Aula 2 - Introduction to Reinforcement Learning

Gabriel Valentim

March 2023

1 Algoritmos de Policy Gradient

Para atingir o objetivo central do *Reinforcement Learning*, utilizamos algoritmos de treinamento que podem ser por meio de métodos que utilizam o gradiente da policy (*Policy Gradient*). Nesse sentido, vejamos a taxonomia dos métodos de treinamento de policy na figura 1.

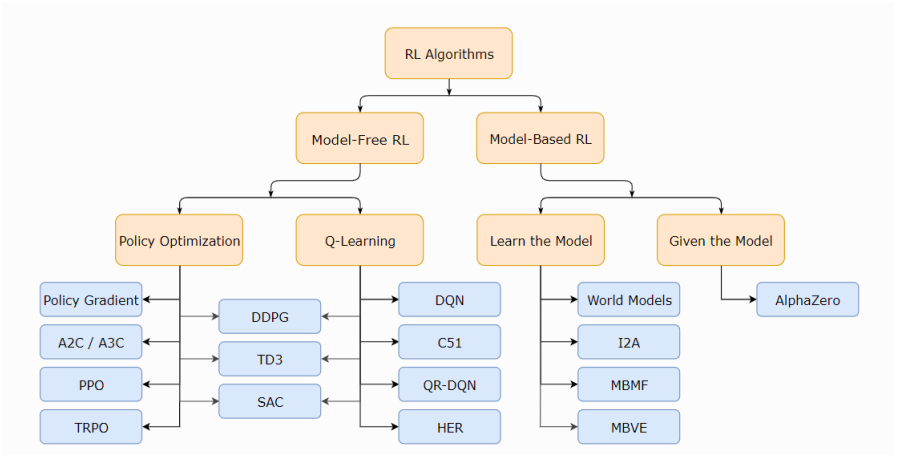

Figure 1: Taxonomia dos algoritmos de RL.

Dentre os inúmeros métodos e modelos existentes, vamos focar nos modelos livres (*Model-free RL*), em específico nos algoritmos de *Policy Gradient*, *PPO* (*Proximal Policy Opmization*) e *Genetic Algorithm*.

2 Derivando um *Policy Gradient* simplista

Como visto anteriormente, queremos maximizar o nosso return esperado que é definido por $J(\pi_\theta) = E_{\tau\sim\pi_\theta}[R(\tau)]$. Iremos optimizar nossa policy através do gradiente ascendente definido por:

$$\theta_{k+1} = \theta_k + \alpha\nabla_\theta J(\pi_{\theta_k}) \tag{1}$$

O gradiente da performance da *policy*, $\nabla_\theta J(\pi_\theta)$, é chamado de ***policy gradient***, os algoritmos que optimizam a *policy* são chamados de ***Policy Gradient Algorithms*** (como os citados aqui, *Policy Gradient simplista* e *Proximal Policy Optimization*). Nesse caso, para chegarmos no valor do *policy gradient*, vamos seguir alguns passos algébricos que vão nos direcionar para uma expressão que pode ser computada.

1. **Probabilidade de uma trajetória**. Dada uma trajetória $\tau = (s_0, a_0, ..., s_{T+1})$ e tomadas ações que seguem uma *policy* π_θ, teremos que:

$$P(\tau|\theta) = \rho_0(s_0) \prod_{t=0}^{T} P(s_{t+1}|s_t, a_t)\pi_\theta(a_t|s_t). \tag{2}$$

2. **Propriedade do log derivativo**. Tomando como base a regra da cadeia, teremos:

$$\nabla_\theta P(\tau|\theta) = P(\tau|\theta)\nabla_\theta \log P(\tau|\theta). \tag{3}$$

3. **Log-Prob de uma trajetória**. O log-prob de uma trajetória é:

$$\log P(\tau|\theta) = \log \rho_0(s_0) + \sum_{t=0}^{T} \Big(\log P(s_{t+1}|s_t, a_t) + \log \pi_\theta(a_t|s_t) \Big). \tag{4}$$

4. **Gradiente de funções de ambiente**: Como estamos derivando para variáveis de θ, funções de ambiente (que não dependem de θ) têm gradiente nulo.

5. **Grad-Log-Prob da trajetória**: O gradiente do log-prob de uma trajetória é:

$$\begin{aligned} \nabla_\theta \log P(\tau|\theta) &= \underbrace{\nabla_\theta \log \rho_0(s_0)}_{=0} + \sum_{t=0}^{T} \Big(\underbrace{\nabla_\theta \log P(s_{t+1}|s_t, a_t)}_{=0} + \nabla_\theta \log \pi_\theta(a_t|s_t) \Big) \\ &= \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t). \end{aligned} \tag{5}$$

Utilizando as construções acima, podemos concluir que:

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta E_{\tau\sim\pi_\theta}[R(\tau)] \\
&= \nabla_\theta \int_\tau P(\tau|\theta)R(\tau)d\tau && \text{Expande esperança} \\
&= \int_\tau \nabla_\theta P(\tau|\theta)R(\tau)d\tau && \text{Insere o gradiente na integral} \\
&= \int_\tau P(\tau|\theta)\nabla_\theta \log P(\tau|\theta)R(\tau)d\tau && \text{Log-derivativo} \\
&= E_{\tau\sim\pi_\theta}[\nabla_\theta \log P(\tau|\theta)R(\tau)] && \text{Esperança do retorno} \\
\nabla_\theta J(\pi_\theta) &= E_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)R(\tau)] && \text{Expressão para o grad-log-prob}
\end{aligned}
$$

O valor encontrado é uma esperança, portanto, podemos estimar esse valor. Se coletarmos um conjunto de trajetórias $\mathcal{D} = \{\tau_i\}_{i=1,\ldots,N}$, onde cada trajetória é feita pelo agente seguindo uma *policy* π_θ, o *policy gradient* pode ser estimado com:

$$\hat{g} = \frac{1}{|\mathcal{D}|} \sum_{\tau\in\mathcal{D}} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)R(\tau), \tag{6}$$

Onde $|\mathcal{D}|$ é o número de trajetórias no conjunto $\mathcal{D}$ (aqui, N). Essa última expressão é a versão mais simples computável que desejamos. Assumindo que tenhamos representado nossa *policy* de maneira que podemos calcular $\nabla_\theta \log \pi_\theta(a|s)$, e se somos capazes de executar a *policy* em um ambiente para coletar um conjunto de dados de trajetórias, nós podemos computar o *policy gradient* e tomar a atualização dos passos (*update step*).