Policy Search

A review

Charles Reizine, Emile Mathieu

January 17, 2017

Ecole des Ponts ParisTech

Problem Statement

Reinforcement Learning

Markov Decision Process

- State space $\boldsymbol{x} \in \mathcal{X}$
- Action space $\boldsymbol{u} \in \mathcal{U}$
- Transition dynamics $\mathcal{P}(\boldsymbol{u_{t+1}}|\boldsymbol{x_t}, \boldsymbol{u_t})$
- Reward function $r(\boldsymbol{x_t}, \boldsymbol{u_t})$
- Initial state probabilities $\mu_o(\boldsymbol{x_t})$

Learning

Adapting the policy $\pi(\boldsymbol{u}|\boldsymbol{x})$

Objective

Find $\pi^\star \in \arg\max_\pi J_\pi$ with $J_\pi = \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$

Learning Paradigms

Value Function :

Estimate the *value function* $V^{\pi}(\boldsymbol{x}) = \mathbb{E}_{\pi}[R_t|\boldsymbol{x}_t = \boldsymbol{x}]$ or *action-value function* $Q^{\pi}(\boldsymbol{u}, \boldsymbol{x}) = \mathbb{E}_{\pi}[R_t|\boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{u}_t = \boldsymbol{u}]$.

Then compute the policy by action selection.

Limits

High dimensional or continuous problems.

Policy Search :

Employ a parametrize policy $\pi_{\boldsymbol{\theta}}$.

Iterate in the parameter space of the policy.

Fit large scale, continuous problems

Two approaches

- Model-based methods
- Model-free methods

Model-Free Policy Search

Model-Free Policy Search

Use samples $\mathcal{D} = \left\{ \left(\boldsymbol{x}_{1:T}^{[i]}, \boldsymbol{u}_{1:T}^{[i]}, r_{1:T}^{[i]} \right) \right\}_{i=1,\ldots,N}$

to directly update the policy π.

Pseudo code

Algorithm 1 Model free policy search

1: **while** has not converged **do**
2: Explore: Generate trajectories $\boldsymbol{\tau}^{[i]}$ from current policy π_k
3: Evaluate: Assess quality of trajectory or actions
4: Update: Compute new policy π_{k+1} from trajectories and evaluations
5: **end while**

Exploration Strategies & Evaluation Strategies

Episode-based

Explore:

In parameter space at the beginning of an episode: $\boldsymbol{\theta}_i \sim \pi_\omega(\theta)$

- Search distribution π_ω over parameter space $\boldsymbol{\theta} \in \Theta$
- Deterministic control policy: $\boldsymbol{u} = \pi_\theta(\boldsymbol{x})$

Evaluate:

The quality of parameter $\boldsymbol{\theta}_i$ by the accumulated reward:
$R^{[i]} = \sum_{t=1}^{T} r_t$, $\mathcal{D} = \left\{\theta_i, R^{[i]}\right\}$

Step-based

Explore:

In action space at each time step: $\boldsymbol{u}_t \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{u}|\boldsymbol{x}_t)$

- Stochastic control policy

Evaluate:

The quality of state-action pairs $(\boldsymbol{x}_t^{[i]}, \boldsymbol{u}_t^{[i]})$ by rewards to come:

$Q_t^{[i]} = \sum_{h=t}^{T} r_h(\boldsymbol{x}_h, \boldsymbol{u}_h)$

$\mathcal{D} = \left\{\boldsymbol{x}_t^{[i]}, \boldsymbol{u}_t^{[i]}, Q_t^{[i]}\right\}$

Policy Update Strategies

Policy gradients methods

Expectation-maximization based methods

Information theoretical approaches

Policy Gradient

Optimize average return $J_{\boldsymbol{\theta}}$ by **gradient ascent**.

