Meeting Tesi

Germano Gabbianelli

07/11/2016

Problema

batch learning. LQG

$$\begin{aligned} S_{t+1} &= S_t + a_t + \epsilon \\ R_t &= -0.5 \cdot \left(S_t^2 + a_t^2\right) \\ \epsilon &\sim \mathcal{N}(0, 0.1) \\ \pi_*(S_t) &= -0.608 \cdot S_t \end{aligned}$$

Fitted Q iteration

Vogliamo valutare le differenze di performance tra Fitted Q iteration e il nostro algoritmo.

Lo svantaggio principale di Fitted Q iteration è la necessità di svolgere un passo di training ad ogni iterazione dell'algoritmo.

Nuovo approccio

Fissiamo uno spazio funzionale per la nostra Q:

$$\begin{aligned} Q_\theta(s,a) &= (a - ks)^2 + b \\ \theta &= [k, b] \end{aligned}$$

Vogliamo ora ottenere una black box f_ρ che dato in ingresso θ_i produca in output θ_{i+1} tale che

$$Q_{\theta_{i+1}}(s,a) \approx r(s,a) + \gamma \max_a Q_{\theta_i}(s,a)$$

Metodi per stimare ρ

- Natural Evolutional Strategy (NES)
- Policy Gradients with Parameter Exploration (PGPE)

pyBrain sembra avere le implementazioni per questi e altri metodi black box già incluse.

Metrica di valutazione

Usiamo Bellman residual come metrica di valutazione

$$e = \left[Q_{\theta_{i+1}}(s,a) - \left(r(s,a) + \gamma \max_a Q_{\theta_i}(s,a) \right) \right]^2$$

Nel paper sul Fitted Q iteration, testano le performance su un set di stati iniziali S^i, per tutte le azioni in U e ne calcolano la media.