Universidade de Aveiro

Inteligência Artificial (*LEI, LECI*)

Tópicos de IA:
Tratamento probabilístico da incerteza

Ano lectivo 2025/2026

Regente: Luís Seabra Lopes

Tópicos de Inteligência Artificial

- Agentes
- Resolução automática de problemas
- Representação do conhecimento
- Tratamento probabilístico da incerteza
  - Redes de Bayes
  - Processos de decisão de Markov

(REVISÃO)

Propriedades do mundo de um agente

- Acessibilidade – o mundo é "acessível" se os sensores do agente permitem obter uma descrição completa do estado do mundo; o mundo será "efectivamente acessível" se é possível obter toda a informação relevante ao processo de escolha das acções.
- Determinismo – o mundo "determinístico" se o resultado de uma acção é totalmente determinado pelo estado actual e pelos efeitos esperados da acção; caso contrário, o mundo é "estocástico".
- Mundo episódico – no caso em que cada episódio de percepção-acção é totalmente independente dos outros.
- Dinamismo – o mundo é "dinâmico" se o seu estado pode mudar enquanto o agente delibera; caso contrário, o mundo diz-se "estático".
- Continuidade – o mundo é "contínuo" quando a evolução do estado do mundo é um processo continuo ou sem saltos; caso contrário o mundo diz-se "discreto".

Tópicos de Inteligência Artificial

- Agentes
- Resolução automática de problemas
- Representação do conhecimento
- Tratamento probabilístico da incerteza
  - Redes de Bayes
  - Processos de decisão de Markov

Conhecimento impreciso

- Já vimos
  - Técnicas de representação baseadas em lógica ou com expressividade equivalente
  - Cada facto só pode ser verdadeiro ou falso
  - **Conhecimento certo**
- Vamos agora ver
  - Como incluir probabilidades na representação do conhecimento
  - **Conhecimento incerto**

Redes de crença bayesianas

- Também conhecidas simplesmente como "redes de Bayes"
- Permitem representar conhecimento impreciso em termos de um conjunto de variáveis aleatórias e respectivas dependências
  - As dependências são expressas através de probabilidades condicionadas
  - A rede é um grafo dirigido acíclico

Axiomas das probabilidades

- Para uma qualquer proposição a, a sua probabilidade é um valor entre 0 e 1:

$$0 \leq P(a) \leq 1$$

- Proposições necessariamente verdadeiras têm probabilidade 1

$$P(true) = 1$$

- Proposições necessariamente falsas têm probabilidade 0

$$P(false) = 0$$

- A probabilidade da disjunção é a soma das probabilidades subtraída da probabilidade da intercepção:

$$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$

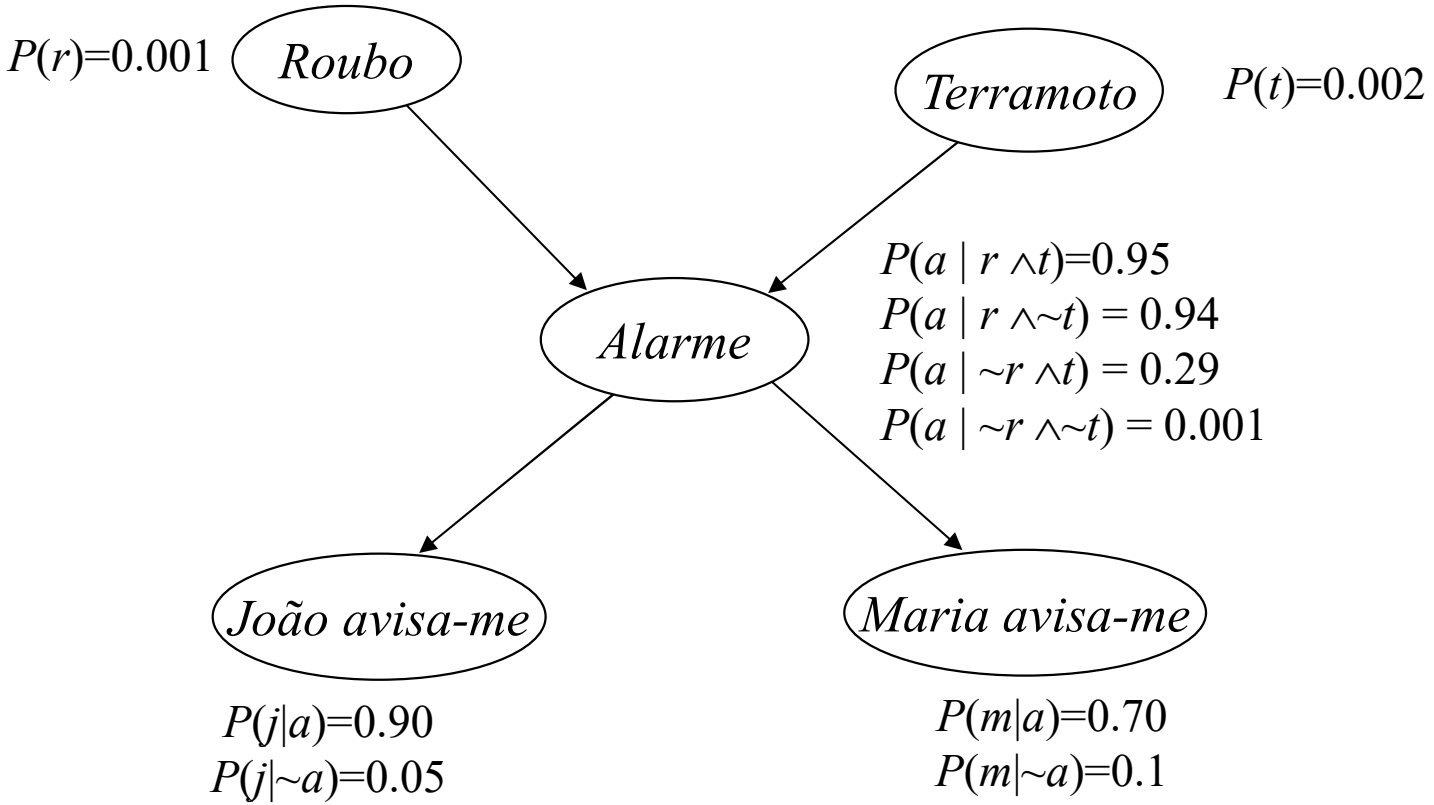
Probabilidades condicionadas

- Uma probabilidade condicionada $P(a|b)$ identifica a probabilidade de ser verdadeira a proposição a na condição de (isto é, sabendo nós que) a proposição b é verdadeira
- Pode calcular-se da seguinte forma:

$$P(a \mid b) = \frac{P(a \wedge b)}{P(b)}$$

Redes de crença bayesianas – exemplo

- Por simplicidade, focamos em variáveis aleatórias booleanas:



