From Policy Gradient to Actor-Critic methods
The Policy Search problem

Olivier Sigaud

Sorbonne Université
http://people.isir.upmc.fr/sigaud
Example: a (cheap) tennis ball collector

- A robot without a ball sensor
- Travels on a tennis court based on a parametrized controller
- Performance: number of balls collected in a given time
- Just depends on robot trajectories and ball positions
Influence of policy parameters

- Controller parameters: proba of turn per time step, travelling speed
- How do the parameters influence the performance?
- Policy search: find the optimal policy parameters
Two sources of stochasticity

- From the environment: position of the balls
- From the policy, if it is stochastic
- The performance can vary a lot $\rightarrow$ need to repeat
- Tuning parameters can be hard
The policy search problem: formalization

- $\tau_i$ is a robot trajectory
- $R(\tau_i)$ is the corresponding return
- $\pi_\theta$ is the parametrized policy of the robot

- We want to optimize $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$, the global utility function
- We tune policy parameters $\theta$, thus the goal is to find

$$\theta^* = \underset{\theta}{\text{argmax}} J(\theta) = \underset{\theta}{\text{argmax}} \sum_{\tau} P(\tau|\theta) R(\tau) \quad (1)$$

- where $P(\tau|\theta)$ is the probability of trajectory $\tau$ under policy $\pi_\theta$

Direct Policy Search is black box optimization

- $J(\theta)$ is the performance over policy parameters
- Choose a $\theta$
- Generate trajectories $\tau_{\theta}$
- Get the return $J(\theta)$ of these trajectories
- Look for a better $\theta$, repeat

- DPS uses $(\theta, J(\theta))$ pairs and directly looks for $\theta$ with the highest $J(\theta)$
From Policy Gradient to Actor-Critic methods

Policy improvement

(Truly) Random Search

Select $\theta_i$ randomly
Evaluate $J(\theta_i)$
If $J(\theta_i)$ is the best so far, keep $\theta_i$
Loop until $J(\theta_i) > \text{target}$

- Of course, this is not efficient if the space of $\theta$ is large
- General “blind” algorithm, no assumption on $J(\theta)$
- We can do better if $J(\theta)$ shows some local regularity

Direct policy search

- Locality assumption: The function is locally smooth, good solutions are close to each other

Variation - selection: Perform well chosen variations, evaluate them

Variations generally controlled using a multivariate Gaussian
**Gradient ascent**: Following the gradient from analytical knowledge

- **Issue**: in general, the function $J(\theta)$ is unknown
- **How can we apply gradient ascent without knowing the function?**
- **The answer is the Policy Gradient Theorem**
- **Next lessons**: Policy Gradient methods
Any question?

Send mail to: Olivier.Sigaud@upmc.fr
References

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al.
A survey on policy search for robotics.

Olivier Sigaud and Freek Stulp.
Policy search in continuous action domains: an overview.