RL INSANE

Esse resumo é e será completamente baseado no livro “Reinforcement Learning An Introduction - 2ed” de Richard S. Sutton e Andrew G. Barto e das aulas e do github do professor Flávio Codeço Coelho.

Se alguém for ler, considere que eu estou aprendendo a matéria, e não a cursei totalmente ainda, provavelmente terão erros.

1 - Introdução

1.1 - Reinforcement Learning

Em vez de ser teórico, o livro gostaria de começar ensinando diretamente como simular o aprendizado computacionalmmente, sem ficar dando exemplos da vida real, como o aprendizado de bebês, etc. “O agente não terá a certeza de qual ação fazer, mas descobrirá qual a ação trará a maior recompensa por tentá-la. Nos desafios mais interessantes, as ações não irão afetar apenas o momento atual, mas sim todas as ações subsequentes”.

O problema de aprendizado por reforço será formalizado usando ideias da Teoria de Sistemas Dinâmicos, especificamente, controle ótimo do processo de decisão incompleto de Markov.

“Um agente de aprendizagem deve ser capaz de sentir o estado de seu ambiente até certo ponto e deve ser capaz de tomar ações que afetem o estado. O agente também deve ter um ou mais objetivos relacionados ao estado do ambiente. Os processos de decisão markovianos visam incluir apenas esses três aspectos — sensação, ação e objetivo — em suas formas mais simples possíveis, sem banalizar nenhum deles. “

Vale lembrar que Aprendizado por Reforço é diferente do Aprendizado Supervisionado, já que o supervisionado é um aprendizado feito com um conjunto de dados já rotulados dado por um supervisor externo (basicamente o que foi feito em Técnicas e Algoritmos para Ciência de Dados, com o Pacannaro). Além disso, Aprendizado por reforço também não é o mesmo que Aprendizado Não Supervisionado, já que essa busca encontrar estruturas escondidas em dados não rotulados.

Aprendizado por reforço é considerado como o terceiro paradigma do Aprendizado de Máquina, ao lado do Aprendizado Supervisionado e Não Supervisionado, e outros paradigmas.

Uma característica específica do Aprendizado por reforço, é o trade-off da exploração de novas possibilidades e da exploração do que já se sabe. “Para obter uma grande recompensa, um agente de aprendizagem por reforço deve preferir ações que já tentou no passado e que se mostraram eficazes na produção de recompensa. Mas, para descobrir tais ações, ele precisa tentar ações que não selecionou antes. O agente precisa explorar o que já experimentou para obter recompensa, mas também precisa explorar para fazer melhores seleções de ações no futuro.”

“Outra característica fundamental do aprendizado por reforço é que ele considera explicitamente todo o problema de um agente orientado a um objetivo interagindo com um ambiente incerto. Isso contrasta com muitas abordagens que consideram subproblemas

sem abordar como eles podem se encaixar em um cenário mais amplo. Por exemplo, mencionamos que grande parte da pesquisa em aprendizado de máquina se concentra no aprendizado supervisionado sem especificar explicitamente como tal habilidade seria útil."

O aprendizado por reforço começa com um agente completo, interativo e objetivo. Todos os agentens tem metas explícitas, podem sentir os aspectos do seu ambiente e conseguem escolher suas ações para influenciar seus resultados.

1.2 Exemplos

1. Um mestre em xadrez faz um movimento. A escolha é baseada tanto no planejamento — antecipando possíveis respostas e contra-respostas — quanto em julgamentos imediatos e intuitivos sobre a conveniência de determinadas posições e movimentos.
2. Um controlador adaptativo ajusta os parâmetros da operação de uma refinaria de petróleo emtempo real. O controlador otimiza a relação rendimento/custo/qualidade com base nos custos marginais especificados, sem se ater estritamente aos pontos de ajuste originalmente sugeridos pelos engenheiros.
3. Uma gazela luta para ficar de pé após acabar de nascer. 20 minutos depois, ela consegue correr a 20 milhas por hora.
4. Phil prepara seu café da manhã. Abrir o armário, selecionar o cereal, pegá-lo, fechar o armário, pegar uma tigela, encontrar o leite, misturar... Uma série de julgamentos são feitos, ele deve se locomover, acessar informações do seu corpo, como força, preferência de alimentação, nível de fome, etc.

Todos esses exemplos envolvem interação entre um agente tomador de decisão e seu ambiente, com o agente buscando atingir uma meta mesmo sem saber tudo sobre seu ambiente.