Redes de crença bayesianas – probabilidade conjunta

- A probabilidade conjunta identifica a probabilidade de ocorrer uma dada combinação de valores de todas as variáveis da rede:

$$P(x_1 \wedge ... \wedge x_n) = \prod_{i=1}^{n} P(x_i \mid pais(x_i))$$

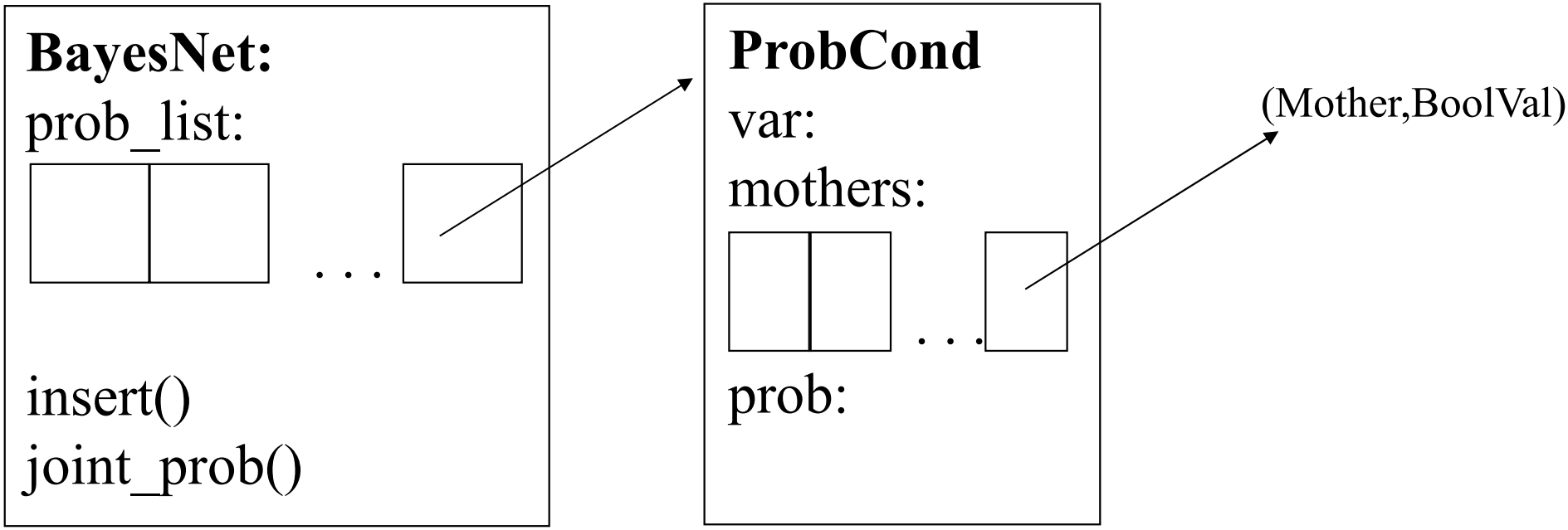
- Assim, no exemplo anterior, a probabilidade de o alarme tocar e o João e a Maria ambos avisarem num cenário em que não há roubo nem terramoto, é dada por:

$$\begin{aligned} & P(j \wedge m \wedge a \wedge \sim t \wedge \sim r) \\ & = P(j \mid a) \times P(m \mid a) \times P(a \mid \sim r \wedge \sim t) \times P(\sim r) \times P(\sim t) \\ & = 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 \\ & = 0.000628 \end{aligned}$$

Redes bayesianas em Python

- Vamos criar uma rede de crença bayesiana, representada com base numa lista de probabilidades condicionadas
  - Classe `BayesNet()`
- A probabilidade condicionada de uma dada variável ser verdadeira, dados os valores (`True` ou `False`) das variáveis mães, é representado pela seguinte classe:
  - Classe `ProbCond(var,mother_vals,prob)`
  - Exemplo: `ProbCond("a", [ ("r",True),  ("t",True)  ], 0.95)`
- Operações principais:
  - `insert` – introduzir uma nova probabilidade condicionada na rede
  - `joint_prob` – obter a probabilidade conjunta para uma dada conjunção de valores de todas as variáveis da rede

Redes bayesianas em Python



- Nota: ver módulo usado nas aulas práticas

Redes de crença bayesianas – probabilidade individual

- A probabilidade individual é a probabilidade de um valor específico (*verdadeiro* ou *falso*) de uma variável
- Calcula-se somando as probabilidades conjuntas das situações em que essa variável tem esse valor específico
- O cálculo das probabilidades conjuntas pode restringir-se à variável considerada e às outras variáveis das quais depende (ascendentes na rede bayesiana)
  - Exemplo: o conjunto dos ascendentes de "João avisa" é { "alarme", "roubo" e "terramoto" }

Redes de crença bayesianas – probabilidade individual

$$P(x_i = v_i) = \sum_{\substack{a_j \in \{v,f\} \\ j=1,\ldots,k}} P(x_i \wedge a_1 \wedge \ldots \wedge a_k)$$

- Seja:
  - $C = \{ x_1, \ldots, x_n \}$ – conjunto de variáveis da rede
  - $x_i \in C$ – uma qualquer variável da rede
  - $v_i \in \{\text{v,f}\}$ – valor de x_i cuja probabilidade se pretende calcular
  - $\{ a_1, \ldots, a_k \} \subset \text{C}$ – conjunto das variáveis da rede que são ascendentes de x_i

