Capítulo 3 - Sutton

Paulo Henrique Albuquerque

2023-04-25

1 O Problema de aprendizagem por reforço

O problema de aprendizagem por reforço é um esqueleto para o problema de aprender a partir de interações para atingir um objetivo. Cada ator do problema recebe um nome especial. O tomador de decisões é o *agente*. O objeto que interaje com o agente é o *ambiente*. O agente interaje sequêncialmente com o ambiente escolhendo ações. Então, o ambiente responde a essa ação e apresenta novas situações para o agente. Além disso, o ambiente fornece recompensas ao agente. As recompensas são valores numéricos que o agente tenta maximizar ao longo do tempo. Uma especificação completa de um ambiente define uma *tarefa*, que é uma instância do problema de aprendizagem por reforço.

O arcabouço para o problema é o seguinte: o agente interaje com o ambiente em cada passo de tempo de uma sequência de passos discretos, $t = 0, 1, 2, \ldots$. A cada passo de tempo t, o agente percebe o estado do ambiente, $S_t \in S$, onde $\mathcal{S}$ é o conjunto de estados possíveis, então decide qual ação $A_t \in \mathcal{A}(S_t)$, onde $\mathcal{A}(S_t)$ é o conjunto de ações possíveis no estado S_t. No próximo passo de tempo, como um consequência de sua ação, o agente recebe uma recompensa numérica, $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$ e transita para um novo estado S_{t+1}.

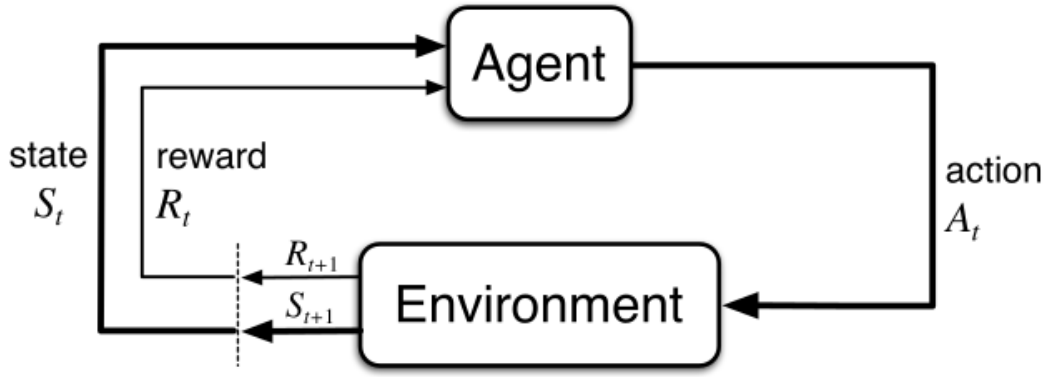

Figure 1:

A cada passo, o agente usa uma política π, que é um mapeamento de estados para as probabilidades de selecionar alguma ação. A notação é: $\pi_t(a|s)$ representa a probabilidade do agente escolher a ação $A_t = a$ no tempo t, dado que $S_t = s$. Os algoritmos de aprendizagem por reforço buscam especificar como o agente deve mudar e melhorar sua política π a medida que vai experienciando. O objetivo do agente, em linhas gerais, é máximar a quantidade total de recompensa que ele recebe durante o processo completo.

A fronteira entre agente e ambiente é abstrata e depende muito da especificação do problema. A regra geral é que qualquer coisa que não pode ser arbitrariamente mudada pelo agente é considera externa a ele e, portanto, faz parte do ambiente. Outro ponto interessante é que o agente pode ter algum ou total conhecimento sobre o ambiente, mesmo sendo entidades separadas. Até em um arranjo onde o agente tem conhecimento total sobre o ambiente, a tarefa pode uma instância de aprendizagem por reforço difícil de resolver. Por exemplo: sabemos exatamento como o puzzle do cubo de Rubik funciona, porém resolvê-lo ainda é uma tarefa difícil. Ou seja, a fronteira agente-ambiente é determinada por aquilo que o agente não tem total controle, e não por aquilo que ele não tem conhecimento.

Em resumo, qualquer problema de aprendizagem dirigida a objetivo pode ser reduzida para três sinais passados "back and forth" entre um agente e um ambiente. Evidentemente, esse arcabouço pode não ser suficiente para representar todos os problemas possíveis, porém, foi provado ser bem útil numa grande variedade de aplicações.

2 Objetivos e Recompensas

Hipotése de Recompensa: Tudo o que queremos dizer com objetivos e propósitos pode ser pensado como a maximização do valor esperado da soma acumulada de um sinal escalar recebido (chamado de recompensa).

Formalizar objetivos através de sinais de recompensa pode parecer limitante a priori. No prática, porém, tem se provado bem flexível e abrangente.