1.3 - Elementos do Aprendizado de Máquina

Por trás do agente e do ambiente, podemos identificar quatro subelementos de um sistema de Aprendizado por reforço:

1. A **política** define o aprendizado do agente e o comportamento que ele deverá ter em dado momento. Grossamente falando, a política é um mapeamento de estados percebidos do ambiente para as ações que podem ser tomadas quando nesse estado. Em geral, pode ser estocástica, determinando probabilidades de cada ação.

2. Um sinal de **recompensa** define a meta do problema de aprendizado. A cada passo, o ambiente envia ao agente um número chamado de recompensa. O objetivo do agente é maximizar essa recompensa por todo o caminho que ele percorre. Logo, o sinal de recompensa mostra o que são eventos ruins e bons para o agente.

    - O sinal de recompensa é a base principal para alterar a política; se uma ação selecionada pela política for seguida de uma recompensa baixa, a política pode ser alterada para selecionar alguma outra ação naquela situação no futuro. Em geral, os sinais de recompensa podem ser funções estocásticas do estado do ambiente e das ações tomadas.

3. Enquanto a recompensa indica o que é bom no senso imediato do agente, a **função de valor** mostra o que é bom no longo prazo. Por exemplo, um estado pode dar uma recompensa baixa de imediato, mas ter um alto valor pois os estados que se seguem trazem recompensas maiores.
   - Podemos dizer que a recompensa são de senso primário, enquanto valores são secundários. No entanto, são os valores que mais nos preocupam ao tomar e avaliar decisões. As escolhas de ação são feitas com base em julgamentos de valor. Por isso, buscamos ações que gerem estados de maior valor, não de maior recompensa, porque essas ações nos proporcionam a maior recompensa a longo prazo.
4. O elemento final é o **modelo do ambiente**. Isso é algo que imita o comportamento do ambiente ou, de forma mais geral, que permite fazer inferências sobre como o ambiente se comportará. Por exemplo, dado um estado e uma ação, o modelo do ambiente deveria prever o próximo estado e a próxima recompensa. Os métodos para resolver problemas de aprendizagem por reforço que utilizam modelos e planejamento são chamados de métodos baseados em modelos, e são o contrário de métodos mais simples sem modelos que são explicitamente aprendizes de tentativa e erro.

1.4 - Limitações e escopo

O estado é um conceito central em aprendizado por reforço, servindo como entrada para a política, a função de valor e o modelo. Informalmente, pode ser entendido como o sinal que descreve “como o ambiente está” em um dado momento.

Embora muitos métodos de aprendizado por reforço se baseiem em funções de valor, isso não é obrigatório. Métodos evolutivos (como algoritmos genéticos e programação genética) não usam funções de valor: eles testam várias políticas em paralelo, selecionam as mais recompensadas e geram novas políticas a partir delas, de forma análoga à evolução biológica. Esses métodos podem ser eficazes em certos contextos (por exemplo, quando o espaço de políticas é pequeno ou quando há bastante tempo disponível), e podem lidar bem com situações em que o agente não percebe o estado completo do ambiente.

1.5 - Tic-Tac-Toe (jogo da velha)

Vamos considerar que empates e perdas são igualmente ruins. Como construir um jogador que encontre as falhas nas jogadas de um oponente(considerando um que consiga perder), que aprenda a maximizar as chances de vitória? Basicamente, esse jogo pode ser resolvido de outra forma, mas não de forma satisfatória sem o Aprendizado por Reforço. Enfim, vamos resolver usando um método de função de valor.

Primeiro fazemos uma tabela de números, um para cada estado possível do jogo. O estado A será escolhido se for maior que o estado B, e, assumindo que sempre jogaremos com os X’s, se tivermos 3 X’s em uma diagonal, linha ou coluna, então nossa probabilidade de ganhar é 1. No caso inverso, onde tem 3 bolas nessas posições, então a probabilidade de vencer é 0. Definimos o restante dos estados, inicialmente valendo 0.5, representando 50% de chance.

Continuamos jogando várias partidas contra o oponente, e para, selecionarmos os estados que iremos escolher, normalmente nos moveremos gananciosamente (greedy), buscando

o maior valor. Por vezes, mudaremos isso e escolheremos um valor aleatório, significando movimentos exploratórios, com o objetivo de experienciar outros estados.