Tópicos de Inteligência Artificial

- Agentes
- Resolução automática de problemas
- Representação do conhecimento
- Tratamento probabilístico da incerteza
  - Redes de Bayes
  - Processos de decisão de Markov

(REVISÃO)

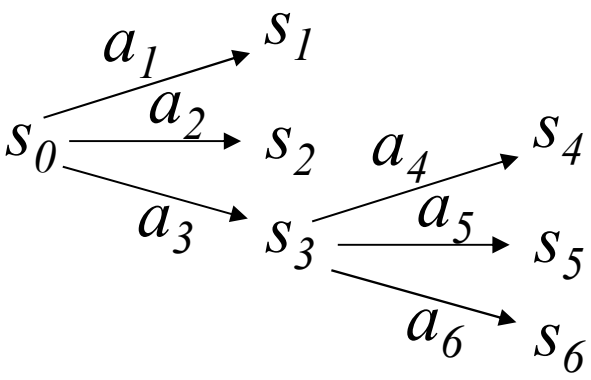
Propriedades do mundo de um agente

- Acessibilidade – o mundo é "acessível" se os sensores do agente permitem obter uma descrição completa do estado do mundo; o mundo será "efectivamente acessível" se é possível obter toda a informação relevante ao processo de escolha das acções.
- Determinismo – o mundo "determinístico" se o resultado de uma acção é totalmente determinado pelo estado actual e pelos efeitos esperados da acção; caso contrário, o mundo é "estocástico".
- Mundo episódico – no caso em que cada episódio de percepção-acção é totalmente independente dos outros.
- Dinamismo – o mundo é "dinâmico" se o seu estado pode mudar enquanto o agente delibera; caso contrário, o mundo diz-se "estático".
- Continuidade – o mundo é "contínuo" quando a evolução do estado do mundo é um processo continuo ou sem saltos; caso contrário o mundo diz-se "discreto".

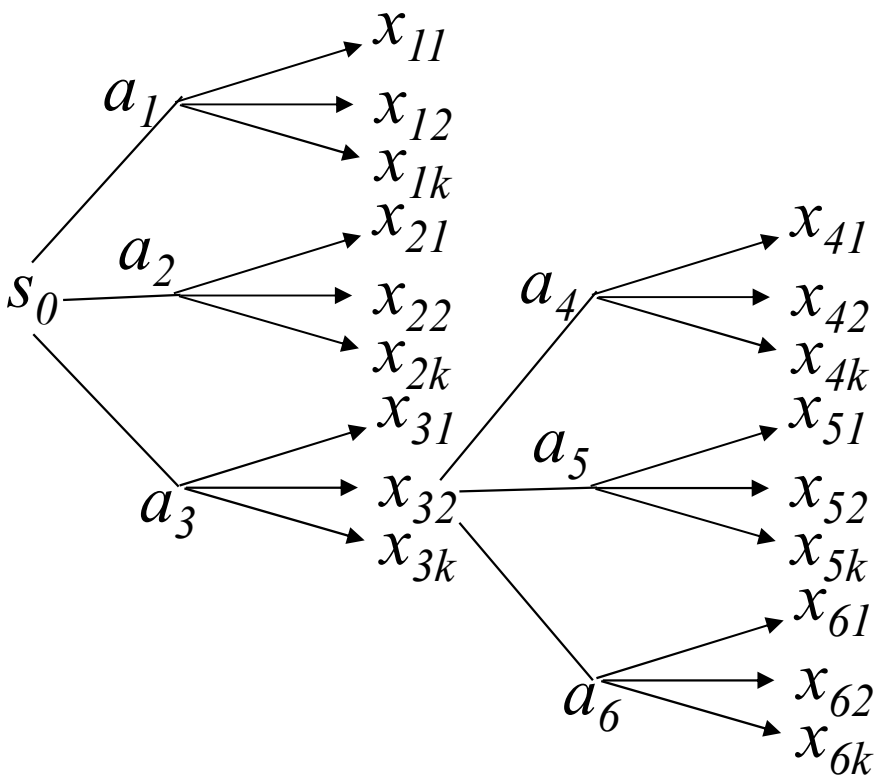
Deliberação em ambientes estocásticos

- Já vimos
  - Resolução automática de problemas para **ambientes determinísticos**, usando
  - Pesquisa em árvore no espaço de estados, assumindo que
  - Cada acção produz sempre os mesmos efeitos, e que
  - Ambiente evolui de forma determinística e previsível
- Vamos agora ver
  - Como tomar decisões em **ambientes estocásticos**
    - Rever acetato "Propriedades do mundo de um agente"
  - Estocástico = não determinístico
    - Efeitos das acções podem variar
    - Ambiente evolui de forma não (completamente) previsível

Pesquisa em árvore para domínios estocásticos?

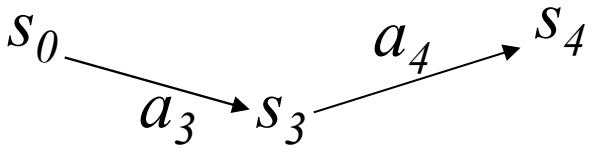


Pesquisa em ambiente determinístico



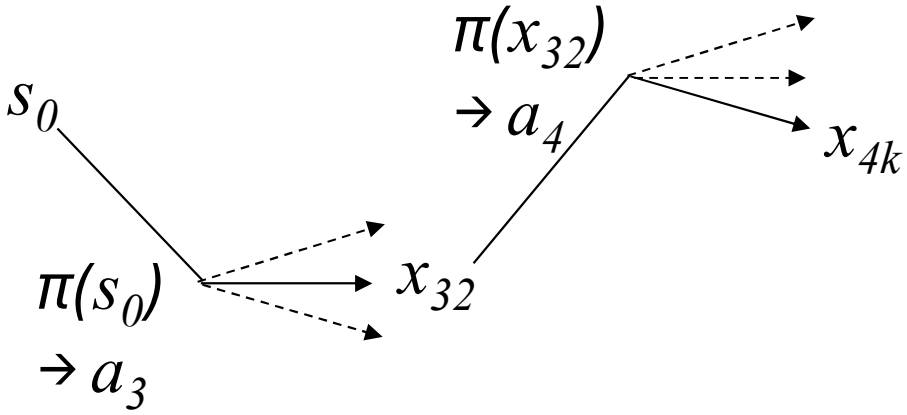
Pesquisa em ambiente estocástico **(inviável)**