Uma observação importante. É crucial que as recompensas representem verdadeiramente o que queremos que o agente atinga. Em particular, o sinal de recompensa não é o local para dar ao agente conhecimento prévio sobre *como* atingir o objetivo. Por exemplo, um agente que joga xadrez deve ser recompensado por ganhar e não por, por exemplo, capturar o rei do oponente. Se esses sub-objetivos forem recompensados, o agente pode encontrar uma forma de atingir eles sem atingir o objetivo real. No exemplo do xadrez, o agente poderia aprender uma forma de capturar o rei do oponente mesmo que isso custe a derrota.

Note que a fronteira do agente está no limite de seu controle, não do seu corpo físico. O objetivo do agente deve ser algo que ele tem controle imperfeito. Portanto, colocamos a fonte de recompensas fora do agente. Isso não impede que o agente defina recompensas internas. Isso é exatamente o que muitos métodos de aprendizagem por reforço fazem.

3 Retornos

Denotamos a sequência de recompensas recebidas após o tempo t por: $R_{t+1}, R_{t+2}, \ldots$ queremos maximizar o *retorno esperado*, onde G_t é definido como alguma função específica da sequência de recompensas. A função mais simples a se pensar é a soma simples das recompensas:

$$G_t = R_{t+1} + R_{t+2} + \ldots + R_T.$$

onde T é o tempo final. Isso faz sentido em problemas em que há uma noção natural de tempo final. Esses problemas são geralmente *episódicos* no sentido que a intereção agente-ambiente é naturalmente quebrada em *episódios*, como partidas de xadrez, passeios em um labirinto, etc. Cada episódio acaba num estado *terminal* seguido por um *reset* de algum tipo. Em tarefas com episódios, o conjunto de todos os estados não terminais é denotado por $\mathcal{S}$ e o conjunto de todos os estados (não terminais+terminal) é denotado por $\mathcal{S}^\star$.

Entretanto, em muitas interações agente-ambiente, não há essa quebra natural em episódios. Esses cenários são conhecidos como tarefas contínuas, por exemplo: um agente robô de longa vida. Em situações desse tipo, o retorno acima é problemático pois G_t pode crescer indefinidamente. Nesses casos, definimos G_t como sendo

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}.$$

onde γ é um parâmetro, $0 \leq \gamma \leq 1$, chamado *taxa de desconto*. A taxa de desconto determina o valor presente de recompensas futuras: uma recompensa recebida depois de k passos vale apenas γ^{k-1} vezes o que valeria se tivesse sido recebida imediatamente.

4 Unindo tarefas episódicas e contínuas

Para unificar a notação para o retorno G_t para tarefas episódicas e contínuas, introduzimos o conceito de *estados absorvedores*. São estados com uma única transição com probabilidade não nula (portanto com probabilidade 1) para ele mesmo. Além disso, a recompensa é sempre 0 nesse estado para qualquer ação. O diagrama abaixo exemplifica uma tarefa episódica com $T = 3$ utilizando um estado absorvedor.

Começando por S_0, recebemos a seguinte sequência de recompensas: $+1, +1, +1, 0, 0, 0, \ldots$. Obtemos o mesmo retorno se somarmos os T primeiros termos ou fazendo a soma infinita. Então, definimos o retorno como sendo

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}.$$

incluindo a possibilidade de $T = \infty$ ou $\gamma = 1$ (mas não ambas).

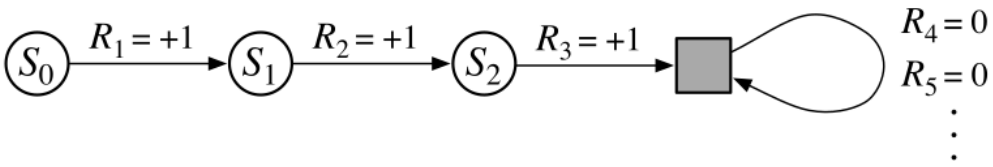

Figure 2: absorving-state.png

5 Propriedade de Markov

Por "estado" queremos dizer qualquer informação disponível para o agente. Assumimos que o estado é dado por algum sistema de preprocessamento que, normalmente, faz parte do ambiente.

O sinal do estado deve conter sensações imediatas, mas pode conter muito mais que isso. Exemplificando em uma situação da vida real, podemos mover nossos olhos sobre um cena de filme, com somente uma pequena fração da tela visível momentaneamente em qualquer tempo, mesmo assim, montamos uma representação rica e detalhada da cena. Num nível mundano, um sistema de controle pode medir a posição de um objeto em dois tempos distintos para obter um estado que representa a velocidade do objeto.

Por outro lado, o estado não deve informar ao agente tudo sobre o ambiente. Por exemplo, se o agente está jogando blackjack, nós não devemos esperar que ele saiba qual a próxima carta do deck. Há informação escondida no ambiente e essa informação seria útil para o agente, mas ele não tem como saber pois ainda não recebeu nenhuma sensação relevante para tal.

