卒業研究

西村　礼恩

January 8, 2019

Contents

第 1 章

要旨

今日計算機が人間の代わりに多くのタスクを代わりにやってくれる．タスクの中には自動運転や株の自動売買などがあり，それぞれ交通事故や予期せぬ急激な株価下落などのリスクがある．人間は多くの場合そのような状況から知識と知恵を用いて回避することができるが，計算機は必ずしも回避できるとは限らない．なぜなら，計算機にはその状況が危機であるかどうかの判断をするような機構を持ち合わせていないからだ．しかし計算機がシミュレーションを大量に試行する中でその危機状況が端末状態に因果関係があることと考えることができるのであればリスク回避できるのではないだろうか．本研究ではマルコフ決定過程の探索困難問題に対してそのような仮説をたてた上で議論する．探索する方針 Behavior Policy とは別に仮説を満たすような新たな方針 Trace Deterministic-Policy を提案し，実験結果から仮説の検証と Trace Deterministic-Policy を加えてないものと比べて精度が高くなっていることを示す．

第 2 章

序論

自動運転技術や二足歩行ロボットなどの制御が困難とされてきた問題 [1] に対して日々新たな解法が試されている．現在はそれぞれの企業が今日まで集成してきたサンプルデータを活用できないかと，解法の一つとして統計的学習が世界中で研究されている．ただ，統計的学習を用いた複雑な制御は今のところ成功例を聞かない．Boston Dynamics 社の Atlas などは二足歩行ロボットとして有名であるが，制御理論を元に作られている．

制御に用いられる統計的学習の一つに強化学習がある．この理論はエージェントが未知の環境からデータをサンプルして，それらを使って，より良い方策を見つけるというフレームワークである．自動運転技術であれば，自動車がエージェントにあたり道路が環境にあたる．

報酬の最大化を行うため，エージェントは過去に試し効率的に報酬が得られた行動を好むが，そのような行動を見つけるには，今まで選択してこなかった行動を試す必要がある．エージェントは最大報酬を得るため既に経験したサンプルのみを活用し，同時により良い行動を行う潜在的に良い方策を探索する必要がある．強化学習はこの探索と活用を相互作用して性能を上げていくが，二律背反の関係にあることもわかる．他にも，環境の報酬がスパースであるとき，エージェントが何をもって行動を決定するかが困難になり，探索困難問題 [2] の一つとされている．報酬がスパースであるとは，方策による環境の探索の際に報酬が短期的に得られない，あるいは環境が目的とするような状態に達成しない限り報酬が与えられない，ような環境の性質を指す．

強化学習を用いた自動運転技術などの制御に応用することが難しいのは，このような問題を抱えているからである．

Our contribution. — 本研究は，どのようにして報酬に依存しない探索を強化学習で実現するか，ということに注力した．より具体的には，報酬に関係なく過去の良い状態を重点的に学ぶことによって，より深い探索に間接的につながる可能性があるという仮説を立てて検証した．本研究の貢献として，（1）強化学習のアルゴリズム内で状態 s_t に対する分散を学ぶことができることを実験的に言えること，具体的にはどのように状態 s_t の分散の推定を VAE の特性 [3] を用いて実現したかということ，（2）サンプルの中から報酬に依存しない状態の分散が大きいものを重点的に学習することによって，より広い範囲を探索する軌跡分散型探索ができること，（3）一つのタイムステップで，分散が大きいサンプルは報酬にボーナスを与えて，報酬がスパースな環境に対しても学習を実現すること，である．

Related work. — 報酬がスパースであるとき，方策勾配ベースのアルゴリズムであれば勾配ベクトルの長さが短くなり，価値関数ベースのアルゴリズムであれば行動価値関数の推定が過大評価により，学習が困難になる．半教師あり学習のように，数少ない教師データを如何に学習するかという動機付けに似ている．そして解決策の一つとして，疑似回数 (pseudo-count)[*1] を使うアルゴリズム [4] がある．端的に言えば状態を擬似的に回数し，同じ状態に訪問したときに罰則を与えるというものである．

[*1] なぜ擬似的に回数するかと言えば，近年の強化学習が扱う環境は状態が連続であったり高次元であるなど一般的な計算機で回数することが不可能であるからである．

第 3 章

準備

強化学習における問題を，形式上，マルコフ決定過程 (Markov decision process; MDP) と呼ぶ．3.1節では，マルコフ決定過程の定義を行い，3.2節では，強化学習における目的関数を定義する．マルコフ決定過程において定義された目的関数が最大となるような方策を見つけることが強化学習の目的である．

