Policy Search

Valdinei Freire

(EACH - USP)

Planejamento e Aprendizado por Reforço

- Value Iteration e Policy Iteration (operador de Bellman em todos estados)
- LAO* e LRTDP (operador de Bellman em estados alcançáveis)
- Monte Carlo Tree Search (amostragem das transições)
- Q-Learning e Sarsa(λ) Tabular (Programação Dinâmica Estocástica)
- Q-Learning e Sarsa(λ) Aproximado (Estados e Ações contínuas)

Problema 1: Qualidade da Política

- Avaliação de Políticas

$$V^{\pi}(s) = \mathsf{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | \pi, s_0 = s\right]$$

- Value-Based RL

$$\min_{\theta} \mathsf{E}_{s\sim\eta}\left[\left(V^{\pi}(s;\theta) - \sum_{t=0}^{\infty} \gamma^t r_t\right)^2 | \pi, s_0 = s\right]$$

$$\min_{\theta} \mathsf{E}_{s\sim\eta}\left[\left(r_t + \gamma V^{\pi}(s_{t+1};\theta) - V^{\pi}(s_t;\theta)\right)^2 | \pi, s_0 = s\right]$$

- Qualidade das Políticas

$$Q(s,a;\theta) \rightarrow \widehat{\pi}^* \rightarrow V^{\widehat{\pi}^*} = ?$$

Problema 2: Operador de Maximização

- Agindo de forma gulosa

$$a_t \leftarrow \max_a Q(s_t, a; \theta)$$

- Custo para ações discretas: $|\mathcal{A}| \times$ custo de acesso a $Q(s_t, a; \theta)$
- Custo para ações contínuas
  - como maximizar $Q(s_t, a; \theta)$?
  - problema de otimização
  - depende da classe de funções $Q(s_t, a; \theta)$
  - envolve alguma busca

Ponto-Fixo e Gradiente Ascendente

Definition. 1 (Equação de Ponto-Fixo)**.** *Dada uma função* $g : \mathbb{R}^d \to \mathbb{R}$ *contínua e diferenciável. Considere que exista uma contração* $G : \mathbb{R}^d \to \mathbb{R}^d$ *tal que:*

$$\nabla_{\mathbf{x}} g(\mathbf{x}^{(t)}) = \mathbf{0} \Leftrightarrow G(\mathbf{x}) = \mathbf{x}.$$

O método do Ponto-Fixo itera nos valores $x^{(t)}$ *seguindo:*

$$x^{(t+1)} = G(x^{(t)}).$$

Definition. 2 (Método do Gradiente Ascendente)**.** *O método do gradiente Ascendente itera nos valores* $\mathbf{x}^{(t)}$ *seguindo:*

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \beta^{(t)} \nabla_{\mathbf{x}} g(\mathbf{x}^{(t)}).$$

Se α *for bem escolhido e existir um ponto de máximo local, o operador acima é uma contração.*

Stochastic Approximation

Seja $H(y|x) = \Pr(Y_x \leqslant y)$ uma família de funções distribuição de probabilidade que correspondem a valores $x \in \mathbb{R}$. Defina:

$$M(x) = \mathrm{E}[Y_x] = \int_{-\infty}^{\infty} y \, dH(y|x).$$

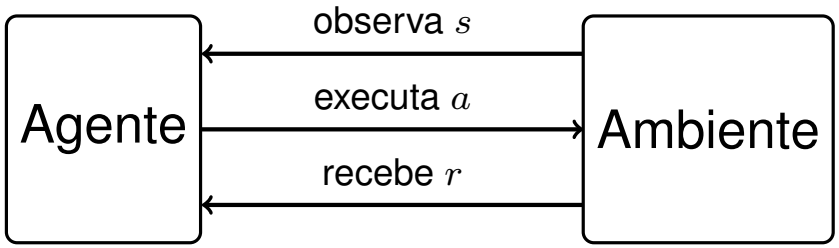
Considere que se possa obter amostras de Y_x para x arbitrários. Deseja-se encontrar θ tal que:

$$M(\theta) = \beta.$$

Dado um x_0 arbitrário, Robbins e Monro propõem o seguinte método:

$$x_{t+1} \leftarrow x_t + \alpha_t(\beta - y_t)$$

Problema de RL



Processo: em cada tempo t

- agente observa estado s_t
- agente escolhe ação a_t e atua no ambiente
- agente recebe recompensa r_t (potencialmente estocástica)
- ambiente transita para um novo estado s_{t+1} (potencialmente estocástica e dependente de s_t e a_t)