O caso ideal está entre as duas situações: o estado resume sensações passadas de forma compacta, mas de uma forma tal que retém todas informações essenciais. Isso normalmente requer mais que sensações imediatas mas nunca mais do que o histórico completo de sensações passadas. Um sinal de estado que sucede em reter todas as informações essenciais é dito Markoviano. A definição formal vem abaixo. Por exemplo, a configuração atual de um tabuleiro de xadrez resume tudo que é importante sobre a sequência de jogadas que levaram até lá. Muita informação é perdida, mas tudo que realmente importa para o futuro do jogo é retido. Aqui, supomos que o conjunto de estados e de recompensas são finitos, para que possamos trabalhar com somas e probabilidades, em vez de integrais e densidades de probabilidade.

A dinâmica de um ambiente pode ser determinada apenas se especificarmos a distribuição de probabilidades completa:

$$\Pr\{S_{t+1} = s', R_{t+1} = r | S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\}.$$

para todo r, s' e todos os possíveis valores do histórico: $S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t$. Se o sinal de estado possui a propriedade de Markov, a resposta do ambiente em $t+1$ depende somente das representação de estado e ação no tempo t, de tal forma que a dinâmica do ambiente pode ser definida especificando somente

$$p(s', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a\}.$$

para todo r, s', S_t e A_t. Ou seja, o sinal de estado possui a propriedade de Markov e é um estado de Markov se, e somente se, as duas expressões acimas são iguais para todo s', r, e para todos históricos.

A propriedade de Markov é importante no contexto de aprendizagem por reforço pois as decisões e os valores são funções somente do estado atual, nesse cenário. Para que eles sejam efetivos e informativos, a representação do estado deve ser informativo.

Além disso, é útil pensar nos estados em cada tempo como sendo uma aproximação de um estado de Markov, mesmo que eles são satisfaçam completamente a propriedade de Markov.

Exemplo de Poker: além do conhecimento das próprias cartas, o estado de um jogo de poker deve incluir as apostas e o número de cartas trocadas pelos outros jogadores. As apostas dos outros jogadores influnciam na sua opinião sobre a mão dos outros jogadores. Na verdade, todo o seu passado com os jogadores faz parte do estado de Markov. Mas isso, na prática, é muito difícil de lembrar e analizar, e terá pouco efeito claro sobre as decisões. Bons jogadores de Poker são bons em lembrar somente os fatores essenciais, mas não conseguem lembrar tudo que é relevante. Como resultado, as representações de estados usadas para tomar decisões de Poker são, inevitavelmente, não Markovianas. Mesmo assim, pessoas tomam boas decisões em tarefas desse tipo. Concluímos, então, que não ter acesso a estados *perfeitamente Markovianos* não é, provavelmente, um problema severo para um agente de aprendizagem por reforço.

6 MDPs

Uma tarefa de aprendizagem por reforço que satisfaz a propriedade de Markov é chamada de *Processo de decisão Markoviano*, ou *MDP*. Um MDP é definido pelos conjuntos de estados e ações e pela dinâmica sequêncial do ambiente. Dado um estado s e uma ação a, a probabilidade de cada possível par de próximo-estado-recompensa, (s', r), é dado por

$$p(s', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = a, A_t = a\}.$$

Essas quantidades especificam completamente a dinâmica de um MDP finito. Com esses valores, podemos calcular tudo sobre o ambiente. Por exemplo, as recompensas esperadas para um par estado-ação,

$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] = \sum_{r\in\mathcal{R}} r \sum_{s'\in\mathcal{S}} p(s', r|s, a).$$

As probabilidades de transição,

$$p(s'|s, a) = \Pr\{S_{t+1} = s'|S_t = s, A_t = a\} = \sum_{r\in\mathcal{R}} p(s', r|s, a).$$

E as recompensas esperadas para triplas estado-ação-próximo-estado,

$$r(s, a, s') = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s'] = \frac{\sum_{r\in\mathcal{R}} rp(s', r|s, a)}{p(s'|s, a)}.$$

Um jeito útil de resumir esses valores é através de um grafo de transições. Nesse grafos, há dois tipos de vértices: vértices de estados e vértices de ações. Há um vértice de estado para cada estado possível e um vértice de ação para cada par estado-ação. Começando em um estado s e tomando a ação a faz você se mover ao longo da linha a partir do vértice de estado s para o vértice de ação (s, a). Então, o ambiente responde com uma transição para algum vértice de estado através de um dos arcos que deixam o vértice (s, a). Cada arco corresponde a uma tripla (s, s', a) e rotulamos cada arco pela probabilidade de transição $p(s'|s, a)$ e a recompensa esperada $r(s, a, s')$. Abaixo temos um exemplo de um grafo de transições.