Compute gradient from samples

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \int_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) R(\boldsymbol{\tau}) d\boldsymbol{\tau} = \int_{\boldsymbol{\tau}} \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) R(\boldsymbol{\tau}) d\boldsymbol{\tau}$$

$$\nabla_{\boldsymbol{\omega}} J_{\boldsymbol{\omega}} = \nabla_{\boldsymbol{\omega}} \int_{\boldsymbol{\theta}} \pi_{\boldsymbol{\omega}}(\boldsymbol{\theta}) \int_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) R(\boldsymbol{\tau}) d\boldsymbol{\tau} d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\omega}} \pi_{\boldsymbol{\omega}}(\boldsymbol{\theta}) \int_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) R(\boldsymbol{\tau}) d\boldsymbol{\tau} d\boldsymbol{\theta}$$

Update control policy parameter

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$$

$$\text{or} \quad \boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \alpha_k \nabla_{\boldsymbol{\omega}} J_{\boldsymbol{\omega}}$$

Finite Differences

Small perturbation

$\boldsymbol{\theta}_k + \delta\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_k$

Change of returns: $\delta R^{[i]} = R(\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}^{[i]}) - R(\boldsymbol{\theta}_k)$

Construct $\delta\boldsymbol{\theta} = [\delta\boldsymbol{\theta}^{[1]}, \ldots, \delta\boldsymbol{\theta}^{[N]}]^T$ and $\delta R = [\delta R^{[1]}, \ldots, \delta R^{[N]}]^T$.

Gradient approximation

Using a first-order Taylor approximation and solving $\nabla_{\boldsymbol{\theta}}^{FD} J_{\boldsymbol{\theta}}$ in the least-square sense yields:

$$\nabla_{\boldsymbol{\theta}}^{\mathrm{FD}} J_{\boldsymbol{\theta}} = (\delta\boldsymbol{\theta}^T \delta\boldsymbol{\theta})^{-1} \delta\boldsymbol{\theta}^T \delta R$$

Likelihood-Ratio Policy Gradients

Injecting the *likelihood ratio* trick $\nabla p_{\boldsymbol{\theta}}(y) = p_{\boldsymbol{\theta}}(y)\nabla \log p_{\boldsymbol{\theta}}(y)$ into $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$ gives:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} &= \int_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}}(y)\nabla \log p_{\boldsymbol{\theta}}(y) R(\boldsymbol{\tau}) d\boldsymbol{\tau} = \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{\tau})}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) R(\boldsymbol{\tau})] \\ &\simeq \frac{1}{N}\sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}^{[i]}) R(\boldsymbol{\tau}^{[i]})\end{aligned}$$

For a stochastic policy: $p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) = p(\boldsymbol{x}_1)\prod_{t=1}^{T} p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)\pi_{\boldsymbol{\theta}}(\boldsymbol{u}_t|\boldsymbol{x}_t, t)$

Hence $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) = \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{u}_t|\boldsymbol{x}_t, t)$

The REINFORCE algorithm uses the policy gradient:

$$\nabla_{\boldsymbol{\theta}}^{\mathrm{RF}} J_{\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{u}_t^{[i]}|\boldsymbol{x}_t^{[i]}, t) R(\boldsymbol{\tau}^{[i]})$$

Natural Policy Gradient

Objective: Achieve a more stable behaviour of the learning process.

Idea: Maintain a limited step-width in the trajectory distribution space, enforced by the constraint:

$$KL(p_{\boldsymbol{\theta}}(\boldsymbol{\tau})||p_{\boldsymbol{\theta}+\delta\boldsymbol{\theta}}(\boldsymbol{\tau})) \simeq \delta\boldsymbol{\theta}^T F_{\boldsymbol{\theta}} \delta\boldsymbol{\theta} \leq \epsilon$$

Optimization program:

$$\delta\boldsymbol{\theta}^{NG} = \arg\max_{\delta\boldsymbol{\theta}} \delta\boldsymbol{\theta}^T \delta\boldsymbol{\theta}^{VG} \quad \text{s.t.} \quad \delta\boldsymbol{\theta}^T F_{\boldsymbol{\theta}} \delta\boldsymbol{\theta} \leq \epsilon$$

