Departamento de Informática
Universidad Carlos III de Madrid

Grado en Ingeniería Informática

Artificial Intelligence

May 2016

General instructions

- Time assigned to the exam is **2.5 hours**
- Teachers will not answer any question about the exam
- You cannot leave the classroom during the exam, unless you have finished it
- Exams cannot be answered using a pencil

Theory questions

We expect very short answers.

1. (2.5 points) We want to control a robot with a production system. The robot has four sensors (S1, S2, S3 and S4) and two wheels (W1 and W2). Each sensor returns an integer value between 1 and 10. Actions can set the speed of each wheel in integer values between 1 and 5. Each rule will check the values of all sensors and set the speed of the two wheels.
    - What would be the working memory composed of?
    - How many different states can we create with that working memory?
    - How many different valid rules can we create?
    - How big would the conflict set be given a particular working memory?
2. (2.5 points) Rachel wants to go from one place to another in a city using a set of L bus lines. For the sake of simplicity, assume places correspond to bus stops. Each bus line has a set of ordered S stops and a set of B buses, each one on a given place (bus stop) at each time step. There can be more than one bus at the same stop and time step only if the buses belong to different bus lines. Buses move from one stop to the next in their bus line at each time step. Bus lines are circular (they do not have start and end stops). Time steps are discretized in ten minutes slots from 6am till 11pm (6:00, 6:10, 6:20, ..., 23:00).
    - Define the corresponding problem space
    - What is the approximate size of the state space?
    - If Rachel does not have any constraint in the quality of the solution (she has plenty of money and she is not in a hurry), select three different search tecniques to provide a correct solution
    - If Rachel wants to arrive at destination as fast as possible, select two different search techniques to provide a correct solution
    - If each bus line has a different price and Rachel wants to obtain the cheapest solution, select two different search techniques to provide a correct solution
3. (2.5 points) How can you pose the Naïve Bayes learning technique as a Bayesian network? Where do the probabilities can from? How can they be computed?
4. (2.5 points) We want to control an autonomous car with fuzzy logic. It has three sensors (front, right and left) and three actuators (brake, gas, wheel).
    - Define the input variables and their values.
    - Provide an example of a membership function.
    - Give an example of a direct input (before fuzzification).
    - Define the output variables and their values.
    - Provide an example of a membership function.
    - Give an example of an output (before defuzzification).

Questions about the project (1 p)

1. (0.5 p) Suppose that in the *gridworld* problem we define a grid as shown in Figure 1, named examGrid. Next, we run: `python gridworld.py -a value -i 1 -g examGrid --discount 1 --noise 0.1`

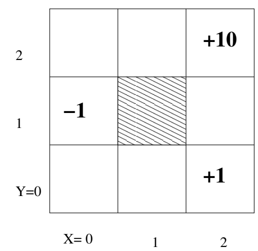

Figure 1: Definition of the grid examGrid.

After the Value Iteration algorithm finishes its execution (with the number of iterations shown in the shell script with -i) answer the following questions and briefly explain your answers.

   (a) What will be the value of the `self.values` variable in the class `ValueIterationAgent`?

   (b) What will the method `computeQValueFromValues` of the class `ValueIterationAgent` return if we pass as input the state `(0,0)` and the action `east`?

   (c) What will the method `computeActionFromValues` of the class `ValueIterationAgent` return if we pass as input the state `(2,1)`?

2. (0.5 p) Detail the variables that you have used to develop the learning in Pacman's MDP transition function.

Solutions

Theory questions

This is just one possible valid way of answering the exam questions. It does not mean they are the only valid answers.

First question

- The working memory will be composed of the current four sensors' values, as well as, optionally, the current wheels'x values. Thus, it will have the structure of {S1,S2,S3,S4,W1,W2} where Si (i=1..4) is a sensor value and Wi (i=1..2) is a wheel value. Each Si can have 10 different values, while each Wi can have 5 different values.
- In case we consider the wheels, the size will be $10^4 \times 5^2$.
- We can create also $10^4 \times 5^2$ different rules of the form:

  If S1=s1 AND S2=s2 AND S3=s3 AND S4=s4 THEN W1=w1 and W2=w2

  Obviously, we cannot have that many rules at the same time in the rule set. For instance, given a specific value for all sensors, there should only be one rule in the ruleset to actuate in the two wheels. Thus, on any ruleset, there can be at most 10^4 rules (all combinations of sensor values).
- Given that each rule checks for a specific configuration of the working memory, there can only be one rule instance in the conflict set at each cycle.