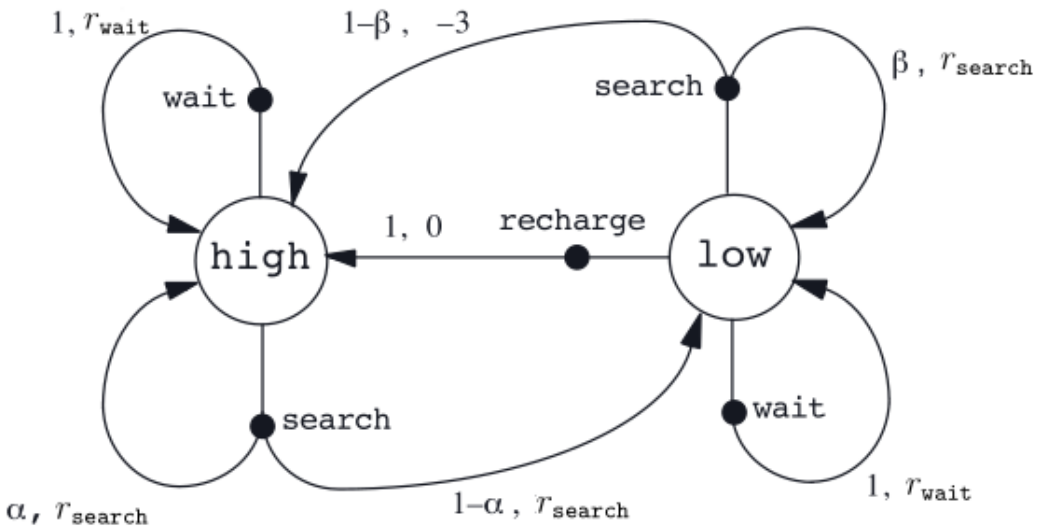

Figure 3: transition-graph.png

7 Funções Valor

Informalmente, funções valor são funções dos estados (ou de pares estado-ação) que estimam o quão bom é para o agente estar em um dado estado (ou o quão bom é realizar uma dada ação em um dado estado). A noção de o "quão

bom" é definida em termos de recompensas futuras (retorno esperado). Como as recompensas futuras dependem das ações que o agente toma, evidentemente, funções valor são definidas de acordo com uma política particular.

O *valor* de um estado s sob uma política π, denotado por $v_\pi(s)$ é o retorno esperado quando o agente começa no estado s e segue a política π após. Para MDPs, formalmente, temos,

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s].$$

onde $E_\pi[.]$ denota o valor esperado de uma variável aleatória dado que o agente segue a política π, e t é um tempo qualquer. A função v_π é chamada de função valor de estado para a política π.

Similarmente, definimos o valor de tomar a ação a no estado s sob a política π, denotado por $q_\pi(s, a)$, como sendo o retorno esperado quando o agente começa do estado s, toma a ação a e segue a política π após:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a].$$

q_π é chamada de função valor de ação para a política π.

As função valor de estado e de ação, v_π e q_π podem ser estimadas por experiência. Por exemplo, se um agente segue a política π e mantém uma média, para cada estado encontrado, dos retornos reais obtidos por aquele estado, então a média irá convergir para o valor do estado, $v_\pi(s)$, a medida que o número de vezes que o estado foi visitado se aproxima de infinito. Métodos de estimativa desse tipo são chamados de métodos de *Monte Carlo*. Se há muitos estados, manter médias separadas para cada estado pode ser impraticável. Em vez disso, o agente pode manter as funções v_π e q_π como funções parametrizadas e ajustar os parâmtros a medida que experiencia, para casar melhormente com os retornos observados.

Funções valor obedecem algumas relações recursivas. Para qualquer política π e estado s, a seguinte relação vale, que relaciona o valor do estado s com os valores de seus possíveis sucessores:

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\
&= \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2}|S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2}|S_{t+1} = s']] \\
\boxed{v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]}
\end{aligned}$$

Nas relações acima, está implícito que as ações são tomadas dentro do conjunto $\mathcal{A}(s)$, os estados dentro de $\mathcal{S}$, e recompensas de $\mathcal{R}$. A expressão final, a equação de *Bellman* para v_π, é facilmente lida como um valor esperado. Uma demonstração mais passo-a-passo é dada na sub-seção abaixo. É uma soma sobre todos os valores possíveis de três variáveis, a, s' e r. Para cada tripla (a, s', r), computamos sua probabilidade, $\pi(a|s)p(s', r|s, a)$, e multiplicamos pelo peso entre colchetes. A equação relaciona o valor de um estado com o valor dos estados sucessores. O diagrama abaixo ilustra o conceito. Cada círculo aberto representa um estado e cada círculo preenchido representa um par estado-ação. Depois da ação, o ambiente responde com um estado s' e uma recompensa r. A equação de Bellman faz a média de todas as possibilidades, com peso igual a probabilidade de uma resposta específica (estado s') acontecer. A equação essencialmente diz que o valor do estado inical é igual a soma do valor do próximo estado esperado (descontado) com a recompensa obtida no caminho para esse estado. O diagrama abaixo é chamado de um diagrama *backup*. A função valor v_π é a solução única para sua equação de Bellman. Essa equação forma uma base para várias formas de computar, aproximar e aprender v_π. Observe a distinção entre diagramas *backup* e grafos de transição. Os vértices de estado nesses diagramas podem aparecer mais de uma vez. Omitimos arcos dirigidos nesses diagramas pois está implícito que a direção do tempo é de cima para baixo.

