L'algorithme TRPO : Trust Region Policy Optimization

Séminaire TALia

Jean Vassoyan

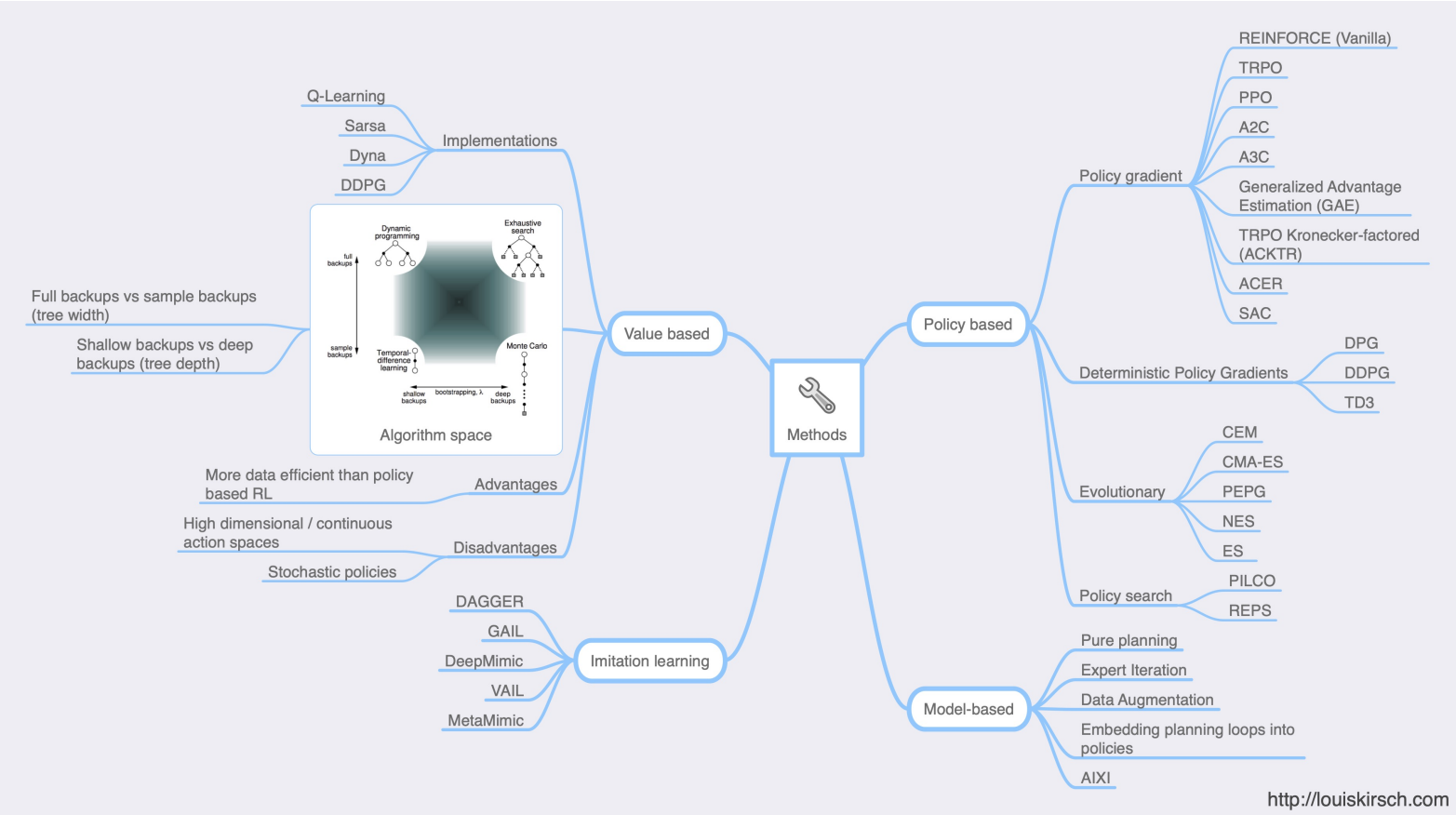
onepoint & Télécom Paris

May 13, 2022

Outline

1. Rappels de RL
2. TRPO en théorie
3. TRPO en pratique
4. Recap

Reinforcement Learning map



Fonction objectif

- Contexte : Policy Gradient methods
- Fonction objectif :