Solution: $\delta\boldsymbol{\theta}^{NG} \propto F_{\boldsymbol{\theta}}^{-1} \delta\boldsymbol{\theta}^{VG}$

Natural policy gradient:

$$\nabla_{\boldsymbol{\theta}}^{NG} J_{\boldsymbol{\theta}} = F_{\boldsymbol{\theta}}^{-1} \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$$

Guided Policy Search

Issue: New trajectories $\boldsymbol{\tau}^{[i]}$ are required at each gradient step to compute: $\mathbb{E}[\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}] \simeq \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}^{[i]}) R(\boldsymbol{\tau}^{[i]})$

Importance Sampling: $\mathbb{E}[J_{\boldsymbol{\theta}}] \simeq \frac{1}{Z(\boldsymbol{\theta})} \sum_{i=1}^{N} \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}^{[i]})}{q(\boldsymbol{\tau}^{[i]})} R(\boldsymbol{\tau}^{[i]})$, with $\boldsymbol{\tau}^{[i]} \sim q$.

q can be a previous policy, or a guiding distribution constructed with differential dynamic programming (DDP).

LQR algorithm: Iteratively optimize a trajectory, with linear reward and quadratic dynamics approximations.

f and r estimated by finite differences.

Yields a deterministic policy: $\boldsymbol{u}_t = g(\boldsymbol{x}_t)$.

Stochastic policy: $q(\boldsymbol{u}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{u}_t|g(\boldsymbol{x}_t), \Sigma)$

Expectation Maximization for policy search

Observed variable: Binary reward event given by $p(R = 1|\boldsymbol{\tau}) = p(R|\boldsymbol{\tau})$, defined from a transformation of $R(\boldsymbol{\tau})$.

Latent variable: Trajectory $\boldsymbol{\tau}$.

We want to find the maximum solution $\boldsymbol{\theta}^{\star}$ for the log marginal-likelihood:

$$\log p_{\boldsymbol{\theta}}(R) = \int_{\boldsymbol{\tau}} p(R|\boldsymbol{\tau}) p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) d\boldsymbol{\tau}$$

M-Step: Yields closed form solution for parameters, for most common policies.

E-Step: Cannot be computed exactly: approximations are needed.

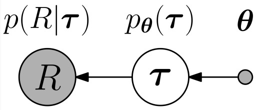

Figure 1: Graphical model for inference-based policy search.

E-Step approximation

Monte-Carlo EM-based Policy Search

Sample based approximation for the auxiliary distribution q:

$$q(\boldsymbol{\tau}) \simeq p(\boldsymbol{\tau}|R) \propto p(R|\boldsymbol{\tau})p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})$$

Since $\boldsymbol{\tau}^{[i]} \sim p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})$: $q(\boldsymbol{\tau}^{[i]}) \propto p(R|\boldsymbol{\tau}^{[i]})$

Expected complete log-likelihood:

$$\mathcal{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta}') \simeq \sum_{\boldsymbol{\tau}^{[i]} \sim p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})} p(R|\boldsymbol{\tau}^{[i]}) \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}^{[i]})$$

Variational Inference-based Policy Search

Use a parametrized auxiliary distribution q_β.

$$\begin{aligned} \beta &\in \arg\min_\beta KL(q_\beta(\boldsymbol{\tau}) || p(R|\boldsymbol{\tau})p_{\boldsymbol{\theta}}(\boldsymbol{\tau})) \\ &\in \arg\min_\beta \sum_{\boldsymbol{\tau}^{[i]}} q_\beta(\boldsymbol{\tau}^{[i]}) \log \frac{q_\beta(\boldsymbol{\tau}^{[i]})}{p(R|\boldsymbol{\tau}^{[i]})p_{\boldsymbol{\theta}}(\boldsymbol{\tau}^{[i]})} \end{aligned}$$

Information Theoretical Approaches

Main idea:

The trajectory distribution after the policy update should not be far from the trajectory distribution before the policy update.

