Unyleya
EDUCACIONAL

O AMBIENTE: PROCESSO DE DECISÃO DE MARKOV

O AMBIENTE: PROCESSO DE DECISÃO DE MARKOV

Resumo da Unidade I

O problema de aprendizagem por reforço envolve dois elementos: o **ambiente** e o **agente**.

A cada instante de tempo, o ambiente se encontra em um estado, cuja evolução é condicionada pelas ações que o agente toma. O agente é capaz de ler, a cada momento, o estado do ambiente, e responder com uma ação.

Após cada ação, o agente recebe um sinal de recompensa do ambiente. O objetivo do agente é maximizar a recompensa acumulada ao longo do tempo.

O AMBIENTE: PROCESSO DE DECISÃO DE MARKOV

Resumo da Unidade I

A separação entre agente e ambiente é flexível, e depende da formulação do problema:

Por exemplo, o agente pode ser:

- Nosso corpo, recebendo sinais físicos através dos sentidos, ou então apenas o cérebro, recebendo sinais eletroquímicos do resto do corpo através dos neurônios.
- Grupo de neurônios interagindo com o resto do cérebro.

Determinada ação executada pelo agente pode levar a períodos intermediários de recompensa negativa, para só mais tarde trazer bons resultados.

O agente deve ser capaz de realizar um planejamento visando não apenas uma boa recompensa imediata, mas uma boa recompensa acumulada a longo prazo. O agente não deve ser guloso em relação à recompensa imediata.

O AMBIENTE: PROCESSO DE DECISÃO DE MARKOV

Cenários representados usando processos de decisão de Markov

Representar situações onde é necessário executar ações em sequência em ambientes com incerteza.

Cenário: pode ser modelado como um processo de **decisão de Markov** abrange um sistema com vários estados, assim como ações que provavelmente modifiquem o estado do sistema e a probabilidade de perceber o resultado de cada ação executada. Dada a descrição do problema, resolvê-lo significa encontrar uma política ótima que determine a cada momento que ação tomar para maximizar a recompensa esperada (minimizar o custo esperado).

Elementos: o **ambiente** é visto como um processo de decisão de Markov, e o **agente** expressa-se em uma política que associa estados do ambiente com probabilidades de executar cada uma das ações pertencentes a um conjunto de ações possíveis.

O AMBIENTE: PROCESSO DE DECISÃO DE MARKOV

***Markov Decision Process* (MDP ou Processo de Decisão de Markov)**

É uma maneira de construir processos onde as transições entre estados são probabilísticas, sendo possível observar em que estado o processo está sendo possível interferir no processo de tempos em tempos executando ações (HAUSKRECHT, 1997; PUTERMAN, 1994).

Ações têm recompensas – ou custo – que decorrem do estado em que o processo se encontra. Recompensas podem ser definidas apenas por estado, sem que essas dependam da ação executada. Chamamos de “Markov” (ou “Markovianos”) os processos que foram construídos obedecendo à propriedade de Markov.

O resultado de uma ação em um estado está sujeito a ação e ao estado atual do sistema e são chamados de “processos de decisão” porque modelam a possibilidade de uma agente (ou “tomador de decisões”) interferir periodicamente no sistema executando ações, diferentemente de **Cadeias de Markov**, onde não se trata de como interferir no processo.

O AMBIENTE: PROCESSO DE DECISÃO DE MARKOV

Referências

HAUSKRECHT, M. Dynamic decision making in stochastic partially observable medical domains: Ischemic heart disease example. In: KERAVNOU, E. et al. (Ed.). 6th Conference on Artificial Intelligence in Medicine. [S.l.]: Springer, (Lecture Notes in Artificial Intelligence, v. 1211), p. 296–299. 1997.

HAUSKRECHT, M. Planning and control in stochastic domains with imperfect information. Tese (Doutorado) — EECS, Massachusetts Institute of Technology, 1997.

PUTERMAN, M. L. Markov Decision Processes: Discrete Stochastic Dynamic Programming. New York, NY: Wiley-Interscience, 1994. ISBN 0471619779.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

EXEMPLO: CADEIA DE MARKOV

EXEMPLO: CADEIA DE MARKOV

Processo de decisão de Markov

Uma maneira de descrever problemas de Aprendizado por Reforço que requerem tomadas de ações com incertezas, como diagnósticos médicos, tratamentos de enfermidades, controle da movimentação de um robô, e muito mais.

Consiste de um conjunto de estados (com estado inicial s_0); um conjunto de ações aplicáveis em cada estado; um modelo de transição e uma função de recompensa R(s).

O resultado de uma ação em um estado está sujeito a ação e ao estado atual do sistema (e não como o processo chegou a este estado).

Chamados de “processos de decisão” porque modelam a possibilidade de uma agente interferir periodicamente no sistema executando ações, diferentemente de **Cadeias de Markov**, onde não se trata de como interferir no processo.

EXEMPLO: CADEIA DE MARKOV

Processo de decisão de Markov

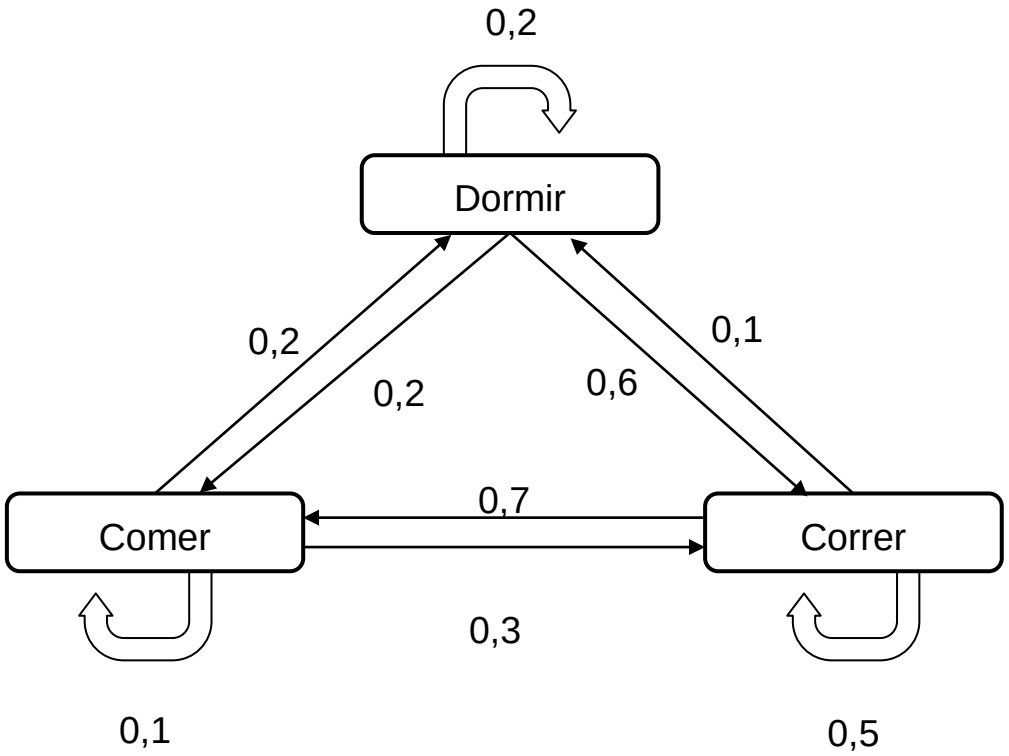
“Dado o presente, o futuro independe do passado”.