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)] = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{H} R_t\right]$$

- Approche standard : Vanilla Policy Gradient

REINFORCE algorithm:

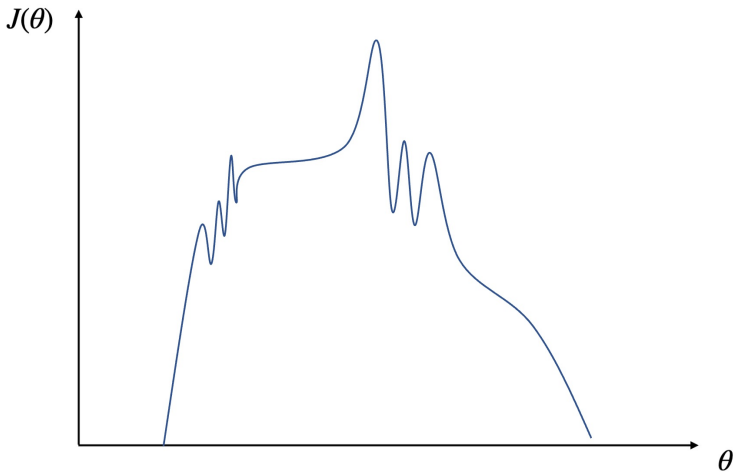
1. sample $\{\tau^i\}$ from $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ (run it on the robot)
2. $\nabla_\theta J(\theta) \approx \sum_i \left(\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)\right)\left(\sum_t r(\mathbf{s}_t^i, \mathbf{a}_t^i)\right)$
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Vanilla Policy Gradient : 2 problèmes

1. Peu *sample efficient*
   - Les trajectoires passées ne sont pas réutilisables

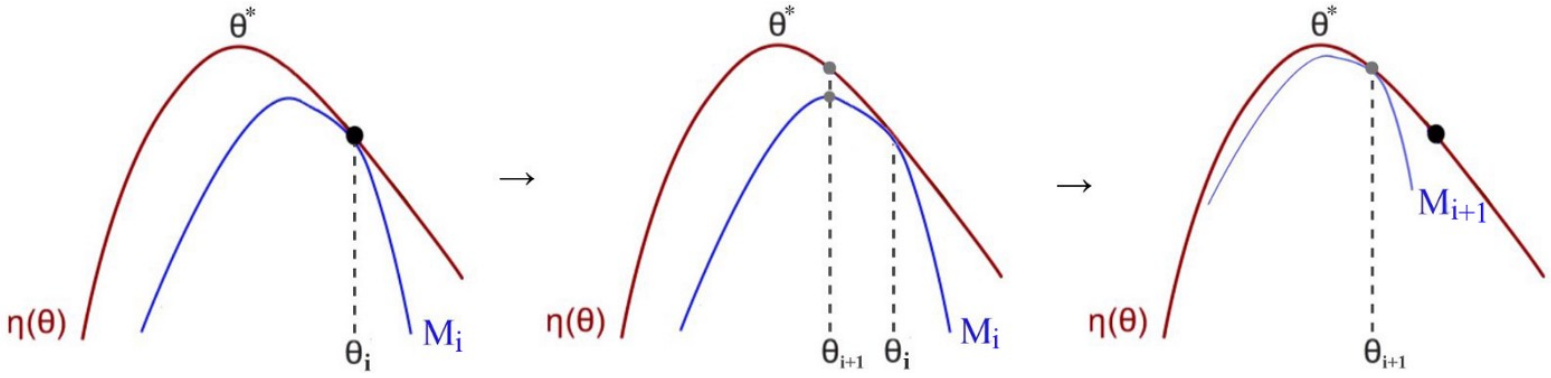
$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

2. Convergence instable
   - La statistique des données recuillies varie avec θ : impossible de trouver un α approprié sur tout l'entrainement



Nouveau problème d'optimisation : l'algorithme MM

- MM = Minorize Maximization
- Minorer localement $J(\theta)$ par une fonction de substitution $M(\theta)$ et résoudre le problème d'optimisation dessus :



- Objectif : trouver une fonction de substitution $M(\theta) \leq J(\theta)$ sur laquelle on peut **résoudre** le problème d'optimisation.