Durante o jogo, os valores dos estados são atualizados continuamente para refletirem melhor a probabilidade de vitória. Isso é feito por meio de um processo de backup: o valor de um estado anterior $V(S_t)$ é ajustado para ficar mais próximo do valor do estado seguinte $V(S_{t+1})$

$$V(S_t) \leftarrow V(S_t) + \alpha[V(S_{t+1}) - V(S_t)] \tag{1}$$

onde α é uma pequena fração positiva chamada de parâmetro de taxa de passo(step-size), que influencia a velocidade do aprendizado. Essa regra de atualização é um exemplo de um método de aprendizado por diferença temporal (temporal-difference learning), assim chamado porque suas mudanças se baseiam em uma diferença, $V(S_{t+1}) - V(S_t)$, entre estimativas em dois instantes sucessivos.

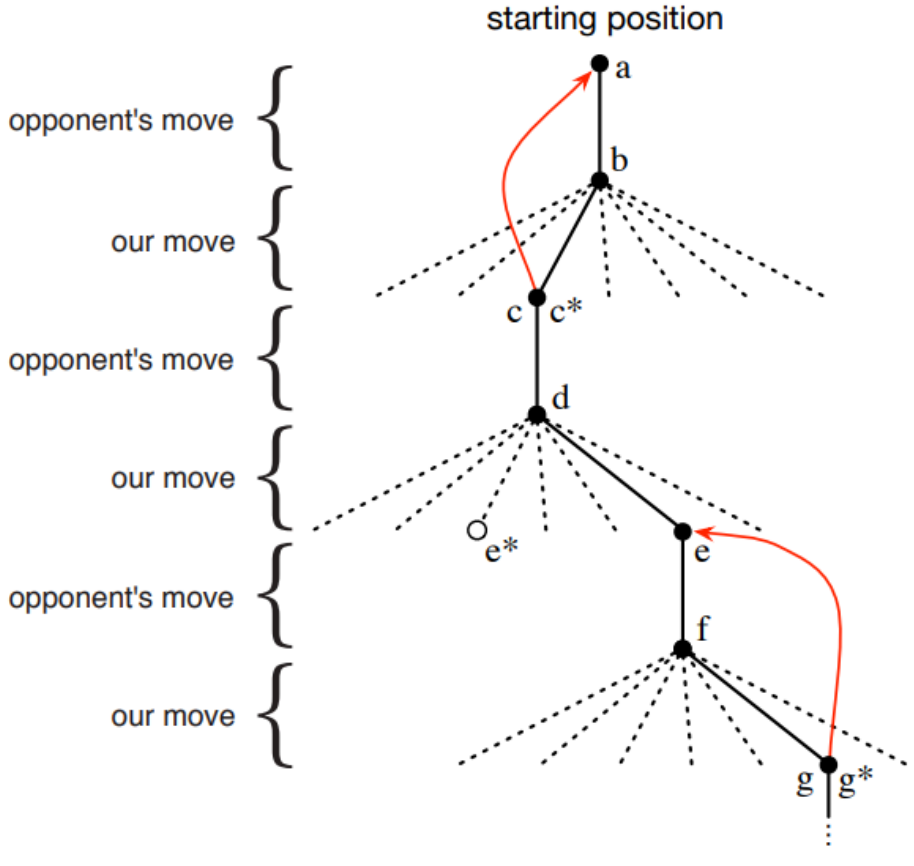

Figura 1: Uma sequência de movimentos no tic-tac-toe. As linhas pretas contínuas representam os movimentos feitos durante uma partida; as linhas tracejadas representam movimentos que nós (o agente) consideramos, mas não realizamos. Nosso segundo movimento foi um movimento exploratório, ou seja, foi realizado mesmo que outro movimento alternativo, aquele que leva a e^* tivesse classificação mais alta. Movimentos exploratórios não resultam em aprendizado, mas cada um dos outros movimentos resulta, causando atualizações conforme sugerido pelas setas vermelhas, nas quais os valores estimados são propagados para cima na árvore, de nós posteriores para nós anteriores.

- talvez um código vai ser colocado em algum lugar simulando isso??

Esse exemplo mostra a diferença entre métodos evolutivos e métodos função de valor(no caso, a solução desse problema aborda o método de função de valor, pois nós atualizamos o valor do estado a cada jogada com base na diferença temporal), onde para evoluir a política usando o método evolutivo nós usamos a mesma política e jogamos diversas vezes contra o oponente, analisando apenas a saída final (vitória/derrota). No fim, ambos os métodos buscam o espaço das políticas, mas aprender a função de valor faz com que se ganhe vantagem das informações disponíveis durante cada jogo (aparentemente, o método evolutivo não faz parte do aprendizado por reforço).

