Inteligência Artificial

Aula 17

Profª Bianca Zadrozny
http://www.ic.uff.br/~bianca/ia

Tomada de decisões complexas

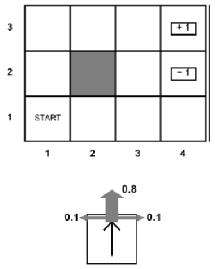
Capítulo 17 – Russell & Norvig

Seções 17.1 e 17.2

Decisão Sequencial

- Cap. 16 = tomada de decisão simples ou instantânea
  - Apropriado para ambientes episódicos não-determinísticos
- Cap. 17 = tomada de decisão sequencial
  - Utilidade do agente depende de uma sequencia de decisões
  - Generalização dos problemas de busca (cap. 3)
    - Agora incluímos incerteza e utilidades

Exemplo



- Agente tem probabilidade de 0.8 de se mover na direção desejada e 0.2 de se mover em ângulo reto.
- Se não houvesse incerteza, poderíamos usar busca para encontrar a solução ótima.
- Os estados finais tem recompensa +1 e -1.
- Todos os outros estados tem recompensa -0.04.
- A medida de desempenho é a soma das recompensas.

Processo de Decisão de Markov (PDM)

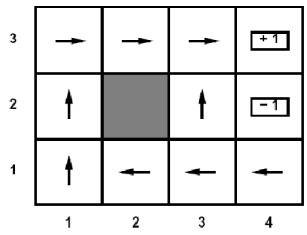
- Especifica um problema de decisão sequencial.
- Definido por:
  - Um conjunto de estados $s \in S$
  - Um conjunto de ações $a \in A$
  - Uma modelo de transição $T(s, a, s')$
    - Probabilidade de se alcançar s' a partir de s se a for executada.
    - Propriedade de Markov: essa probabilidade depende apenas de s e a e não do histórico de estados e ações.
  - Uma função de recompensa $R(s)$
  - Um estado inicial (ou distribuição inicial)
  - (Talvez) Um ou mais estados terminais

Resolvendo PDMs

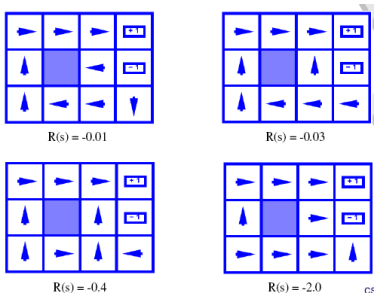
- Num ambiente determinístico com um único agente, a solução é um **plano** = sequencia ótima de ações.
- Num ambiente não-determinístico, a solução é uma **política** = especifica uma ação para cada estado.
  - A **política ótima** é a que produz a utilidade esperada mais alta possível.
  - Define um agente de reflexo simples.

Exemplo

- Política ótima quando os estados não-terminais tem recompensa $R(s)$ = -0.04.



Exemplo



Utilidades das Sequencias

- Para formalizar a função de utilidade temos que definir a utilidade de uma sequencia de estados.
  - Usamos a notação $U_h([s_0, s_1, \ldots, s_n])$
  - No exemplo, a utilidade era a soma das recompensas de cada estado, mas essa não é a única possibilidade.

Utilidade das Sequências

- Teorema: Se o agente tiver preferências estacionárias, ou seja,

$$[s_0, s_1, s_2, \ldots] \succ [s_0, s'_1, s'_2, \ldots]$$
$$\Leftrightarrow$$
$$[s_1, s_2, \ldots] \succ [s'_1, s'_2, \ldots]$$

então só existem duas possibilidades para a utilidade de uma sequência:

  - Recompensas aditivas
$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \cdots$$
  - Recompensas descontadas
$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$$

onde γ é um número entre 0 e 1 chamado de fator de desconto

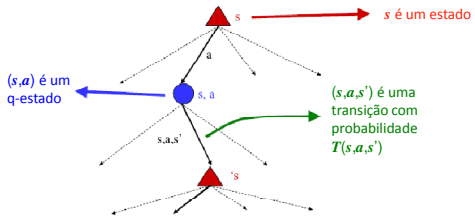
Utilidades Infinitas?

- Problema: sequências infinitas com soma de recompensa infinita.
- Soluções:
  - Horizonte finito: terminar episódios depois de T passos.
    - Gera um política não-estacionária (depende de quantos passos faltam para o fim).
  - Garantir que toda política sempre alcança um estado final.
  - Usar recompensas descontadas.

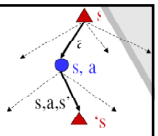
$$U_h([s_0, s_1, s_2, \ldots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = R_{\max}/(1-\gamma)$$

- Quanto menor o valor de γ menor o "horizonte"

Árvore do MDP

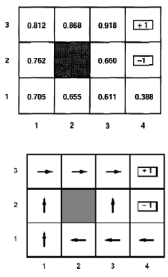


Utilidades Ótimas



- Operação fundamental: Calcular a utilidade ótima de cada estado s.
  - Valores ótimos definem políticas ótimas!
- Definir a utilidade de um estado s.
  - $U(s)$ = retorno esperado de se começar em s e agir de forma ótima.
- Definir a política ótima.
  - $\pi^*(s)$ = ação ótima a partir do estado s.

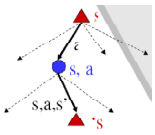
$$\pi^*(s) = \operatorname*{argmax}_a \sum_{s'} T(s,a,s')U(s')$$



Equação de Bellman

- Equação recursiva definindo a utilidade de um estado:

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$



- É a recompensa imediata correspondente a esse estado + a utilidade descontada esperada do próximo estado, supondo que o agente escolha a ação ótima.

Resolvendo a Equação de Bellman

- Por que não usar algoritmos de busca?
  - Árvore pode ser infinita
  - Teríamos que fazer uma busca pra cada estado
  - Repete muitas vezes os mesmos cálculos sempre que o mesmo estado for alcançado.
- Ideia: Iteração de valor
  - Calcular valores de utilidade ótimos para todos os estados simultaneamente, usando aproximações sucessivas.

Iteração de Valor

- Calcular estimativas $U_i(s)$
  - Retorno esperado de se começar no estado s e agir de forma ótima por i passos.
  - Começamos com $i = 0$ e vamos aumentando o valor de i até a convergência (isto é, valores não mudam de i para $i + 1$).
    - A convergência é garantida com horizonte finito ou recompensas descontadas.

Iteração de Valor

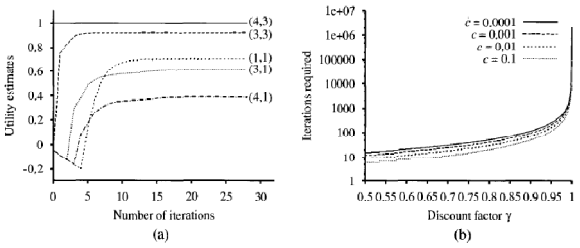
- Inicializar $U_0(s) = 0$.
- Calcular $U_{i+1}(s)$ a partir de $U_i(s)$ usando a equação:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U_i(s')$$

chamada de atualização de Bellman.

- Repetir o passo 2 até convergência isto é $U_{i+1}(s) \approx U_i(s)\ \forall s$

Exemplo: Iteração de Valor



Erro: $\epsilon = c \cdot R_{\max}$

Exemplo: Iteração de Valor

- Ver demo em:
  http://people.cs.ubc.ca/~poole/cs522/2000/mdpapplet/vi.htm