Execução de sequências de acções



Determinístico

Execução de planos



Estocástico

Execução de políticas

Política de um agente

- Em geral, para actuar racionalmente, um agente poderá precisar de considerar em cada momento t toda a história anterior de percepções o_τ (observações) e acções a_τ com vista a determinar a nova ação a_t:

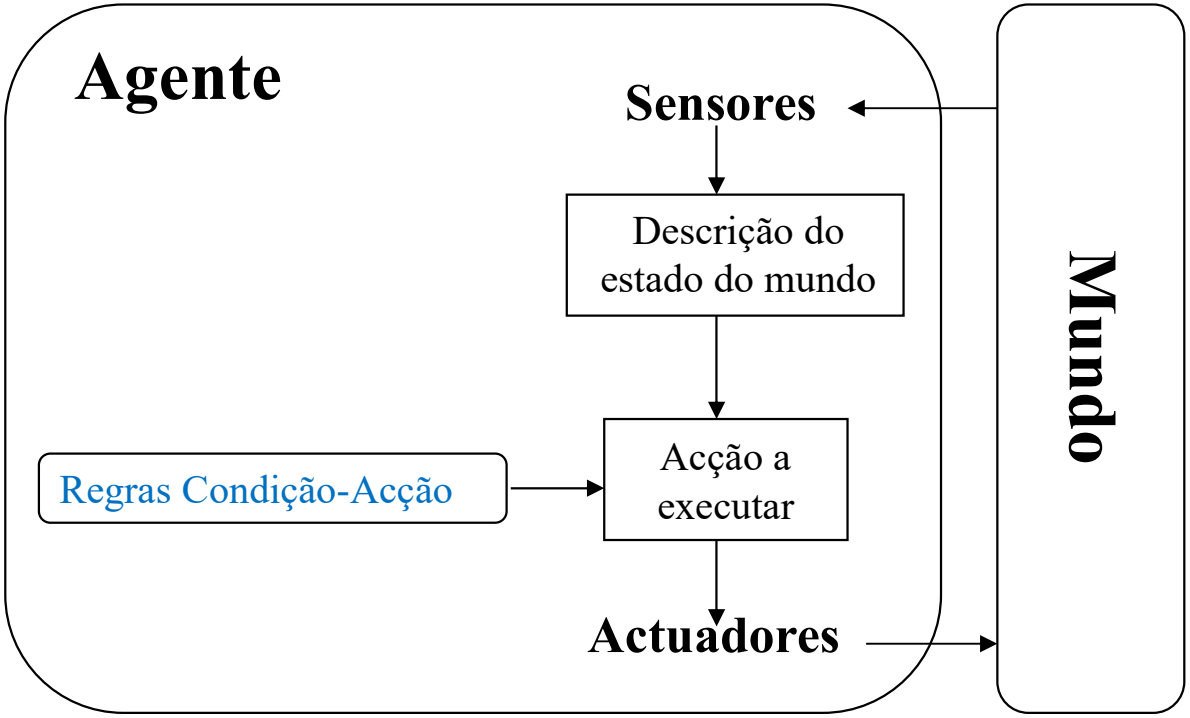
$$a_t = \pi(a_0, o_1, a_1, o_2, \ldots, o_t)$$

- A função π é chamada a **política do agente**

- Será fácil definir uma política?

(REVISÃO)

Agente reactivo: simples



Política reactiva

- Um agente reactivo determina a nova acção a_t com base apenas na percepção actual o_t

$$a_t = \pi(o_t)$$

- Trata-se de uma política sem memória
  - Ignora o passado!
- Caso particular de ambiente observável, i.e. acessível
  - Ver acetato sobre "propriedades do mundo de um agente"
  - Percepção actual o_t revela o estado actual do mundo s_t

$$a_t = \pi(s_t)$$

Mundo de Markov

- Num mundo (ou ambiente) acessível, a percepção revela o estado, mas o estado pode não conter toda a informação relevante sobre o passado
- Premissa de Markov: o estado actual depende apenas de um número finito de estados anteriores
- Processos de Markov são processos em que é viável adoptar essa premissa

Processos de Markov

- Processo de Markov de primeira ordem é um processo em que cada estado depende apenas do estado anterior
  - Considera-se que a descrição do estado imediatamente anterior contém toda a informação relevante sobre o passado
- Processo de Markov de segunda ordem é um processo em que cada estado depende apenas dos dois estados imediatamente anteriores

Determinístico *versus* estocástico

- Num mundo determinístico, o estado x resultante da execução de uma acção a é totalmente determinado pelo estado actual, s, e pelos efeitos esperados da acção

$$(s,a) \rightarrow x$$

- Num mundo estocástico, o modelo do mundo representa os efeitos da acção através de uma distribuição probabilística sobre estados

$$(s,a) \rightarrow P(x|s,a)$$

- $P(x|s,a)$ é a probabilidade de se atingir um dado estado x quando se executa a acção a no estado s, em que:

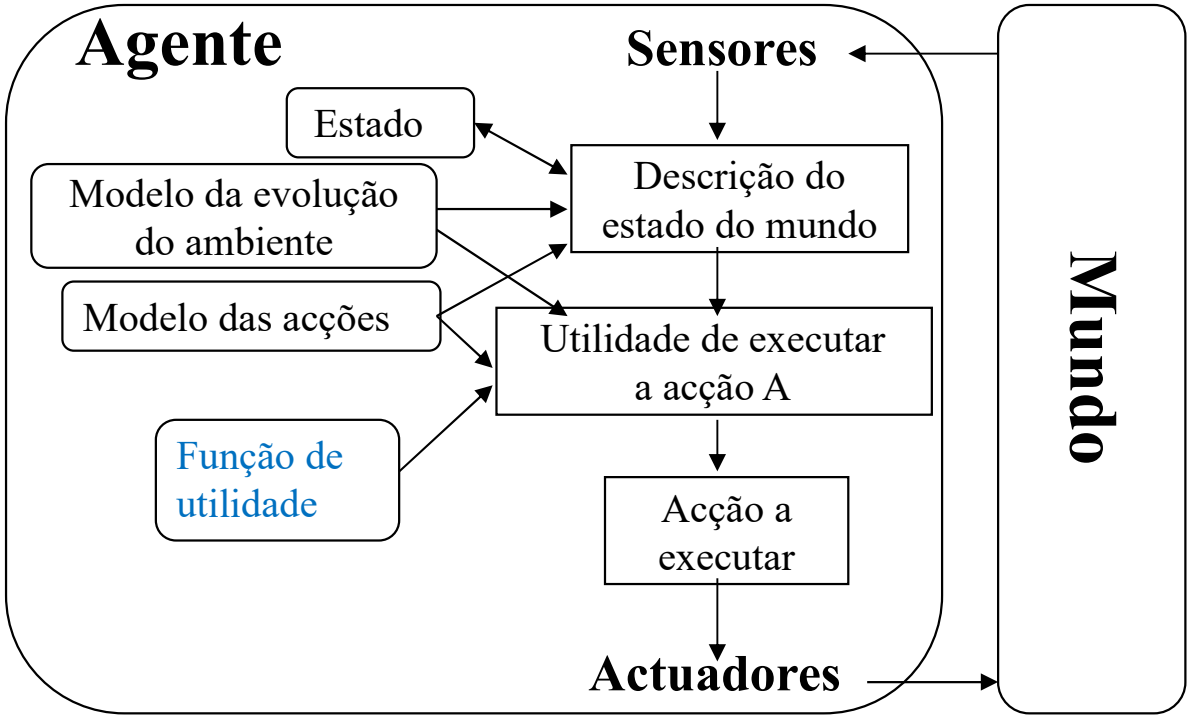
$$\sum_{x \in S} P(x \mid s,a) = 1$$

Episódico *versus* sequencial

- Problemas de decisão episódica – tomam decisões isoladas, referentes à execução de uma única ação
- Problemas de decisão sequencial – tomam múltiplas decisões que resultam na execução sequencial de múltiplas ações
- Mais uma vez: ver acetato sobre "propriedades do mundo de um agente"

(REVISÃO)

Agente deliberativo: orientado por função de utilidade



Preferências e funções de utilidade

- Notação:
  - $A > B$ – O agente prefere o estado A ao estado B
  - $A \sim B$ – Ao agente, é-lhe indiferente estar em A ou em B
  - $A \geq B$ – O agente prefere ou é-lhe indiferente estar em A ou em B
- As preferências de um agente sobre os diferentes estados do mundo são descritas através de uma **função de utilidade**
- A utilidade é uma função real sobre o conjunto de todos os estados possíveis, S

$$u : S \rightarrow \Re$$

Axiomas da teoria da utilidade

- Ordenabilidade: $(A > B) \vee (B > A) \vee (A \sim B)$
- Transitividade: $(A > B) \wedge (B > C) \Rightarrow (A > C)$
- Continuidade: $A > B > C \Rightarrow \exists p\ [p, A;\ 1\text{-}p, C] \sim B$
- Substitubilidade: $A \sim B \Rightarrow [p, A;\ 1\text{-}p, C] \sim [p, B;\ 1\text{-}p, C]$
- Monotonicidade: $A > B \Rightarrow (p \geq q \Leftrightarrow [p, A;\ 1\text{-}p, B] \geq [q, A;\ 1\text{-}q, B]$
- Decomponibilidade:

$$[p, A;\ 1\text{-}p, [q, B;\ 1\text{-}q, C]] \sim [p, A;\ (1\text{-}p)q, B;\ (1\text{-}p)(1\text{-}q), C]$$

- Em que:
  - p e q são probabilidades
  - A, B e C são estados
  - $[p_1, s_1; \ldots, p_n, s_n]$ é uma lotaria, em que $s_1 \ldots, s_n$ são os estados possíveis e $p_1 \ldots, p_n$ são as respectivas probabilidades

Preferências versus utilidades

- De acordo com os princípios da teoria da utilidade, existe uma função de utilidade u() tal que
  - $u(A) > u(B) \Leftrightarrow A > B$
  - $u(A) = u(B) \Leftrightarrow A \sim B$
- A utilidade de uma lotaria é a utilidade esperada dada por:
  - $u([p_1: s_1; ..., p_n: s_n]) = \Sigma_i \, p_i \, u(s_i)$
- Um agente racional toma decisões na presença de incerteza maximizando a sua utilidade esperada.

Utilidade do dinheiro

- O que prefere?
  - Opção A: 3000€ com uma probabilidade de 50%, ou
  - Opção B: 1000€
- Pode depender do dinheiro que já tem
- Suponha que já tem um montante M
  - $U(A) = 0.5 \times u(M) + 0.5 \times u(M+3000)$
  - $U(B) = u(M+1000)$
- Em geral, constata-se que a utilidade de uma certa quantia adicional de dinheiro é proporcional ao logarítmo da quantia total com que se vai ficar!