É possível refletir e aplicar o conceito de Cadeias de Markov de uma maneira mais leve de se entender, imagine que um estado seguirá essa propriedade caso todos os estados anteriores a ele não influenciarem na decisão do próximo estado, o único que influencia é apenas o estado atual.

EXEMPLO: PROCESSO DE DECISÃO DE MARKOV

Considere algumas atividades completamente independentes para realizar. Com elas identifique também os graus de probabilidade de ocorrência de uma, dada à outra.

Figura 1: Exemplo da Cadeia de Markov simplificado.



Fonte: a autora

EXEMPLO: PROCESSO DE DECISÃO DE MARKOV

Cadeia de Markov

A ideia é simples, vamos analisar as probabilidades das ações continuarem ou serem revertidas em outras atividades. No caso, avaliar as três ações, particularmente independentes, mas que podem levar uma a outra.

O conjunto de ações é:

$$S = \{Comer, Dormir, Correr\}$$

EXEMPLO: PROCESSO DE DECISÃO DE MARKOV

Cadeia de Markov

As probabilidades associadas, no exemplo, mostra que a correlação probabilística entre as ações é a seguinte:

- P(Correr) = 0,5;
- P(Comer) = 0,1;
- P(Dormir) = 0,2;
- P(Comer | Dormir) = 0,2;
- P(Dormir | Comer) = 0,2;
- P(Correr | Dormir) = 0,1;
- P(Dormir | Correr) = 0,6;
- P(Comer | Correr) = 0,3;
- P(Correr | Comer) = 0,7;

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

POLÍTICA E FUNÇÃO VALOR DE UM MDP

POLÍTICA E FUNÇÃO VALOR DE UM MDP

Política e função valor

A maioria dos algoritmos de aprendizagem por reforço envolvem a estimativa de valor de funções - funções de estados (ou de pares de ação do estado) que estimam quão bom é para o agente estar em um determinado estado (ou como é bom executar uma determinada ação em um determinado estado).

O termo “quão bom” pode ser definido em termos de recompensas futuras que podem ser esperadas ou para ser mais preciso, em termos de retorno esperado.

As recompensas que o agente pode esperar receber no futuro dependem das ações.

Função valor pode ser definida como as formas particulares de agir, chamadas **políticas**.

POLÍTICA E FUNÇÃO VALOR DE UM MDP

Definição: Política para um MDP

A regra de decisão para um processo de decisão de Markov em uma época de decisão k pode ser definida pela função $d_k : S \mapsto A$, que estabelece a ação a ser efetivada, dado o estado do sistema.

Uma política para um MDP é uma sequência de regras de decisão $\pi = \{d_0, d_1, \ldots, d_{z-1}\}$, uma para cada época de decisão.

É necessário também que seja definida a ideia de *valor* de uma política, determinada por uma **função valor**.

POLÍTICA E FUNÇÃO VALOR DE UM MDP

Definição: função valor de uma política para um MDP

Uma função valor da política π para um MDP $M = (S, A, T, R)$ é definida como a função $V^{\pi} : S \mapsto R$, tal que $V^{\pi}(s)$ atribuído ao valor que se espera da recompensa para esta política, consoante com o critério de otimalidade.

POLÍTICA E FUNÇÃO VALOR DE UM MDP

Política ótima e função valor ótima de um MDP

Dada política processada a partir do estado inicial, a natureza estocástica do ambiente pode levar a um histórico de ambiente distinto.

A qualidade de uma política pode ser medida pela utilidade esperada dos históricos de possíveis ambientes concebidos por essa política. Uma **política ótima** é, portanto, uma que fornece a utilidade que se espera mais alta (RUSSELL e NORVIG, 2010).

Resolver uma tarefa de aprendizado de reforço significa encontrar uma política que alcance muita recompensa a longo prazo.

POLÍTICA E FUNÇÃO VALOR DE UM MDP

Definição: política ótima para um MDP

Podemos definir precisamente uma política ótima da seguinte forma para MDPs finitos:

Um MDP com horizonte z pode ser definido como $M = (S, A, T, R)$.

Uma função valor V^* é ótima para $M \forall \cup \in v, \forall s \in S, V_k\ (s) \geq U_k\ (S)$ para todo $k \in \aleph$ tal que $0 \leq k < z$.

Uma política π^* é ótimo para M se $\forall \pi' \in \Pi, \forall s \in S, Q_k^{\pi^*}\ (s, d_k^{\pi^*}(s)) \geq Q_k^{\pi'}(s, d_k^{\pi'}(s))$ para todo $k \in \aleph$ tal que $0 \leq k < z$.

Ao menos uma política é melhor ou igual a todas as outras (política ótima).

POLÍTICA E FUNÇÃO VALOR DE UM MDP

Definição: política ótima para um MDP

A função do agente é representada explicitamente por uma política. Políticas ótimas compartilham a mesma função valor de estado chamada de função valor ótima, denotada por e definida como:

$$v_*(s) = max_\pi v_\pi(s)$$

Para todo $s \in S$.

As políticas ótimas também compartilham a mesma função valor ótima, denotada por:

$$q_*(s,a) = max_\pi q_\pi(s,a)$$

Para todo $s \epsilon S$ e a ϵ A(s). Para o par da ação (s, a), a função fornece o retorno esperado para a ação a no estado s e depois seguindo uma política ótima.

Podemos $\boldsymbol{q}^*$ escrever em termos de $\boldsymbol{v}^*$, como segue:

$$q_*(s,a) = E[R_{t+1} + \gamma v_*(S_t + 1)|S_t = s,, A_t = a]$$

POLÍTICA E FUNÇÃO VALOR DE UM MDP

Referências

RUSSELL, S.; NORVIG, P. **Artificial Intelligence:** A Modern Approach. 3. ed. New Jersey: Pearson Education, 2010.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

ALGORITMOS PARA SOLUÇÃO DE MDPs

ALGORITMOS PARA SOLUÇÃO DE MDPs

Algoritmos: iteração por política e iteração de valor

Iteração por política: sugere um caminho alternativo para encontrar políticas ótimas, que mostra que, quando uma ação é claramente melhor que todas as outras, a magnitude exata das utilidades nos estados envolvidos não precisa ser exata.