Vale lembrar que o Aprendizado por reforço é mais geral que apenas esse exemplo e que abrange vários problemas maiores com características diferentes, como os que não tem um adversário, desafios sem uma separação exata(sem começo e fim), com recompensas a qualquer momento e que também pode ser aplicado a um cenário contínuo.

“O quão bem um sistema de aprendizado por reforço pode funcionar em problemas com conjuntos de estados tão grandes está intimamente ligado à sua capacidade de generalizar adequadamente a partir de experiências passadas. É nesse papel que temos a maior necessidade de métodos de aprendizado supervisionado com aprendizado por reforço.”

Por fim, existem métodos que não precisam de nenhum tipo de modelo de ambiente em geral. O jogo da velha por si próprio é model-free no contexto de jogador, já que não há nenhum tipo de modelo para o oponente do agente. “Como os modelos precisam ser razoavelmente precisos para serem úteis, os métodos sem modelo podem ter vantagens sobre métodos mais complexos quando o verdadeiro gargalo na resolução de um problema é a dificuldade de construir um modelo de ambiente suficientemente preciso.”

1.6 - Sumário

O Aprendizado por reforço é uma abordagem computacional para entender e automatizar o aprendizado e a tomada de decisão orientados a objetivos, diferenciando-se por enfatizar o aprendizado de um agente a partir da interação direta com o ambiente, sem supervisão explícita nem modelos completos. Baseia-se no formalismo de Processos de Decisão de Markov, que descrevem a interação em termos de estados, ações e recompensas, capturando causa e efeito, incerteza e a presença de metas. Um conceito central é a função de valor, fundamental para guiar a busca eficiente no espaço de políticas, sendo justamente o uso dessas funções o que distingue os métodos de aprendizado por reforço dos métodos evolutivos, que avaliam apenas políticas inteiras.

1.7 Pequena história do Aprendizado por Reforço

Desculpe, mas eu sei tão pouco sobre isso que resumir seria um pecado, pois não sei nada de história.

Part I - Soluções de métodos tabulares

Nessa parte do livro iremos falar sobre a resolução de problemas que podem ser representados por arrays e tabelas(por isso o nome métodos tabulares). Ou seja, o espaço das ações e os estados são pequenos o suficiente para serem armazenados, cada espaço possível. Nesse caso, os métodos podem achar o valor exato das soluções, ou seja, o valor exato do valor ótimo da função e a política exata.

O primeiro capítulo dessa parte vai falar sobre problemas tabulares onde há apenas um único estado, chamado de bandit problems(ou bandidos de um braço só, ou multi-armed bandits). O segundo capítulo descreve problemas mais gerais onde iremos falar sobre processos de Markov finitos e suas principais ideias, etc.

2 - Bandidos de muitos braços (Multi-armed bandits)

A característica mais importante do Aprendizado por Reforço que a difere de outros tipos de aprendizados é que ela utiliza informações de treino que avalia as ações já tomadas, ou seja, enquanto o Aprendizado Supervisionado dá um feedback instrutivo, isto é, o feedback não depende da ação tomada, enquanto no Aprendizado por Reforço, o feedback é instrutivo, ou seja, o feedback depende interiramente da ação tomada.

Nesse capítulo vamos ver o aspecto do feedback avaliativo simplificado, que não envolve aprender a agir em mais de uma situação, ou seja, teremos sempre apenas um estado, e depois generalizaremos.

2.1 - O problema do bandido k-armado

Considere o seguinte problema: você seguidamente tem que escolher entre k opções, ou ações, e depois de cada escolha você recebe uma recompensa de uma distribuição de probabilidade estacionária que depende da sua escolha.

Nota: o jogo da velha, explicado no capítulo passado, não é um problema que se enquadra como bandido k-armado, já que cada estado depende dos anteriores, e cada jogada muda o estado do tabuleiro. Como exemplo simples de um problema do bandido 1-armado, pense apenas num caça níquel, onde só existe uma ação(puxar o braço), as recompensas são sempre diferentes, e só existe um estado possível.