Exemplo da Grade: Considere a grade retangular abaixo, onde cada célula da grade representa um estado. Para cada célula, há quatro ações possíveis: **norte**, **sul**, **leste** e **oeste**, que faz, deterministicamente, o agente mover para a célula correspondente à direção. Ações que fariam o agente se mover para fora da grade deixa a posição do agente inalterada, e geram um sinal de recompensa -1. Outras ações geram sinal 0, exceto para movimentos a partir dos

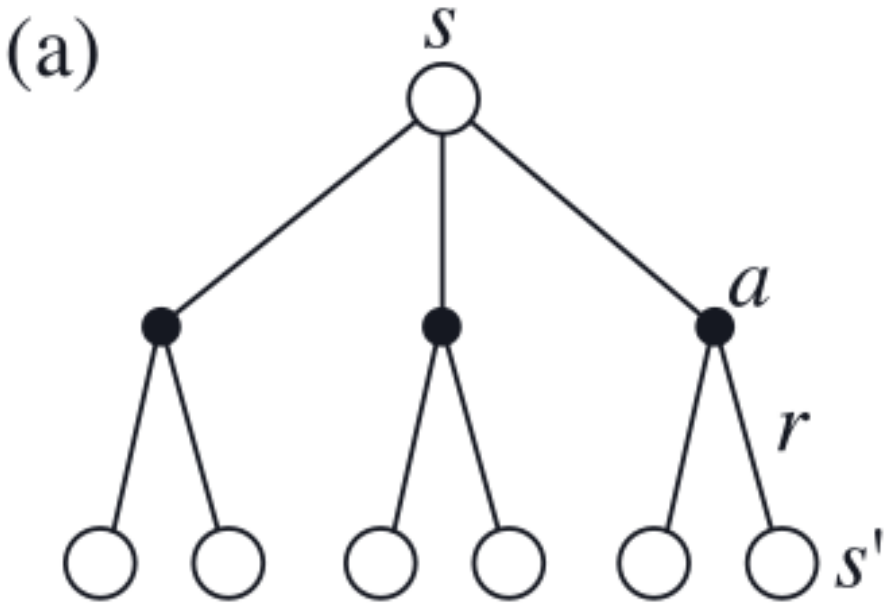

Figure 4: backup-diagram1.png

estados especiais A a B : a partir de A, todas as quatro ações geram sinal $+10$ e levam o agente para A'. A partir de B, todas as ações geram $+5$, e levam o agente para B'.

A política do agente seleciona as quatro ações com igual probabilidade, isto é $\pi(\textbf{norte}|s) = \pi(\textbf{sul}|s) = \pi(\textbf{leste}|s) = \pi(\textbf{oeste}|s) = \frac{1}{4}$, para todo estado s. A figura abaixo mostra a função valor, v_π, para essa política, com $\gamma = 0.9$. A função foi computada resolvendo a equação de Bellman para v_π.

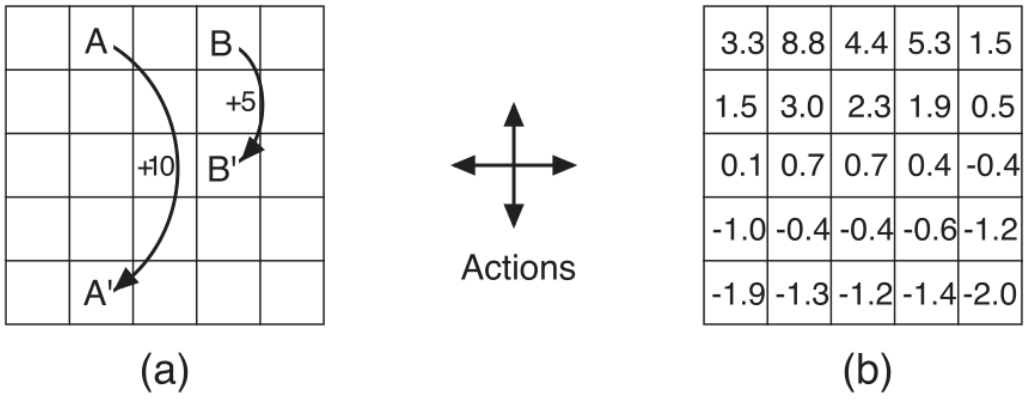

Figure 5: grid.png

Algumas observações práticas: os valores dos estados perto das quinas inferiores são negativas pois a partir desses estados é provável que o agente esbarre na fronteira sob a política aleatória. O estado A é o melhor para se estar, mas observe que seu valor é menor que $+10$. De fato, quando o agente transita de A para A', é provável que ele esbarre em uma fronteira.