3.1　マルコフ決定過程

本研究では, 強化学習は有限マルコフ決定過程 $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \pi, \gamma)$ を仮定し, 準備としてここから説明していく．環境の状態は $s_0 \sim d_0; s_0 \in \mathcal{S}$ から開始し，それぞれのステップでエージェントが現在の方策 π に従って行動 $a_t \sim \pi(a_t|s_t); a_t \in \mathcal{A}$ を選択する．環境は遷移確率関数 $p(s_{t+1} = s', r_{t+1} = r|s_t = s, a_t = a)$ に従って，状態 $s_{t+1} \in \mathcal{S}$ と報酬 $r_{t+1} \in \mathcal{R}$ を返す．同時確率はそれぞれ $p(r|s,a) = \sum_{s'} p(s', r|s, a)$ と $p(s'|s,a) = \sum_r p(s', r|s, a)$ であり，期待値報酬は $r(s,a) = \sum_r p(r|s,a)$ と定義する．r を有界であると認める．$G_t := \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}$ は割引因子 $\gamma \in [0, 1]$ に関する割引き付き報酬の総和であり，単に収益やリターンと呼ぶ．今後はリターンと呼ぶことにする．強化学習の慣習で軌跡 (trajectory) を $\tau := \{(s_t, a_t)\,;\, t \in \{1, 2, \cdots\}\}$ とし，暗黙的に $\mathbb{E}_\pi[\,\cdot\,] := \mathbb{E}_{\tau\sim\pi}[\,\cdot\,]$ を使う．エントロピー正則化の文脈 [5] ではこのリターンをエントロピー拡張し，改めて $R_t := \sum_{t'=t}^{\infty} \gamma^{t'-t}(r_{t'} - \kappa D_{\mathrm{KL}}[\pi(\cdot|s_{t'})\|\bar{\pi}(\cdot|s_{t'})])$ とする．κ は温度と呼ばれるスカラー係数であり，$D_{\mathrm{KL}}[\pi(\cdot|s_t)\|\bar{\pi}(\cdot|s_t)]$ は時刻 t における現在の方策 π と参照方策 $\bar{\pi}$ のカルバック-ライブラ (Kullback-Leibler; KL と呼ぶ) ダイバージェンスである．
表記を簡略化するため $D_{\mathrm{KL}}(s) := D_{\mathrm{KL}}[\pi(\cdot|s_t)\|\bar{\pi}(\cdot|s_t)]$，$\mathrm{KL}_t^\pi := D_{\mathrm{KL}}(s_t)$ とする．KL_t^π を KL ペナルティと呼ぶ．

3.2　目的関数と価値関数

強化学習における目的は目的関数 (objective function)$J^{\mathrm{std}}(\pi) := J^{\mathrm{std}}(d_0, \pi)$ を最大化するような (エージェントの) 方策 π を見つけることである．目的関数は初期状態 $s_0 \sim d_0$ から開始とするときの割引付き総和報酬の期待値であり，

$$J^{\mathrm{std}}(d_0, \pi) := \mathbb{E}_\pi[G_0|d_0, \pi] = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t|d_0, \pi] \tag{3.2.1}$$

$$= \int_{\mathcal{S}} d_\gamma^\pi(s) \int_{\mathcal{A}} \pi(a|s) \int_{\mathcal{S}} p(s'|s,a) r(s,a) \mathrm{d}s' \mathrm{d}a \mathrm{d}s. \tag{3.2.2}$$

ここで $d_\gamma^\pi(s) := \sum_{t=0}^{\infty} \gamma^t p(s_t = s|d_0, \pi)$ は方策 π における割引付き状態分布 (discounted state distribution) である．$p(s_t = s)$ は確率密度関数であることを留意していただきたい．

環境が未知であるため $J^{\mathrm{std}}(\pi)$ もまた未知である．価値関数と呼ばれる初期状態や行動が決定されたときに推定す

る値を考える必要がある．一般的に価値関数は三つある．行動価値関数 (action value function)

$$Q^\pi(s,a) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a], \tag{3.2.3}$$

は状態 s と行動 a に依存した価値関数である．状態価値関数 (state value function)

$$V^\pi(s,a) := \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)] \tag{3.2.4}$$

$$= \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s], \tag{3.2.5}$$

は状態 s にのみ依存した価値関数であり，方策 π に従ったときの行動価値関数の期待値である．これらは再帰形式で表すことができ，それぞれ

$$Q^\pi(s,a) = \mathbb{E}_{a'\sim\pi(\cdot|s'),s'\sim p(\cdot|s,a)}[r(s,a) + \gamma Q^\pi(s',a')], \tag{3.2.6}$$

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s),s'\sim p(\cdot|s,a)}[r(s,a) + \gamma V^\pi(s')], \tag{3.2.7}$$

