First ideas

Félicien Hêche

Heig-VD

22 juin 2021

Content

Content

Introduction

Suppose you are a dispatcher and you have to manage an emergency.
Which vehicle should you send ?

- An ambulance and/or a helicopter ?
- Which *precise* ambulance/vehicle ?

Greedy approach : always send the vehicle which could arrive with the smallest among of time.
Is it a good idea ?

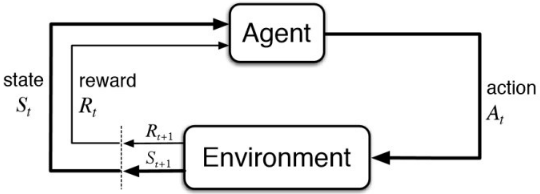
- Send an helicopter for low priority emergencies.
- Obviously not optimal !

How to determine an optimal strategy ?

- Natural framework to attack this problem : reinforcement learning (RL)

Content

RL : general framework



The goal of the agent is to find a policy $\pi(a|s) = \mathbb{P}(a|s)$ which maximizes the *expected return* defined as

$$\mathbb{E}_\pi[\sum_{k\geq 0} \gamma^k R_{t+k+1} | S_t = s, A_t = a]$$

with $\gamma \in [0,1[$ and we call $G_t := \sum_{k\geq 0} \gamma^k R_{t+k+1}$ the return.

RL : general framework

We can also define the *state-value function* V_π as

$$V_\pi(s) = \mathbb{E}_\pi[\sum_{k\geq 0} \gamma^k R_{t+k+1} | S_t = s]$$

where $s \in S$ is a state of the environment.
Therefore we define the *action-value* function Q as

$$Q_\pi(s, a) = \mathbb{E}_\pi[\sum_{k\geq 0} \gamma^k R_{t+k+1} | S_t = s, A_t = a]$$

RL : general framework

There are 3 different categories of RL depending on the possible interactions agent-environment.

- **Online RL.** The agent can interact with environment as he wants.
- **Off-policy RL.** The agent can interact with an environment but through a behaviour policy. He can only observe the result of a behaviour policy with the the environment.
- **Offline RL.** The agent can not interact with the environment. He only has access to a dataset which contains experiments of a behaviour policy.

RL : general framework

How to determine the optimal policy $\pi_\star$?
Two main frameworks to attack real world problem.

- Q-learning
- Actor-critic

The idea of Q-learning is to learn the optimal Q function $Q_{\pi_\star}$.
For example in DQN, they use a neural network to estimate this function.

RL : actor-critic

Idea : Directly parametrize the policy (typically by a neural network)

$$\pi_\theta(s|a) = \mathbb{P}[a|s, \theta]$$

Goal : find θ that maximizes a function $J(\theta)$.
Which function can we choose ? Depending on the context but typically something like

$$J(\theta) = \mathbb{E}_{\pi_\theta}[\sum_{k=1}^{T} \gamma^k R_{t+k}] = \mathbb{E}_{\pi_\theta}[G_t]$$

RL : actor-critic

How to maximize $J(\theta)$?
SGD !
How to compute the gradient ?
Policy gradient Theorem

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log(\pi_\theta(S, A)) Q_{\pi_\theta}(S, A)]$$

RL : actor-critic

How to estimate $Q_{\pi_\theta}(s, a)$?

- Sampling (Mote-Carlo Policy Gradient REINFORCE)
  $\Rightarrow$ High variance
- Estimate this action-value function with a neural network
  $Q_w(s, a) \sim Q_{\pi_\theta}(s, a)$

This leads to the actor-critic algorithm
In each iteration

- Updates action-value function parameters w
- Updates policy parameters θ in direction suggested by the critic.

RL : Application to our problem

Environment : We could model our region with a graph. Each vertex will represent a given area of the region. There will be an edge between two vertices v_1, v_2 if we can drive from v_1 to v_2. Furthermore, on all vertices v_i we could add some features (time, weather condition, temperature, traffic condition, ...).