Relative Entropy Policy Search (REPS):

$$\begin{aligned} \underset{\pi}{\text{maximize}} \quad & \int \pi(\boldsymbol{\theta}) R(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ \text{subject to} \quad & \pi(\boldsymbol{\theta}) \log \frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \leq \epsilon \\ & \int \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1 \end{aligned}$$

Solution via Lagrangian: $\pi(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta}) \exp(\frac{R(\boldsymbol{\theta})}{\eta})$.

Model-based Policy Learning

General Setup 1

Objective:

$$\pi_\theta^* \in \arg\max_\pi J_\theta = \arg\max_\pi \sum_{t=1}^{T} \gamma^t \mathbb{E}[r(\boldsymbol{x}_t, \boldsymbol{u}_t) | \pi_\theta], \quad \gamma \in [0, 1] \quad (1)$$

Assumption:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w}$$

General Setup 2

Hypothesis :

The model is easier to learn than the policy.

Interest of this approach :

Enable complex policy learning using computer simulation.

Pipeline :

- Generate trajectories
- Use measurements to update model
- Use model to update policy
- Use policy to return to first step

Learning a model : Locally Weighted Bayesian Regression

Locally:

$$\boldsymbol{x}_{t+1} = [\boldsymbol{x}_t, \boldsymbol{u}_t]^T \psi + \boldsymbol{w}$$

Using Bayes' theorem:

$$\mathbb{E}[\boldsymbol{\psi}|\widetilde{X}, \boldsymbol{y}] = S\widetilde{X}B\boldsymbol{\Omega}^{-1}y$$

$$\text{cov}(\boldsymbol{\psi}|\widetilde{X}, \boldsymbol{y}) = S - S^T\widetilde{X}B\boldsymbol{\Omega}^{-1}B\widetilde{X}^TS$$

where $\widetilde{X} = [X, U]$, $\boldsymbol{\Omega} = B\widetilde{X}^TS\widetilde{X}B + \boldsymbol{\Sigma}_w$ and $B = diag(b_i)$.

Finally :

$$\boldsymbol{\mu}_{\boldsymbol{x}}^{t+1} = [\boldsymbol{x}_t, \boldsymbol{u}_t]^T\mathbb{E}[\boldsymbol{\psi}|\widetilde{X}, \boldsymbol{y}]$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}}^{t+1} = [\boldsymbol{x}_t, \boldsymbol{u}_t]^T\text{cov}[\psi|\widetilde{X}, \boldsymbol{y}][\boldsymbol{x}_t, \boldsymbol{u}_t]$$

Learning a model : Gaussian Process Regression

Gaussian prior characteristics :

$$m = 0$$

$$k(\widetilde{\boldsymbol{x}}_p, \widetilde{\boldsymbol{x}}_q) = \sigma_f^2 \exp(-\frac{1}{2}(\widetilde{\boldsymbol{x}}_p - \widetilde{\boldsymbol{x}}_q)^T \Lambda^{-1}(\widetilde{\boldsymbol{x}}_p - \widetilde{\boldsymbol{x}}_q)) + \delta_{pq}\sigma_w^2$$

Prediction :

$\boldsymbol{x}_{t+1}$ is Gaussian distributed :

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{x}_{t+1}|\boldsymbol{\mu}_{t+1}^x, \boldsymbol{\Sigma}_{t+1}^x)$$

where

$$\boldsymbol{\mu}_{t+1}^x = \mathbb{E}_f[f(\boldsymbol{x}_t, \boldsymbol{u}_t)] = \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{y}$$

$$\boldsymbol{\Sigma}_{t+1}^x = var_f[f(\boldsymbol{x}_t, \boldsymbol{u}_t)] = k_{**} - \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{k}_*$$

with $\boldsymbol{k}_* := k(\widetilde{\boldsymbol{X}}, \widetilde{\boldsymbol{x}}_t)$, $k_{**} = k(\widetilde{\boldsymbol{x}}_t, \widetilde{\boldsymbol{x}}_t)$ and $\boldsymbol{K}$ is the kernel matrix with entries $\boldsymbol{K}_{ij} = k(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)$.