Policy Search

- Model-Based

$$\widehat{T}(s, a, s'), \widehat{R}(s, a) \Rightarrow \widehat{Q}(s, a) \Rightarrow \pi(s)$$

- Value-Based

$$\widehat{Q}(s, a) \Rightarrow \pi(s)$$

- Policy Search: busca direta no espaço de políticas probabilística

$$\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

Policy Search

Ideia geral:

1. escolhe modelo de política probabilística parametrizada por $\theta \in \mathbb{R}^d$

2. defina avaliador $J(\theta)$ da política π_θ

3. escolha θ_0 tal que $\pi_\theta(s, a) > 0$ para todo par (s, a)

4. itere sobre:

$$\theta_{t+1} = \theta_t + \alpha_t \nabla_\theta J(\theta_t)$$

Avaliando uma Política

Lembre-se que:

$$\begin{aligned} V^{\pi}(s) &= \mathrm{E}\left[\sum_{t=0}^{T} \gamma^t r_t \middle| s_0 = s, \pi\right] \\ &= \sum_{a \in \mathcal{A}} \pi(s,a) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} T(s,a,s') V^{\pi}(s') \right) \\ &= \sum_{a \in \mathcal{A}} \pi(s,a) Q^{\pi}(s,a) \end{aligned}$$

Considerando uma distribuição $\eta_0 : \mathcal{S} \rightarrow [0,1]$ sobre estados iniciais, podemos avaliar uma política como:

$$J(\theta) = \sum_{s \in \mathcal{S}} \eta_0(s) V^{\pi_\theta}(s)$$

Gradiente de uma Política

Pode-se demonstrar que:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \sum_{t=0}^{\infty} \sum_{s \in \mathcal{S}} \gamma^t \Pr(s_t = s | \eta_0, \pi) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(s,a) Q^{\pi_\theta}(s,a) \\
&= \sum_{t=0}^{\infty} \gamma^t \mathsf{E}_{s_t, a_t \sim \pi_\theta, \eta_0} \left[V_t \frac{\nabla_\theta \pi_\theta(s_t, a_t)}{\pi_\theta(s_t, a_t)} \right] \\
&= \sum_{t=0}^{\infty} \gamma^t \mathsf{E}_{s_t, a_t \sim \pi_\theta, \eta_0} \left[V_t \nabla_\theta \log \pi_\theta(s_t, a_t) \right]
\end{aligned}$$

onde $V_t = \sum_{k=t}^{\infty} \gamma^{k-t} r_k$ é a recompensa acumulada descontada a partir do tempo t.

Algoritmo REINFORCE

1. Inicialize θ arbitrariamente
2. Repita para todo episódio:
   (a) gere um episódio $s_0, a_0, r_0, \ldots, s_T, a_T, r_T$ seguindo a política π_θ
   (b) para cada tempo $t = 0, 1, \ldots, T$, faça:
      i. $V \leftarrow \sum_{k=t}^{T} \gamma^{k-t} r_k$
      ii. $\theta \leftarrow \theta + \alpha_t \gamma^t V \nabla_\theta \log \pi_\theta(s_t, a_t)$

Distribuição de Boltzman (Softmax)

Considere uma função $H : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, pode-se construir a seguinte política probabilística:

$$\pi(s,a) = \frac{e^{\tau H(s,a)}}{\sum_{a' \in \mathcal{A}} e^{\tau H(s,a')}},$$

onde τ é o parâmetro de temperatura. Se $\tau = 0$, temos uma distribuição uniforme.