Iteração de valor: para calcular uma política ótima. O propósito é calcular a utilidade de cada estado. Em seguida, utilizar as utilidades do estado para eleger uma ação ótima em cada estado.

ALGORITMOS PARA SOLUÇÃO DE MDPs

Iteração por política

O algoritmo de iteração política alterna a **avaliação de política** e a **melhoria de política** começando com uma política inicial π_0.

Avaliação de política

Dada uma política π_i, calcular $U_i = U^{\pi i}$ se a utilidade de cada estado fosse executada. A avaliação de política pode ser expressa como uma versão simplificada da equação de Bellman, relacionando a utilidade de s (sob π_i), às utilidades de seus vizinhos:

$$U_i(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

ALGORITMOS PARA SOLUÇÃO DE MDPs

Avaliação de política

Espaços de estados pequenos, a avaliação de política usando métodos de solução exata em geral é a abordagem mais eficiente. Espaços de estados grandes, o tempo do algoritmo sendo $O(n^3)$ talvez não seja o recomendado.

Não é necessário fazer a avaliação de política exata. Podemos executar algum número de passos de iteração de valor simplificada para fornecer uma aproximação razoavelmente boa das utilidades. Nesse caso, a atualização de Bellman simplificada para esse processo é:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

Repetida k vezes para produzir a próxima estimativa de utilidade.

ALGORITMOS PARA SOLUÇÃO DE MDPs

Avaliação de política iterativa – algoritmo que implementa a atualização de Bellman

Figura 1: Algoritmo de avaliação de política iterativa.

função AVALIAÇÃO-DE-POLÍTICA-ITERATIVA

entrada: a política a ser avaliada

variáveis locais: um limiar pequeno > 0 determinando a precisão da estimativa

inicialize U(s), para todo s ϵ S, exceto V=0

repita

$\Delta \leftarrow 0$

para cada estado s ϵ S faça

$u \leftarrow (s)$

$u \leftarrow U(s) = R(s) + \gamma \sum_{s'} P(s' \mid s,(s))U(s')$

$\Delta \leftarrow max(\Delta, \mid u - U(s) \mid)$

até $\Delta < 0$

Fonte: adaptado de Sutton e Barto (2018)

ALGORITMOS PARA SOLUÇÃO DE MDPs

Avaliação de política iterativa – algoritmo que implementa a atualização de Bellman

Observe como o algoritmo lida com a terminação.

Formalmente a avaliação de política iterativa converge apenas no limite, mas na prática deve ser interrompida.

Dessa forma, o algoritmo testa a quantidade $\max_{s \in S} | U_{i+1}(s) - U_i(s)|$ depois de cada varredura e para quando é suficientemente pequeno.

ALGORITMOS PARA SOLUÇÃO DE MDPs

Referências

SUTTON, R. S.; BARTO, A. G. Reinforcement learning: an introduction, 2ªEd. MIT Press, Cambridge, Massachusetts, EUA, 2018.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

MELHORIA DE POLÍTICA

MELHORIA DE POLÍTICA

Melhoria de política

Consiste no processo de fazer uma nova política que melhora uma política original, tornando-a gulosa no que diz respeito à função valor da política original.

Suponha que a nova política gulosa, π' é tão boa quanto, mas não melhor que, a política original π. Em outras palavras, vamos considerar a nova política gulosa π' , dada por:

$$\pi'(s) = argmax_a \Sigma_{s',r} P(S', r \mid S, a)[r + \gamma v_\pi (S')]$$

Onde, $argmax_a$ denota o valor de a em que a expressão que se segue é maximizada.

A política gulosa toma a ação que parece melhor a curto prazo, de acordo com v_π .

MELHORIA DE POLÍTICA

Teorema da melhoria de política

Seja π e π' qualquer par de políticas determinísticas tais que, para todo $s \in S$,

$$q_\pi(S, \pi'(S)) \geq v_\pi(S)$$

A política deve ser tão boa ou melhor.

Podemos verificar que a política gulosa satisfaz as condições do teorema da melhoria política, e é tão boa quanto, ou melhor, que a política original.

MELHORIA DE POLÍTICA

Teorema da melhoria de política

Suponha que a nova política gulosa π' , é tão boa quanto, mas não melhor que a política original π . Então $v_\pi = v_{\pi'}$, para todo $s \epsilon S$, temos:

$$v_{\pi'}(S) = max_a \Sigma_{S',r} P(S', r \mid S, a)[r + \gamma v_{\pi'}(S')]$$

Melhoria de política, deve nos dar uma política estritamente melhor, exceto quando a política original já é ótima.

MELHORIA DE POLÍTICA

Algoritmo da iteração de política

A maneira para encontrar uma política ótima é chamada de iteração de política.

Uma vez que uma política π tenha sido melhorada usando para produzir uma política melhor π', podemos então calcular v_π e melhorá-la para obter π'' uma ainda melhor. Podemos, obter uma melhor sequência de políticas e funções valor:

$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$$

Onde $\xrightarrow{E}$ denota uma avaliação de política e $\xrightarrow{I}$ denota uma melhoria de política.

Cada política é garantidamente uma melhoria estrita em relação à anterior. Como um MDP finito tem apenas um número finito de políticas, esse processo deve convergir para uma política ótima e uma função valor ótima em um número finito de iterações.

MELHORIA DE POLÍTICA

Algoritmo da iteração de política

Cada avaliação de política é iniciada com a função valor da política anterior.

Figura 1: Algoritmo de iteração de política para calcular uma política ótima.

função ITERAÇÃO-DE-POLÍTICA(**mdp) retorna** uma política

entradas: *mdp*, um MDP com estados *S*, ações *A(s)*, modelo de transição *P(s' | s, a)*

variáveis locais: *U*, um vetor de utilidades para estados em *S*, inicialmente zero e,

π, um vetor de política indexado pelo estado, inicialmente aleatório

repita

U ← AVALIAÇÃO-DE-POLÍTICA (π, *U*, *mdp*)

inalterado? ← verdadeiro

para cada estado *s* **em** *S* **faça**

se $max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\, U[s'] > \sum_{s'} P(s' \mid s, \pi[s])\, U[s']$ **então faça**

$\pi[s] \leftarrow argmax_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\, U[s']$

inalterado? ← falso

até inalterado?

retornar π

Fonte: Russell e Norvig (2010)

Resulta em um grande aumento na velocidade de convergência da avaliação de políticas (pois a função valor muda pouco de uma política para outra).

