Inteligência Artificial

Fabrício Olivetti de França

07 de Junho de 2018

Teoria da Decisão e Utilidade

Teoria da Decisão

Teoria da Decisão estuda a escolha entre um conjunto de ações baseados no desejo de um resultado **imediato**.

Teoria da Decisão

Tomem como exemplo jogos de auditório em que você deve tomar decisões baseado em eventos probabilísticos.

Imagine o jogo Show do Milhão em que você pode escolher a categoria das perguntas, se irá parar de jogar, responder uma alternativa ou escolher alguma ajuda.

Suas ações, sendo um agente racional, deve ser aquelas que te tragam o maior valor esperado imediato.

Teoria da Decisão

Assumindo $R(a)$ como uma variável aleatória com o resultado de uma ação a, temos que a probabilidade de atingir um estado s' dada uma evidência de estado e é:

$$P(R(a) = s' \mid a, e)$$

Dada uma função $U(s)$ denominada **função utilidade**, que mapeia um estado para um valor numérico representando o desejo em chegar a esse estado, temos a **utilidade esperada** definida como:

$$EU(a \mid e) = \sum_{s'} P(R(a) = s' \mid a, e) U(s')$$

Teoria da Decisão

O princípio da **utilidade máxima esperada** diz que um agente racional deve escolher a ação que maximize a utilidade esperada do agente:

$$a = argmax_a EU(a \mid e)$$

Notação:

$A \succ B$ indica que o agente prefere A sobre B.

$A \sim B$ indica que o agente não tem preferência entre A e B.

$A \succeq B$ indica que o agente prefere A sobre B ou não tem preferência.

Exemplo: imagine que A e B representam escolher prato de massa ou de carne em um vôo. Você não sabe exatamente qual massa ou qual carne é oferecida, nem se o prato de massa não foi esquentado corretamente, ou se carne passou do ponto. É uma loteria.

Teoria da Utilidade

Uma função de utilidade deve modelar um processo de preferência com certas restrições.

Ordenação: um agente não pode evitar de fazer uma escolha:
$A \succ B, B \succ A$ ou $A \sim B$.

Transitividade: dada três escolhas, se um agente prefere A em relação a B e B em relação a C, ele deve preferir A em relação a C:

$$(A \succ B) \land (B \succ C) \Rightarrow (A \succ C)$$

Continuidade: se existe uma decisão B entre A e C na ordem de preferência, tem uma probabilidade p tal que o agente racional será indiferente entre escolher B com toda certeza e a escolha A com prabilidade p ou C com probabilidade $1 - p$:

$$A \succ B \succ C \Rightarrow \exists p[pA, 1 - pC] \sim B$$

Teoria da Utilidade

Substituição: se um agente é indiferente entre A e B, então ele também é indiferente ao substituir uma escolha que pode levar A por uma escolha que pode levar B com a mesma probabilidade: $A \sim B \Rightarrow [pA; 1 - pC] \sim [pB; 1 - pC]$.

Monotônica: se o agente prefere A sobre B e tem duas escolhas de ações que podem levar tanto para A como para B, ele irá escolher a de maior probabilidade:
$A \succ B \Rightarrow (p > q \Leftrightarrow [pA; 1 - pB] \succ [qA; 1 - qB])$.

Decomposição: uma escolha composta pode ser quebrada em escolhas simples:

$$[pA; 1-p[qB; 1-qC]] \sim [pA; (1-p)qB; (1-p)(1-q)C]$$

Teoria da Utilidade

A função utilidade $U(s)$ deve ser equivalente as restrições postas anteriormente tal que:

$$U(A) > (B) \Leftrightarrow A \succ B$$

$$U(A) = (B) \Leftrightarrow A \sim B$$

Teoria da Utilidade

A **utilidade esperada de uma loteria** é a soma ponderada pela probabilidade de utilidade de cada resultado.

Exercício

Suponha que após o término de um programa de auditório você recebeu um prêmio de R$1.000.000,00, nesse momento te é dada a opção de jogar um jogo de cara ou coroa tal que se der cara você perde tudo, mas se der coroa você ganha R$2.500.000,00.

Qual a ação racional a ser feita?