Second question

- A problem space is composed of
  - a set of states. In this case, each state will contain the complete information at each time point. It will include:
    * time step
    * position of Rachel at that time step
    * position (bus stop) of each of the $L \times B$ buses at that time step
    * the map of the bus infrastructure: which bus stops are included in each bus line and how they are ordered
  - a set of actions. There can be many different ways to represent the actions in this domain. A possibility would be to have the following two actions: get-on-bus and get-off-bus (possibly another move action)
- The size would be approximately $T \times S^{L\times B} \times L \times (S+B) + L \times S$ where T is the number of possible time steps, $S^{L\times B}$ is the number of possible different states for the positions of all buses, $L \times (S + B)$ is the maximum number of positions of Rachel and $L \times S$ is the size of the map
- Any search technique could solve that kind of task, preferably those that are complete
- Only admissible search techniques with arbitrary costs (time): A*, Dijkstra, ...
- Only admissible search techniques with arbitrary costs (price): A*, Dijkstra, ...

Third question

- Naïve Bayes can be posed as a Bayesian Network with the structure in Figure 2
- Since Naïve Bayes is a machine learning technique, probabilities will come from the input examples.
- Probabilities can be computed as (meaning is in the slides)
  - $P(A_j = V_{ji}|C_k) = \frac{n_{ijk}}{n_k}$
  - $P(C_k) = \frac{n_k}{n}$

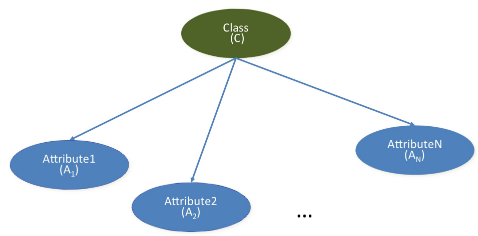

Figure 2: Naïve Bayes as a Bayesian Network.

Fourth question

- Input variables: front (values: close, medium, far), right (values: close, medium, far), left (values: close, medium, far)
- Front=close could be represented as in Figure 3

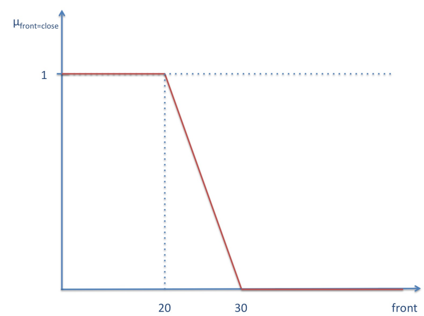

Figure 3: Membership function of front=close.

- front=25.6, right=20.3, left=2.4
- Output variables: brake (values: no, light, strong), gas (values: no, light, strong), wheel (values: straight, left, right)
- Wheel=straight could be represented as in Figure 4
- The result of fuzzy reasoning before defuzzification could be as the one in Figure 5

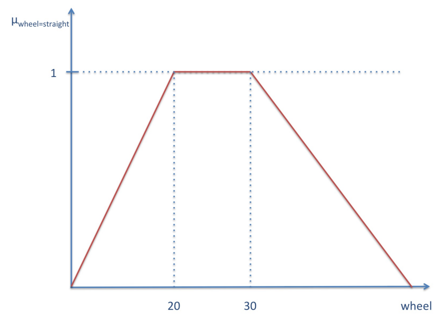

Figure 4: Membership function of wheel=straight.

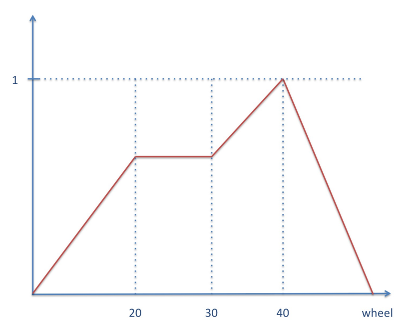

Figure 5: Result of fuzzy reasoning.