MELHORIA DE POLÍTICA

Referências

RUSSELL, S.; NORVIG, P. **Artificial Intelligence:** A Modern Approach. 3. ed. New Jersey: Pearson Education, 2010.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

O ALGORITMO DE ITERAÇÃO DE VALOR

O ALGORITMO DE ITERAÇÃO DE VALOR

A equação de *Bellman*

A utilidade de uma sequência de estados é a **soma** das **recompensas descontadas** obtidas durante a sequência. Possibilita comparar políticas, comparando as utilidades esperadas obtidas quando as executamos.

Há uma relação direta entre a utilidade de um estado e a utilidade de seus vizinhos: “a utilidade de um estado é a recompensa imediata correspondente a esse estado mais a utilidade descontada esperada do próximo estado, assumindo que o agente escolha a ação ótima” (RUSSELL e NORVIG, 2010).

O ALGORITMO DE ITERAÇÃO DE VALOR

A equação de *Bellman*

A utilidade de um estado s é conhecida como **Equação de *Bellman*** e pode ser definida da seguinte forma:

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

Onde:

- γ é o fator de desconto;
- $U(S)$ é a recompensa total a "longo prazo" de s;
- $R(s)$ é a recompensa a "curto prazo" por estar em s.

Toda função valor satisfaz essa equação, dada uma política

O ALGORITMO DE ITERAÇÃO DE VALOR

O algoritmo de interação de valor

A equação de *Bellman* é a base do algoritmo de iteração de valor para a solução de MDPs. Caso haja n estados possíveis, teremos n equações de *Bellman,* sendo uma para cada estado. Assim, n equações contêm n incógnitas - as utilidades dos estados (RUSSELL e NORVIG, 2010).

Há um problema ao tentar **resolver** essas **equações simultaneamente** para encontrar as utilidades: as **equações não são** lineares porque o **operador** max **não** é um operador **linear**.

Em sistemas de equações lineares normalmente são solucionados rapidamente usando técnicas de álgebra linear (sistemas de equações não lineares são mais difíceis de ser solucionados).

Sanar esse problema: **abordagem iterativa**.

O ALGORITMO DE ITERAÇÃO DE VALOR

O algoritmo de interação de valor

Iniciaremos com valores aleatórios para as utilidades. Calculado o lado direito da equação e o resultado será inserido no lado esquerdo, atualizando a utilidade de cada estado a partir das utilidades de seus vizinhos. Repete esse processo até que se chegue a um equilíbrio.

Seja (s) o valor de utilidade para cada estado s na *i-ésima* iteração. Essa fase de iteração, é conhecida como **atualização de *Bellman***, podendo ser assim definida:

$$U_{i+1}(S) \leftarrow R(S) + \gamma max_{a \in A(S)} \Sigma_{s'} P(S' \mid S, a) U_i(S')$$

Assumimos que a atualização será aplicada juntamente a todos os estados em cada iteração. Aplicada com frequência infinita, podemos garantir o alcance de um equilíbrio e os valores finais de utilidade devem ser soluções para as equações de Bellman. Na prática, eles igualmente são as únicas soluções, e a política equivalente é ótima (RUSSELL e NORVIG, 2010).

O ALGORITMO DE ITERAÇÃO DE VALOR

O algoritmo de interação de valor

Figura 1: Algoritmo de iteração valor para calcular utilidades de estados.

função ITERAÇÃO-DE-VALOR(mdp, ε) **retorna** uma função utilidade

entradas: *mdp*, um MDP com estados *S*, ações *A(s)*, modelo de transição *P(s' | s, a)*, recompensa R*(s)*, desconto e o erro máximo permitido na utilidade de cada estado

variáveis locais: *U, U'*, vetores de utilidades para estados em *S*, inicialmente zero e, a mudança máxima na utilidade de qualquer estado em uma iteração

repita

$U \leftarrow U'; \leftarrow 0$

para cada estado s **em** S **faça**

$$U'(s) \leftarrow R[s] + \gamma max_{a \grave{o} A(s)} \sum_{s'} P(s' \mid s, a) U[s']$$

se $|U'[s] - U[s]| > \delta$ **então** $\delta \leftarrow |U'[s] - U[s]|$

até $\delta < \varepsilon(1-\gamma)/\gamma$

retornar U

Fonte: Russell e Norvig (2010)

O ALGORITMO DE ITERAÇÃO DE VALOR

O algoritmo de interação de valor

O algoritmo de iteração de valor pode ser usado para resolver MDPs com horizonte infinito, porque converge para uma função valor ótima.

Seja um horizonte finito, a equação de otimalidade pode ser assim definida:

$$\begin{aligned} U(s) &= HU(s) \\ &= max_{a \in A} h(s, a, U) \\ &= max_{a \in A}[R(s, a) + \gamma \Sigma_{s' \in S} P(S' \mid S, a) U(S')] \end{aligned}$$

Onde: H é definido como um operador H: $U \mapsto U$ de melhoria da política, determinando a melhor ação de um estado usando os valores U dos estados numa precedente época de decisão.

O ALGORITMO DE ITERAÇÃO DE VALOR

O algoritmo de interação de valor

Seja o operador H, e a equação de otimalidade para processos de decisão de Markov determinada por uma função valor em uma época de decisão a contar da função valor da época de decisão anterior.

Horizonte finito: o critério de parada é quando o número de iterações é igual ao horizonte do problema. O algoritmo produz uma política para cada época de decisão, sendo a política não estacionária. A complexidade assintótica do algoritmo é dada por $O(z\ |A|\ |S|^2)$.

Horizontes infinitos (ou indefinidos): o algoritmo é interrompido quando os valores para cada estado tiverem convergido. Na prova de convergência dos algoritmos de iteração de valores define-se um conceito essencial, conhecido como **contração em um espaço *Banach*.**

O ALGORITMO DE ITERAÇÃO DE VALOR

Referências

RUSSELL, S.; NORVIG, P. **Artificial Intelligence:** A Modern Approach. 3. ed. New Jersey: Pearson Education, 2010.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

CONTRATAÇÃO EM UM ESPAÇO DE BANACH

CONTRATAÇÃO EM UM ESPAÇO BANACH

Definição: espaço de *Banach*

Uma norma $||.||$ num espaço vetorial X é um espaço de *Banach* se toda sequência de *Cauchy* $(x_0, x_1, \dots, x_n)$ de elementos de X tem um limite $x_k \in X$.

Definição: contração

Dado X um espaço de *Banach*. Um operador $F: X \mapsto X$ é uma contração quando para quaisquer dois elementos $U, V \in X$.

$$\big||FV - FU|\big| \leq \beta||V - U||$$

Onde $0 \leq \beta \leq 1$ é o fator de contração.

