Reinforcement Learning

Nicolas Maia
linkedin/NicolasSMaia

Presença

- Linktree: Presente na bio do nosso instagram
- Presença ficará disponível até 1 hora antes da próxima aula
- É necessário 70% de presença para obter o certificado

Presença

Recompensa Média

Formas de Retorno

- Definição básica: soma das recompensas
- Reflete o quão bem o nosso agente foi ou pode ir

Formas de Retorno

- Forma Episódica

$$G_t = \sum_{i=0}^{n} R_{t+i}$$

Formas de Retorno

- Forma Descontada

$$G_t = \sum_{i=0}^{n} \gamma^i R_{t+i}$$

Recompensa Média

- Nova forma de definir o Retorno
- Fornece uma métrica clara da qualidade da política atual
- Pode ser aplicada tanto no caso contínuo quanto no episódico
- Não é preciso ficar escolhendo valores de gamma, o que também implica que não ficamos impactando quais ações damos preferência: curto ou longo prazo

Recompensa Média

- Fórmula:

$$G_t = R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + R_{t+3} - r(\pi) + \ldots$$

Recompensa Média

- Fórmula:

$$G_t = R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + R_{t+1} - r(\pi) + ...$$

Onde $r(\pi)$ é a recompensa média daquela política

Semi-Gradient Sarsa

Algoritmo

Entrada: Uma função diferenciável $\hat{q}$ para aproximar q_π

Parâmetros de incremento $\alpha, \beta > 0$

Vetor de pesos w, inicializado arbitrariamente

Valor inicial para a aproximação da Recompensa Média de π: $\bar{R}$

Inicializa em um estado S com uma ação A

repeat

    Executa A e observa R e S'

    Escolhe A' de acordo com $\hat{q}(S', \cdot, w)$ (pode ser ϵ-greedy)

    $\delta \leftarrow (R - \bar{R}) - (\hat{q}(S, A, \mathbf{w}) - \hat{q}(S', A', \mathbf{w}))$

    $\bar{R} \leftarrow \bar{R} + \beta\delta$

    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\nabla\hat{q}(S, A, \mathbf{w})$

    $S \leftarrow S'$

    $A \leftarrow A'$

until convergência =0

Pontos a Destacar

- Delta é o erro TD que mede o quão diferente a realidade foi da nossa expectativa
- O algorítmo corrige nossa estimativa na direção da surpresa: se a surpresa foi positiva, aumentamos, se foi negativa, diminuímos

Policy Approximation

Definição

- Até então, trabalhamos com *action-value methods* que escolhiam ações baseando-se no valor estimado delas
- Agora, vamos trabalhar com políticas parametrizadas que escolhem ações sem consultar uma função valor, podendo ainda a função valor ser usada para atualizar os parâmetros da nossa política

$$\pi(a|s,\boldsymbol{\theta}) = \Pr\{A_t = a \mid S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}$$

- Nos baseamos no gradiente de uma medida de performance J(θ) com respeito a θ, tentando melhorar essa performance

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\widehat{\nabla J(\boldsymbol{\theta}_t)},$$

- Esse são os *policy-gradient methods.* Métodos que aproxima a política E a função valor são chamadas de *actor-critic*.

Exemplo

- Nos policy-gradient methods, a política pode ser parametrizada de qualquer jeito contanto que π(a | s,θ) seja diferenciável em respeito a θ.

- Se o número de ações é discreto e não tão grande, a parametrização levá forma de uma preferência parametrizada h(s,a,θ) em que a probabilidade de uma ação ser escolhida é dada pela soft-max

$$\pi(a|s,\boldsymbol{\theta}) \doteq \frac{e^{h(s,a,\boldsymbol{\theta})}}{\sum_b e^{h(s,b,\boldsymbol{\theta})}}$$

- Essa preferência parametrizada pode ser feita de várias formas, como uma rede neural cujos pesos são θ, ou simplesmente por métodos lineares

$$h(s,a,\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}(s,a)$$

Vantagens

Capacidade de Aproximar Políticas Determinísticas (Políticas Puras)

- A parametrização soft-max nas preferências de ação ($h(s,a,\theta)$) permite que a política se aproxime do determinismo (probabilidades de 0 ou 1), diferente de métodos epsilon-greedy.

- As preferências de ação são guiadas ao longo do treinamento: se a política ótima é determinística, as preferências das ações ótimas são levadas a serem infinitamente maiores.

Capacidade de Aprender Políticas Estocásticas Ótimas

- A parametrização permite a seleção de ações com probabilidades arbitrárias ($\pi(a|s,\theta)$), o que é essencial em problemas onde a política ótima é inerentemente estocástica (ex: Poker).

- Métodos baseados em valor de ação não possuem um mecanismo natural para encontrar políticas estocásticas ótimas.

