

CS2180 Artificial Intelligence Lab (Jan-May 2023)

Department of Computer Science and Engineering

Indian Institute of Technology Palakkad

Assignment 5: Knowledge & Reasoning (Given: 21 Mar 2023, Due: 9 Apr 2023)

General instructions

- Solutions are to be typed in the `.ipynb` file provided and uploaded in the lab course page in Moodle on the due date.
- Your code should be well commented and should be compatible with python3.
- For this assignment, you are allowed to import the libraries `pycosat`, `matplotlib`, `random` and `numpy` of python3. No other libraries may be imported.
- To start the jupyter notebook in the lab machine, enter `source /opt/anaconda3/bin/activate base` (note the space after `source` and before `base`) in the terminal and then type `jupyter notebook`
- Again, in the lab machines, if you are getting an error when importing `pycosat` saying that package is not installed, run `!pip3 install pycosat` in a jupyter notebook and install.

1 Island of Knights and Knaves

Consider the island of Knights and Knaves where each inhabitant is either a knight or a knave. A knight always tells the truth: if knight states a sentence, then that sentence is true. A knave always lies: if a knave states a sentence, then that sentence is false. Now, given a set of statements mentioned by some inhabitants, the problem is to determine whether each of these inhabitants is a knight or a knave. This problem can be expressed as a Satisfiability problem of a Boolean formula.

For example, consider the following statements by three people,

- A says either “I am a knight” or “I am a knave”, but you don’t know which.
- B says “A said I am a knave”
- B says “C is a knave”
- C says “A is a knight”

We will now show how to solve this puzzle by expressing facts about the island and the statements made by the inhabitants as propositional statements in CNF (Conjunctive Normal Form) and solve for a satisfying assignment by calling a standard SAT solving algorithm. We will use the `pycosat` which is wrapper for *picosat* SAT solver written in C.

- (a) For each person X introduce appropriate variables to capture whether X is a knight or a knave. Using these variables, introduce appropriate propositional statements that (1) capture that each person is either a knight or a knave, and (2) exactly matches the statements that each

of them makes about themselves or others. Write these statements as text (in [markdown](#)) in the cell below.

—Type the answer here—

- (b) Convert the above expressions to CNF formulas in the format used in `pycosat`.
- (c) Invoke the solver in `pycosat` and find who among A, B and C are knights and knaves.
- (d) Repeat parts (a)-(c) for the following sets of statements. In these statements, there are only two inhabitants of the island.
 - i. A says “At least one of us is a knave” and B says nothing.
 - ii. A says “We are the same kind”, B says “We are of different kinds”

2 Binary Town

Consider a town that contains only kids and adults. Each person in this town is associated with a height. Also, the probability of a random person in this town being a kid is $pKid$ and that of being an adult is $pAdult = 1 - pKid$.

- (a) Simulate 1000 people in this town by taking $pKid = 0.4$. Assume that the height of any person is a value from the set $htSet = \{4.9, 5.0, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8\}$. Also, the probability that a person has a specific height given that the person is a kid is given by $pHtKid = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$ and the probability that a person has a specific height given that the person is an adult is given by $pHtAdult = [0.02, 0.02, 0.02, 0.02, 0.02, 0.18, 0.18, 0.18, 0.18, 0.18]$. That is, the probability that a kid has height 5.1 is 0.1, the probability that an adult has height 5.4 is 0.18 and so on. Show the histograms of heights of both the groups in different colors in a single plot.
- (b) Implement an agent that classifies each sample generated in part (a) as kid or adult using Bayes’ Rule. Measure the accuracy of the classifier in terms of the fraction of correct predictions. You may assume that the agent knows $pKid$, $htSet$, $pHtKid$ and $pHtAdult$.
- (c) Assume that the height of a person can take on any value in \mathbb{R} . Repeat parts (a) and (b) when probability distributions of height given kid and height given adult are given by