CONTRATAÇÃO EM UM ESPAÇO BANACH

Teorema do ponto fixo de *Banach*

Seja X um espaço de *Banach*. Seja F: $X \mapsto X$ uma contração e seja $N = (x_0, x_1, \dots, x_n)$ uma sequência com um ponto inicial arbitrário$x_0, \in X$, tal que $x_i, = F_{x_i-1}$. Então:

- F tem um único ponto fixo x^*tal que $Fx^* = x^*$;
- A sequência N converge para x^*.

Seja $||.||$ uma norma no espaço V tal que $||V|| = max_{s \in S} |V(s)|$. V é um espaço de *Banach*.

O teorema de *Banach* garante que H tem um ponto fixo único V^*e que a sequência $V_k = HV_{k-1}$, iniciando de uma função valor qualquer, converge para este ponto fixo.

Alcançar uma função valor precisa do algoritmo, não é suficiente apenas garantir a convergência. Precisa determinar quando deve interromper o cálculo de aproximações sucessivas da função valor. O teorema do erro de *Bellman* é um critério de convergência bastante conhecido para isso.

CONTRATAÇÃO EM UM ESPAÇO BANACH

Teorema do erro de *Bellman*

Se a diferença entre $V_k(S) = V_{k-1}(S)$ é no máximo δ para todo estado s, então $V_k(S)$ nunca difere de $V^*(s)$ por mais de $\frac{\delta}{1-\gamma}$.

Assim, o critério de parada deve ser:

$$\forall\, s \in S, |V(S) - V'(S)| \leq \frac{\varepsilon(1-\gamma)}{2\gamma}$$

Para que a precisão da solução seja no mínimo ε.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

EXEMPLO DO SUPER MÁRIO

EXEMPLO DO SUPER MÁRIO

Jogo do Super Mario

Leva em consideração o Mario em uma fase do jogo, com o objetivo de salvar a Princesa das garras do Bowser.

O objetivo somente será atingido se ele seguir algumas regras e caminhos: ele precisa passar por várias fases, derrotando os inimigos que aparecem pelo caminho e pegando os brindes que aparecerem.

Para passar de uma fase no jogo, há a necessidade de um algoritmo treinado, imagine que é sabido quando, quantas vezes e quais botões, em quais sequências é preciso apertar para cada tipo de desafio na fase.

EXEMPLO DO SUPER MÁRIO

O algoritmo de interação de valor

Figura 1: Cadeia de Markov no jogo do Mario.

Fonte: Vasconcellos, 2018.

EXEMPLO DO SUPER MÁRIO

Jogo do Super Mario

Como já é sabido, na Cadeia de Markov, o estado que o Mario vai estar no futuro só depende do estado em que ele está no momento.

Considere que ele está parado, o fato dele está parado vai indicar no seu próximo movimento e estado, mas como ele estava antes de estar parado não importa.

Isto é, não interessa se Mario ficou pulando loucamente para passar um espaço vago, se ele está parado nos últimos minutos, é o que precisa ser analisado.

EXEMPLO DO SUPER MÁRIO

Jogo do Super Mario

Tabela 1. Tabela de transição do jogo do Mario com as suas probabilidades de mudança de estados.

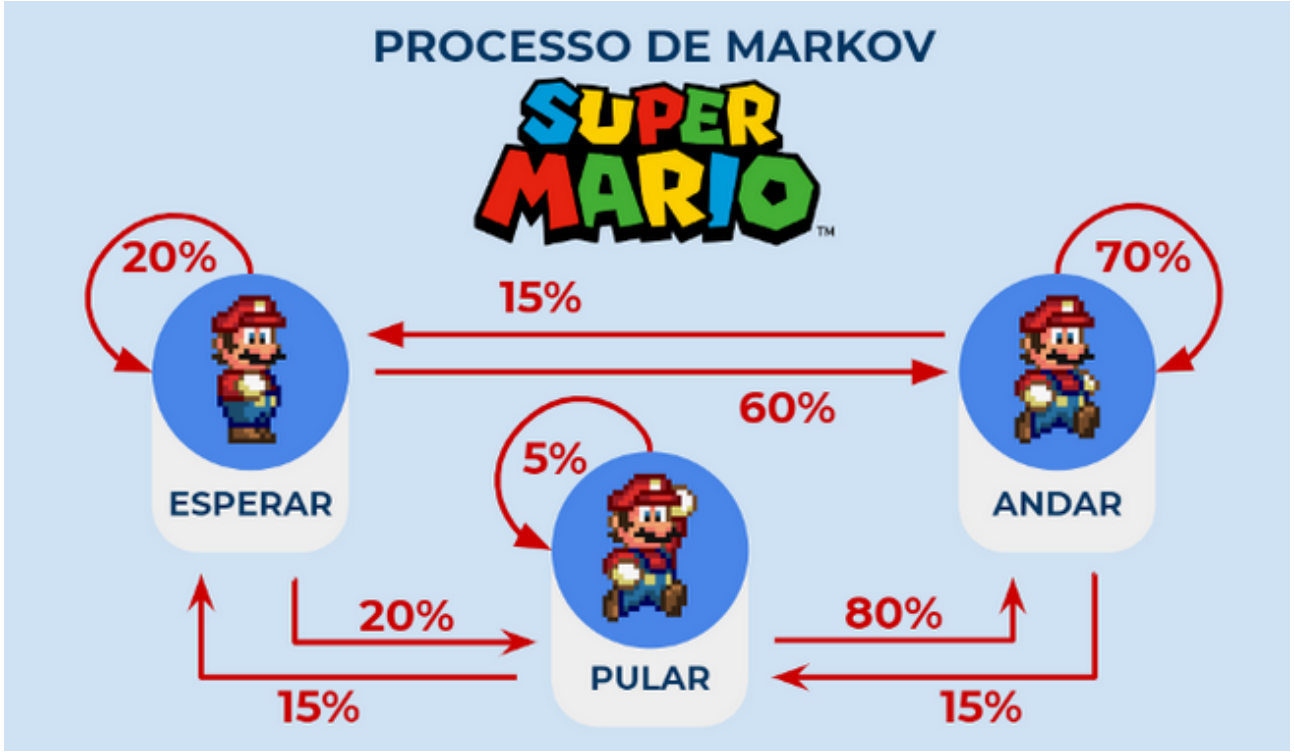
	Matriz de transição de Mario		
	Andar	**Ficar parado**	**Pular**
Andar	0,7	0,15	0,15
Ficar parado	0,6	0,2	0,2
Pular	0,8	0,15	0,05

Fonte: o autor.

EXEMPLO DO SUPER MÁRIO

O algoritmo de interação de valor

Figura 2: Processo de Markov para o jogo do Mario.



Fonte: Vasconcellos, 2018.