Recherche d'un bon minimiseur $M(\theta)$

- Objectif : se servir des trajectoires passées pour trouver un bon approximateur de $J(\theta)$:

Lemme 1

$$J(\theta) = J(\theta_{\text{old}}) + \mathbb{E}_{s_0, a_0, \cdots \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t A_{\pi_{\theta_{\text{old}}}} (s_t, a_t) \right]$$

D'où :

$$J(\theta) = J(\theta_{\text{old}}) + \sum_s \rho_{\pi_\theta}(s) \sum_a \pi_\theta(a \mid s) A_{\pi_{\theta_{\text{old}}}}(s, a)$$

Avec $\quad \rho_{\pi_\theta}(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots$

Recherche d'un bon minimiseur $M(\theta)$

$$J(\theta) = J(\theta_{\text{old}}) + \sum_s \rho_{\pi_\theta}(s) \sum_a \pi_\theta(a \mid s) A_{\pi_{\theta_{\text{old}}}}(s, a)$$

Approximation :

$$L_{\theta_{\text{old}}}(\theta) = J(\theta_{\text{old}}) + \sum_s \rho_{\pi_{\theta_{\text{old}}}}(s) \sum_a \pi_\theta(a \mid s) A_{\pi_{\theta_{\text{old}}}}(s, a)$$

$L_{\theta_{\text{old}}}(\theta) \approx J(\theta)$ (approximation du 1er ordre)

Recherche d'un bon minimiseur $M(\theta)$

Theorem

$$J(\theta) \geq L_{\theta_{\text{old}}}(\theta) - \frac{4\epsilon\gamma}{(1-\gamma)^2} D_{\text{TV}}^{\max}(\pi_\theta, \pi_{\theta_{\text{old}}})^2$$

$$\textit{avec :} \quad \epsilon = \max_{s,a} \left| A_{\pi_{\theta_{\text{old}}}}(s,a) \right|$$

Où :

$$D_{\text{TV}}^{\max}(\pi_\theta, \pi_{\theta_{\text{old}}}) = \max_s \left[\frac{1}{2} \sum_a |\pi_\theta(a \mid s) - \pi_{\theta_{\text{old}}}(a \mid s)| \right]$$

$\rightarrow$ Difficile à estimer !

Trick : $D_{\text{KL}}(p\|q) \geq D_{TV}(p\|q)^2$

Recherche d'un bon minimiseur $M(\theta)$

On obtient finalement que :

$$J(\theta) \geq L_{\theta_{\text{old}}}(\theta) - C\, D_{\text{KL}}^{\max}(\pi_\theta, \pi_{\theta_{\text{old}}})$$

$$\text{où} \quad C = \frac{4\epsilon\gamma}{(1-\gamma)^2} \qquad \text{et} \quad \epsilon = \max_{s,a} \left| A_{\pi_{\theta_{\text{old}}}}(s,a) \right|$$

D'où le minimiseur :

$$M_{\theta_{\text{old}}}(\theta) = L_{\theta_{\text{old}}}(\theta) - \frac{4\gamma \max_{s,a} \left| A_{\pi_{\theta_{\text{old}}}}(s,a) \right|}{(1-\gamma)^2}\, D_{\text{KL}}^{\max}(\pi_\theta, \pi_{\theta_{\text{old}}})$$

Algorithme TRPO idéal

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return η

Initialize π_0.
for $i = 0, 1, 2, \ldots$ until convergence **do**
Compute all advantage values $A_{\pi_i}(s, a)$.
Solve the constrained optimization problem

$$\pi_{i+1} = \arg\max_{\pi} \left[L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

where $C = 4\epsilon\gamma/(1-\gamma)^2$

and $L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$

end for

Formulation en région de confiance

$$\underset{\theta}{\text{maximize}}\,[L_{\theta_{\text{old}}}(\theta) - C\, D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)] \qquad \text{où} \qquad C = \frac{4\epsilon\gamma}{(1-\gamma)^2}$$

