Algorithms for Inverse Reinforcement Learning

池田 春之介

Contents

- Abstract
- 背景と目的
- 逆強化学習(IRL)
- アルゴリズムの説明
- 結論と今後の取り組み

Abstract

- MDPにおける逆強化学習問題を扱う
- 3つのアルゴリズムについて説明
  1. 有限な状態空間におけるテーブル形式の報酬関数表現
  2. 無限な状態空間での報酬関数の線形近似
  3. 観測されたサンプル集合でのみをもちいる現実的な場合
- “degeneracy”が問題になる

degeneracyとは

観測された方策が最適となるような報酬関数の集合が大きいこと

最適な方策と他の方策の区別が最大限にできるような報酬関数を求める

背景と目的

- 動物や人間の学習を計算モデルとして使うことと強化学習の関連性
  - 蜂の採餌などで強化学習が起きていることが証明されている
  - 報酬関数が既知で固定されていると仮定(ex. 蜜の飽和度)

⇒ 人間や動物の行動を調べる際には，経験に基づいて報酬関数を考慮すべきではないか

- 特定のドメインにおいて，優れた振る舞いのできる知的エージェントを構成
  - エージェントの設計者は報酬関数の最適化が所望の振る舞いをするという大雑把な考えのため，使えない可能性 (ex. 運転)

⇒ エキスパートの報酬関数を模倣する

逆強化学習とは

強化学習 (RL)

目的　報酬rをもとに計算される価値関数V^{π} or Q^{π}を最大にするような最適方策 π^{*}を見つける

$$\pi(s) \in \underset{a \in A}{\operatorname{argmax}} Q^{\pi}(s, a)$$

逆強化学習 (IRL)

目的　方策π によって生成される系列に基づいて、最適な報酬関数を推定すること

条件

- エージェントの振る舞いが観測できる
- 必要ならば，エージェントに対しての知覚的な入力
- 必要ならば，環境のモデル

MDPの基本的な特性

- Theorem 1 (Bellman Equations)

  MDP $M = (S, A, \{P_{sa}\}, \gamma, R)$ と方策 $\pi : S \longmapsto A$ を所与とすると

  $\forall s \in S,\ \forall a \in A$ で以下が成立

$$V^{\pi}(s) = R(s) + \gamma \sum_{s'} P_{s\pi(s)}(s') V^{\pi}(s') \tag{1}$$

$$Q^{\pi}(s,a) = R(s) + \gamma \sum_{s'} P_{sa}(s') V^{\pi}(s') \tag{2}$$

- Theorem 2 (Bellman Optimality)

  方策 π が M で最適である

$$\Longleftrightarrow \pi(s) \in \underset{a \in A}{\operatorname{argmax}} Q^{\pi}(s,a) \tag{3}$$

逆強化学習

解集合の特徴づけ (有限な状態空間)

- Theorem 3

有限状態空間 S, 行動の集合 $A = \{a_1, \ldots, a_k\}$, 遷移確率行列$\{P_{sa}\}$, 割引率γ は所与とする. $\pi(s) \equiv a_1$ を最適方策とすると,$a = a_2, \ldots, a_k$ で報酬$\boldsymbol{R}$ は以下をみたす.

$$(\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1}\boldsymbol{R} \succeq 0 \qquad (4)$$

[Proof]

$\pi(s) \equiv a_1$ より (1)式は $\boldsymbol{V}^\pi = \boldsymbol{R} + \gamma \boldsymbol{P}_{a1}\boldsymbol{V}^\pi$ と書き換えられるので

$$\boldsymbol{V}^\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_{a1})^{-1}\boldsymbol{R} \qquad (5)$$

(2)式の代わりにTheorem2から(3)式を用いると,$\pi(s) \equiv a_1$が最適方策

$$\Longleftrightarrow a_1 \equiv \pi(s) \in \operatorname*{argmax}_{a \in A} \sum_{s'} P_{sa}(s')V^\pi(s') \quad \forall s \in S, \forall a \in A$$

$$\Longleftrightarrow \sum_{s'} P_{sa_1}(s')V^\pi(s') \geq \sum_{s'} P_{sa}(s')V^\pi(s') \quad \forall a \in A \setminus a_1$$

$$\Longleftrightarrow \boldsymbol{P}_{a1}\boldsymbol{V}^\pi \succeq \boldsymbol{P}_a\boldsymbol{V}^\pi \quad \forall a \in A \setminus a_1$$

(5)式を代入して

逆強化学習の問題点

- 前スライドの定理はIRLの解となるすべての報酬関数の集合を特徴づける

2つの問題が生じる

1. $\boldsymbol{R} = \mathbf{0}$(実際は定数)が常に解になる

   もし報酬が行動によらず同じなら、最適方策を含むどんな方策も最適となる

   $\pi(s) \equiv a_1$ が唯一の最適方策ならば問題は緩和(満足ではない)

2. (4)式をみたす報酬関数$\boldsymbol{R}$は数多く存在する

線形計画法(LP)定式化とペナルティ項

アルゴリズム1

LP定式化とペナルティ項

- LP定式化
  - (4)式は制約条件
  - πを最適とし、他の方策との差異をできるだけ大きくする

最大化 →

$$\sum_{s\in S}\left(Q^{\pi}(s,a_1)-\max_{a\in A\setminus a_1}Q^{\pi}(s,a)\right) \tag{6}$$