となりこれらの再帰形式をベルマン方程式 (Bellman Equation) と呼ぶ．アドバンテージ関数 (advantage function)

$$A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s), \tag{3.2.8}$$

も価値関数の一つである．この価値関数は方策 π に従う代わりに状態 s において行動 a を選択したときの優位性を推定するものになっている．状態価値関数は名前からわかるように，その状態のみに注目したときにどれほどの価値があるかを推定するものになっている．行動価値関数は状態-行動価値関数 (state-action value function) とも呼ばれるようにある状態においてある行動を行うときにどれほどの価値があるかを推定するものになっている．ベルマン方程式は縮小写像であるため様々な強化学習のアルゴリズムは縮小写像の性質から収束性の証明を行っている．あと，一般的に $Q^\pi(s,a)$ は一つのサンプル τ に対する価値の推定を行うが，$Q^\pi(s_t,a_t) := \mathbb{E}_\pi[G_t|s_t,a_t] = \mathbb{E}_\pi[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}|s_t,a_t]$ はサンプル $\tau_t := \{(s_{t'},a_{t'})\,;\, t' \in \{t, t+1, \cdots\}\}$ に対する価値の推定を行う．

π に関するマルコフ連鎖の定常分布 $d^\pi(s) := \lim_{t\to\infty} p(s_t = s, \pi)$ を仮定したとき，目的関数 $J^{\text{std}}(\pi)$ と価値関数は次のような関係になる．

$$J^{\text{std}}(\pi) = \int_{\mathcal{S}} d^\pi(s) V^\pi(s) \mathrm{d}s = \int_{\mathcal{S}} d^\pi(s) \int_{\mathcal{A}} \pi(a|s) Q^\pi(s,a) \mathrm{d}a\, \mathrm{d}s. \tag{3.2.9}$$

エントロピー正則化目的関数の定義は6.1節で述べる．そして，これらの定義と本研究のベースラインとなるアルゴリズムについては6.2節で説明する．

第 4 章

提案手法

有限マルコフ決定過程における目的関数 (3.2.2) の最大化の解法として，
policy gradient based — 直接的に方策勾配を求めて最適方策を見つける手法 (e.g. 6.2.1)
value function based — 間接的に行動価値関数を最適化し，最適方策を見つける手法 (e.g. [6])
actor-critic based — 方策勾配と行動価値関数の両方の空間で最適化を行う手法 (e.g. 6.2.2，6.2.3)
が存在する．本研究では actor-critic based のアルゴリズムにサンプリングの方法と追加の活用を行い，既存手法 (6.2.2，6.2.3) より良いリターンが得られることを期待する．4.1 節では，報酬に依存しない変数を用いる新たなサンプリングの手法を提案する．4.2節では，TDP の振る舞いに関係する状態の不確実性を表すパラメータ (本研究では状態の分散) をどのように学習するかを提案する．4.3節では，A2C(6.2.2) をベースラインとし，TDP を加えたアルゴリズムについて提案する．

4.1 Trace Deterministic-Policy

この節では本研究の重要な概念となる trace deterministic policy(TDP) について定義し，説明する．環境でデータをサンプリングする行動方策 (Behavior Policy) と TDP の違いは，前者は状態を与えたときの行動確率分布であるのに対して，後者は状態に依存する変数がある閾値を超えない限り決定論的に最適な行動をし，閾値を超えたときは行動方策と同様の働きをするものである．

既存の強化学習における方策は，状況に関係なく基本的に探索を行うため，探索の必要でない状況でも確率的に行動する．すなわち，一つの軌跡サンプル τ に本来不必要なステップ (s,a)[*1]が複数存在することによって，サンプル効率が悪くなってしまう．よって，本研究ではあるパラメータによって探索の必要ない状況を決定論的に行動するものを考える．パラメータの選択として，状態の不確実性を表すパラメータが相応しいと仮定する．すなわち，不確実性を示すパラメータが閾値を超えない限り決定論的に行動して，逆に超えれば確率的に様々な行動を探索することによって，無駄な探索を省きながら不確実な状況に対応できるような方策となると仮説を立て，TDP を定義する．

Definition 4.1 (Trace Deterministic Policy). $\pi(a|s)$ を行動確率分布, 状態依存する変数を $\Sigma(s_t)$, 閾値を $\Sigma = \mathbb{E}[\Sigma(s)]$ とする．このとき trace deterministic policy を

$$\mu(s) = \begin{cases} \arg\max_{a \in \mathcal{A}} \pi(a|s) & \text{if } \Sigma(s_t) < \Sigma \\ a \sim \pi(a|s) & \text{otherwise.} \end{cases} \tag{4.1.1}$$

と定義する．本研究では $\Sigma(s_t)$ をどのように実現するかということに注目する．閾値 Σ をどのように設定するかによって精度は間違いなく変わるが，本研究では $\Sigma(s)$ に対する期待値とする．

