Planejamento Probabilístico

MAC499 - Trabalho de Formatura Supervisionado

Helton Massato Kishi
n.o USP - 4987051

Supervisora: Leliane Nunes de Barros

O que é Planejamento Probabilístico?

Planejamento é o processo de escolher e organizar ações através da antecipação de seus efeitos, tendo como objetivo satisfazer uma meta pré-estabelecida.

No Planejamento Probabilístico, os efeitos das ações são incertos. Para cada ação existe uma distribuição de probabilidade dos efeitos da ação.

Markov Decision Process

Markov Decision Process (MDP) é um modelo matemático usado para resolver o problema de planejamento probabilístico.

- **S**, um conjunto de estados;
- **A**, um conjunto de ações;
- **P**, uma função de transição de estados P: A x S x S -> [0,1], onde **P(a,s,s')** é a probabilidade do agente ir do estado s para o estado **s'** após da execução da ação **a**;
- **R(s)**, uma função recompensa atribuída para cada estado.
- **C(s,a)**, é a função do custo da ação **a** no estado **s**.

Algoritmo de Iteração por Valor

Resolver um MDP é o problema de encontrar uma política pi: S -> A. Uma política ótima indica qual é a melhor ação que o agente deve executar em cada estado do MDP. Para encontrarmos tal política é preciso escolher ações que minimizem o valor V(s) calculado da seguinte maneira:

$$V(s) = \min_{a \in A} \{C(a,s) - R(s) + \sum_{s' \in S} \gamma P_a(s'|s) V(s')\}$$

Esta fórmula é chamada de equação de Bellman.

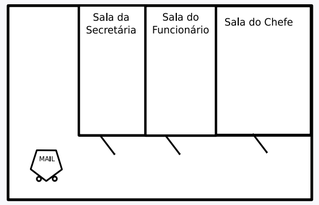
$$\pi(s) = \arg\min_{a \in A} \{C(a,s) - R(s) + \sum_{s' \in S} \gamma P_a(s'|s) V(s')\}$$

O algoritmo IV garante encontrar uma política ótima usando programação dinâmica. A idéia do algoritmo é inicializar um vetor V(s) com valores arbitrários e, a cada iteração, usa-se a equação de bellman para atualização dos valores de V(s) e cálcula-se o valor do residual, que é a diferença entre os valores de V(s) antes e depois da iteração. Se o valor do residual for menor do que epsilon, um valor muito pequeno, o algoritmo termina, obtendo assim a política ótima. Caso contrário, o algoritmo executa novamente a iteração.

Robô de entrega de cartas.

Considere um robô que faz a entrega de cartas dentro de um andar de escritórios composto por três salas: a sala do chefe, do funcionário e da secretária. O robô deve decidir, em um dado instante, para quem deve entregar a carta. O robô nem sempre consegue obter sucesso na entrega das cartas, pois existe a possibilidade de o destinatário não estar presente na sala quando o robô tenta fazer uma entrega. Há também prioridades diferentes. É muito mais importante entregar a carta ao chefe do que entregar a carta para a secretária.

O estado pode ser definido usando três variáveis booleanas, uma para cada destinatário, indicando se o robô entregou ou não a carta para o respectivo destinatário. São eles HD0 (Have Delivered 0), HD1 (Have Delivered 1) e HD2 (Have Delivered 2), sendo o destinatário 0 a secretária, 1 o funcionário e 2 o chefe. O custo da ação é definido pelo custo de energia gasta pelo robô para entregar a carta (que é a mesma para todos os destinatários). A recompensa é definida de acordo com a prioridade de entrega.



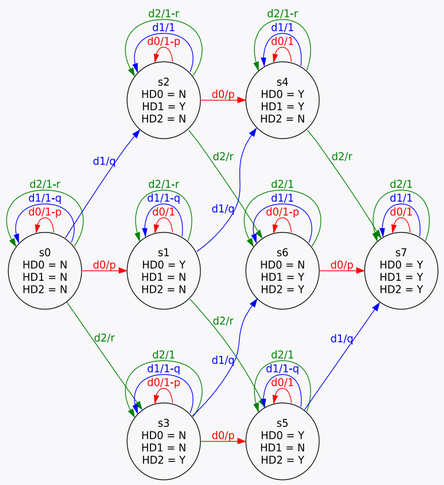
Ações possíveis:
d0 = deliver0 = entregar uma carta para a secretária
d1 = deliver1 = entregar uma carta para o funcionário
d2 = deliver2 = entregar uma carta para o chefe

Probabilidades:
p = prob. de encontrar a secretária em sua sala = 0,9
q = prob. de encontrar o funcionário em sua sala = 0,7
r = prob. de encontrar o chefe em sua sala = 0,5

Recompensa
R[] = [0, 1, 2, 4, 3, 5, 6, 7]

Função de custo:
O custo de todas as ações é igual a 10.

Função de multa:
M[d0] = [10, 9, 8, 6, 7, 5, 4, 3]
M[d1] = [10, 9, 8, 6, 7, 5, 4, 3]
M[d1] = [10, 9, 8, 6, 7, 5, 4, 3]



Grafo da matriz de transição do problema. Uma aresta d0/p significa que a aresta tem probabilidade p para a ação d0. Para facilitar a visualização, as arestas com mesma cor são as arestas de uma mesma ação.

Política IV:

s0	-> d2
s1	-> d2
s2	-> d2
s3	-> d1
s4	-> d2
s5	-> d1
s6	-> d0
s7	-> d0

Política RTDP:

s0	-> d0
s1	-> d1
s2	-> null
s3	-> null
s4	-> d2
s5	-> null
s6	-> null
s7	-> null

Política LRTDP:

s0	-> d2
s1	-> d0
s2	-> d0
s3	-> d1
s4	-> d2
s5	-> d1
s6	-> d0
s7	-> d0

SSP: Caminho Estocástico Mínimo

O SSP (do inglês, Shortest Stochastic Path) é um sub-problema do problema de MDP em que, além dos estados, ações, função de transição, precisamos definir o estado inicial e o conjunto dos estados meta. Assim, um problema de planejamento probabilístico pode ser naturalmente descrito como um problema de SSP.

Algoritmo RTDP e LRTDP

O algoritmo RTDP (Real-Time Dynamic Programming) e LRTDP (Labeled Real-Time Dynamic Programming) resolvem o problema de SSP, devolvendo uma política parcial.

O algoritmo RTDP devolve uma política parcial que não é ótima.

O algoritmo LRTDP devolve uma política parcial ótima.

Conclusão

Soluções clássicas de planejamento probabilístico usando MDPs geram políticas totais, isto é, mapeiam ações para todo estado do MDP. No entanto, para problemas de planejamento, em que são dados o estado inicial e o estado meta, o problema do MDP se reduz a um problema do tipo SSP. Com a implementação dos algoritmos IV, RTDP e LRTDP foi possível verificar para o problema do robô uma redução no número de atualizações de V(s).

Estado	# de atualiz. RTDP	# de atualiz. IV	# atualiz. LRTDP
s0	1	14	62
s1	2	14	1
s2	0	14	1
s3	0	14	16
s4	2	14	1
s5	0	14	1
s6	0	14	7
s7	0	14	1

RTDP é o algoritmo que faz menor número de atualizações, mas não devolve uma política ótima.
Já o IV é o algoritmo que devolve uma política total ótima, no entanto faz o maior número de atualizações.
O LRTDP devolve uma política parcial ótima e faz menos atualizações que o IV e RTDP.