Utilidade esperada e decisão episódica

- Utilidade esperada

$$U(s,a) = \sum_{x \in S} P(x \mid s,a) u(x)$$

- Política baseada na **máxima utilidade esperada**

$$\pi(s) = \arg\max_{a \in A} U(s,a)$$

  - Ou seja: calculam-se as utilidades esperadas das várias acções possíveis e escolhe-se a acção que tiver maior utilidade esperada
- Informação sobre estado – o estado será *t*, e não *s*, quanto maior for a seguinte medida

$$U(t,\pi(t)) - U(t,\pi(s))$$

Problemas de decisão sequencial

- Problemas em que a função de utilidade do agente depende de uma sequência de acções
  - Podem ser vistos com uma generalização, para ambientes estocásticos, dos problemas tratados através das técnicas clássicas de pesquisa e planeamento
  - De cada vez que uma dada política é executada (resultando na execução de uma determinada sequência de acções), atingir-se-á um estado diferente
  - A qualidade da política é medida pela utilidade média esperada, considerando todos os estados a que a política pode conduzir e respectivas probabilidades

Problemas de decisão sequencial

- Focamos em processos de decisão de Markov
  - Ambiente completamente observável
  - Premissa de Markov de primeira ordem

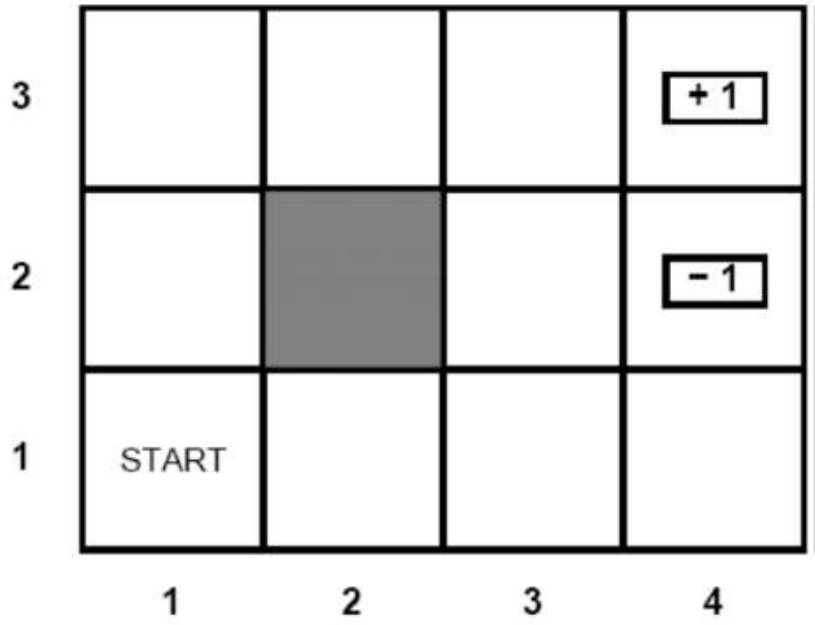
→ MDP = "Markov Decision Process"

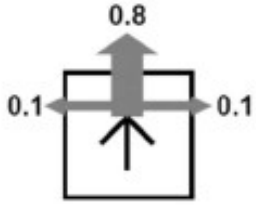
Processos de decisão de Markov (MDP)

- Formulação do problema
  - Estado inicial: s_0
  - Modelo de transição de estados: $T(s,a,x) = P(x|s,a)$
  - **Função de recompensas**: $R(s)$
    - Estipula a recompensa dada ao agente em cada estado
    - A recompensa pode ser vista como uma "utilidade de curto prazo"