Vantagens

Simplificação da Função a ser Aproximada

- Em certos problemas (e.g., Tetris), a política em si é uma função mais simples de modelar do que a complexa função de valor de ação Q(s,a).

- Nesses casos, a aproximação de política permite um aprendizado mais rápido e uma política assintótica superior.

Injeção de Conhecimento Prévio (Prior Knowledge)

- A escolha da forma da parametrização de $\pi(a|s,\theta)$ permite incorporar conhecimento prévio sobre a estrutura desejada da política.

- Isso pode ser a razão mais importante para selecionar um método baseado em política, direcionando o sistema de RL para a forma de solução esperada.

Policy Gradient Theorem

Função de otimização

Função que define quão boa é nossa Policy atual

$$J(\theta) = \sum_{s \in S} d^{\pi}(s) v^{\pi}(s) = \sum_{s \in S} d^{\pi} \sum_{a \in A} \pi_{\theta}(a|s) Q^{\pi}(s, a)$$

Policy Gradient-Theorem

Ideia: usar gradiente ascendente

- problema: gradiente de d(s)
- solução: reestruturar o problema

$$\nabla J(\theta) = \nabla \Big[\sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} Q^{\pi}(s,a)\, \pi_{\theta}(a|s) \Big]$$

$$\nabla J(\theta) \propto \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} Q^{\pi}(s,a)\, \nabla \pi_{\theta}(a|s)$$

Monte Carlo REINFORCE

Modificamos o gradiente anterior, chegando ao seguinte update:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla_{\boldsymbol{\theta}} \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)}$$

Monte Carlo REINFORCE

Algorithm 1 REINFORCE, um método de Gradiente de Política Monte Carlo (episódico)

Entrada: uma parametrização de política diferenciável $\pi(a|s, \boldsymbol{\theta})$
Inicialize o parâmetro da política $\boldsymbol{\theta} \in \mathbb{R}^{d'}$
repeat
    Gere um episódio $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, seguindo $\pi(\cdot|\cdot, \boldsymbol{\theta})$
    for cada passo do episódio $t = 0, \ldots, T-1$ **do**
        $G \leftarrow$ retorno a partir do passo t
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\gamma^t G \nabla_{\boldsymbol{\theta}} \ln \pi(A_t|S_t, \boldsymbol{\theta})$
    end for
until convergência =0

Métodos Actor-Critic

Problemas do REINFORCE

- Utiliza o retorno completo, o que dificulta o treinamento online
- Alta variância, o que aumenta o tempo até a convergência
- Se baseia em episódios

Métodos Actor-Critic

- Resolve esses problemas do mesmo modo que os métodos TD resolveram os problemas com os métodos de Monte Carlo, por meio do *bootstrapping*
- É composto por duas partes:
  - O Ator (Actor)
  - O Crítico (Critic)

Métodos Actor-Critic

- O Ator é responsável por controlar como o agente se comporta. Ele corresponde à política parametrizada $\pi(a|s, \theta)$
- O Crítico é responsável por avaliar a ação tomada pelo Ator. Ele aprende uma função de valor $\hat{v}(s, w)$

Métodos Actor-Critic

- A interação funciona da seguinte forma: o Ator executa uma ação A_t no estado S_t
- Crítico observa a recompensa e o próximo estado e calcula o erro TD

$$\delta_t = R_{t+1} - \bar{R}_{t+1} + \hat{v}(S_{t+1}, w_t) - \hat{v}(S_t, w_t)$$

- Se delta for positivo o ator aumenta a probabilidade dessa ação, se for negativo ele diminui

Pontos a Destacar

- O Crítico guia os ajustes feitos no Ator
- Reduz a variância pois usa *bootstrapping*
- Ótimo para aprendizado online

Política para Ações Contínuas

Ações Contínuas

- **Desafios:** Existem infinitas ações possíveis

- **Métodos de Valor da Ação falham:**
  - **Impossível** calcular um valor *Q(s, a)* para cada uma das infinitas ações
  - *max_a Q(s, a)* -> problema de otimização

Aprender uma Distribuição

- Aprender os **parâmetros de uma distribuição de probabilidade** sobre o espaço de ações
  - Ação amostrada a partir da distribuição

- **Distribuição Normal (Gaussiana):**
  - **Média:** *‘melhor aposta’*
  - **Desvio padrão:** *nível de exploração*

Definir e Aprender a Política

- **Definição da política**
  - Aprende a calcular a média e o desvio padrão para cada estado
  - Funções para mapear o estado *s* - média (linear) e desvio padrão (exponencial)
- **Ciclo Ação-Aprendizagem**
  - Observa o estado e calcula a distribuição
  - Amostra uma ação, executa e observa
  - Atualiza os pesos

O que veremos na próxima aula

data@icmc.usp.br

@data.icmc

/c/DataICMC

/icmc-data

data.icmc.usp.br

obrigado!