Apprenticeship Learning via Inverse Reinforcement Learning

- Key insight : Instead of learning the policy from the "expert", it learns a reward function

- Preliminaries :

  S: states, A: actions, $T = \{P_{sa}\}$ transition probability, $\gamma \in [0,1)$ discount factor,

  D: initial-state distribution, $R: S \mapsto A$ reward function ≤ 1. MDP \ R (without a reward function)

  ϕ: {features} $S \to [0,1]^k$, $R^*(s) = w^* \cdot \phi(s)$, $w^* \in \mathbb{R}^k$ true reward function

  π : policy $S \mapsto D_A$, $\mathbb{E}_{s \sim D}[V^\pi(s_0)] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t w \cdot \phi(s_t) | \pi] = w \cdot \underbrace{\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi]}_{\mu(\pi)}$

  $= w \cdot \mu(\pi)$

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

  Note, R is a linear combination of ϕ.

  - Let $\pi_1, \pi_2 \in \Pi$, π_3 be $\pi_1 + \pi_2$ with $P_r(\pi_1) = \lambda$, $P_r(\pi_2) = 1 - \lambda$. $\Rightarrow \mu(\pi_3) = \lambda \mu(\pi_1) + (1-\lambda)\mu(\pi_2)$
  - Generally, $\pi_1, \dots \pi_d \in \pi$, $\to$ convex combination $\sum_{i=1}^{n} \lambda_i \mu(\pi_i)$, $(\lambda_i \geq 0, \sum_i \lambda_i = 1)$
  - π_E : "expert" policy, can be viewed as optimal $R^* = w^{*T} \phi$
  - Estimator : $\mu_E = \mu(\pi_E)$, m trajectories $\{s_0^{(i)}, s_1^{(i)}, \dots\}_{i=1}^{m}$ generated by expert

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

- Algorithm:

  Goal : find $\tilde{\pi}$ s.t. $\|\mu(\tilde{\pi}) - \mu_E\|_2 \leq \varepsilon$

  $$\left| \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi_E] - \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \tilde{\pi}] \right|$$

  $$= |w^T \mu(\tilde{\pi}) - w^T \mu_E| \leq \|w\|_2 \, \|\mu(\tilde{\pi}) - \mu_E\|_2 \leq 1 \cdot \varepsilon = \varepsilon$$

  Problem turns into finding a policy $\tilde{\pi}$ that makes $\mu(\tilde{\pi})$ close to μ_E

  Steps: (QP-based)

  1. Randomly pick $\pi^{(0)}$, compute $\mu^{(0)} = \mu(\pi^{(0)})$, set $i = 1$.
  2. Compute $t^{(i)} = \max_{w: \|w\|_2 \leq 1} \min_{j \in \{0..(i-1)\}} w^T(\mu_E - \mu^{(i)})$, let $w^{(i)}$ be the max
  3. If $t^{(i)} \leq \varepsilon$, terminate
  4. Compute the optimal policy $\pi^{(i)}$ using $R = (w^{(i)})^T \phi$
  5. Compute $\mu^{(i)} = \mu(\pi^{(i)})$
  6. Set $i = i+1$, go to step 2

My comment, this is kinda a trial & error algorithm

Projection-based method: replace step (2) by

set $\bar{\mu}^{(i-1)} = \bar{\mu}^{(i-2)} + \frac{(\mu^{(i-1)} - \bar{\mu}^{(i-2)})^T (\mu_E - \bar{\mu}^{(i-2)})}{(\mu^{(i-1)} - \bar{\mu}^{(i-2)})^T (\mu^{(i-1)} - \bar{\mu}^{(i-2)})} (\mu^{(i-1)} - \bar{\mu}^{(i-2)})$

set $w^{(i)} = \mu_E - \bar{\mu}^{(i-1)}$, $t^{(i)} = \|\mu_E - \bar{\mu}^{(i-1)}\|_2$

- Theoretical results
  - Theorem 1: Let an MDP\R, features $\phi : S \mapsto [0,1]^k$. and any $\varepsilon > 0$ be given. Then the apprenticeship learning algorithm will terminate with $t^{(i)} \leq \varepsilon$ after at most

    $$n = O\left(\frac{k}{(1-\gamma)^2 \varepsilon^2} \log \frac{k}{(1-\gamma)\varepsilon}\right)$$ iterations.

  - Theorem 2: lower bound for m, (samples required).

    $$m \geq \frac{2k}{(\varepsilon(1-\gamma))^2} \log \frac{2k}{\delta}.$$

Proofs are skipped.