Northeastern University
CS5100 – Foundations of Artificial Intelligence
Fall 2022, Richard Hoshino

Course Synthesis #2

Name: ____

Problem	Points
Problem #1 - Constraint Satisfaction	/10
Problem #2 - Nash Equilibrium	/10
Problem #3 - MDP and RL	/10
Problem #4 - Bayesian Networks	/10
Problem #5 - Hidden Markov Models	/10
Problem #6 - Machine Learning	/10
Problem #7 - Pac-Man Project	/10
Total	/50

Instructions

- This Synthesis will be marked out of 50. There are seven problems. You are welcome to attempt as many of them as you wish, but only your *top five* will be counted. For example, if your marks on the seven problems are 8, 0, 6, 10, 8, 3, 10 then your grade will be 8+6+10+8+10 = 42 out of 50, since your lowest two scores will be dropped.
- Think of this Course Synthesis as a week-long individual take-home exam where you may consult your notes, the course textbook, anything on the Canvas Page, and any websites linked from the Canvas Page. However, you may NOT consult your classmates or look at **any** other online resources unless explicitly approved by me beforehand. Please post your questions on the Canvas Discussion Forum if you would like any clarifications or hints.
- Submit individual files (e.g. in .pdf/.py/.java/.c) for each of the problems you attempt.

(10 pts.) Problem #1 - Constraint Satisfaction

Sommer took Richard's Discrete Structures (CS 5002) course in Fall 2020. The 21 students had 10 assessments where they worked together in groups. Richard found a way to split the 21 students into 3-person groups so that *every student worked with every other student exactly once.*

Here are the first three rows of Richard's optimal solution, known as a "combinatorial block design".

Problem Set 1	Zhe Emmanuel Mitchell	Clara JP Nassi	Guy Saurav David	Arghavan Zohreh Katie	Carey Bingyan Susan	Moni Jill Ian	Sommer Kelvin Alina
Problem Set 2	Arghavan Clara Bingyan	Mitchell Nassi David	Emmanuel JP Zohreh	Carey Saurav Katie	Guy Sommer Ian	Zhe Jill Alina	Moni Kelvin Susan
Problem Set 3	Guy Carey Zhe	Moni Clara Saurav	Emmanuel Bingyan Kelvin	Sommer David Zohreh	Arghavan Nassi Alina	JP Jill Susan	Mitchell Katie Ian

Let (n, d) be an ordered pair of positive integers. We say that the pair (n, d) is *happy* if it is possible to divide the n students into 3-person groups on d different days so that every student works with each of the other $n - 1$ students on exactly one day. Notice how this is a Constraint Satisfaction Problem.

For example, my solution above shows that the pair $(21, 10)$ is happy.

(a) Explain why n must be a multiple of 3, and why $d = \frac{n-1}{2}$. Use this result to show that if (n, d) is happy, and $n \leq 20$, then n must be one of 3, 9, or 15.

(b) Show that the pair $(n, d) = (9, 4)$ is happy by arranging the $n = 9$ students into three 3-person groups on $d = 4$ different days so that each pair of students s_i and s_j (with $i \neq j$) appear in the same group exactly once. Clearly explain how you constructed your solution.

(c) Create a Python program that solves a Constraint Satisfaction Problem to show that $(n, d) = (15, 7)$ is happy. To do this we define $X_{s,g,d}$ to be the binary variable that equals 1 if student s is assigned to group g on day d, and is equal to 0 otherwise. The skeleton of your Python program has been provided on the Canvas site (see "Notes from Class" and look under "Class #13").

Add the following three constraints to this Python program: (i) on each day, every student belongs to exactly one group; (ii) on each day, every group contains exactly three students; (iii) students s_i and s_j must be in the same group exactly once.

Alternatively, the third constraint can be modelled this way: for each four-tuple (s_i, s_j, d_k, d_l) with $1 \leq i < j \leq 15$ and $1 \leq k < l \leq 7$, students s_i and s_j cannot be in the same group on both days d_k and d_l. Do you see how that is equivalent to constraint (iii)?

For part (c), just submit your completed .py file.

(10 pts.) Problem #2 - Nash Equilibrium

In a 2-player game (with players Rose and Colin), each player chooses one of three options: Paper, Rock, or Scissors.

If both players select the same option, neither player gains any money.

The player who selects PAPER wins $\$x$ from the person who selects ROCK.

The player who selects ROCK wins $\$y$ from the person who selects SCISSORS.

The player who selects SCISSORS wins $\$z$ from the person who selects PAPER.

Notice that this game is *zero-sum*. For example, if Rose picks PAPER and Colin picks ROCK, then Rose gains $\$x$ and Colin loses $\$x$.

In the game of Rock-Paper-Scissors, we have $x = y = z = 1$.

(a) Let p_1, r_1, s_1 be the probabilities that Rose selects each of the three options Paper, Rock, Scissors, in that order. Similarly, let p_2, r_2, s_2 be the probabilities that Colin selects each of the three options.

If $x = y = z = 1$, determine the payoff function for Rose, which will be a function in terms of these six variables.