Considere que $\theta \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ e que $\theta_{s,a} = H(s,a)$, temos que:

$$\frac{\partial \log \pi_\theta(s_t, a_t)}{\partial \theta_{s,a}} = \begin{cases} \tau - \tau\pi(s,a) & , \text{ se } a = a_t \text{ e } s = s_t \\ -\tau\pi(s,a) & , \text{ se } a \neq a_t \text{ e } s = s_t \\ 0 & , \text{ caso contrário} \end{cases}$$

Distribuição de Boltzman (Softmax)

Considere d funções bases $f_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ e $\theta \in \mathbb{R}^d$. Pode-se construir a seguinte política probabilística:

$$\pi(s,a) = \frac{\exp(\tau \sum_{i=1}^{d} f_i(s,a)\theta_i)}{\sum_{a' \in \mathcal{A}} \exp(\tau \sum_{i=1}^{d} f_i(s,a')\theta_i)},$$

onde τ é o parâmetro de temperatura.

Temos que:

$$\frac{\partial \log \pi_\theta(s_t,a_t)}{\partial \theta_i} = \tau f_i(s_t,a_t) - \tau \sum_{a' \in \mathcal{A}} f_i(s_t,a')\pi(s,a')$$

Distribuição Normal

Como considerar problemas quando ação é contínua?

Exemplo: Distribuição Normal

$$\pi_\theta(s,a) = \frac{1}{\sigma(s,\theta)\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s,\theta))^2}{2\sigma(s,\theta)^2}\right)$$

Pode-se considerar, por exemplo:

$$\mu(s,\theta) = \sum_{i=1}^{d} f_i(s,a)\theta_{\mu,i}$$

e

$$\sigma(s,\theta) = \sum_{i=1}^{d} f_i(s,a)\theta_{\sigma,i}$$

REINFORCE com Baseline

Considere uma função $b : \mathcal{S} \to \mathbb{R}$, temos que:

$$\sum_{a \in \mathcal{A}} b(s) \nabla_\theta \pi_\theta(s, a) = b(s) \nabla_\theta \sum_{a \in \mathcal{A}} \pi_\theta(s, a) = b(s) \nabla_\theta 1 = 0$$

Portanto:

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_{t=0}^{\infty} \sum_{s \in \mathcal{S}} \gamma^t \Pr(s_t = s | \eta_0, \pi) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(s, a) (Q^{\pi_\theta}(s, a) - b(s)) \\ &= \sum_{t=0}^{\infty} \gamma^t \mathsf{E}_{s_t, a_t \sim \pi_\theta, \eta_0} \left[(V_t - b(s_t)) \nabla_\theta \log \pi_\theta(s_t, a_t) \right] \end{aligned}$$

Se $b(s)$ é escolhido apropriadamente, pode-se diminuir a variância do gradiente. Usualmente estima-se:

$$b(s) = V^{\pi_\theta}(s)$$

Algoritmo REINFORCE com Baseline

1. Inicialize θ e $b(s)$ arbitrariamente
2. Repita para todo episódio:
   (a) gere um episódio $s_0, a_0, r_0, \ldots, s_T, a_T, r_T$ seguindo a política π_θ
   (b) para cada tempo $t = 0, 1, \ldots, T$, faça:
      i. $V \leftarrow \sum_{k=t}^{T} \gamma^{k-t} r_k$
      ii. $\delta \leftarrow V - b(s_t)$
      iii. $b(s_t) \leftarrow b(s_t) + \beta_t \delta$
      iv. $\theta \leftarrow \theta + \alpha_t \gamma^t \delta \nabla_\theta \log \pi_\theta(s_t, a_t)$

Actor-Critic

Lembre-se que $\nabla_\theta J(\theta)$ depende de $Q^{\pi_\theta}(s, a)$.

O algoritmo REINFORCE utiliza Monte-Carlo e considera apenas amostras de $V_t \sim Q^{\pi_\theta}(s_t, a_t)$.