7.1 Demonstração da Equação de Bellman para v_π

Nessa seção faço um passo-a-passo da demonstração da equação de Bellman para v_π.

Partimos da definição de $v_\pi(s)$, o valor de um estado.

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1}|S_t = s] + \gamma \mathbb{E}_\pi[G_{t+1}|S_t = s] \end{aligned}.$$

Podemos calcular o primeiro termo da seguinte forma:

$$\mathbb{E}_\pi[R_{t+1}|S_t = s] = \sum_{r\in\mathcal{R}} \Pr\{R_{t+1} = r|S_t = s\}.$$

Temos que,

$$\Pr\{R_{t+1} = r|S_t = s\} = \sum_{a\in\mathcal{A}(s)} \Pr\{R_{t+1} = r|S_t = s, A_t = a\} \Pr\{A_t = a|S_t = s\}.$$

A ação que o agente toma é função somente do estado atual: $\Pr\{A_t = a|S_t = s\} = \pi(a|s)$. Então,

$$\Pr\{R_{t+1} = r|S_t = s\} = \sum_{a\in\mathcal{A}(s)} \Pr\{R_{t+1} = r|S_t = s, A_t = a\}\pi(a|s).$$

Por outro lado, podemos utilizar as funções probabilidades conjuntas da dinâmica do ambiente para escrever,

$$\Pr\{R_{t+1}|S_t = s, A_t = a\} = \sum_{s'\in\mathcal{S}} p(s', r|s, a).$$

Portanto,

$$\mathbb{E}_\pi[R_{t+1}|S_t = s] = \sum_{r\in\mathcal{R}} r \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s', r|s, a),$$

que pode ser escrita de forma mais compacta:

$$\mathbb{E}_\pi[R_{t+1}|S_t = s] = \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s',r} p(s', r|s, a)r$$

.

Agora, calculamos o segundo termo de $v_\pi(s)$, $\gamma\mathbb{E}_\pi[G_{t+1}|S_t = s]$.

$$\mathbb{E}_\pi[G_{t+1}|S_t = s] = \sum_g g \Pr\{G_{t+1} = g|S_t = s\} = \sum_g g \sum_{s'\in\mathcal{S}} \Pr\{G_{t+1} = g|S_{t+1} = s'\} \Pr\{S_{t+1} = s'|S_t = s\}.$$

Além disso,

$$\Pr\{S_{t+1} = s'|S_t = s\} = \sum_{a\in\mathcal{A}(s)} \Pr\{S_{t+1} = s'|S_t = s, A_t = a\} \Pr\{A_t = a|S_t = s\} = \sum_{a\in\mathcal{A}(s)} \pi(a|s) \Pr\{S_{t+1} = s|S_t = s, A_t = a\},$$

com,

$$\Pr\{S_{t+1} = s'|S_t = s, A_t = a\} = \sum_{r\in\mathcal{R}} \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\} = \sum_{rin\mathcal{R}} p(s', r|s, a).$$

Então,

$$\Pr\{S_{t+1} = s'|S_t = s, A_t = a\} = \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{rin\mathcal{R}} p(s', r|s, a)$$

.

Finalmente,

$$\mathbb{E}_\pi[G_{t+1}|S_t = s] = \sum_g g \sum_{s' \in \mathcal{S}} \Pr\{G_{t+1} = g | S_{t+1} = s'\} \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{r \in \mathcal{R}} p(s', r|s, a).$$

Podemos trocar a ordem das somas:

$$\mathbb{E}_\pi[G_{t+1}|S_t = s] = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s', r} p(s', r|s, a) \sum_g g \Pr\{G_{t+1} = g | S_{t+1} = s'\}.$$

Mas $\sum_g g \Pr\{G_{t+1} = g|S_{t+1} = s'\} = \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s'] = v_\pi(s')$

□

8 Função valor ótima

A grosso modo, resolver uma tarefa de aprendizagem por reforço é achar uma política π que faz o agente atingir muita recompensa ao longo do processo completo. Para MDPs finitos, podemos definir precisamente o conceito de política ótima.