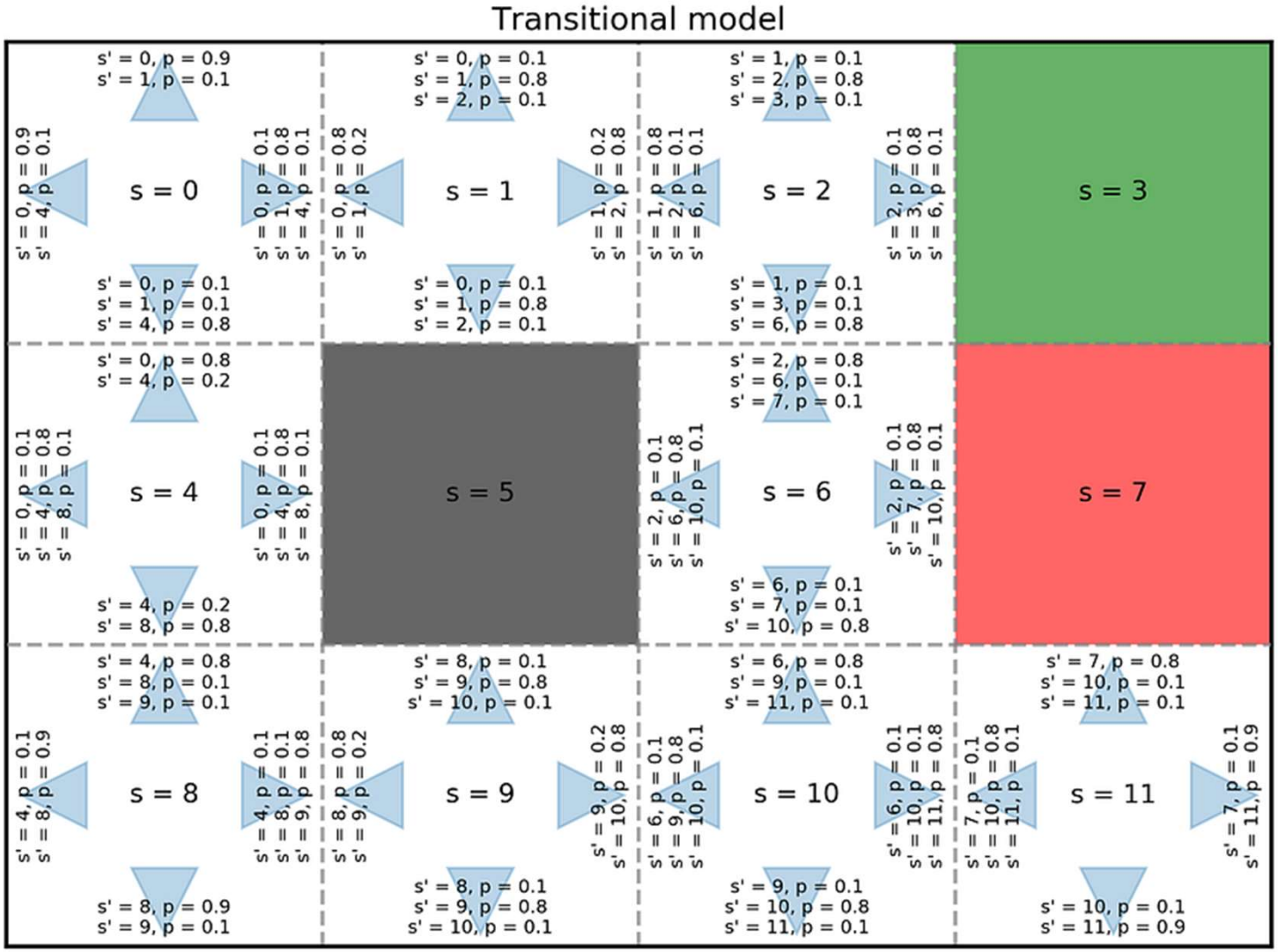
Exemplo de mundo estocástico

- Cada célula é um estado
- Objetivo com recompensa de +1
- Ações:
  - cima, baixo, esquerda, direita
  - Com uma probabilidade de 0.8, cada ação produz o efeito desejado
- A probabilidade de atingir o objetivo com [cima, cima, direita, direita, direita] é de $0.8^5 = 0.32768$.





Exemplo de mundo estocástico



Modelo de transição de estatos (probabilidades)

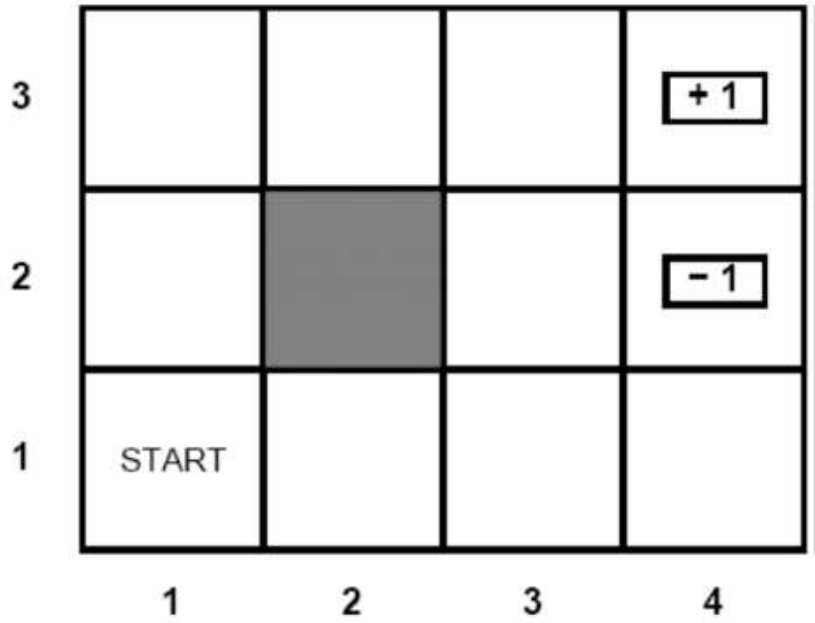
NOTA:

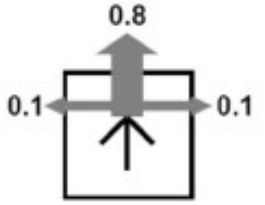
Nesta figura, os estados estão numerados de 0 a 11

Fonte: https://medium.com/@ngao7/markov-decision-process-basics-3da5144d3348

Exemplo de mundo estocástico

- $R(4,3) = +1$
- $R(4,2) = -1$
- $R(s) = -0.04$ (para os restantes estados s)
- Recompensa total para a sequência [cima, cima, direita, direita, direita]: $-0.04-0.04-0.04-0.04-0.04+1 = 0.8$





Processos de decisão de Markov (MDP)

- Focamos em situações de horizonte infinito e portanto numa política estacionária.
- Horizonte finito consistiria em impôr um limite ou prazo para atingir o objectivo
  - Neste caso, a política óptima seria não estacionária porque a acção óptima em cada estado dependeria do tempo disponível

MDP: recompensas descontadas

- Em políticas estacionárias, a utilidade de uma sequência de estados é definida da seguinte forma

$$U([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma\, R(s_1) + \gamma^2 R(s_1) + \ldots$$

em que o **factor de desconto** γ:

  - Varia entre 0 e 1
    - Típicamente, terá um valor ligeiramente inferior a 1
  - No caso particular de γ=1, tem-se um esquema de recompensas aditivas
    - Estratégia gulosa ("greedy"): dar preferência aos estados mais próximos pode conduzir o agente a uma solução sub-óptima

MDP: factor de desconto

- Descreve a preferência do agente por recompensas actuais em detrimento de recompensas futuras
  - Quanto menor γ, maior será essa preferência, porque maior é o desconto nas recompensas futuras
  - Compensa a incerteza no tempo disponível (modela a mortalidade)
  - Com $\gamma = 0.90$, a promessa de receber 3000€ no próximo ano vale apenas 90% da promessa de receber os mesmos 3000€ agora
  - Quando $\gamma = 0$, as recompensas futuras não significam nada
- Impede recompensas infinitas
  - Sem o fator de desconto, as sequências infinitas, em horizontes infinitos, terão utilidades infinitas

Definição recursiva da utilidade

- **Equação de Bellman**

$$u(s) = R(s) + \gamma \max_a \sum_x P(x|s,a)u(x)$$

- Ou seja:
  - A utilidade de um estado é a recompensa colhida nesse estado somada à utilidade descontada no estado seguinte, no pressuposto de ser escolhida a acção óptima
  - No exemplo apresentado, teremos um sistema com
    - 11 equações de Bellman, uma para cada estado
    - 11 incógnitas, que são as utilidades dos estados