$$f_{h|k}(x) = \begin{cases} 1 & \text{if } 4.5 \leq x \leq 5.0 \\ 2 & \text{if } 5.0 \leq x \leq 5.25 \\ 0 & \text{otherwise} \end{cases} \text{ and } f_{h|a}(x) = \begin{cases} 2 & \text{if } 5.0 \leq x \leq 5.25 \\ 1 & \text{if } 5.25 \leq x \leq 5.75 \\ 0 & \text{otherwise} \end{cases}$$
- (d) Assume that each person in the town has two attributes - height and weight that take on values from \mathbb{R} . The probability distributions of height given kid and height given adult are $N(\mu_{11}, \sigma_{11}^2)$ and $N(\mu_{21}, \sigma_{21}^2)$, respectively. Similarly, The probability distributions of weight given kid and weight given adult are $N(\mu_{12}, \sigma_{12}^2)$ and $N(\mu_{22}, \sigma_{22}^2)$, respectively. Here $N(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and standard deviation σ . Assume that the attributes height and weight are conditionally independent given the category (adult or kid). Simulate 1000 people in this town assuming $pKid = .4$, $\sigma_{11} = \sigma_{21} = 1$, $\sigma_{12} = \sigma_{22} = 10$, $\mu_{11} = 2.5$, $\mu_{12} = 20$, $\mu_{21} = 5$, $\mu_{22} = 40$. Note that you may have to discard a sample (x_1, x_2) if either $x_1 < 0$ or $x_2 < 0$. Plot the scatter plot of the samples (height in the X-axis and weight in the Y-axis).

- (e) Let S denote the set of samples generated in part (d). Given a height-weight pair (x, y) , write a function that finds k nearest points in S for each $k \in \{5, 10, 15\}$. The distance between (x, y) and (x_i, y_i) is defined as $\sqrt{(x_i - x)^2 + (y_i - y)^2}$. For each k , show a scatter plot with (x, y) in one color, its k nearest points in another color and the other points in S in a third color. Also, use the k nearest points of (x, y) and the majority function to classify (x, y) as belonging to a kid or an adult.

3 Optimal First Batting Strategy

Recall the cricket game simulation that we discussed in Assignment 2. Consider a one-day cricket match. Each innings consists of 300 balls ignoring no balls, runouts and overthrows. At any point of time, treat the pair of batsmen playing as a single player. That is, there are 10 players, namely, $(1, 2), (2, 3), \dots, (10, 11)$. For every player, there are 5 possible shots: $A = \{1, 2, 3, 4, 6\}$. Each of these shots is associated with the risk of the player getting dismissed and this risk varies from player to player. The probabilities of getting dismissed for Player $(1, 2)$ is $pOutMin = [0.01, 0.02, 0.03, 0.1, 0.3]$ and for Player $(10, 11)$ is $pOutMax = [0.1, 0.2, 0.3, 0.5, 0.7]$, where the i th entry is for the i th action. If there are w wickets in hand, then use the formula

$$pOut(a, w) = pOutMax(a) + (pOutMin(a) - pOutMax(a)) \times (w - 1)/9$$

to calculate the probabilities of Player $(11 - w, 11 - w + 1)$ getting dismissed for where $w \in [10]$. Note that $pOut(a, 1) = pOutMin(a)$ and $pOut(a, 10) = pOutMax(a)$. You may assume that if the player does not get out, then the runs corresponding to the shot played is scored.

Let us formulate an optimal strategy for the team batting first. Let b be the number of balls left and w be the wickets in hand. Let $BestScore(b, w)$ be the maximum expected score under an optimal policy in the remaining b balls with w wickets in hand. Note that no matter what the actual score is, the player should maximize the expected score in the remaining part of the game. Use the boundary conditions $BestScore(0, w) = 0$ for each w and $BestScore(b, 0) = 0$ for each b .

Hint: For $b \in [300]$ and $w \in [10]$, $BestScore(b, w) = \max_a(a + BestScore(b - 1, w))$ if the player does not get out in the b th ball and $BestScore(b, w) = BestScore(b - 1, w - 1)$ if the player gets out in the b th ball.

- Implement an optimal batting-first strategy by computing $BestScore(b, w)$ for each $b \in [300]$ and $w \in [10] \cup \{0\}$ using dynamic programming. Simulate 10 matches and bat according to the optimal policy. Output the optimal action (the recommended shot $BestShot(b, w)$) and the value (total expected score $BestScore(b, w)$ in the remainder of the game) for each pair (b, w) where b is the number of balls left and w is the number of wickets left. Note that $BestShot(b, w)$ denotes the shot to be taken when there are b balls and w wickets left so that $BestScore(b, w)$ is attained.
- What is your observation on the optimal strategy? Is it a good idea to bat slowly during the early part of the game, keeping wickets in hand? This usually allows a steady increase in the scoring rate, and often results in a last-minute rush of runs and wickets during the final part of the innings. Or is it better to score quickly in the early part of the innings? How would the optimal strategy change if there is a chance that a player does not score the runs for a particular shot despite not getting dismissed. Taking $pRunMin = 0.5$, and $pRunMax = 0.8$,

when the player is not getting out, the probability of successfully obtaining the runs for that shot is given by $pRun(w) = pRunMin + (pRunMax - pRunMin) \times ((w - 1)/9)$.