Démarche

Mehdi Mounsif

10 avril 2018

1 Récap : The day before

- Finalisation de $\mathbb{A}^\star$et tests
- Rédaction d'un rapport pour le Gang avec les résultats de $\mathbb{A}^\star$et propositions (CNN, LSTM) pour la suite
- Lecture de [1]
- Préparation de [2]
- Lecture de : Probabilistic Programming and Bayesian Methods for Hackers

2 GAIL

Premiers tests (concluants) sur CartPole. Un expert (un agent PPO) fournit un certain nombre de trajectoires. Celles-co sont transformées en tenseurs, et passée dans une classe spéciale pour le chargement de dataset. L'algorithme fonctionne ainsi :

1. L'agent mime joue un épisode et observe les récompenses données par la fonction $\mathbb{D}$
2. A la fin de l'épisode, $\mathbb{D}$ doit minimiser

$$\mathcal{L}_D = BCE(\mathbb{D}(\tau_\pi), 1) + BCE(\mathbb{D}(\tau_E), 0)$$

3. Amélioration de l'agent mime *via* PPO

La structure fonctionne et l'algorithme converge. A noter cependant que comme la récompense de CartPole est toujours 1, il suffit de donner une récompense $r \geq 0$ pour pousser l'agent dans le bon sens (l'environnement est trop trivial). En revanche, on remarque que D affecte une récompense plus basse lorsque l'agent s'éloigne du centre et lorsque le bâton s'écarte de la vertical. C'est un signe encourageant. Pour la suite, l'environnement Reacher a été modifié. 8 actions sont désormais possibles (les 8 directions du stick directionnel) et la fonction `step_gail_discrete()` retourne une liste avec :

- La meilleure action (celle choisie par l'expert)
- Le vecteur d'observation

On se servira de cet environnement pour tester GAIL dans un environnement moins trivial. En cas de succès, étendre le concept à Reacher avec régression sur les angles (différences avec SuperBot ?)

Références

[1] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[2] Ho, J., and Ermon, S. Generative adversarial imitation learning. *CoRR abs/1606.03476* (2016).