Démarche

Mehdi Mounsif

26 février 2018

J'ai étudié le rapport de Matthew Alger [2] au sujet de l'apprentissage par renforcement inverse. En substance, IRL (Inverse Reinforcement Learning) vise à déterminer la fonction de récompense de l'environnement à partir de l'observation des trajectoires optimales $\pi^\star$ fournies par un agent expert [3]. Cette approche a plusieurs avantages :

- On évite ainsi de devoir définir une fonction de récompense qui peut être complexe.
- Les politiques générées de cette manière plutôt que *via* l'évolution du comportement, peuvent être plus robustes et mieux s'adapter aux perturbations de l'environnement [1]
- D'un point de vue de modélisation, il peut être intéressant de découvrir la fonction de récompense qui explique le comportement d'un agent

Par exemple, dans le cas de la conduite, comment déterminer les compromis entre maintien de la trajectoire, respect de la distance de sécurité, conserver une vitesse raisonnable... ? Dans [1], on mentionne des tentatives d'apprentissages supervisé qui, bien qu'efficace dans les cas décrits par les données d'entrainements, sont inadaptés aux perturbations de l'environnement.

Considérons les notations suivante :

- MDP composé de (S, A, T, γ, D, R). Où
  - S représente les états
  - A les actions disponibles
  - T les probabilités de transitions $T = \{P_{s,a}\}$
  - γ facteur discount
  - D Distribution des états initiaux
  - R fonction de récompense, dont la valeur absolue est bornée par 1
- MDPwR : MDP sans fonction de récompense
- Admettons qu'il existe $\phi : S \to [0,1]^k$ un vecteur de *features*
- Et une *vraie* fonction de récompense $R^\star = w^\star \cdot \phi(s)$

Toujours dans le cas de la conduite, ϕ pourrait correspondre aux différents paramètres que le cherche à pondérer. Ainsi, $w^\star$ serait précisément cette pondération. On peut alors écrire la valeur d'une politique π

$$\mathbb{E}_{s_0 \sim D} = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi]$$

$$\mathbb{E}_{s_0 \sim D} = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t w \cdot \phi(s_t) | \pi]$$

$$\mathbb{E}_{s_0 \sim D} = w\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi]$$

On définit alors *expected discounted accumulated feature value vector* $\mu(\pi)$, aussi nommé **features expectations** :

$$\mu(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi] \in \mathbb{R}^k$$

Avec cette formulation, $\mu(\pi)$ détermine complètement la valeur de la politique. Prenons l'exemple suivant : Soit Π un jeu de politique pour un MDP donné. Soient $\pi_1, \pi_2 \in \Pi$, on peut construire π_3 en mélangeant les deux politiques précédente. Si π_3 agit choisit d'agir suivant π_1 avec une probabilité λ et suivant π_2 avec une probabilité $1 - \lambda$ alors clairement :

$$\mu(\pi_3) = \lambda\mu(\pi - 1) + (1 - \lambda)\mu(\pi_2)$$

A noter que la sélection entre π_1 et π_2 ne s'effectue qu'au début de la trajectoire, et non pas à chaque pas. Ainsi, plus généralement, si on a une collection de politiques $\pi_1, \pi_2, ..., \pi_n$, alors on peut déterminer une nouvelle politique dont l'expectation value est donnée par un vecteur représentant les probabilités de choisir chacune des politiques.

Admettons maintenant qu'on dispose des démonstrations d'un expert π_E (i.e : les observations des états et les actions). On peut calculer l'estimation des **features expectations** de l'expert à partir des trajectoires $\{s_0^{(i)}, s_1^{(i)}, ..., s_n^{(i)}\}_{i=1}^m$

$$\hat{\mu_E} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)})$$

Le problème le plus proéminent concernant l'apprentissage inverse est la multiplicité des fonctions de récompenses qui pourraient expliquer une politique optimale. En particulier, la fonction $R(s) = const$ rend toutes les politiques π optimales. En pratique, il existe plusieurs approches qui visent à découvrir $R(s)$. Parmi lesquelles :

- Programmation linéaire
- Utilisation de l'entropie maximum [5]
- Son équivalent ANN : Maximum Entropy Deep Inverse Reinforcement Learning [4]

***

***

Pour l'entropie maximum, une méthode de correspondance est mise en place entre les trajectoires observées et la trajectoire optimale. Intuitivement :

Soit une politique optimale π_o pour une fonction de récompense découverte (recovered). Alors, en moyenne, π_o génère les même trajectoires que la politique optimale $\pi^\star$ pour la vraie fonction de récompense.

On peut ainsi découvrir, grâce aux algorithmes classiques d'apprentissage par renforcement, une approximation de la fonction de récompense qui peut ensuite être utilisée pour déduire une politique similaire à la politique optimale. Pour incorporer l'entropie maximum, il faut reformuler le problème de manière probabilistique : Soit une distribution de trajectoires sur le MDP (environnement). Ce dernier étant un environnement stochastique, il peut exister plusieurs chemins qui répondent aux critères de correspondance. L'entropie maximum explique que les chemins les plus rentables doivent être exponentiellement plus susceptibles d'être sélectionés :

$$P(\Psi) = \frac{1}{Z} e^{R(\Psi)}$$

Où Z est nommée fonction de partition et joue le rôle d'une constante de normalisation. L'approche probabiliste permet de gérer le bruit sur les trajectoires et donc augmente la robustesse de la fonction de récompense.

*** Approche avec approximation de fonctions

Un des points faible de la précédente formulation avec entropie maximum est l'hypothèse que la fonction de récompense peut être représentée linéairement.

$$R(s) = \alpha \cdot \Phi(s)$$

Egalement, l'aspect **features count** est ambigu et la problématique de la suboptimalité est irrésolue.

Références

[1] Abbeel, P., and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. 1–.

[2] Alger, M. Inverse reinforcement learning, 2016.

[3] Russell, S. Learning agents for uncertain environments (extended abstract). 101–103.

[4] Wulfmeier, M., Ondruska, P., and Posner, I. Maximum entropy deep inverse reinforcement learning. *CoRR abs/1507.04888* (2015).

[5] Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. 1433–1438.