エントロピー RL まとめ

北村俊徳

May 9, 2023

目次

このスライドで説明する内容

いろんな価値反復法の解説

1. Value Iteration (VI)
2. Soft Value Iteration (SVI)[Haarnoja et al., 2017]
3. Advantage learning (AL)[Bellemare et al., 2016]
4. Dynamic Policy Programming (DPP)[Azar et al., 2012]
5. Conservative Value Iteration (CVI)[Kozuno et al., 2019]

を解説するよ.
後で説明するように CVI は他のアルゴリズムの一般形になっているため, CVI についての解析で他のアルゴリズムの特徴を説明するよ.

表記 I

- $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$:
  状態空間, 行動空間, 遷移確率, 方策関数, 割引率
- 行動価値関数: $Q^\pi(s,a) := \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \mid S_0 = s, A_0 = a \right]$
- 状態価値関数: $V^\pi(s) := \mathbb{E}^\pi \left[Q^\pi(s, A) \right]$
- ベルマン方程式:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) V^\pi\left(s'\right) \right) \tag{1}$$

- 方策 π のアドバンテージ: $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$

表記 II

式を簡単化するため, 期待値は行列形式で表す.
例えば以下の $\boldsymbol{P}$ は $|\mathcal{S}||\mathcal{A}| \times \mathcal{S}$ の行列.

- 状態遷移についての期待値: $(\boldsymbol{P}V)(s,a) := \sum_{s' \in \mathcal{S}} P(s' \mid s,a)\, V(s')$
- 方策についての期待値: $(\boldsymbol{\pi} Q)(s) := \sum_{a \in \mathcal{A}} \pi(a \mid s) Q(s,a)$

また, max 作用素については以下のように表す:

- 普通の max: $(\boldsymbol{m}Q)(s) := \max_{a \in \mathcal{A}} Q(s,a)$
- mellow-max[1]: $\left(\boldsymbol{m}_\beta Q\right)(s) := \frac{1}{\beta} \log \left(\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \exp(\beta Q(s,a)) \right)$

簡易化したベルマン作用素

- ベルマン最適作用素: $\boldsymbol{T} : Q \in \mathcal{Q} \mapsto r + \gamma \boldsymbol{P}\boldsymbol{m}Q \in \mathcal{Q}$
- soft ベルマン最適作用素: $\boldsymbol{T}_\beta : Q \in \mathcal{Q} \mapsto r + \gamma \boldsymbol{P}\boldsymbol{m}_\beta Q \in \mathcal{Q}$

[1] mellow-max は softmax とほぼ同じ [Kozuno et al., 2019].

いろいろな価値反復法 I

ここでは近似がない場合の価値反復法を紹介する.
近似がない場合の価値の更新は全て要素ごとに実行される.

Value Iteration

行動価値関数を次式で置換:

$$Q_{k+1} := \boldsymbol{T} Q_k \tag{2}$$

備考: 近似が入った場合の性能が悪い (後で解説).

いろいろな価値反復法 II

Soft Value Iteration

行動価値関数を次式で置換:

$$Q_{k+1} := \boldsymbol{T}_\beta Q_k \tag{3}$$

備考: VI の max を softmax に置き換えたもの.
近似誤差などの誤差項に強くなるが, 最終性能が下がる. β によって最終性能を調節できる. 実は Entropy regularized な MDP では soft-VI と方策勾配法が一緒になる [Schulman et al., 2017].

いろいろな価値反復法 III

Advantage Learning

行動価値関数を次式で置換 ($\alpha \in [0, 1]$):

$$Q_{k+1} := \boldsymbol{T}_{AL,\alpha} Q_k := \boldsymbol{T} Q_k + \alpha \left(Q_k - \boldsymbol{m} Q_k \right) \tag{4}$$

備考: $Q_k - \boldsymbol{m} Q_k$ (gap-increasing 項と呼ぶ) によって行動間の Q 値の差を広げる. VI では最適行動以外の価値が非定常方策の価値になってしまう問題があるが, AL を使うと解決できる. CVI の証明では α によってノイズに対する頑健性と収束の速さを調節できる.