Par dualité Lagrangienne, ce problème d'optimisation est équivalent à :

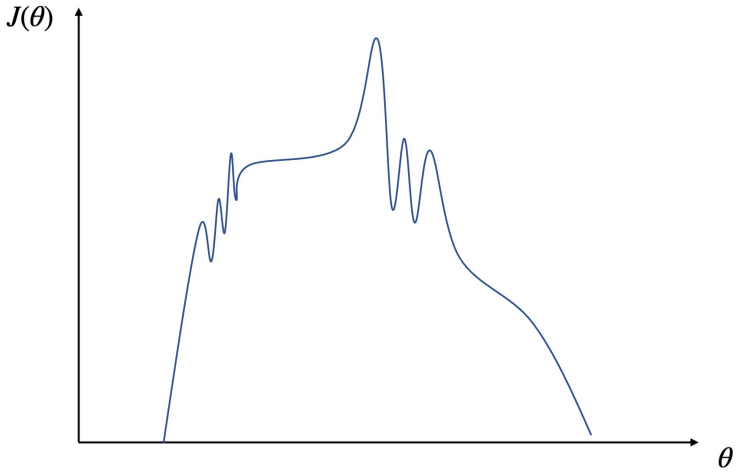
$$\underset{\theta}{\text{maximize}}\; L_{\theta_{\text{old}}}(\theta)$$
$$\text{tel que} \quad D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta$$

où δ dépend de C et donc de θ_{old}.

Limites du TRPO idéal

$$\underset{\theta}{\text{maximize}}\ L_{\theta_{\text{old}}}(\theta)$$
$$\text{tel que} \quad D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta$$

En pratique, si l'on utilisait le coefficient de pénalité C (ou δ) exigé par la théorie, les *stepsizes* seraient minuscules.



Approximation 1 : paramétrisation de δ

Pour laisser au système la possibilité d'effectuer des steps plus "larges", on paramétrise δ :

$$\underset{\theta}{\text{maximize}}\ L_{\theta_{\text{old}}}(\theta)$$
$$\text{tel que} \quad D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta$$

Note : le δ théorique était "adaptatif", puisqu'il dépendait de θ_{old}. Il est désormais fixé.

$\rightarrow$ on relaxe la contrainte sur les régions de confiance !

Approximation 2 : KL divergence

La contrainte $D_{\mathrm{KL}}^{\max}(\theta_{\mathrm{old}}, \theta) \leq \delta$ impose une borne en **chaque** point de l'espace.

$\rightarrow$ difficile à vérifier en pratique

On utilise à la place la KL divergence moyenne :

$$\bar{D}_{\mathrm{KL}}^{\rho}(\theta_1, \theta_2) := \mathbb{E}_{s \sim \rho}\left[D_{\mathrm{KL}}\left(\pi_{\theta_1}(\cdot \mid s) \| \pi_{\theta_2}(\cdot \mid s)\right)\right]$$

$$\underset{\theta}{\text{maximize}}\ L_{\theta_{\mathrm{old}}}(\theta)$$

$$\text{tel que} \quad \bar{D}_{\mathrm{KL}}^{\rho_{\theta_{\mathrm{old}}}}(\theta_{\mathrm{old}}, \theta) \leq \delta$$

Problème d'optimisation final

Calcul de $L_{\theta_{\text{old}}}$:

$$
\begin{aligned}
L_{\theta_{\text{old}}} &\approx \sum_s \rho_{\theta_{\text{old}}}(s) \sum_a \pi_\theta(a \mid s) A_{\theta_{\text{old}}}(s,a) \\
&= \frac{1}{1-\gamma} \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}} \sum_a \pi_\theta(a \mid s) A_{\theta_{\text{old}}}(s,a) \\
&= \frac{1}{1-\gamma} \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}} \sum_a \pi_{\theta_{\text{old}}}(a \mid s) \frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)} A_{\theta_{\text{old}}}(s,a) \qquad (1) \\
&= \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)} A_{\theta_{\text{old}}}(s,a) \right] \\
&\approx \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)} Q_{\theta_{\text{old}}}(s,a) \right]
\end{aligned}
$$