Actions : At each emergency call the agent (our algorithm) will have to make some actions. Which precise vehicle should he send ? Should he make a strategic move ?

Rewards : We can imagine a lot of different function for the reward. For example

$$R_{t+1} = -\alpha_i t_r - \beta g(a, s) - c(a, s)$$

where i is the level of the emergency's priority, t_r is the time to arrive to the emergency location, α_i, β are hyperparameters, $g(a, s)$ is a binary function which is one if we send an helicopter and zero otherwise and finally $c(a, s)$ is the cost to make the strategic move chosen by the action a.

RL : Application to our problem

Our problem can be seen as a offline RL problem.
But...

- In our problem, will we really want to maximize the *expected* reward ?
- For example if we take $\gamma = 0$ and suppose we are in a environment s. The agent could make two different actions a_1, a_2.
  - With action a_1 he will get a reward of 100 with a probability of 0.8 and -100 with probability 0.2. $\Rightarrow \mathbb{E}[G_t | a_1, s] = 60$
  - With actions a_2 he will get a reward of 50 with probability 1. $\Rightarrow \mathbb{E}[G_t | a_2, s] = 50$
- Thus the optimal policy $\pi_\star$ will choose the action a_1, although it could lead to a potential bad situation with a quite high probability (0.2).
- Is it really a good idea in our problem ?

Content

Risk-sensitive RL : general idea

In risk-sensitive RL the framework is similar to the classical RL.
But instead to try to maximize the expected return, we try to maximize a function

$$\mathcal{D}(G_t)$$

called risk distortion.
The choice of the operator $\mathcal{D}$ depends on the context.
For example

- Exponential utility

$$V := \frac{1}{\beta} \log \left(\mathbb{E}_\pi [e^{\beta G_t}] \right)$$

  With Taylor we have $V = \mathbb{E}[G_t] + \frac{\beta}{2} Var(G_t) + O(\beta^2)$

- Conditional value at risk (CVaR_α)

$$\mathrm{CVaR}_\alpha = \mathbb{E}_\pi [G_t | G_t \leq x_\alpha]$$

  where x_α denotes the α-quantile of G_t. i.e.
  $x_\alpha := \inf\{x \in \mathbb{R} | \alpha \leq F_{G_t}(x)\}$

Risk-sensitive RL : general idea

To resume, our problem is the following : we want to make offline risk-sensitive RL.

Good news :

- In the begin of this year [UCK21] present O-RAAC (Offline Risk Averse Actor-Critic) algorithm which solves this kind of problem.

Risk-sensitive RL : O-RAAC

We define

$$Z_\pi(s, a) :=_D G_t$$

and we denote $\tau \mapsto Z_\pi(s, a; \tau)$ the quantile function of $Z_\pi(s, a)$.
Idea : Use an actor-critic framework.

- The critic will be a neural network $Z_\pi^w(s, a; \tau)$ which approximates $Z_\pi(s, a; \tau)$.
- The actor will be a neural network ϵ_θ which we will use to build the policy π_θ.

Quantile regression

Let X a r.v. with distribution function F and probability density function f.
Suppose that f is continuous and with $\text{supp}(f) = \mathbb{R}$.
We pose x_τ the τ-quantile of X and

$$\rho_\tau(u) = (\tau - \mathbb{1}_{\{u \leq 0\}})u$$

Remark that since f is continuous, we get

$$x_\tau := F^{-1}(\tau)$$

Claim

$$x_\tau = \text{argmin}_q \quad \mathbb{E}[\rho_\tau(X - q)]$$

Proof. First remark that we have

$$\begin{aligned}\frac{\partial}{\partial q}\mathbb{E}[\rho(X-q)] &= \frac{\partial}{\partial q}\int_{\mathbb{R}} \rho_\tau(x-q)f(x)dx \\ &= \int_{\mathbb{R}} \frac{\partial}{\partial q}(\rho_\tau(x-q))f(x)dx\end{aligned}$$