EXEMPLO DO SUPER MÁRIO

Jogo do Super Mario

O processo de decisão de markov é representado por algumas variáveis, que são:

- **S (conjunto finito de estados)** - o estado que o seu personagem ou o seu objeto pode se encontrar. No jogo do Mario, os exemplos de estados foram está parado, andando ou pulando. Porém, lembre-se que isto é avaliado para cada caso, imagine que eu estou jogando xadrez, o estado “pulando” não iria se encaixar.
- **A (conjunto finito de ações)** - igualmente intuitivo, a ação é o ato de fazer algo acontecer. Se quero que o Mario pule, eu preciso tomar alguma iniciativa, fazer alguma ação para que ele vá para o estado “pulando”. Como o assunto é o jogo do Mario, as ações são os movimentos que o jogador faz para que seu avatar (Mario, Luigi) vá para um estado, por exemplo, eu cliquei no botão “↑↑↑”, essa ação faz com que meu Mario pule 3 vezes.

EXEMPLO DO SUPER MÁRIO

Jogo do Super Mario

- **P (modelo de probabilidade)** - a probabilidade de uma ação que será escolhida pelo jogador para um estado futuro, ser baseada no estado atual do Mario. Se a fase é aquela em que tenho que “ir para o céu”, então a probabilidade de eu escolher a ação “↑” e ir para o estado pulando é muito maior do que a ação “←” e ir para o estado andando.
- **R (recompensa)** - é uma variável que representa um número que o Mario vai receber depois de executar a ação passada para ele e ele entrar no estado atual. Neste caso tem-se duas possibilidades como vimos nos conceitos, uma de recompensa positiva e uma de recompensa negativa. Imagine que você tentou pular para ir para o céu mas errou e caiu no abismo, sua ação levou a uma recompensa negativa, você foi punido, você perdeu uma vida! Mas se você pulou certo e chegou num bloco que te deu um cogumelo, sua ação te levou a uma recompensa positiva, e você foi beneficiado com esta ação.

EXEMPLO DO SUPER MÁRIO

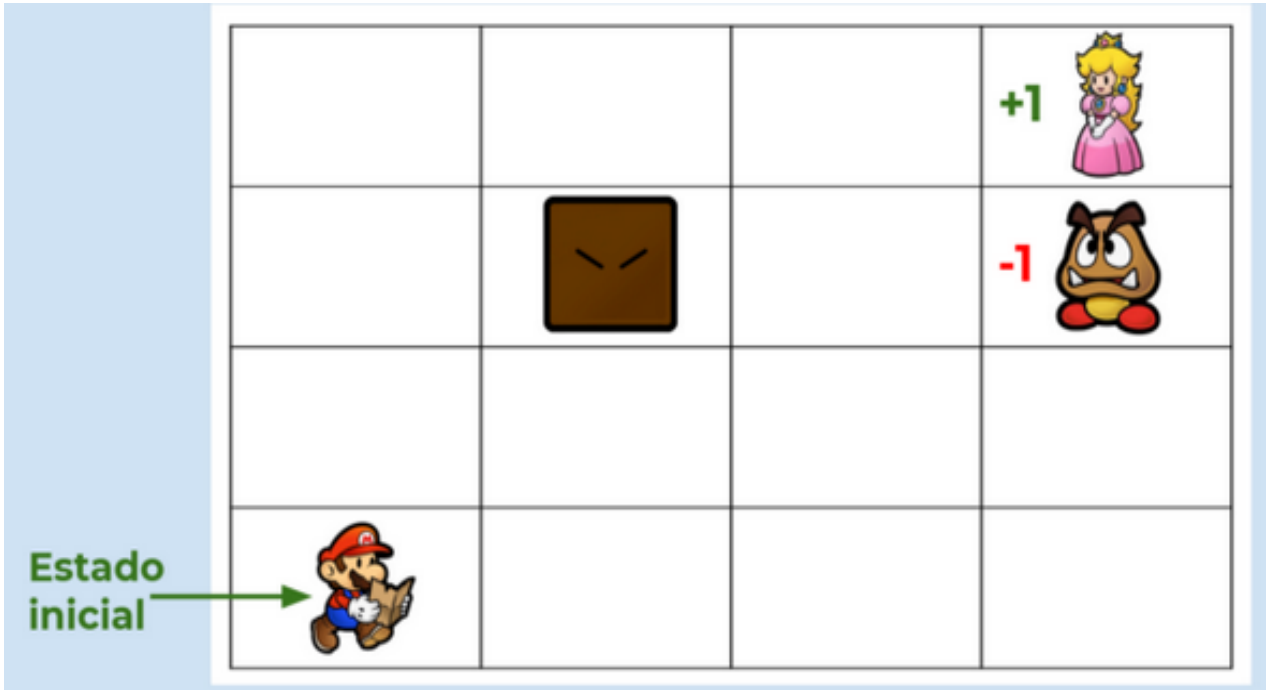
Jogo do Super Mario

- γ**(fator de desconto)** - é um número, que normalmente varia entre 0 e 1, que molda o total de recompensa que o agente vai ganhar no futuro. Vamos supor que no jogo, se Mario pegar o caminho A qualquer, que tem um fator de desconto de 0,7 e pode pegar 100 moedas. Mas se ele escolher o caminho B, que tem um fator de desconto 0,9, ele pode pegar 300 moedas. Isto significa dizer que suas ações atuais (que não tem influência nas ações futuras), ainda sim, podem interferir nelas - maximizar a recompensa futura, não só a atual.

EXEMPLO DO SUPER MÁRIO

O algoritmo de interação de valor

Figura 3: Matriz de estados do jogo do Mario de forma simplificada.