いろいろな価値反復法 IV

Dynamic Policy Programming

Action preference 関数を次式で置換:

$$\Psi_{k+1} := \boldsymbol{T}_{DPP,\beta}\Psi_k := \boldsymbol{T}_\beta\Psi_k + \Psi_k - \boldsymbol{m}_\beta\Psi_k \tag{5}$$

$$\pi_{k+1}(a \mid s) = \frac{\exp\left(\beta\Psi_{k+1}(s,a)\right)}{\sum_{b\in\mathcal{A}}\exp\left(\beta\Psi_{k+1}(s,b)\right)} \tag{6}$$

備考: Action preference 関数は行動価値関数と同じく, 状態と行動を引数にスカラーを返す関数. この更新則の導出は後で解説. soft-max と gap-increasing を同時に使うと誤差項とノイズのどちらにも強くなり, 最終性能も良くなる.

いろいろな価値反復法 V

Conservative Value Iteration

Action preference 関数を次式で置換:

$$\Psi_{k+1} = \boldsymbol{T}_\beta \Psi_k + \alpha \left(\Psi_k - \boldsymbol{m}_\beta \Psi_k \right) \tag{7}$$

$$\pi_{k+1}(a \mid s) = \frac{\exp\left(\beta \Psi_{k+1}(s, a)\right)}{\sum_{b \in \mathcal{A}} \exp\left(\beta \Psi_{k+1}(s, b)\right)} \tag{8}$$

備考: これは VI, SVI, AL, DPP の全ての一般形になっているため, CVI を解析することは全ての解析に繋がる.
実は natural actor critic (NAC) とも繋がっており, β が NAC における学習率に対応する. β を減らすと NAC における学習率が下がり, 学習が安定する.

CVI の特徴まとめ

CVI の理論的な性能の上界を見ると, 以下のことがわかった

- gap-increasing とシンプルな max を使う手法はノイズには強いが, エラーには弱い
- softmax を使った手法は全てのエラーに対して頑健だが, 収束時の性能が弱い
- gap-increasing と softmax を使った手法は良いとこ取りするので強い

これを見ていこう.

目次

概要

最初に説明したように, CVI は他の価値反復法の一般形である:

CVI と他のアルゴリズムの関係

$$\Psi_{k+1} = \boldsymbol{T}_\beta \Psi_k + \alpha \left(\Psi_k - \boldsymbol{m}_\beta \Psi_k \right)$$

この更新則は

- $\alpha = 0$: Soft Q learning と同じ

$$Q_{k+1} := \boldsymbol{T}_\beta Q_k$$

- $\beta \to \infty$: mellow-max が max と等しく, Advantage learning と同じ

$$Q_{k+1} := \boldsymbol{T} Q_k + \alpha \left(Q_k - \boldsymbol{m} Q_k \right)$$

- $\beta = 1$: Dynamic policy programming と同じ

$$\Psi_{k+1} := \boldsymbol{T}_\beta \Psi_k + \Psi_k - \boldsymbol{m}_\beta \Psi_k$$

CVI のアルゴリズムの導出 I

以下の CVI の更新則を導出してみよう.

$$\Psi_{k+1} = \boldsymbol{T}_\beta \Psi_k + \alpha \left(\Psi_k - \boldsymbol{m}_\beta \Psi_k \right)$$

これはエントロピーと KL ダイバージェンスによるボーナス項が報酬に入った価値反復法を考えることで導出できる.

CVI の目標 (次ページで詳しく解説)

各状態で, 以下の量を最大化する方策を求めたい:

$$\sum_a \pi_{k+1}(a \mid s) \left[r(s,a) + \gamma \left(\boldsymbol{P} W_{\pi_k}^{\pi_{k+1}} \right)(s,a) + i_{\pi_k}^{\pi_{k+1}}(s) \right] \tag{9}$$

ここで,

- $i_{\tilde{\pi}}^{\pi}(s) := \sum_{a \in \mathcal{A}} \pi(a \mid s) \left[-\sigma \log \pi(a \mid s) - \tau \log \frac{\pi(a|s)}{\tilde{\pi}(a|s)} \right]$
- $W_{\tilde{\pi}}^{\pi}(s) := \sum_{t=0}^{\infty} \gamma^t \mathbb{E}^{\tilde{\pi}} \left[r(S_t, A_t) + i_{\tilde{\pi}}^{\pi}(S_t) \mid S_0 = s \right]$

CVI のアルゴリズムの導出 II

$$i^{\pi}_{\widetilde{\pi}}(s) := \sum_{a \in \mathcal{A}} \pi(a \mid s) \left[-\sigma \log \pi(a \mid s) - \tau \log \frac{\pi(a \mid s)}{\widetilde{\pi}(a \mid s)} \right]$$

は以下で構成されている:

