Coping with uncertainty

©Luís Seabra Lopes

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

I Objectives

The present list of exercices is dedicated to probabilisting representations and algorithms for coping with uncertainty in the world. We focus on Bayesian networks and Markov decision processes.

This list of exercices is used in the courses of *Artificial Intelligence* (*Licenciatura em Engenharia Informática*, *Licenciatura em Engenharia de Computadores e Informática*), and *Intelligent Systems I* (*Mestrado Integrado em Robótica e Sistemas Inteligentes I*).

This work will be done in 2 to 3 practical classes. ***For a fruitful use of the classes, exercises that are within the thematic scope of a given class should be completed before the next class.***

II Bayesian Networks

1 Presentation of the initial modules

The bayes_net module, attached hereto, exports a class to represent Bayes networks (BayesNet). A method is already implemented joint_prob (conjunction) to calculate the joint probability. The bn_example has the 'alarm' example of theoretical classes.

2 Exercises

1. Create a new Bayes network to represent the knowledge given in exercise III.11 of the *Theoretical-Practical Guide.*

2. Develop a new method that, given a variable in the network, returns a list with all predecessor variables.

3. Develop a new method that, given a network variable and a Boolean value, calculate its probability individual.

4. Using the network from the previous paragraph, calculate the probability of a user needing help.

III Markov decision processes

1 Presentation of the initial modules

Module `gridworld` implements the class `GridWorld` with methods to create and use grid-like environments, where we can have one or more success states, one or more failure states and some obstacles. We initialize the world with the grid dimensions (width,height), the locations of success and failure states as well as obstacles, and the rewards to apply in different states.

Module `mdp` constains a class `MDP`, which can be used to compute utilities and policies for given environments.

2 Exercices

1. Implement the value iteration algorithm as a method in the `MDP` class. This method returns the utilities of all states.
2. Develop a method tin the `MDP` hat, given the utilities of all states, computes a policy that always prefers to move to states with higher utilities.