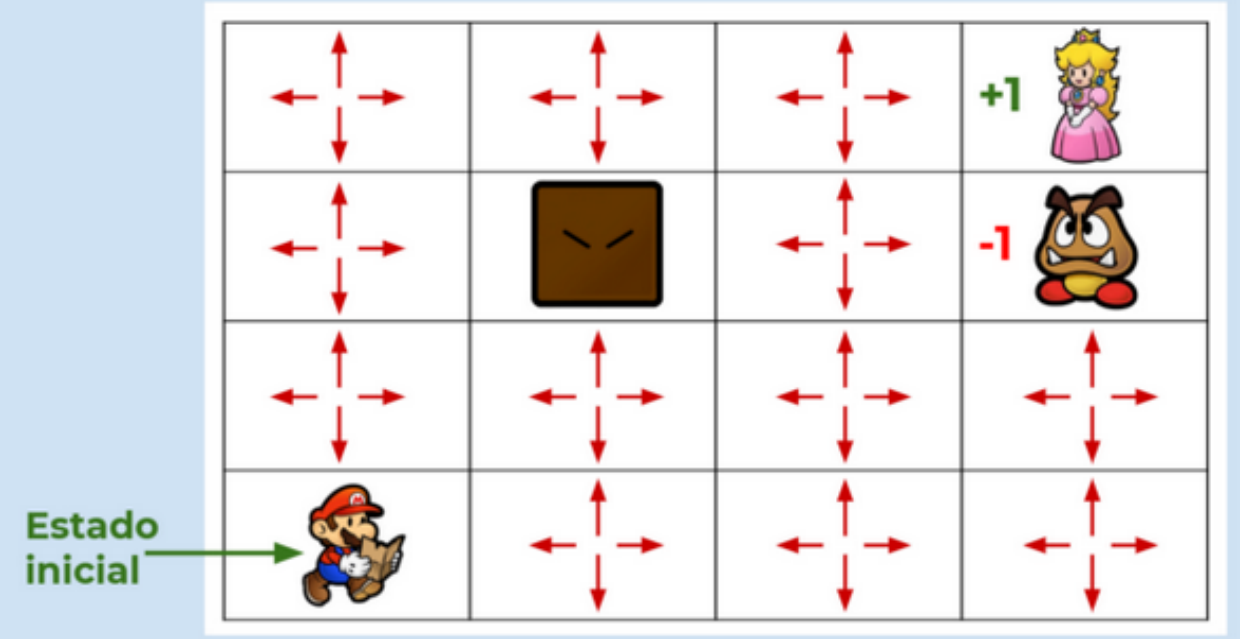


Fonte: Vasconcellos, 2018.

EXEMPLO DO SUPER MÁRIO

O algoritmo de interação de valor

Figura 4: Matriz do jogo do Mario com todas as ações possíveis em todos os estados

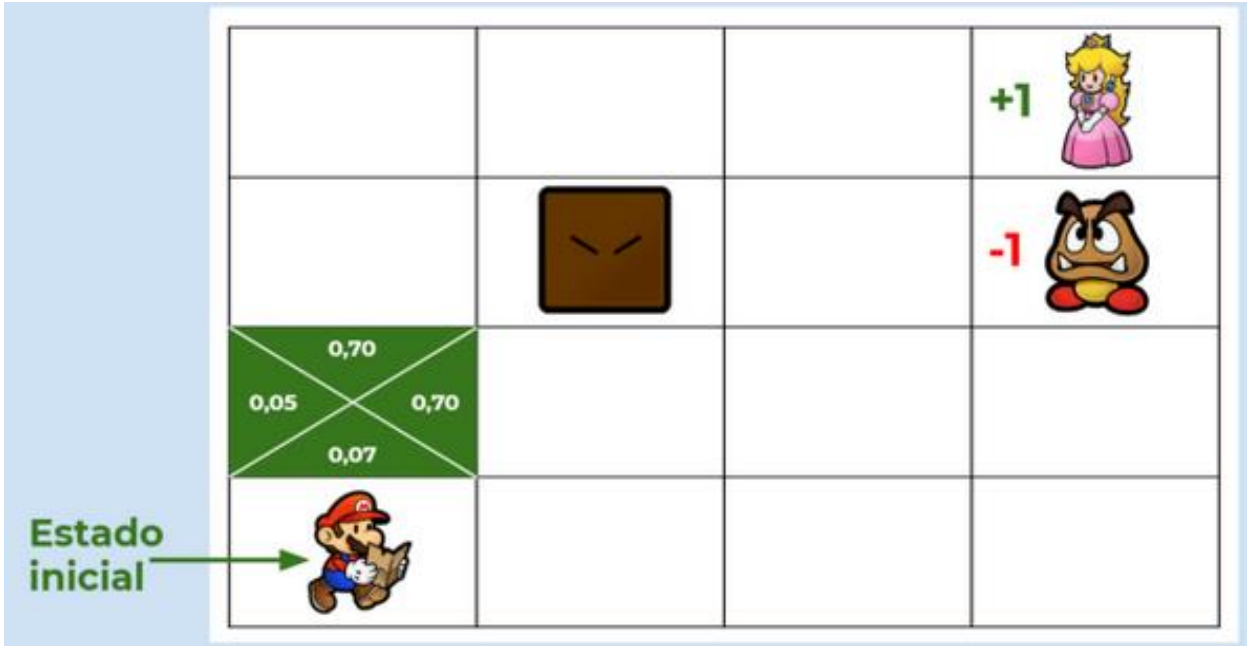


Fonte: Vasconcellos, 2018.

EXEMPLO DO SUPER MÁRIO

O algoritmo de interação de valor

Figura 5: Matriz do jogo do Mario com a posição 3 x 1 mostrando os graus de recompensa por cada possível ação.



Fonte: Vasconcellos, 2018.

EXEMPLO DO SUPER MÁRIO

Desafio!

Que tal você mesmo tentar definir alguns valores a cada um dos estados do nosso exemplo e traçar caminhos diferentes para calcular o valor de sua recompensa? É tipo um jogo de labirinto, mas que você já sabe o caminho para te levar a recompensa, você só não sabe se vai optar pelo que te concede a melhor recompensa.

EXEMPLO DO SUPER MÁRIO

Referências

VASCONCELLOS, PAULO. Explicando Deep Reinforcement Learning com Super Mario ao invés de matemática. 2018. https://paulovasconcellos.com.br/explicando-deep-reinforcement-learning-com-super-mario-ao-inv%C3%A9s-de-matem%C3%A1tica-4c77392cc733.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

PROCESSOS DE DECISÃO DE MARKOV PARCIALMENTE OBSERVÁVEIS

PROCESSOS DE DECISÃO DE MARKOV PARCIALMENTE OBSERVÁVEIS

Conceito

Um processo de decisão de Markov parcialmente observável (POMDP) é uma generalização de MDPs na qual o estado atual do sistema não é obrigatoriamente conhecido.

O agente recorda das ações que ele efetuou e das observações que reparou ao longo do tempo, tentando utilizar ainda essas informações para perceber sua próxima decisão.

Por exemplo:

- É provável que ao contrário de um “estado atual do sistema”, uma distribuição de possibilidades seja mantida durante o tempo em que as decisões são tomadas.

POMDPs são mais difíceis de solucionar que os MDPs, no entanto, são mais significativos.

PROCESSOS DE DECISÃO DE MARKOV PARCIALMENTE OBSERVÁVEIS

POMDPs usado para solucionar problemas em diversas áreas

Manutenção de máquinas, navegação de robôs, controle de elevadores, visão computacional, modelagem de comportamento em ecossistemas, aplicações militares, diagnóstico médico, educação e diversas outras áreas.