Cálculo iterativo de utilidades

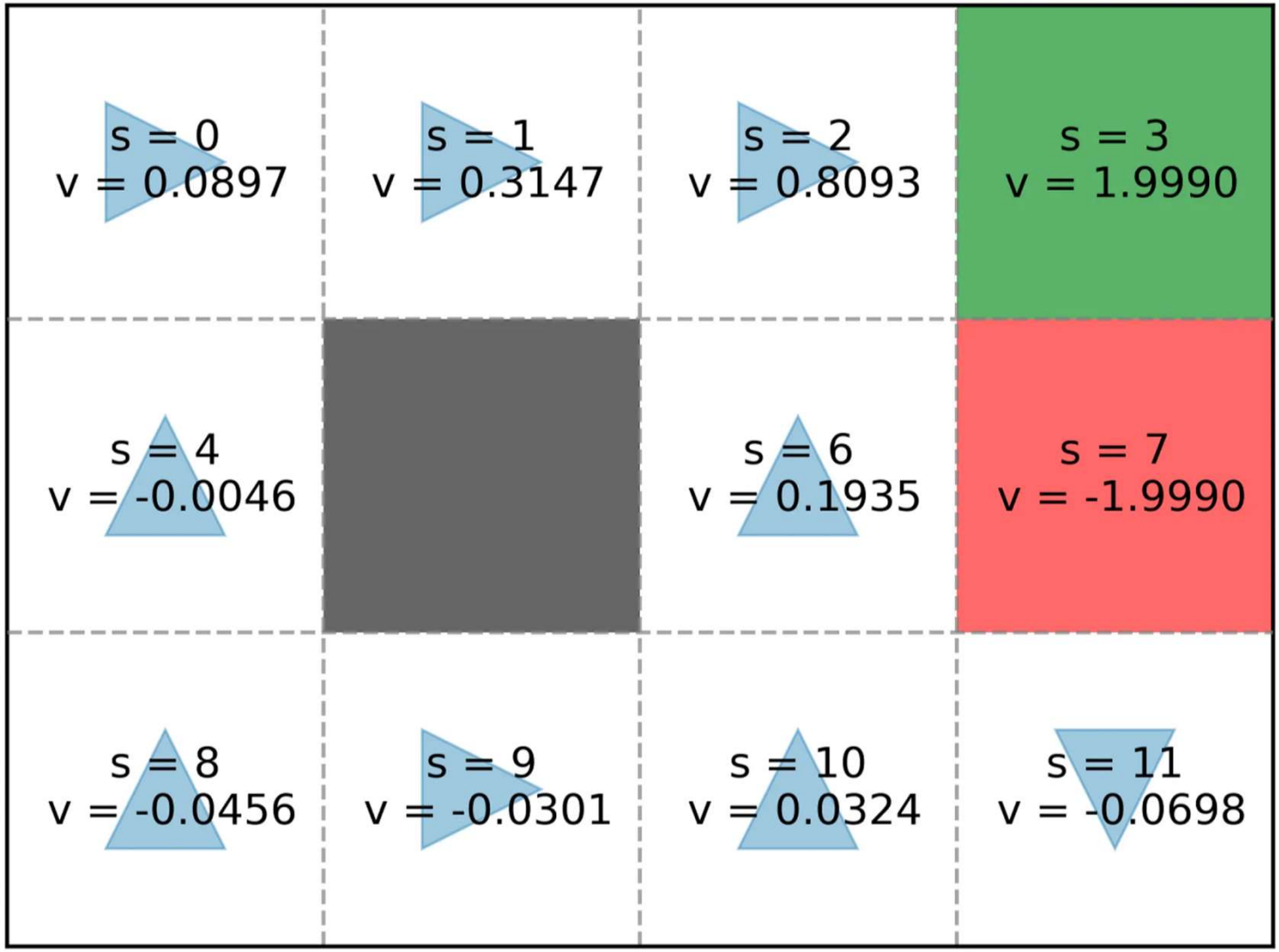
- Algoritmo que converge para uma solução ótima única da equação de Bellman

repeat
  $u \leftarrow v$
  $\delta \leftarrow 0$
  for each state $s \in S$ **do**:
    $v[s] \leftarrow R[s] + \gamma \max_a \Sigma_x P(x|s,a)\, u[x]$
    if $|v[s]-u[s]| > \delta$ **then** $\delta \leftarrow |v[s]-u[s]|$
until $\delta < \varepsilon \times (1-\gamma)/\gamma$
return u

Em que:

- S – conjunto de todos os estados possíveis
- R – função de recompensa
- γ – fator de desconto
- u e v – vetores de utilidade para todos os estados s (inicialmente 0)
- ε – erro máximo permitido na utilidade de um estado
- δ – variação máxima da utilidade de um estado numa determinada iteração

Exemplo de mundo estocástico



Utilidades calculadas, com $\gamma = 0.50$, ao fim de 11 iterações, e a correspondente política

NOTA: estados estão numerados de 0 a 11

Fonte: https://medium.com/@ngao7/markov-decision-process-value-iteration-2d161d50a6ff

MPD parcialmente observáveis (POMDP)

- POMDP = "Partially Observable Markov Decision Processes"
- São, em geral, muito mais difíceis de tratar do que os MDP observáveis
- A formulação de um POMDP inclui os elementos já considerados na formulação de um MDP e ainda um modelo de observação:
  - $O(s,o) = P(o|s)$
  - Ou seja: especifica as probabilidades de obter diferentes observações (i.e. percepções) em diferentes estados

MPD parcialmente observáveis (POMDP)

- No tratamento destes problemas, surge a noção de estado de crença (*belief state*), $b(s)$:
  - b é uma distribuição de probabilidade sobre todos os estado possíveis
  - $b(s)$ é a probabilidade de se estar no estado real s segundo o estado de crença b

POMDP: actualização do estado de crença

- A actualização do estado de crença é feita de acordo com o seguinte passo iterativo, para qualquer estado x

$$b(x) \leftarrow \alpha\, O(x,o) \sum_s P(x|s,a) b(s)$$

  - Em que: α é uma constante de normalização que permite a soma das crenças em todos os estados possíveis ser 1
- Um POMDP no espaço de estados real pode ser reduzido a um MDP observável no espaço de estados de crença
- No entanto, sendo uma distribuição de probabilidade o espaço de estados de crença é contínuo
  - O algoritmo iterativo dado anteriormente não resolve
  - Um solução (parcial) consiste em dividir o espaço de crenças em regiões, associando a cada região uma acção óptima