Uncertainty

We have spent some time looking at planning as a process when the output of an action is known when the world that the planning operates within is closed such that the agent knows the environment and can rely upon the outcome of any particular action.

We know, however, in the real world that the outcomes from any action are far from certain. The best that we sometimes have is a strong likelihood of and outcome from an action. For example, assume for a moment that you were a stockbroker and that you were tracking the progress of Hewlett Packard (HP) stock and the stock had been steadily gaining for many months. You might make a decision that since the stock price of HP has been on the rise for many months that there is a strong likelihood that it will continue to rise. When purchasing stock, you have no guarantee that the price of the stock will remain strong. In fact, you could purchase the stock and then an event that you did not nor could not plan for (such as a key executive leaving, or a supply issue with memory chips, or a natural disaster in a key market) could occur and the price of the stock goes down.

This is a great example of an event that is uncertain. You may have a belief that the stock price will continue to remain strong and your belief expressed as a probability might be high (for example you may believe that the probability that the stock price will remain strong is 90%) however there is always the potential that your belief and any decisions you make based upon that belief (such as buying the stock) could prove to be wrong.

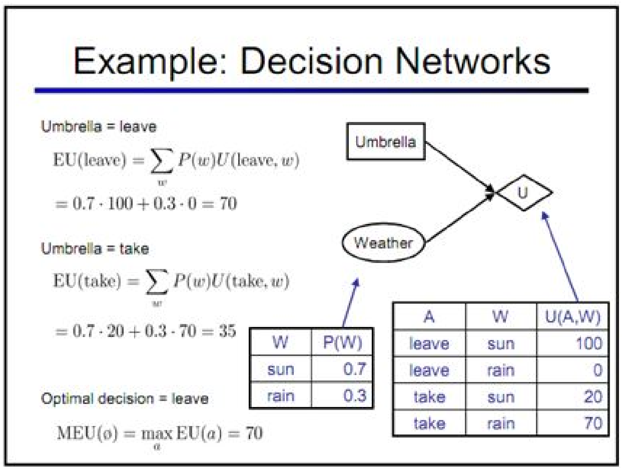
In such uncertain situations we act based upon two factors. The first is our beliefs and the second is our preferences. The belief is an estimate of the probability that an event will occur and the preference is the value of the goal that we are trying to attain.

For example, assume that a concert will be performed in your town and that in order to get tickets you will need to stand in line for many hours. Further, assume that since so many people want to see the concert that there will not be enough tickets to go around and that your probability of getting tickets is only 25%. If you REALLY like the group that will be performing then you may be willing to take the risk that you will not be able to get tickets and stand in line. However, if you see the group as ok and you wouldn't be all that upset if you didn't get to see the concert, you might not make the investment to stand in line because you don't have a strong preference to see the concert.

Utility can be seen as a measure of this relationship between belief and preference.

Decision making when there is uncertainty is essentially a process of searching for a goal using the utility of any action as a cost factor in the process. In this process, the agent would section an action that would result in the greatest utility.

Decision networks are belief networks that have been extended to include the utility of actions. As we know, belief networks are graphs that incorporate potential actions and the probability (belief) that the event or outcome of the action will occur.



These probabilities can then be evaluated in the context of the utility (utility * probability) to determine the best course of action based upon the action that produces the greatest utility.

Reading Assignment

Poole, D. L., & Mackworth, A. K. (2017). *Artificial Intelligence: Foundations of computational agents*. Cambridge University Press. https://artint.info/2e/html/ArtInt2e.html

Read the following chapters:

- Chapter 9 – Planning with Uncertainty

Discussion Assignment

Define “Decision Networks” in your own words in terms of variables and utility.

Then consider the following scenario:

A robot is dedicated to serving coffee for office employees. But it spills 20% of the time while pouring the coffee and on the way to the employee’s desk, making a mess. There are some decisions the robot can take for a better outcome. For example, improving its precision, or it can do nothing extra for a medium outcome. Complete the following:

1. Draw a Decision Network based on chance nodes, decision nodes, and a utility node.
2. Explain your Decision Network. For your response, you must design an approach for decision planning for the robot.
3. Describe what optimization solution you would employ and why?