- ペナルティ項
  - 報酬が小さい解の方が単純で望まれる
  - L1ノルム：$-\lambda||R||_1$

定式化 ←

最適化問題

$\boldsymbol{P}_a$のi行目

$$\text{maximize} \sum_{i=1}^{N}\min_{a\in\{a_2,\ldots,a_k\}}\{(\boldsymbol{P}_{a_1}(i)-\boldsymbol{P}_a(i))(\boldsymbol{I}-\gamma\boldsymbol{P}_{a_1})^{-1}\boldsymbol{R}\}-\lambda\|\boldsymbol{R}\|_1$$

$$\text{s.t. } (\boldsymbol{P}_{a_1}-\boldsymbol{P}_a)(\boldsymbol{I}-\gamma\boldsymbol{P}_{a_1})^{-1}\boldsymbol{R}\succeq 0,\ |\boldsymbol{R}_i|\le R_{max}\ \ i=1,\ldots,N$$

アルゴリズム2

線形関数近似(大規模な状態空間)

大規模な状態空間では報酬関数を直接計算して求めることが困難なため,
以下のように近似する

$$R(s) = \alpha_1\phi_1(s) + \alpha_2\phi_2(s) + \cdots + \alpha_d\phi_d(s) \quad (8)$$

$\phi_1, \ldots, \phi_d$ は基底関数, $\alpha_1, \ldots, \alpha_d$ は求めたい未知のパラメータ

期待値の線形性より，(8)式は価値関数の定義から
$(V^\pi(s_1) = \mathrm{E}[R(s_1) + \gamma R(s_2) + \gamma^2 R(s_3) + \cdots | \pi])$

$$V^\pi = \alpha_1 V_1^\pi + \alpha_2 V_2^\pi + \cdots + \alpha_d V_d^\pi \quad (9)$$

これとTheorem2を用いると(4)式は次のようになる

$$\mathrm{E}_{s' \sim P_{sa1}}[V^\pi(s')] \geq \mathrm{E}_{s' \sim P_{sa}}[V^\pi(s')] \quad (10)$$

アルゴリズム2

線形関数近似 → 2つの問題点が生じる

1. 大規模な状態空間では(10)式の制約が多すぎる
   - 大きいが有限な部分集合 S_0でのみ制約を考える
2. 報酬関数が線形近似できない可能性がある

定式化

最適化問題

$$\text{maximize} \quad \sum_{i=1}^{N} \min_{a \in \{a_2, \ldots, a_k\}} \{p(\mathrm{E}_{s' \sim P_{sa1}}[V^{\pi}(s')] - \mathrm{E}_{s' \sim P_{sa}}[V^{\pi}(s')])\}$$

$$\text{s.t.} \quad |\alpha_i| \leq 1, \ i = 1, \ldots, d$$

$$p(x) = \begin{cases} x & \text{if } x \geq 0 \\ 2x & \text{otherwise} \end{cases}$$

アルゴリズム3

アルゴリズム1，2では遷移確率などのモデルが必要であった

より現実的に

サンプル系列から報酬関数を推定する

仮定

- 初期状態の分布 D を固定する
- 最適方策やそれ以外の方策によってMDPにおけるサンプルを生成できる

目標

方策 π が $\mathrm{E}_{s_0 \sim D}[V^{\pi}(s_0)]$ を最大化するような報酬関数 R を見つける

アルゴリズム3

手順

方策 π のもとでm-モンテカルロでサンプルを生成

m-モンテカルロから得られた収益によって $\hat{V}_i^{\pi}$ を定義 $i = 1, \ldots, d$

$\hat{V}^{\pi}(s_0)$ は次式のように表現できる

$$\hat{V}^{\pi}(s_0) = \alpha_1 \hat{V}_1^{\pi}(s_0) + \alpha_2 \hat{V}_2^{\pi}(s_0) + \cdots + \alpha_d \hat{V}_d^{\pi}(s_0) \quad (11)$$

(12)式をみたすように，求める

$$V^{\pi^*}(s_0) \geq V^{\pi_i}(s_0) \quad i = 1, \ldots, k \quad (12)$$

定式化

最適化問題

$$\text{maximize} \quad \sum_{i=1}^{k} p\left(V^{\pi^*}(s_0) - V^{\pi_i}(s_0)\right)$$

$$\text{s.t.} \quad |\alpha_i| \leq 1, \quad i = 1, \ldots, d$$

$$p(x) = \begin{cases} x & \text{if } x \geq 0 \\ 2x & \text{otherwise} \end{cases}$$

結論と今後の取り組み

結論

- 少なくとも状態空間の大きさがそれほどでもない離散や連続な領域では、逆強化学習は解くことができる

今後の取り組み

- Potential-based shaping rewards(Ng et al., 1999)は最適性を好んで用いずに解を学習することを劇的に容易にした報酬関数を生みだした

  ⇒ より簡潔な報酬関数を取り出すIRLアルゴリズムを設計する

- 実問題では、ノイズが混入したり、最適な方策の一部しか所与でない場合がある ⇒ どのような評価指標を用いるか

- もし振る舞いが最適性とかけ離れている場合，状態空間の特定の部分に対して部分的な報酬関数を特定する

- 部分的に環境が観測できる場合に，提案したアルゴリズムがどの程度うまくいくか