Exercício

Nesse caso temos que a utilidade esperada de arriscar é:

$$0.5 \times 2.5 \cdot 10^6 + 0.5 \times 0 = 1.25 \cdot 10^6$$

Que é maior do que a utilidade de não arriscar: $1 \cdot 10^6$.

Exercício

E se sua função utilidade for definida como $U(S_n)$ onde n é composto pela soma de quanto você possui atualmente (fora do jogo) e quanto você ganhará (no jogo). Agora temos que:

$$EU(arriscar) = 0.5 \times U(S_k) + 0.5 \times U(S_{k+2500000})$$

$$EU(recusar) = U(S_{k+1000000})$$

Exercício

Na realidade, a utilidade monetária não é proporcional ao prêmio, mas a quanto você possui. A utilidade do seu primeiro milhão é muito maior do que do seu décimo milhão, pois é o ponto em que você tem uma mudança de qualidade de vida maior.

Exercício

Se determinarmos, por exemplo, $U(S_k) = 5, U(S_{k+2500000}) = 9, U(S_{k+1000000}) = 8$, temos que:

$$EU(arriscar) = 0.5 \times U(S_k) + 0.5 \times U(S_{k+2500000}) = 7$$

$$EU(recusar) = U(S_{k+1000000}) = 8$$

Agentes Autônomos

Agentes Autônomos

Um *Agente Autônomo* é um agente com inteligência artificial que executa uma tarefa envolvendo uma ou mais decisões sem interferência externa.

Estudado principalmente em Inteligência Artificial e utiliza Aprendizado de Máquina para automatizar a tarefa de *ensinar* o agente.

Características de um Agente Autônomo:

- **Autonomia:** não requer interferência externa para a tomada de decisão.
- **Localização:** é capaz de se situar no ambiente em que age.
- **Interação:** interage com o ambiente para adquirir novas informações.
- **Inteligência:** consegue agir de acordo com o estado do ambiente para atingir um determinado objetivo.

Basicamente, queremos um sistema em que o agente recebe o estado atual do ambiente e execute uma ação nesse ambiente para então receber um novo estado.

Ou seja, queremos encontrar uma função $f : S \to A$, sendo S o conjunto de estados possíveis do sistema e A o conjunto de possíveis ações que podem ser executadas pelo agente.

Agentes Autônomos

Considerem os seguintes exemplos de agentes autônomos:

- Automóvel sem motorista.
- Casa Inteligente.
- Visita de rotina a pacientes.
- Sistema anti-invasão de computadores.

Agentes Autônomos

Considerem os seguintes exemplos de agentes autônomos:

- Automóvel sem motorista.
- Casa Inteligente.
- Visita de rotina a pacientes.
- Sistema anti-invasão de computadores.

Como você coletaria um conjunto de exemplos para o aprendizado?

Agentes Autônomos

Em alguns desses casos existem múltiplas ações possíveis.

- Automóvel sem motorista.
- Casa Inteligente.
- Visita de rotina a pacientes.
- Sistema anti-invasão de computadores.

Em outros não temos como gerar contra-exemplos de ações que **não** deveriam ser realizadas.

- Automóvel sem motorista.
- Casa Inteligente.
- Visita de rotina a pacientes.
- Sistema anti-invasão de computadores.

Também temos casos em que só saberemos a ação correta após uma série de ações serem aplicadas.

- Automóvel sem motorista.
- Casa Inteligente.
- Visita de rotina a pacientes.
- Sistema anti-invasão de computadores.

Uma outra forma de entender um ambiente de um agente autônomo é quando temos pelo menos uma recompensa instantânea associada a ação realizada.

Essa recompensa pode ser positiva ou negativa.

Quais recompensas podemos atribuir para possíveis ações dos sistemas de exemplo?

- Automóvel sem motorista.
- Casa Inteligente.
- Visita de rotina a pacientes.
- Sistema anti-invasão de computadores.

Quais recompensas podemos atribuir para possíveis ações dos sistemas de exemplo?

- Automóvel sem motorista:
  $r_t = d(x, x_{pista})^{-1} - d(x, x_{obstculo})^{-1}$.

Agentes Autônomos

Quais recompensas podemos atribuir para possíveis ações dos sistemas de exemplo?