Risk-sensitive RL : O-RAAC

and then

$$\begin{aligned}\frac{\partial}{\partial q}\mathbb{E}[\rho_\tau(X-q)] &= -\int_{-\infty}^{q}(\tau-1)f(x)dx - \int_{q}^{+\infty}\tau f(x)dx \\ &= \int_{\infty}^{q} f(x)dx - \int_{\infty}^{q}\tau f(x)dx - \int_{q}^{+\infty}\tau f(x)dx \\ &= F(q)-\tau = 0 \Rightarrow q = F^{-1}(\tau)\end{aligned}$$

And since

$$\frac{\partial^2}{\partial^2 q}\mathbb{E}[\rho_\tau(X-q)] = \frac{\partial}{\partial q}F(q)-\tau = f(q) > 0$$

we get as expected

$$x_\tau = \mathrm{argmin}_q \quad \mathbb{E}[\rho_\tau(X-q)]$$

Critic loss How can we learn the quantile function ?
First remark that

$$Z_\pi(s,a) =_D R(s,a) + \gamma Z_\pi(S',A')$$

Hence for a sampling (s,a,r,s',a') we can define the TD-error

$$\delta_{\tau,\tau'} = R(s,a) + \gamma Z_\pi^{w'}(s',a',\tau') - Z_\pi^{w}(s,a,\tau)$$

Risk-sensitive RL : O-RAAC

Moreover we define the τ-quantile Huber-loss

$$\mathcal{L}_k(\delta,\tau) = |\tau - \mathbb{1}_{\{\delta<0\}}| \cdot \begin{cases} \frac{1}{2k}\delta^2 & \text{if } |\delta| \leq k \\ |\delta| - 2k & \text{otherwise} \end{cases}$$

With this function, we can define the critic loss

$$\mathcal{L}_{\text{critic}}(w) = \mathbb{E}_{\substack{(s,a,r,s')\sim d^\beta \\ a'\sim\pi_\theta(\cdot|s')}}[\frac{1}{NN'}\sum_{i=1}^{N}\sum_{j=1}^{N'}\mathcal{L}_k(\delta_{\tau_i,\tau'_j};\tau_i)]$$

Risk-sensitive RL : O-RAAC

Actor-loss

$$\mathcal{L}_{actor} = \mathbb{E}_{s\sim d^b(\cdot)}[\mathcal{D}(Z^w_{\pi_\theta}(s,\pi_\theta(s),\tau))]$$

Remark

$$\begin{aligned}\mathcal{D}(Z^w_{\pi_\theta}(s,\pi_\theta(s),\tau)) &= \int Z^w_{\pi_\theta}(s,\pi_\theta(s),\tau)\mathbb{P}_{\mathcal{D}}(\tau)d\tau \\ &\simeq \frac{1}{K}\sum_{k=1}^{K} Z^w_{\pi_\theta}(s,\pi_\theta(s),\tau_k)\end{aligned}$$

Acerbi's formula

$$\mathrm{CVaR}_\alpha(Z^w_\pi(s,\pi(s),\tau)) = \frac{1}{\alpha}\int_0^\alpha Z^w_\pi(s,a,\tau)d\tau$$

Offline : controlling the bootstrap error.

In the offline setting the bootstrapping error appears : when evaluating the TD-error, the Z-value target will be evaluated at actions where there is no data.

How can we manage this problem ?

- A few different ideas have been introduced to manage the error of the imitation policy.
- But with O-RAAC : learn a generative model (VAE, GAN) π^{IL} which imitates the behaviour policy.

Finally we pose

$$\pi_\theta(s) = b + \lambda\epsilon_\theta(\cdot|s, b) \qquad \text{such that } b \sim \pi^{IL}(\cdot|s)$$

where ϵ_θ is a neural network trained with the actor-loss and λ an hyperparameter.

Algorithm 1 O-RAAC

Input : Data set, Critic Z_w and critic-target $Z_{w'}$, VAE$_\phi$ Perturbation model ϵ, modulation parameter λ , Distortion operator $\mathcal{D}$ or distortion sampling distribution $\mathbb{P}_{\mathcal{D}}$, critic-loss parameters N, N', k, mini-batch size B, learning rate η, soft update parameter μ.

for $t = 1, \dots$ **do**

Sample B transitions (s, a, r, s') from dataset.