Exemplos:

- Hauskrecht (1997) modelou doenças isquêmicas do coração;
- Pineau (2004) usou POMDPs para moldar o comportamento de um robô que ajudou idosos a se recordarem de seus compromissos, acompanhando-os e guiando-os em (de maneira limitada) diálogos;
- Poupart (2005), que modelou um sistema que acompanhava o comportamento de pacientes com demência, utilizando uma câmera para monitorá-los, além de ajuda-los com os passos para lavar as mãos.

PROCESSOS DE DECISÃO DE MARKOV PARCIALMENTE OBSERVÁVEIS

Definição de POMDP

Um POMDP ou processo de decisão de Markov parcialmente observável é uma sequência ordenada (tupla) (S, A, T, R, Ω, O), em que:

- S representa um conjunto de estados em que o processo pode se encontrar;
- A representa um conjunto de ações que podem ser realizadas em diversas épocas de decisão;
- T: $S \times A \times S \mapsto [0, 1]$ é uma função que oferta a probabilidade do sistema ir para um estado s', dado que estava no estado s e a ação executada foi a;
- R: $S \times A \mapsto R$ é uma função que oferta a recompensa por tomar uma decisão a quando o processo está em um estado s;
- Ω é um conjunto de observações que são conseguidas em cada época de decisão;
- O: $S \times A \times \Omega \mapsto [0, 1]$ é uma função que dá a probabilidade de uma observação ser verificada, dado um estado s e a última ação a executada.

PROCESSOS DE DECISÃO DE MARKOV PARCIALMENTE OBSERVÁVEIS

Definição de POMDP

Em um POMDP não temos a oportunidade de verificar de forma direta o estado em que o sistema se encontra em um certo momento, ou seja, o agente não conhece o estado atual s (de forma análoga ao que acontece num problema construído como MDP).

Cada ação resulta em alguma percepção que é probabilisticamente relacionada ao estado do sistema.

Como o estado atual do sistema não é acessível ao agente, é possível usar o histórico anterior de ações e percepções para escolher a melhor ação.

PROCESSOS DE DECISÃO DE MARKOV PARCIALMENTE OBSERVÁVEIS

Referências

HAUSKRECHT, M. Dynamic decision making in stochastic partially observable medical domains: Ischemic heart disease example. In: KERAVNOU, E. et al. (Ed.). 6th **Conference on Artificial Intelligence in Medicine**. [S.l.]: Springer, (Lecture Notes in Artificial Intelligence, v. 1211), p. 296–299. 1997.

PINEAU, J. Tractable Planning Under Uncertainty: Exploiting Structure. Tese (Doutorado)— Robotics Institute, Carnegie-Mellon University, 2004.

POUPART, P. Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes. Tese (Doutorado)— University of Toronto, 2005.

Obrigada!

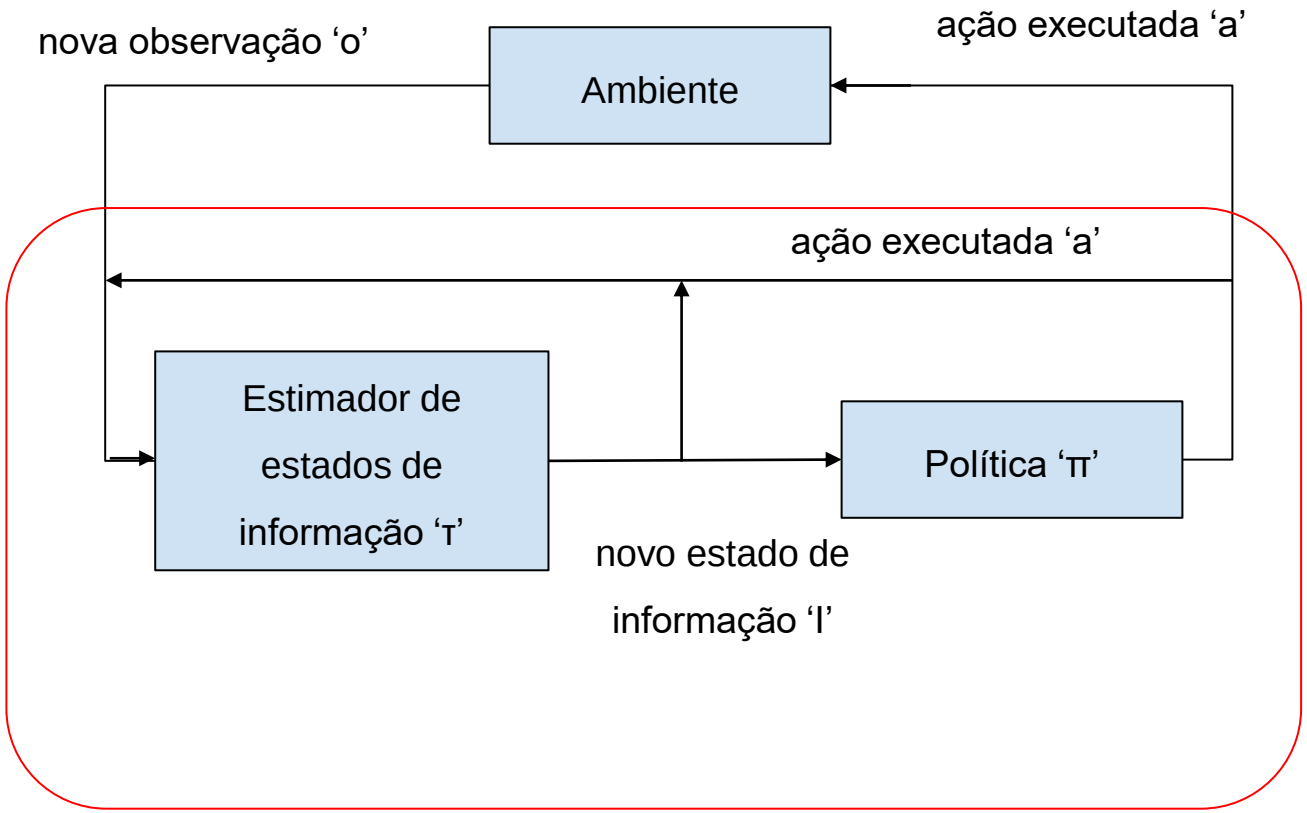
hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

DINÂMICA DE SISTEMAS MODELADOS COMO POMDPS

Dinâmica de sistemas modelados como POMDPs

Figura 1: Funcionamento de um sistema modelado com POMDP.



Fonte: Pellegrini e WAINER (2007)

Dinâmica de sistemas modelados como POMDPs

Referências

PELLEGRINI, JERÔNIMO. WAINER, JACQUES. Processos de Decisão de Markov: um tutorial. Instituto de Computação, Unicamp, Caixa Postal 6176, CEP 13084-971, Campinas – SP, Brazil. RITA. Volume XIV. Número 2. 2007.

Obrigada!

hulianeufrn@gmail.com