- Casa Inteligente: $r_t = s(x) - c(x)$ (sensores corporais indicam relaxamento, número de chutes nos aparelhos inteligentes).

Quais recompensas podemos atribuir para possíveis ações dos sistemas de exemplo?

- Visita de rotina a pacientes
  $r_t = diag(x) + conforto(x) - alter(x)$.

Quais recompensas podemos atribuir para possíveis ações dos sistemas de exemplo?

- Sistema anti-invasão de computadores
  $r_t = -falhas(x) - recurso(x)$.

Nesse caso queremos encontrar uma função $f : S \times A \to \mathbb{R}$, ou seja, uma função que recebe um estado e uma ação e retorna a recompensa instantânea.

O aprendizado dessa função caracteriza o Aprendizado por Reforço, pois desejamos aprender apenas a recompensa instantânea que reforça ou desestimula uma determinada ação executada.

Exercício

Que tipo de recompensa poderia ser implementada para o jogo Super Mario World?

Exercício

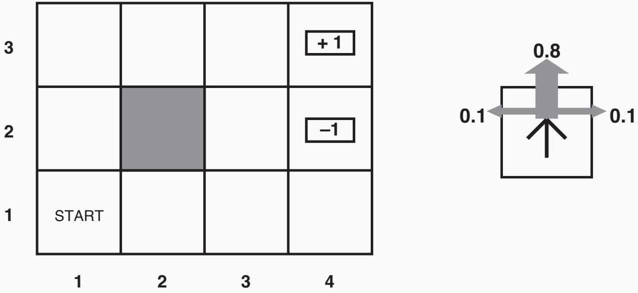
E para problemas de busca como 8-rainhas ou 8 Puzzle?

Exercício

E para o jogo da velha?

Problemas de Decisão Sequencial

Imagine que um agente esteja no seguinte ambiente:



(a) (b)

Problemas de Decisão Sequencial

Começando do *início* o agente deve escolher uma sequência de ações até chegar a um dos estados objetivos: $+1$ ou -1.

Problemas de Decisão Sequencial

Nesse exemplo $Acoes(s) = \{Cima, Baixo, Esquerda, Direita\}$.

Vamos assumir um ambiente totalmente observável.

Se o ambiente for determinístico, a solução é simples: *{Cima, Cima, Direita, Direita, Direita}*.

Digamos que cada **tentativa** de executar uma ação faz a ação requisitada com probabilidade $p = 0.8$.

Com $p = 0.1$ o agente executa uma ação perpendicular a ação requisitada, e $p = 0.1$ para a outra ação perpendicular.

Exemplo: ao requisitar que o agente ande para cima, ele terá 80% de chances de executar tal ação, 10% de chances de andar para a esquerda e 10% de chances de andar para a direita.

Caso o agente bata em uma das paredes, ele permanece no local atual.

O quadrado em cinza é uma parede e não pode ser acessada.

Pergunta: qual a probabilidade de o agente atingir o estado $+1$ com a solução *{Cima, Cima, Direita, Direita, Direita}*?

Ele deve executar a ação correta em cada uma das 5 ações na sequência: $0.8^5 = 0.32768$

Ele deve executar a ação correta em cada uma das 5 ações na sequência: $0.8^5 = 0.32768$

Mas também ele pode seguir a sequência *{Direita, Direita, Cima, Cima, Direita}* que tem probabilidade: $0.1^4 \times 0.8 = 0.32776$.

Modelo de Transição

Um **modelo de transição** descreve o resultado de cada ação em um determinado estado.

No nosso caso, ele é descrito como uma função de probabilidade $P(s'|s,a)$, ou a probabilidade de atingir o estado s' dado o estado atual s e a ação a.

Modelo de Markov

Esse tipo de modelo geralmente assume que as transições são **Markovianas**, ou seja, a probabilidade de chegar em s' a partir de s depende apenas e somente apenas de s, não levando em consideração todo o histórico de estados.

Função Utilidade

A **função utilidade** para um problema sequencial deve contabilizar **todo o histórico** de ações feitas até o estado final.

Função Utilidade

Para isso, definimos uma função de **recompensa** $R(s)$ que determinar um valor especificamente para o estado s. Esse valor pode ser positivo ou negativo, mas deve ser limitado.