Pode-se utilizar o algoritmo TD(0) e substituir V_t por $r_t + \gamma V^{\pi}(s_{t+1})$.

A política π_θ é um **actor**, enquanto, nesse caso, o algoritmo TD(0) seria um **critic**.

Algoritmo Actor-Critic com TD(0)

1. Inicialize θ e $b(s)$ arbitrariamente
2. Repita para todo episódio:
   (a) gere um episódio $s_0, a_0, r_0, \ldots, s_T, a_T, r_T$ seguindo a política π_θ
   (b) para cada tempo $t = 0, 1, \ldots, T$, faça:
      i. $\delta \leftarrow r_t + \gamma V(s_{t+1}) - V(s_t)$
      ii. $V(s_t) \leftarrow V(s_t) + \beta_t \delta$
      iii. $\theta \leftarrow \theta + \alpha_t \gamma^t \delta \nabla_\theta \log \pi_\theta(s_t, a_t)$

Variações

Nas versões com TD(0) e baseline foram apresentadas versões sem aproximações, mas obviamente aproximações também podem ser utilizadas para essas funções.

O algoritmo REINFORCE é a versão Monte-Carlo de Policy Search. Pode-se construir uma versão Actor-Critic com TD(λ) também.

Policy Iteration

Ideia:

- avaliação: executa N episódios com a mesma política π_θ
- melhoria: resolve algum problema de otimização

Exemplos:

- Cross-Entropy
- Trusted Region

Cross Entropy Method

Ideia:

- executa N episódios com a mesma política π_θ
- escolhe os ρ% melhores episódios
- otimiza π_θ para apresentar a menor cross entropy

Cross Entropy para duas distribuição p e q é dada por:

$$H(p, q) = \mathsf{E}_p[-\log q] = -\sum p(x) \log q(x)$$

Cross Entropy Method

Avaliação: cada episódio i é avaliado por:

$$V^i = \sum_{t=0}^{T_i} \gamma^t r_t^i$$

Melhoria: considere $\mathcal{I}_\rho$ como o conjunto dos ρ% melhores episódios e faça

$$\theta = \arg\max_{\tilde{\theta}} \sum_{i \in \mathcal{I}_\rho} \sum_{t=0}^{T_i} \log \pi_{\tilde{\theta}}(s_t^i, a_t^i)$$

Trusted Region Optimization Policy

Considere $\eta(s)$ a esperança da quantidade de ocorrências descontada do estado s, isto é,

$$\eta^{\pi}(s) = \mathsf{E}_{s_t \sim \pi, \eta_0}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}_{s_t=s}\right] = \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s | \eta_0, \pi)$$

onde $\mathbb{1}_{s_t=s}$ é a função indicadora para a condição $s_t = s$.

Considere $A_{\pi}(s, a)$ como a vantagem de aplicar a ação a no lugar da ação indicada por π, isto é,

$$A_{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$$

Trusted Region Optimization Policy

Pode-se provar que:

$$J(\tilde{\theta}) = J(\theta) + \sum_{s \in \mathcal{S}} \eta^{\pi_{\tilde{\theta}}}(s) \sum_{a \in \mathcal{A}} \pi_{\tilde{\theta}}(s,a) A_{\theta}(s,a)$$

Considere a aproximação considerando amostragem em π_θ:

$$L_{\theta}(\tilde{\theta}) = J(\theta) + \sum_{s \in \mathcal{S}} \eta^{\pi_{\theta}}(s) \sum_{a \in \mathcal{A}} \pi_{\tilde{\theta}}(s,a) A_{\theta}(s,a)$$