Podemos definir uma ideia de ordem parcial para políticas. Uma política π é melhor que outra π' se o seu retorno esperado é maior ou igual daquele de π' para todos os estados s. Ou seja, $\pi \geq \pi' \leftrightarrow v_\pi(s) \geq v_{\pi'}(s)\ \forall s \in S$. Há pelo menos uma política que é melhor ou igual a todas outras políticas. Denotamos todas políticas ótimas por $\pi_\star$. Elas compartilham da mesma função valor de estado, chamada de *função valor de estado ótima*, denotada por $v_\star$, definida por

$$v_\star(s) = \max_\pi v_\pi(s),$$

para todo estado $s \in s$. A equação acima é verdadeira mediante a definição de política ótima. Políticas ótimas também compartilham da mesma *função valor de estado-ação ótima*, denotada por $q_\star$ e definida por

$$q_\star(s, a) = \max_\pi q_\pi(s, a), .$$

para todo estado $s \in \mathcal{S}$ e $a \in \mathcal{A}(s)$. Para cada par (s, a), essa função fornece o retorno esperado ao tomar a ação a no estado s e seguir uma política ótima após. Portanto, podemos escrever $q_\star$ em termos de $v_\star$:

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

Evidentemente, por ser uma função valor para uma política, $v_\star$ satisfaz a equação de Bellman. Por ser o caso especial para a política ótima, podemos rescrever a equação de Bellman para esse caso sem fazer menção à nenhuma política:

$$\begin{aligned}
v_\star(s) &= \max_{a \in A(s)} q_{\pi_\star}(s, a) \\
&= \max_a \mathbb{E}_{\pi_\star}[G_t|S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_\star}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_\star}[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2}|S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_\star(S_{t+1})|S_t = s, A_t = a] \\
\boxed{v_\star(s) = \max_{a \in S} \sum_{s', r} p(s', r|s, a)[r + \gamma v_\star(s')]}&.
\end{aligned}$$

Essa equação é chamada de *equação de Bellman ótima*. Intuitivamnte, essa equação expressa o fato de que o valor de um estado sob uma política ótima deve ser igual ao retorno esperado da melhor ação a partir daquele estado.

A equação de Bellman ótima para $q_\star$ é

$$\begin{aligned}
q_\star(s, a) &= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_\star(S_{t+1}, a')|S_t = s, A_t = a] \\
\boxed{q_\star(s, a) = \sum_{s', r} p(s', r|s, a)[r + \gamma \max_{a'} q_\star(s', a')]}&.
\end{aligned}$$

O diagram *backup* abaixo mostra graficamente os *spans* de estados e ações futuras consideradas nas suas equações de Bellman ótima. São semelhantes ao diagrama visto anteriormente, com a diferença do arco na escolha do agente: ele representa o fato de que o máximo sobre essas escolhas é tomado, em vez do valor esperado dada alguma política.

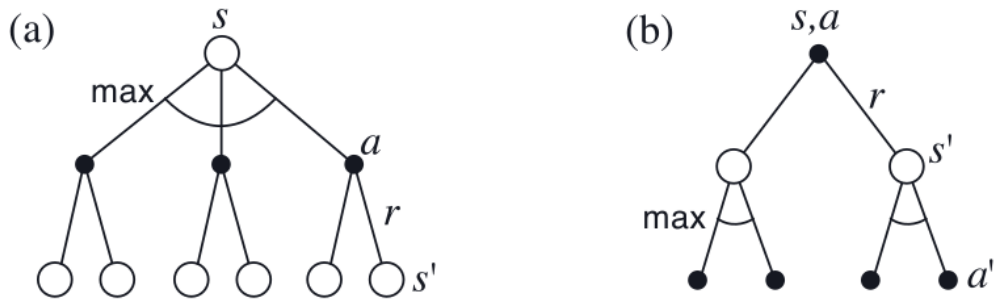

Figure 6: backup-diagram-max.png

Para MDPs finitos, a equação de Bellman ótima tem uma solução única independente da política. Se as dinâmicas do ambiente são conhecidas, isto é, se temos acesso a $p(s', r|s, a)$, em princípio, podemos resolver o sistema de equações não linear de Bellman para $v_\star$ e $q_\star$. Determinado $v_\star$ é fácil encontrar uma política ótima. Para cada estado s, haverá uma ou mais ações nas quais o máximo é obtido na equação de Bellman ótima. Qualquer política que atribui probabilidades não nulas somente a essas ações é uma política ótima. Podemos pensar nisso como uma busca de um passo. Se você tem a função valor ótima $v_\star$, então as ações que parecem melhores após uma busca de um passo serão ações ótimas. Essa é uma estratégia gulosa que funciona, pois $v_\star$ já leva em conta as consequências relacionadas a recompensas de todo comportamento futuro possível.

Para $q_\star$ é ainda mais fácil: para qualquer estado s, deve-se simplemente achar qualquer ação que maximize $q_\star(s, a)$. A função valor estado-ação efetivamente engloba os resultados de todas as buscas de um passo. Com essa função, podemos escolher ações ótimas sem olhar para ações futuras. Então, conhecendo $q_\star$, não precisamos saber nada sobre a dinâmica do ambiente.