Função Utilidade

No nosso exemplo, vamos definir que $R(s) = -0.04$ para qualquer estado exceto os estados finais em que $R(s) = +1, R(s) = -1$.

O significado dessa função R é a de que quanto mais ações precisamos para chegar ao final, menor será o valor da recompensa final.

Função Utilidade

A função utilidade pode ser definida como:

$$U(h) = \sum_{s \in h} R(s).$$

Processo de Decisão de Markov

Quando temos um problema de decisão sequencial que é:

- Totalmente observável
- Estocástico
- Modelo de transição Markoviano
- Recompensa aditiva

Denominamos de **Processo de Decisão de Markov (MDP)**.

Formalização Matemática

$$MDP = (S, A, R, \mathbb{P}, \gamma)$$

com:

- S: conjunto de possíveis estados.
- A: conjunto de ações.
- $R : S \to \mathbb{R}$: mapa de recompensa para cada estado.
- $\mathbb{P}$: probabilidade de transição de um estado para outro dada uma ação.
- γ: fator de desconto.

Markov Decision Process

1. No instante $t = 0$, inicia em um estado inicial $s_0 \sim p(s_0)$.
2. O agente seleciona uma ação a_t.
3. Amostra o próximo estado $s_{t+1} \sim p(s_t, a_t)$.
4. Agente recebe a recompensa r_t e o novo estado s_{t+1}.
5. Se s_{t+1} não for um estado final, retorna ao passo 2.

Markov Decision Process

Uma política $\pi : S \rightarrow A$ é uma função que dado um estado retorna a próxima ação a ser executada pelo agente.

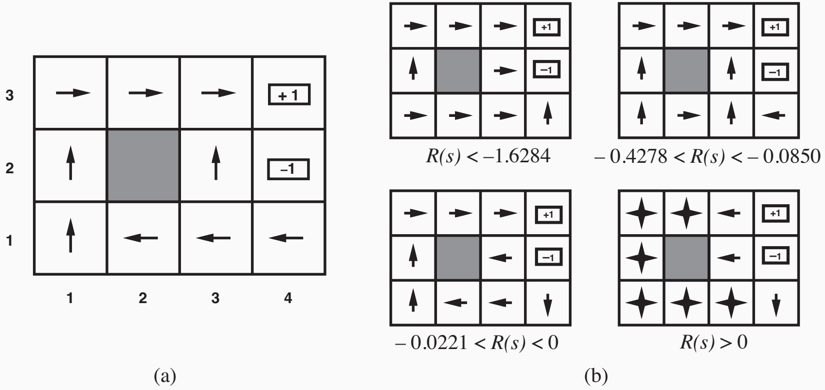
Markov Decision Process

O objetivo do nosso processo de decisão é encontrar uma política ótima π^* que maximiza a recompensa cumulativa descontada:

$$\sum_{t \geq 0} \gamma^t r_t$$

.

O valor de $R(s)$ pode influenciar as decisões tomadas pelo agente nas políticas ótimas:



Função de Utilidade Temporais

A função utilidade como a soma das recompensas do histórico dos estados não é a única possibilidade de função.

Dependendo da situação, outras funções podem ser mais interessantes. Para isso precisamos saber mais sobre nosso problema.

Horizonte

O nosso problema pode ter um **horizonte finito** ou **horizonte infinito**.

Horizonte finito: a quantidade de ações executadas é finita.

$$U(h = \{s_0, s_1, \cdots, s_{N+k}\}) = U(h = \{s_0, s_1, \cdots, s_N\})$$

Estacionaridade

A política ótima para um problema com *horizonte finito* é **não-estacionário**, ou seja, para um mesmo estado s a política ótima π^* pode agir diferente dependendo do tempo restante.

Estacionaridade

Para o caso de *horizonte infinito* o ambiente é **estacionário** e $\pi(s)$ não depende do tempo.

Funções Utilidade para ambientes Estacionários

Utilidade Aditiva

A **utilidade aditiva** é:

$$U(h = \{s_0, s_1, s_2, \cdots) = R(s_0) + R(s_1) + R(s_2) + \cdots.$$

Utilidade com Desconto

A **utilidade com desconto** é:

$$U(h = \{s_0, s_1, s_2, \cdots) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots.$$

com $0 \leq \gamma \leq 1$.