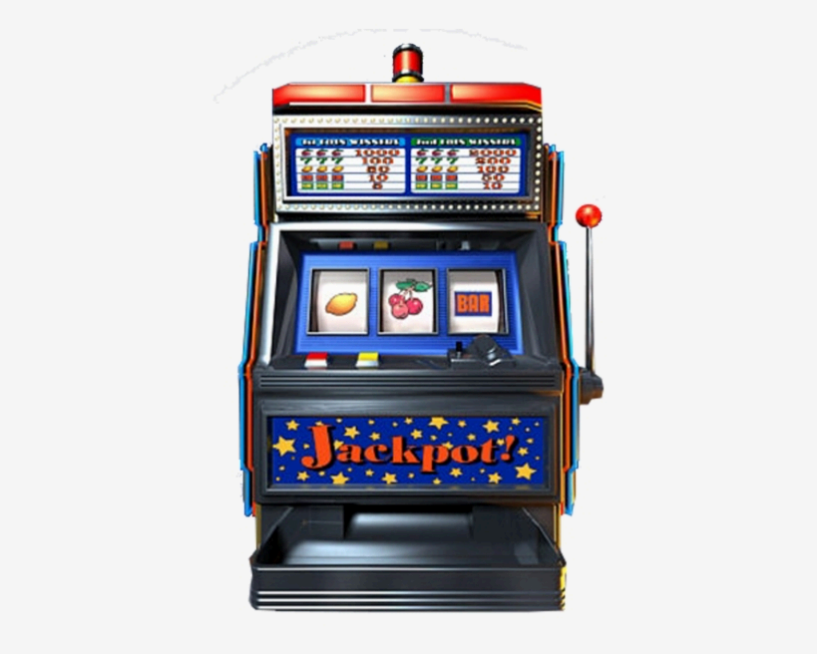

Figura 2: Caça-níquel

Nesse problema do bandido k-armado, cada uma das k-ações tem uma média esperada dada a ação selecionada(vamos chamar isso de valor da função). Nós iremos denotar a ação selecionada no tempo t como A_t, e a sua respectiva recompensa como R_t. Então, o

valor de uma ação arbitrária a, denotado $q_*(a)$, é o valor esperado da recompensa dado que a foi escolhido:

$$q_*(a) \doteq \mathbb{E}[R_t \mid A_t = a] \tag{2}$$

Se soubermos o valor de cada ação, então seria trivial para resolver o problema do bandido k-armado: basta selecionar a ação com maior recompensa. Porém, em geral, não sabemos o valor exato da ação, embora podemos ter estimadores. Denotamos o valor estimado do valor de uma ação a no tempo t como $Q_t(a)$. Nós claramente gostaríamos que $Q_t(a)$ fosse próximo de $q_*(a)$.

Se mantivermos estimativas dos valores das ações, então, em qualquer tempo t, existe pelo menos uma ação cujo valor estimado é o maior. Por isso, chamamos essas de ações gananciosas (greedy actions). Quando selecionamos uma dessas ações, dizemos que está explorando(exploiting) o conhecimento atual dos valores das ações.

Se, em vez disso, selecionarmos uma das ações não gananciosas, então dizemos que estamos explorando(exploring), porque permite melhorar sua estimativa do valor dessa ação não gananciosa. A exploração (exploitation) é a escolha correta para maximizar a recompensa esperada em um único passo, mas a exploração (exploration) pode gerar uma recompensa total maior a longo prazo.

Por exemplo, suponha que o valor de uma ação gananciosa seja conhecido com certeza, enquanto várias outras ações são estimadas como quase tão boas, mas com bastante incerteza. A incerteza é tal que pelo menos uma dessas outras ações provavelmente é, na verdade, melhor que a gananciosa, mas o agente não sabe qual, pois não explorou ainda.

Se o agente ainda tiver muitos passos futuros para escolher ações, pode ser melhor explorar as ações não gananciosas e descobrir quais delas são melhores que a gananciosa. A recompensa será menor no curto prazo, durante a exploração, mas maior no longo prazo porque, depois de descobrir as melhores ações, você poderá explorá-las repetidamente. Como não é possível explorar e explorar ao mesmo tempo em uma única escolha de ação, fala-se frequentemente no “conflito entre exploração(exploitation) e exploração(exploration)”.

O livro enfatiza que esse problema de balanceamento entre exploitation e exploration é recorrente, já que não podemos escolher duas ações diferentes ao mesmo tempo. Em geral, existem métodos especificos para rebalancear isso, mas normalmente são necessários fortes afirmações sobre conhecimentos do modelo que são impossíveis de verificar em aplicações completas de Aprendizado por Reforço.

2.2 - Métodos baseados em valores de ações