If Rose discovers that Colin's strategy is to play PAPER with probability 34%, SCISSORS with probability 33%, and ROCK with probability 33%, determine the optimal strategy (p_1, r_1, s_1) for Rose that will maximize her expected payoff.

(b) If $x = y = z = 1$, prove that the Nash Equilibrium of the Rock-Paper-Scissors game is $(p_1, r_1, s_1) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ for Rose and $(p_2, r_2, s_2) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ for Colin. Explain why this result makes sense.

(c) Suppose that we change the rules of Rock-Paper-Scissors so that $x = 1$, $y = 2$, and $z = 3$. For example, if Rose selects SCISSORS and Colin selects PAPER, then Rose wins $z = 3$ dollars and Colin loses $z = 3$ dollars.

Determine the Nash Equilibrium of this modified Rock-Paper-Scissors game. Clearly justify your answer.

(10 pts.) PROBLEM #3 - MDP AND RL

We play the following game, where you start with \$0. On each turn, I roll a fair 6-sided die.

If I roll a 6, you get \$3. And if I roll any other number (1,2,3,4,5), you get \$1.

You decide when to stop playing. If you choose to stop playing, the game is over and your payoff is your current total. However, if your total becomes more than 5 dollars, then you lose everything, and your payoff becomes 0 dollars. (The maximum payoff for this game is 5 dollars.)

For example, say I roll a 6, roll a 5, and then you stop. Then your payoff is $3 + 1 = 4$ dollars. However, say you decide to continue playing and I then roll a 6. Then $3 + 1 + 3 = 7 > 5$, and so your payoff is \$0.

(a) We model this as a Markov Decision Process (MDP), and use Value Iteration to analyze this game.

For each state s, let $V_k(s)$ be the value of the state at step k.

Using the Bellman Equation, clearly explain why $V_{k+1}(s) = \max\left\{s,\ \frac{5}{6}V_k(s+1) + \frac{1}{6}V_k(s+3)\right\}$ for all non-negative integers k and s.

(b) Use Value Iteration to determine $V^*(0)$, the expected utility for a rational agent playing this game, starting with 0 dollars. From your results, determine the optimal policy for this rational agent.

(c) Q-Learning is a powerful model-free Reinforcement Learning (RL) technique that enables us to directly learn the q-values of each state-action pair (s, a), without the need to know state values, transition functions, or reward functions.

Suppose you run Q-Learning with a very small learning rate (e.g. $\alpha = 0.001$) with many samples (e.g. $k = 100,000$) to determine the set of values $Q(s, a)$.

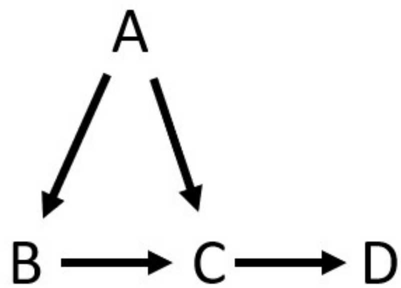
Determine what the q-values in the table below *must* converge to. Carefully justify your answer.

State	STOP	PLAY
0		
1		
2		
3		
4		
5		

For this entire problem, do note use or write a computer program.

(10 pts.) Problem #4 - Bayesian Networks

Consider the following Bayesian Network.



In this Bayesian Network, there are four states: A, B, C, D.

Each variable is either True (+) or False (−).

We are given the following probability values:

$$P(+A) = 0.1, \quad P(+B| + A) = 0.3, \quad P(+B| - A) = 0.5, \quad P(+D| + C) = 0.7, \quad P(+D| - C) = 0.9$$
$$P(+C| + A, +B) = 0.2, \quad P(+C| + A, -B) = 0.4, \quad P(+C| - A, +B) = 0.6, \quad P(+C| - A, -B) = 0.8.$$

For example, $P(+C \mid - A, +B) = 0.6$ is the probability that C occurs, given that A does not occur and B does occur. By definition, $P(-C \mid - A, +B) = 1 - 0.6 = 0.4$.

(a) Determine $P(+B)$, $P(+C)$, and $P(+D)$. Clearly show the steps in your calculations.

(b) Determine $P(+D| + A)$ and $P(+A| + D)$. Clearly show the steps in your calculations.

(c) Provide a real-life scenario where we would be interested in calculating $P(+A|+D)$, using the above 4-variable Bayesian Network.

In your response, clearly explain your scenario, describe what each of your variables A, B, C, D represents, and estimate the nine probability values for your scenario - i.e., $P(+A)$, $P(+B| + A)$, $P(+B| - A)$, and so on.

Finally, estimate the value of $P(+A| + D)$ using Bayesian Sampling, in the programming language of your choice. Submit your completed program (.py/.java/.c) in your solution.

(10 pts.) Problem #5 - Hidden Markov Models

In Chennai, India, each day's weather is one of three states: Sunny, Rainy, and Cloudy.

If the weather is Sunny on day t, then on day $t+1$ the probability of Sunny is 0.8, the probability of Cloudy is 0.1, and the probability of Rainy is 0.1.