Quality of a Policy given a model 1

How to estimate :

$$J_\theta = \sum_{t=0}^{T} \gamma^t \mathbb{E}[r(\boldsymbol{x}_t)|\pi_\theta]$$

Approach based on sampling :

PEGASUS (Policy Evaluation-of-Goodness And Search Using Scenarios)

Deterministic approaches :

Assumption :

$$p(\boldsymbol{x}_t, \boldsymbol{u}_t) = \mathcal{N}([\boldsymbol{x}_t, \boldsymbol{u}_t]|\boldsymbol{\mu}_t^{xu}, \boldsymbol{\Sigma}_t^{xu})$$

Problem to solve :

$$p(\boldsymbol{x}_{t+1}) = \iiint p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) p(\boldsymbol{x}_t, \boldsymbol{u}_t) d\boldsymbol{x}_t d\boldsymbol{u}_t d\boldsymbol{w}_t$$

where $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w}$ with f a non parametric functions.

Hard to solve

Quality of a Policy given a model 2

Hypothesis

$$\mathcal{N}(\boldsymbol{x}_{t+1}|\boldsymbol{\mu}^{x}_{t+1}, \boldsymbol{\Sigma}^{x}_{t+1})$$

How to estimate the predicted distribution parameters?

Moment Matching

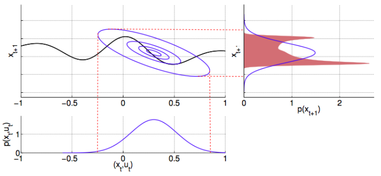

Figure 2: Moment Matching process

Best unimodal distribution

Hard to compute

Quality of a Policy given a model 3

Approximations

Linearization

Locally approximate f around $(\boldsymbol{\mu}_t^x, \boldsymbol{\mu}_t^u)$.

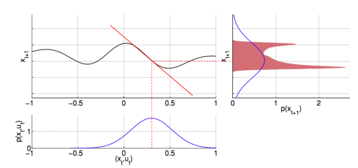

Figure 3: Linearisation Process

Sigma Points

Approximate $p(\boldsymbol{x}_t, \boldsymbol{u}_t)$ by a finite number of points.

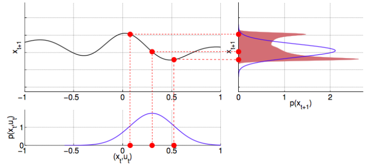

Figure 4: Sigma Point Process

Policy Update

No gradient information

Heuristics such as Nelder-Mead (simplex) or hill-climbing methods (simulated annealing) can be used.

With gradient information

Gradient descent or other popular optimization approach.

How to estimate $dJ_\theta(\theta)/d_\theta$?

Estimation using finite difference

Policy Update

Analytic policy gradient

Advantage

Exact computation of the gradient.

Deterministic model

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) = f(\boldsymbol{x}_t, \pi_\theta(\boldsymbol{x}_t, \boldsymbol{\theta})).$$

$$J_\theta = \sum_t \gamma^t r(\boldsymbol{x}_t)$$

$$\frac{\mathrm{d}J_{\boldsymbol{\theta}}}{\mathrm{d}\boldsymbol{\theta}} = \sum_t \gamma^t \mathrm{d}r(\boldsymbol{x}_t) = \sum_t \gamma^t \frac{\partial r(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}\boldsymbol{\theta}}$$

Using the chain-rule we find that:

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}\boldsymbol{x}_{t-1}}{\mathrm{d}\boldsymbol{\theta}} \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{x}_{t-1}} + \frac{\mathrm{d}\boldsymbol{u}_{t-1}}{\mathrm{d}\boldsymbol{\theta}} \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{u}_{t-1}}$$

Stochastic model

Same with $\mathbb{E}[r(x_t)]$ using $\widetilde{p}(\boldsymbol{x}_t)$ is known.

Conclusion

Thank you for your attention !

Questions?