Seja θ^* tal que $L_\theta(\theta^*) > L_\theta(\theta)$, então, existe $\alpha \in (0, 1]$ tal que:

$$J((1-\alpha)\theta + \alpha\theta^*) > J(\theta)$$

Trusted Region Optimization Policy

Theorem. 3. *Seja* $\epsilon = \max_{s,a} |A_\theta(s,a)|$, *então o seguinte limite existe:*

$$J(\tilde{\theta}) \geqslant L_\theta(\tilde{\theta}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} \max_s D_{KL}(\pi_\theta(s,\cdot) \parallel \pi_{\tilde{\theta}}(s,\cdot)) = \tilde{J}_\theta(\tilde{\theta})$$

onde $D_{KL}(p \parallel q) = -\sum_i p(i) \log \frac{q(i)}{p(i)}$ *é a divergência de Kullback-Leibler.*

Dessa forma, dado uma política inicial π_{θ_0}, pode-se resolver o seguinte problema:

$$\theta_{t+1} = \arg\max_{\tilde{\theta} \in \mathbb{R}^d} \tilde{J}_{\theta_t}(\tilde{\theta})$$

Trusted Region Optimization Policy

O algoritmo TRPO resolve estocasticamente uma aproximação do problema anterior.

Considere:

$$\bar{D}^{\eta}_{KL}(\theta_1, \theta_2) = \frac{\eta(s)}{\sum_{s' \in \mathcal{S}} \eta(s')} D_{KL}(\pi_{\theta_1}(s, \cdot) \parallel \pi_{\theta_2}(s, \cdot))$$

Escolha um delimitador de região $\delta > 0$ e resolva o seguinte problema:

$$\begin{aligned} \max_{\theta} \quad & L_{\theta_{old}}(\theta) \\ \text{sujeito a} \quad & \bar{D}^{\eta_{\theta_{old}}}_{KL}(\theta_{old}, \theta) \leqslant \delta \end{aligned}$$

Trusted Region Optimization Policy

Maximizar $L_{\theta_{old}}(\theta)$ é equivalente a maximizar:

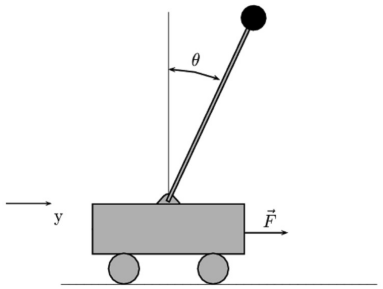
$$\sum_{s \in \mathcal{S}} \eta^{\pi_{\theta_{old}}}(s) \sum_{a \in \mathcal{A}} \pi_\theta(s,a) Q_{\theta_{old}}(s,a)$$

Colocando no formato de esperança, temos:

$$\begin{aligned} \max_\theta \quad & \mathrm{E}_{s,a \sim \pi_{\theta_{old}}} \left[\frac{\pi_\theta(s,a)}{\pi_{\theta_{old}}(s,a)} Q_{\theta_{old}}(s,a) \right] \\ \text{sujeito a} \quad & \mathrm{E}_{s,a \sim \pi_{\theta_{old}}} \left[D_{KL}(\pi_{\theta_{old}} \parallel \pi_\theta) \right] \leqslant \delta \end{aligned}$$

Relatório 2 - Reinforcement Learning

- Implementar pelo menos três algoritmos:
  - value-based
  - policy search
  - sample efficient



- utilizar o AI Gym (gym.openai.com)

- testá-lo no ambiente de Cart-Pole

Relatório 2 - Reinforcement Learning

- considere avaliar primeiro as implementações dos algoritmos em um ambiente discreto
- os algoritmos não precisam se restringir aos vistos em sala de aula
- pode aproveitar implementações prontas (paperswithcode.com)
- data de entrega: 6 de Julho

Referências

Richard Sutton and Andrew Barto. Reinforcement Learning: An Introduction Second Edition, MIT Press, Cambridge, MA, 2018

Csaba Szepesvári. Algorithms for Reinforcement Learning, Morgan & Claypool Publishers, 2009

Shie Mannor, Reuven Rubinstein and Yohai Gat. The Cross Entropy method for Fast Policy Search, ICML, 2003

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan and Pieter Abbeel. Trust Region Policy Optimization, ICML, 2015