- π のエントロピー $-\sigma \sum_{a \in \mathcal{A}} \pi(a \mid s) [\log \pi(a \mid s)]$ によるボーナス: 方策はなるべくランダムな行動を取るようになる
- π と $\widetilde{\pi}$ の KL ダイバージェンス $\tau \sum_{a \in \mathcal{A}} \pi(a \mid s) \left[\log \frac{\pi(a|s)}{\widetilde{\pi}(a|s)} \right]$ によるペナルティ: なるべく前回の方策よりも離れないようになる

CVI のアルゴリズムの導出 III

CVI では $\sum_a \pi_{k+1}(a \mid s) \left[r(s,a) + \gamma \left(\boldsymbol{P} W_{\pi_k}^{\pi_{k+1}} \right)(s,a) + i_{\pi_k}^{\pi_{k+1}}(s) \right]$ を最大化させたいので,
VI と同じように

$$W_{k+1} := \boldsymbol{\pi}_{k+1} \left(r + \gamma \boldsymbol{P} W_k \right) + i_{\pi_k}^{\pi_{k+1}} \tag{10}$$

なる価値関数の更新において, 方策 π_{k+1} が式 (10) を最大化させれば良さそう.

CVI のアルゴリズムの導出 IV

π_{k+1} の導出

π_{k+1} は次の最大化問題の解

$$\underset{\pi}{\text{maximize}}\ \mathbb{E}^{\pi}\left[r(s,A)+\gamma\left(\boldsymbol{P}W_k\right)(s,A)+i^{\pi}_{\pi_k}(s)\right] \tag{11}$$

このとき $\sum_a \pi_{k+1}(a \mid s)=1$ と $1 \geq \pi_{k+1}(a \mid s) \geq 0$ が制約条件
この問題の解は

$$\pi_{k+1}(a \mid s)=\frac{\pi_k(a \mid s)^{\alpha}\exp\left(\beta\left(r+\gamma \boldsymbol{P}W_k\right)(s,a)\right)}{Z(s)} \tag{12}$$

ここで, $\alpha := \tau/(\tau+\sigma)$, $\beta := 1/(\tau+\sigma)$,
$Z(s) := \sum_{a\in\mathcal{A}} \pi_k(a \mid s)^{\alpha}\exp\left(\beta\left(r+\gamma \boldsymbol{P}W_k\right)(s,a)\right)$

価値反復法と近似 I

価値反復法の適用が難しい場合, 以下のような近似を行う:

- 状態や行動空間が大きい時:
  テーブル形式の更新ができないので, 関数近似を行う. 関数空間に射影する際に誤差が乗る.
- ダイナミクスが未知の時:
  サンプリングによって確率的近似を行う. ノイズが乗る.

この近似誤差をまとめて ε_k とすると, 一回の更新は $Q_{k+1} := \boldsymbol{T}Q_k + \varepsilon_k$ と扱える.

価値反復法と近似 II

収束した際の価値関数と最適価値関数のノルムの差はアルゴリズムの評価指標の一つ:

VI における近似誤差と性能

$$\|Q^* - Q^{g_K}\|_\infty \leq 2\gamma^{K+1} V_{\max} + \frac{2\gamma}{1-\gamma}\mathcal{E}_K \tag{13}$$

ここで, g_K は Q_K に対する greedy 方策であり,
$\mathcal{E}_K := \sum_{k=0}^{K-1} \gamma^k \|\varepsilon_{K-k-1}\|_\infty$.

式 (13) は VI が近似に対して弱い説明になる.

VI における近似の影響

VI では,

1. 誤差項の係数が $1-\gamma$ なので, ホライゾンが長い環境では誤差項の影響がとても大きくなる.
2. 誤差項が ∞ ノルムの和であり, 誤差がランダムなノイズであったとしても大きくなる.

そのため, VI は近似と相性が悪い.

CVI で見る性能の上界

References


Azar, M. G., Gómez, V., and Kappen, H. J. (2012).
Dynamic policy programming.
ArXiv, abs/1004.2027.


Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P. S., and Munos, R. (2016).
Increasing the action gap: New operators for reinforcement learning.
In *AAAI*.


Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017).
Reinforcement learning with deep energy-based policies.
In *ICML*.


Kozuno, T., Uchibe, E., and Doya, K. (2019).
Theoretical analysis of efficiency and robustness of softmax and gap-increasing operators in reinforcement learning.
In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2995–3003. PMLR.