No exemplo da grade, podemos calcular $v_\star$ e determinar uma política ótima:

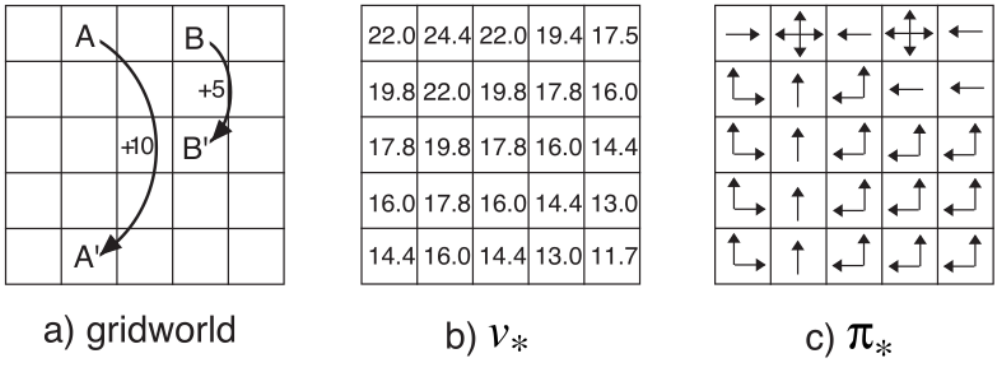

Figure 7: grid-solution.png

9 Aproximação

Claramente, um agente que aprende uma política ótima realizou sua tarefa com sucesso. Na prática, isso raramente acontece. Para as tarefas usuais que estamos interessados, uma política ótima é obtida com um extremo custo computacional. Mesmo que tenhamos uma descrição completa do ambiente, usualmente é inviável computar uma política ótima resolvendo a equação de Bellman ótima. Por exemplo, o jogo de xadrez é apenas uma fração pequena da experiência humana, mesmo assim, computadores poderosos não conseguem computar ações ótimas. A memória disponível também é outro fator limitante nesse contexto. Uma grande quantidade de memória geralmente é necessária para armazenar estimativas/aproximações de funções valor, políticas e modelos. Em tarefas com pequenos conjuntos de estados e ações é possível formar essas aproximações usando tabelas. Métodos que usam tabelas desse

tipo são chamados de métodos *tabulares*. Entretanto, em muitos casos de interesse, esses métodos são impraticáveis e é necessário o uso de funções com algum tipo de parametrização para fazer aproximações.

Em alguns casos, há muitos estados com baixa probabilidade de acontecerem e selecionar ações sub-ótimas para esses estados tem baixo impacto no retorno total. A natureza *on-line* da aprendizagem por reforço faz ser possível aproximar políticas ótimas de forma que mais esforço é colocado em estados mais frequentes, no custo de menos esforço em estados improváveis. Essa é uma propriedade chave que distingue aprendizagem por reforço de outros métodos para resolver aproximadamente MDPs.

10 Resultados Importantes

Retorno esperado com desconto:

$$\boxed{G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}}.$$

Propriedade de Markov

$$\boxed{\Pr\{S_{t+1} = s', R_{t+1} = r | S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\} = \Pr\{S_{t+1} = s', R_{t+1} = r | S_t, A_t\}}.$$

Dinâmica do ambiente:

As recompensas esperadas para um par estado-ação,

$$\boxed{r(s,a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r | s, a)}.$$

As probabilidades de transição,

$$\boxed{p(s'|s,a) = \Pr\{S_{t+1} = s' | S_t = s, A_t = a\} = \sum_{r \in \mathcal{R}} p(s', r | s, a)}.$$

E as recompensas esperadas para triplas estado-ação-próximo-estado,

$$\boxed{r(s,a,s') = \mathbb{E}[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s'] = \frac{\sum_{r \in \mathcal{R}} r p(s', r | s, a)}{p(s'|s,a)}}.$$

Função valor:

Função valor de estado:

$$\boxed{v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s]}.$$

Função valor de estado-ação:

$$\boxed{q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a]}.$$

Equação de Bellman:

$$\boxed{v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r | s, a)[r + \gamma v_\pi(s')]}.$$

$$\boxed{q_\pi(s,a) = \sum_{s',r} p(s', r | s, a)[r + \gamma \sum_{a' \in \mathcal{A}(s')} \pi(a'|s') q_\pi(s', a')]}.$$

Equação de Bellman ótima:

$$\boxed{v_\star(s) = \max_{a \in S} \sum_{s',r} p(s', r|s, a)[r + \gamma v_\star(s')]}.$$

$$\boxed{q_\star(s, a) = \sum_{s',r} p(s', r|s, a)[r + \gamma \max_{a'} q_\star(s', a')]}.$$