Problème d'optimisation final

$$\underset{\theta}{\text{maximize}} \quad \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)} Q_{\theta_{\text{old}}}(s,a)\right]$$

$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\mathrm{KL}}\left(\pi_{\theta_{\text{old}}}(\cdot \mid s) \| \pi_\theta(\cdot \mid s)\right)\right] \leq \delta$$

Résolution approximative du problème

- On note : $\mathcal{L}_{\theta_{\text{old}}}(\theta) := \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} Q_{\theta_{\text{old}}}(s,a)\right]$
- Par développement de Taylor :

$$\mathcal{L}_{\theta_{\text{old}}}(\theta) \approx \overbrace{\mathcal{L}_{\theta_{\text{old}}}(\theta_{\text{old}})}^{0} + \nabla_\theta \mathcal{L}_{\theta_{\text{old}}}(\theta)|_{\theta_{\text{old}}}(\theta - \theta_{\text{old}}) + \dots$$

$$\bar{D}_{KL}(\theta\|\theta_{\text{old}}) \approx \overbrace{\bar{D}_{KL}(\theta_k\|\theta_{\text{old}})}^{0} + \overbrace{\nabla_\theta \bar{D}_{KL}(\theta\|\theta_{\text{old}})|_{\theta_{\text{old}}}(\theta - \theta_{\text{old}})}^{0}$$
$$+ \frac{1}{2}(\theta - \theta_{\text{old}})^T \nabla_\theta^2 \bar{D}_{KL}(\theta\|\theta_{\text{old}})|_{\theta_{\text{old}}}(\theta - \theta_{\text{old}}) + \dots$$

Résolution approximative du problème

On obtient donc :

$$\mathcal{L}_{\theta_{\text{old}}}(\theta) \approx g^T (\theta - \theta_{\text{old}}) \qquad g \doteq \nabla_\theta \mathcal{L}_{\theta_{\text{old}}}(\theta)|_{\theta_{\text{old}}}$$

$$\bar{D}_{KL}(\theta \| \theta_{\text{old}}) \approx \frac{1}{2} (\theta - \theta_{\text{old}})^T H (\theta - \theta_{\text{old}}) \qquad H \doteq \nabla_\theta^2 \bar{D}_{KL}(\theta \| \theta_{\text{old}})\big|_{\theta_{\text{old}}}$$

D'où le nouvel objectif d'optimisation :

$$\begin{aligned} \theta_{k+1} = \ & \arg\max_\theta \ g^T (\theta - \theta_{\text{old}}) \\ & \text{s.t. } \frac{1}{2} (\theta - \theta_{\text{old}})^T H (\theta - \theta_{\text{old}}) \leq \delta \end{aligned}$$

Solution exacte

Objectif :

$$\theta_{k+1} = \arg\max_{\theta} \; g^T (\theta - \theta_{\text{old}})$$
$$\text{s.t. } \frac{1}{2} (\theta - \theta_{\text{old}})^T H (\theta - \theta_{\text{old}}) \leq \delta$$

Solution exacte :

$$\theta_{k+1} = \theta_{\text{old}} + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

Problème : le calcul de H^{-1} est coûteux (surtout si la policy a beaucoup de paramètres) et instable (l'approximation amplifie les erreurs sur H^{-1}).

Approximation de la solution

On va approximer cette solution en cherchant un x_k tel que :

$$x_k \approx \hat{H}_k^{-1} \hat{g}_k$$

Cela revient à chercher x_k tel que $\hat{H}_k x_k \approx \hat{g}$, ce qui peut se ramener au problème d'optimisation suivant :