If the weather is Cloudy on day t, then on day $t+1$ the probability of Sunny is 0.2, the probability of Cloudy is 0.6, and the probability of Rainy is 0.2.

If the weather is Rainy on day t, then on day $t+1$ the probability of Sunny is 0.3, the probability of Cloudy is 0.3, and the probability of Rainy is 0.4.

Let $P(X_t) = \begin{bmatrix} S_t \\ C_t \\ R_t \end{bmatrix}$ be a 3-dimensional vector representing the probability of the weather being Sunny, Cloudy, Rainy on day t. Richard's prior probability is $P(X_0) = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$.

On each day, Niranjan's mood is either Happy or Angry. His mood is influenced by that day's weather in his home city of Chennai. Specifically,

> If it is Sunny, there is a 80% chance Niranjan is Happy and a 20% chance Niranjan is Angry.
>
> If it is Cloudy, there is a 70% chance Niranjan is Happy and a 30% chance Niranjan is Angry.
>
> If it is Rainy, there is a 60% chance Niranjan is Happy and a 40% chance Niranjan is Angry.

Richard and Niranjan see each other on three consecutive days, and Richard notices that Niranjan is Angry on Day 1, Happy on Day 2, and Angry on Day 3. Thus, $E_1 = Angry$, $E_2 = Happy$, $E_3 = Angry$.

(a) Define $F_t = P(X_t|e_1, e_2, e_3, \dots e_t)$. Clearly explain why $F_{t+1} = \alpha_{t+1} \cdot O_{t+1} \cdot T \cdot F_t$, where α_{t+1} is the scaling factor, O_{t+1} is the diagonal emission matrix on day $t+1$, and T is the transition matrix.

Apply your recursive formula above to determine the filtering probability $P(X_3 \mid E_1 = Angry, E_2 = Happy, E_3 = Angry)$. Show all of your calculations.

(b) Consider the prediction probability $P(X_n \mid E_1 = Angry, E_2 = Happy, E_3 = Angry)$ for $n > 3$. Determine the prediction probability for $n = 6$, showing all of your calculations.

Then determine what the prediction probability must be for $n = 10000$. Clearly justify your answer.

(c) Determine the smoothing probability $P(X_1 \mid E_1 = Angry, E_2 = Happy, E_3 = Angry)$. Show all of your calculations.

(10 pts.) PROBLEM #6 - MACHINE LEARNING

In this question, you will create a computer program that will solve a Machine Learning problem of your choice.

You will select a data set of your choice (either one you find online or one you make up yourself), and you will select <u>one</u> Machine Learning technique of your choice.

For your Machine Learning technique, you may use one of the algorithms from the Machine Learning Seminars (either yours or one of your classmates), or a completely different algorithm. The choice is yours.

IMPORTANT NOTE: you may **not** recycle any work you have done in previous courses. For example, suppose you performed a Machine Learning analysis of a data set in a previous course, either at Northeastern or during your undergraduate degree. Then you may not re-use the same data set for this Course Synthesis.

(a) Tell me about your data set, clearly explaining each of your independent variables $x_1, x_2, x_3, x_4, \ldots$. If there is a dependent variable y, explain what your model aims to predict. Clearly explain which Machine Learning technique you are using to analyze your data, and explain *why* you are choosing this particular Machine Learning technique.

(b) Separate your data set into a Training Set and a Testing Set, where 80% of your data forms the Training Set and the remaining 20% forms the Testing Set.

Create a computer program to analyze your Training Set, building one or more models using your chosen Machine Learning technique.

Assess the quality of your model(s) on the Testing Set, providing some basic statistics on how well your model(s) predicted or classified each element in the Testing Set.

If possible, I strongly encourage you to **NOT** use a ready-made package such as *sci-kit learn* to create or test your model; write your own code to do so.

For part (b), just submit your completed program (.py/.java/.c), as well as your input Data Set so that the TA can run your program on his computer.

(c) Write a short reflection (minimum 200 words, maximum 400 words) on the strengths and limitations of your analysis, explaining why your model performed strongly or poorly. Provide at least two different ways that the accuracy of your model would be improved through further analysis.

(10 pts.) Problem #7 - Pac-Man Project

In this question, you will complete an additional Pac-Man programming project.

This project will be marked out of 10. (In other words, if you complete the Pac-Man project and get a perfect score, you will be given 10 out of 10 on this question.)

You have two options.

(i) PROJECT 4 – Ghostbusters (Bayesian Networks and Hidden Markov Models)

(ii) PROJECT 5 – Machine Learning (Perceptron and Neural Networks)

The details of the two projects are available at

`https://inst.eecs.berkeley.edu/~cs188/fa21/project4/`
`https://inst.eecs.berkeley.edu/~cs188/fa21/project5/`

Submit the required .py files for this question.

NOTE: in previous Pac-Man projects, I asked you to also submit answers to various questions, such as writing a short reflection on the question you found most difficult. For this Course Synthesis question, you do **not** need to hand in any such additional work. If you complete the Pac-Man project and get 25 out of 25 on the autograder, you will get 10 out of 10 on this question.