

# Guião 4

## Tratamento probabilístico da incerteza

Departamento de Electrónica, Telecomunicações e Informática  
Universidade de Aveiro

### I Objectivos

O presente guião é dedicado a representações e algoritmos para o tratamento da incerteza, focando em redes de Bayes e processos de decisão de Markov.

Este guião é usado nas disciplinas de *Inteligência Artificial*, da *Licenciatura em Engenharia Informática* e da *Licenciatura em Engenharia de Computadores e Informática*, e *Sistemas Inteligentes I*, do *Mestrado Integrado em Robótica e Sistemas Inteligentes I*.

Este guião será realizado em 2 a 3 aulas práticas. ***Como sempre, para um bom aproveitamento das aulas, os exercícios que estejam no âmbito temático de uma dada aula devem ser completados antes da aula seguinte.***

### II Redes de Bayes

#### 1 Apresentação dos módulos iniciais

O módulo `bayes_net`, fornecido em anexo, exporta uma classe para representar redes de Bayes (BayesNet). Está já implementado um método `joint_prob` (conjunction) para calcular a probabilidade conjunta. O módulo `bn_example` tem o exemplo das aulas teóricas.

#### 2 Exercícios

1. Crie uma nova rede de Bayes para representar o conhecimento dado no exercício V.3 do *Guião Teórico-Prático*.
2. Desenvolva um método que, dada uma variável da rede, devolve a lista das variáveis ascendentes.
3. Desenvolva um novo método que, dada uma variável da rede e um valor booleano, calcule a respectiva probabilidade individual.
4. Usando a rede da alínea anterior, calcule a probabilidade de um utilizador precisar de ajuda.

### III Processos de decisão de Markov

#### 1 Apresentação dos módulos iniciais

O módulo `gridworld` implementa a classe `GridWorld`, com métodos para criar e utilizar ambientes na forma de grelha, onde podemos ter um ou mais estados de sucesso, um ou mais estados de falha e alguns obstáculos. Inicializamos o mundo com as dimensões da grelha (largura, altura), as localizações dos estados de sucesso e de falha, bem como dos obstáculos, e as recompensas a aplicar em diferentes estados.

O módulo `mdp` contém uma classe `MDP`, que pode ser utilizada para calcular utilidades e políticas para ambientes dados.

#### 2 Exercícios

1. Implemente o algoritmo de cálculo iterativo de utilidades na forma de um método na classe `MDP`. Este método devolve as utilidades de todos os estados.
2. Desenvolva um método na classe `MDP` que, dadas as utilidades de todos os estados, calcule uma política que prefere sempre deslocar-se para estados com utilidades mais elevadas.