Your Discussion should be at least 250 words in length, but not more than 750 words. Use APA citations and references for the textbook and any other sources used.

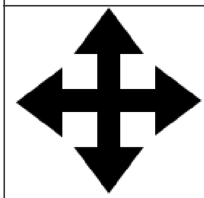
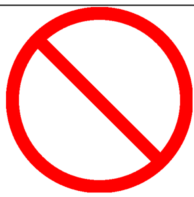
Programming Assignment

In this assignment, you will write pseudo-code for Markov Decision Process.

A Markov Decision Process also known as MDP model contains the following set of features:

- A set of possible states S.
- A set of Models.
- A set of possible actions A.
- A real valued reward function R (s, a).
- A solution of Markov Decision Process.

Consider the following Grid (3 by 3):

 Fire		 Diamond	3
			2
 Start		 Blocked	1
1	2	3	

An agent lives in a grid. It starts at grid number (1 * 1) and can roam around in the grid using the following actions:

UP, DOWN, LEFT, RIGHT

The goal of the agent is to reach the grid number (3 * 3) with the diamond state.

The agent must avoid the fire state at grid number (3 * 1) at any cost.

Also, there is a block grid at (1 * 3) state, which the agent can't pass and must choose an alternate route.

The agent cannot pass a wall. For example, in the starting grid (1 * 1), the agent can only go either UP or RIGHT.

Based on the above information, write a pseudo-code in Java or Python to solve the problem using the Markov decision process.

Your pseudo-code must do the following

1. Implementation of a static environment (grid) using an array or other data structure that will represent the above grid.

2. A function/method to determine what action to take. The decision should be based upon Markov Decision Process.

3. Consideration of reward policy that incorporates the action costs in addition to any prizes or penalties that may be awarded.

4. A function/method to calculate the optimal policy when a blocked state is encountered.

5. A function/method to calculate the optimal policy when the fire state is encountered.

6. A function/method to test if the desired goal is achieved or not.

You will be graded on the following:

1. In your pseudo-code, is the data structure appropriate for such an environment?
2. In your pseudo-code, does the algorithm take decisions based on Markov Decision Process?
3. In your pseudo-code, does it calculate and identify the optimal solution when a blocked state is encountered?
4. In your pseudo-code, does it calculate and identify the optimal solution when a fire state is encountered?
5. In your pseudo-code, does the algorithm consider reward policy while taking steps?

Learning Journal

The Learning Journal is a tool for self-reflection on the learning process. The Learning Journal will be assessed by your instructor as part of your Final Grade.

Your Learning Journal entry must be a reflective statement that considers the following questions:

1. Describe what you did. This does not mean that you copy and paste from what you have posted or the assignments you have prepared. You need to describe what you did and how you did it.

2. Describe your reactions to what you did.

3. Describe any feedback you received or any specific interactions you had while participating discussion forum or the programming assignment, discuss how they were helpful.

4. Describe your feelings and attitudes.

5. Describe what you learned. You can think of one or more topics from your week's lesson and explain your understanding in writings. Feel free to add any diagram or coding example if that helps you explain better.

6. Did you face any challenges while doing the discussion assignment or the Programming assignment? Were you able to solve it by yourself?

The Learning Journal entry should be a minimum of 400 words and not more than 750 words. Use APA citations and references if you use ideas from the readings or other sources.

Self-Quiz

The Self-Quiz gives you an opportunity to self-assess your knowledge of what you have learned so far.

The results of the Self-Quiz do not count towards your final grade, but the quiz is an important part of the University's learning process and it is expected that you will take it to ensure understanding of the materials presented. Reviewing and analyzing your results will help you perform better on future Graded Quizzes and the Final Exam.

Please access the Self-Quiz on the main course homepage; it will be listed inside the Unit**.**

Checklist

- Read the Learning Guide and Reading Assignments
- Participate in the Discussion Assignment (post, comment, and rate in the Discussion Forum)
- Complete and submit the Programming Assignment
- Complete an entry in the Learning Journal
- Take the Self-Quiz