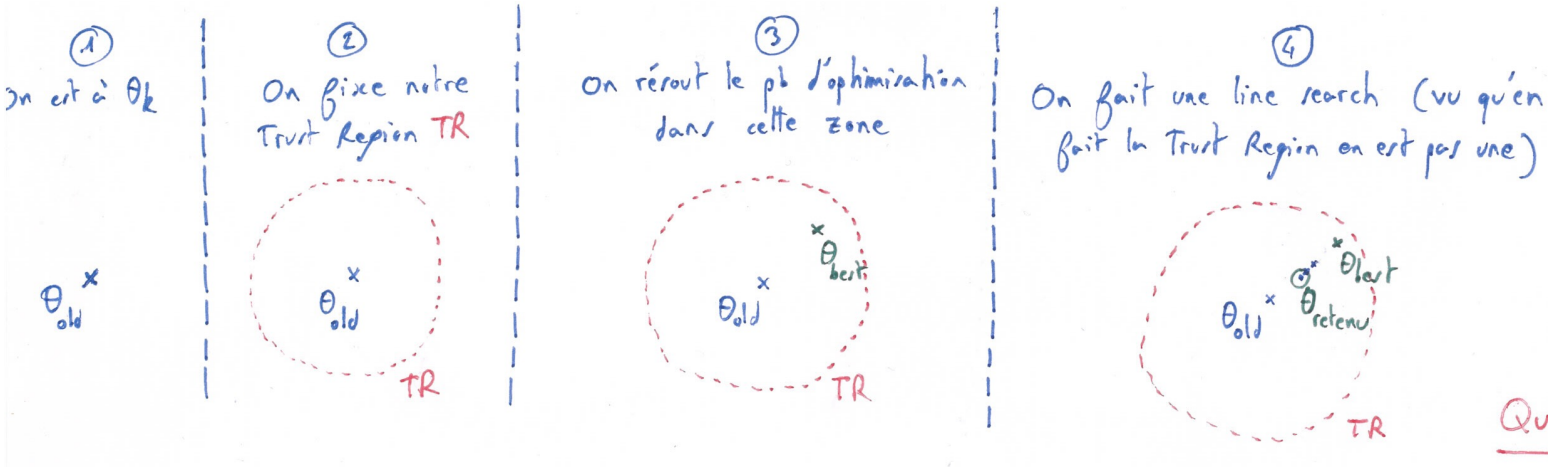
$$x_k = \text{argmin}_{x \in \mathbb{R}^n} \frac{1}{2} x^T H x - g^T x$$

En pratique, on résout ce problème par *conjugate gradient method*, qui est similaire à la gradient ascent, mais avec moins d'itérations (N itérations, grâce à une recherche orthogonale).

Line search

- Problème : comme la Trusted Region était trop petite, on a relaxé sa taille avec un hyperparamètre (C ou δ). De plus, l'approximation quadratique qu'on a faite réduit également l'accuracy. Pour ces deux raisons, la performance peut se dégrader d'un step à l'autre.
- Solution : avant de changer la policy, on fera un check via une line search. Si le check n'est pas concluant, on réduit le learning rate ($1 \to \alpha \to \alpha^2 \to \alpha^3$ ...) : on part de la limite de la trust region et on rentre petit à petit (perso je fixerais α pas loin de 1).

Recap


1
On est à θ_k
θ_{old}
2
On fixe notre Trust Region TR
θ_{old}
TR
3
On résout le pb d'optimisation dans cette zone
θ_{old}
θ_{best}
TR
4
On fait une line search (vu qu'en fait la Trust Region en est pas une)
θ_{old}
θ_{best}
θ_{retenu}
TR
Qu

Algo complet

Algorithm 1 Trust Region Policy Optimization

Input: initial policy parameters θ_0, initial value function parameters ϕ_0
Hyperparameters: KL-divergence limit δ, backtracking coefficient α, maximum number of backtracking steps K
for $k = 0, 1, 2, \ldots$ **do**
  Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
  Compute rewards-to-go $\hat{R}_t$.
  Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
  Estimate policy gradient as

$$\hat{g}_k = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta_k} \hat{A}_t.$$

  Use the conjugate gradient algorithm to compute

$$\hat{x}_k \approx \hat{H}_k^{-1} \hat{g}_k,$$

  where $\hat{H}_k$ is the Hessian of the sample average KL-divergence.
  Update the policy by backtracking line search with

$$\theta_{k+1} = \theta_k + \alpha^j \sqrt{\frac{2\delta}{\hat{x}_k^T \hat{H}_k \hat{x}_k}} \hat{x}_k,$$

  where $j \in \{0, 1, 2, \ldots K\}$ is the smallest value which improves the sample loss and satisfies the sample KL-divergence constraint.
  Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_\phi \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

  typically via some gradient descent algorithm.
end for