Utilidade com Desconto

Quando γ se aproxima de 0, as recompensas futuras são pouco importantes, quando $\gamma = 1$ temos a utilidade aditiva.

Utilidade com Desconto

Como estamos lidando com horizonte infinito, a utilidade com desconto evita a divergência do valor utilidade.

Utilidade Esperada

Em ambientes estocásticos, dada uma política π, a utilidade esperada partindo de um estado s é:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)\right]$$

de acordo com a função de probabilidade da execução de uma ação.

Politica Ótima

Com isso podemos encontrar a política ótima como:

$$\pi^* = argmax_{\pi} U^{\pi}(s)$$

Utilidade de um Estado

E a utilidade de um estado s como sendo:

$$U(s) = U^{\pi^*}(s)$$

Utilidade de Estado e Recompensa

A recompensa $R(s)$ indica a recompensa imediata do estado s enquanto a utilidade $U(s)$ indica a recompensa esperada do ponto s em diante.

Utilidade de um Estado

3	0.812	0.868	0.918	+ 1
2	0.762		0.660	–1
1	0.705	0.655	0.611	0.388
	1	2	3	4

Função de Política Ótima

Finalmente, a escolha de uma ação a dado um estado s em uma política ótima é:

$$\pi^*(s) = argmax_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s')$$

Resta determinarmos $U(s)$...

Equação de Bellman

Equação de Bellman

Richard Bellman descreveu a relação entre a utilidade de um estado s com as utilidades dos estados vizinhos:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s')$$

Calcule $U(1,1)$ utilizando seus vizinhos para a seguinte tabela de utilidade:

3	**0.812**	**0.868**	**0.918**	$\boxed{+1}$
2	**0.762**		**0.660**	$\boxed{-1}$
1	**0.705**	**0.655**	**0.611**	**0.388**
	1	**2**	**3**	**4**

Equação de Bellman

$$U(1,1) = -0.04 + \gamma max[0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1) \tag{1}$$

$$0.9U(1,1) + 0.1U(1,2) \tag{2}$$

$$0.9U(1,1) + 0.1U(2,1) \tag{3}$$

$$0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1)] \tag{4}$$

$$\tag{5}$$

Algoritmos para determinar a Utilidade

Algoritmo Iteração-Valor

Processo iterativo para determinar os valores de utilidade de cada estado. A cada iteração atualiza os valores de U como:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s')$$

Algoritmo Iteração-Valor

```
def valueIteration(mdp, eps):
    delta  = eps*(1 - gamma)/gamma + 1.

    while delta > eps*(1 - gamma)/gamma:
        U     = U’
        delta = 0
        for s in S:
            Uprime[s] = R[s] + gamma * maxAct(U, s)
            delta = max(delta, abs(U’[s] - U[s]))

    return U
```

Algoritmo Iteração-Valor

Esse algoritmo propaga os valores estimados de cada estado para seus vizinhos até encontrar um equilíbrio.

Note que o algoritmo converge para solução única apenas quando $\gamma < 1$.

Algoritmo Iteração-Política

Uma outra abordagem é estimar a melhor política e determinar o valor de U a partir dela através de dois passos:

- **Avaliação de política:** dada uma política π, calcule U^π, assumindo que π é ótimo.
- **Melhoria da política:** atualize a política π baseado nos valores de U^π.

Algoritmo Iteração-Política

```
def policyIteration(mdp):

    while mudou(U):
        U       = policyEvaluation(pi, U, mdp)
        for s in S:
            maxU = maxAct(U, s)
            maxE = expected(U, s)
            if maxU > maxE:
                pi[s]   = actioMaxExp(U, s)
    return pi
```

```
def policyEvaluation(pi, U, mdp):
    for s in S:
        U[s] = R[s] + gamma * expected(U, s)
    return U
```

Conclusão

Resumo

- **Processos de Decisão de Markov** são definidos por um modelo de transição, uma função de probabilidade do resultado de ações e uma função de recompensa para cada estado.
- A **utilidade** de uma sequência de estados é a soma, possivelmente descontada, das recompensas obtidas nessa sequência.
- A **utilidade de um estado** é a utilidade esperada ao executar uma política ótima partindo desse estado.