Sample N quantiles τ and N' quantile τ' from $\mathcal{U}(0, 1)$ and compute $\delta_{\tau,\tau'}$

Compute policy $\pi = b + \lambda\epsilon(s, b)$ such that $b \sim \text{VAE}(s, a)$.

Compute critic loss $\mathcal{L}_{\text{critic}}(w)$, actor loss $\mathcal{L}_{\text{actor}}(\theta)$, VAE loss $\mathcal{L}_{\text{VAE}}(\phi)$.

Gradient state : $w \leftarrow w - \eta\nabla\mathcal{L}_{\text{critic}}(w)$, $\theta \leftarrow \theta - \eta\nabla\mathcal{L}_{\text{actor}}(\theta)$, $\phi \leftarrow \phi - \eta\nabla\mathcal{L}_{\text{VAE}}(\phi)$.

Perform soft-update on $w' \leftarrow \mu w + (1 - \mu)w'$.

end for

Risk-sensitive RL : O-RAAC

Did we solve our problem ?
O-RACC is tested on datasets provided by OpenAI.
For example Half Cheetah dataset has the following parameters.

- Action space $\mathbb{R}^6$.
- Environment space $\mathbb{R}^{17}$.
- Millions of samples.

In our problem :

- Environment space is huge !
- Not a lot of samples.
- A lot of state-action will be unobservable.

Content

Is this hopeless ?

- No fully confident answer to this question.
- But we believe we could solve it.
- [NKL18] solves almost the same problem with less data.

Open questions

In [NKL18] they use Multi-agent RL to solve a problem similar to our : they try to determine the optimal real-time location of the police patrol to minimize the response time to an emergency.

- Each patrol is seen as a agent which has access only at a local observation.
- Data : time and zone of incident (24 possibilities) for 31 days.
- Not risk-sensitive.

This paper leads to an idea to solve our problem :

- Reduce the size of the environment and using a multi-agent approach. Each vehicle would have access at a local representation of the environment.
- Limitation : not an optimal uses of all data.

Does it really help ?

Open questions

How to reduce the environment space ?

Observation : we are close to the few-shot learning problem.
Idea : use methods develop for few-shot learning.
For example : use a neural network ϕ_θ to encode the environment states s.
After that apply O-RAAC algorithm to $\phi_\theta(S)$ environment.
How to learn ϕ_θ ?

- Idea : two states which lead to the same action will be encode closely.
- For a state s take two others action s_+ and s_- where s_+ will lead to the same action a as s and s_- to another. After that we define the loss as

$$\mathcal{L}(s, s_+, s_-) = \max\{\|\phi_\theta(s) - \phi_\theta(s_+)\|^2 - \|\phi_\theta(s) - \phi_\theta(s_-)\|^2 + \alpha, 0\}$$

Open questions

How to add samples in our dataset ?

We could add synthetic data to our dataset.
Emergencies are independent of our policy !
But it could be costly to build and some errors would be unavoidable.
How to deal with ?

- Generative Teaching Network ([SRL+20]) ?
- Could we determine a method to add data only to *important* (s, a, r, s') cases ?
- Core-set selection : This problem considers a fully labeled dataset and tries to choose a subset of it such that the model trained on the selected subset will perform as closely as possible to the model trained on the entire dataset.

Open questions

Other ideas.

- Transfer learning ?
- ...

Bibliographie

Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau.
Credit assignment for collective multiagent rl with global rewards.
2018.

Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune.
Generative teaching networks : Accelerating neural architecture search by learning to generate synthetic training data.
In *International Conference on Machine Learning*, pages 9206–9216. PMLR, 2020.

Núria Armengol Urpí, Sebastian Curi, and Andreas Krause.
Risk-averse offline reinforcement learning.
arXiv preprint arXiv :2102.05371, 2021.