[*1] 探索の必要のない s に対して探索した a とのペア

第 5 章

実験

a

第 6 章

付録

6.1 エントロピー正則化

一般的な強化学習とエントロピー正則化強化学習の違いは目的関数 (3.2.2) をエントロピー正則化させることと，現在の方策 π をボルツマン方策 (Boltzmann policy)$\pi_Q^{\mathcal{B}}$ に変えることである．

$$J^{\text{ent}}(d_0, \pi) := \mathbb{E}_\pi[R_0|d_0, \pi] = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t (r_t - \kappa \text{KL}_t^\pi)|d_0, \pi]. \tag{6.1.1}$$

標準のエントロピーボーナスと同じ効果を果たすために $\bar{\pi}$ を一様分布とする．すなわち，KL ペナルティの代わりにエントロピーボーナス $\mathcal{H}_t^\pi := -\int_{\mathcal{A}} \pi(a|s_t) \log \pi(a|s_t) \mathrm{d}a$ を使うことはエントロピー正則化の文脈では定数ほどの違いしかないが，一般化のため $\bar{\pi}$ を用いた KL ペナルティを考える．

続けて一般の強化学習と同様に価値関数を定義するが，価値関数をエントロピー正則化されているかは文脈によって変えるものとする．

$$Q^\pi(s, a) := \mathbb{E}_\pi[r_0 + \sum_{t=1}^{\infty} \gamma^t (r_t - \kappa \text{KL}_t^\pi)|s_0 = s, a_0 = a], \tag{6.1.2}$$

$$\pi_Q^{\mathcal{B}}(a|s) := \arg\max_\pi \{\mathbb{E}_{a\sim\pi}[Q^\pi(s, a)] - \kappa D_{\text{KL}}[\pi\|\bar{\pi}](s)\} \tag{6.1.3}$$

$$= \bar{\pi}(a|s) \exp(Q^\pi(s, a)/\kappa) / \mathbb{E}_{a'\sim\bar{\pi}}[\exp(Q^\pi(s, a')/\kappa)], \tag{6.1.4}$$

$$V_Q(s) := \kappa \log \mathbb{E}_{a'\sim\bar{\pi}}[\exp(Q^\pi(s, a')/\kappa)] \tag{6.1.5}$$

$$= \mathbb{E}_{a\sim\pi_Q^{\mathcal{B}}(\cdot|s)}[Q^\pi(s, a)] - \kappa D_{\text{KL}}[\pi_Q^{\mathcal{B}}\|\bar{\pi}](s). \tag{6.1.6}$$

行動価値関数は目的関数 (6.1.1) から自明に求まるが，状態価値関数はボルツマン方策 $\pi_Q^{\mathcal{B}}$ を考慮する必要があるので $\pi_Q^{\mathcal{B}}$ の定義も含めた．

6.2 強化学習のアルゴリズム

方策勾配 (Policy Gradient と呼ぶ) ベース Actor-Critic アルゴリズムは低-分散勾配推定を用いた方策探索が可能であるため現実世界に応用しやすいとされている．本研究では，この Actor-Critic を発展させた Advantage Actor-Critic(A2C と呼ぶ) と Proximal Policy Optimization(PPO と呼ぶ) を用いるため，ここではその説明をする．

6.2.1 Policy Gradient

Policy Gradient は強化学習の中で様々なアルゴリズムの土台となっている．Policy Gradient-based なアルゴリズムは直接目的関数の最大化を行い，Value Function-based なアルゴリズムは価値関数の最大化をするなどの間接

的に目的関数を最大化する手法もある．Policy Gradient は名前の通り方策の勾配を求めるのだが，今まで定義に用いていた方策は単なる確率分布である．ここでは問題を，パラメータ付き方策 (parameterized policy) のクラス $\Pi_\theta := \{\pi_\theta; \theta \in \mathbb{R}^m\}$ の中から期待割引付き報酬が最大になるような方策 (パラメータ) を見つける，とする．そして，方策とともにその勾配の関係を表した方策勾配定理 [7] を次に説明する．この定理の式変形に log trick と呼ばれる技工がある．

$$\nabla_\theta \ln \pi_\theta(a \mid s) = \frac{\partial \ln \pi_\theta}{\partial \pi_\theta} \frac{\partial \pi_\theta}{\partial \theta} = \frac{1}{\pi_\theta} \nabla_\theta \pi_\theta(a \mid s) \tag{6.2.1}$$

方策勾配定理は，直感的に π_θ に関する目的関数の勾配 $\nabla_\theta J(\pi_\theta)$ はパラメータ付き方策の勾配方向 $\nabla_\theta \ln \pi_\theta(a|s)$ に長さが $Q^{\pi_\theta}(s,a)$ だけあると云える．強化学習の慣習として，目的関数の勾配推定を $\widehat{g_\theta} := \nabla_\theta J(\pi_\theta)$ とする．

$$\begin{aligned}
\widehat{g_\theta} &= \nabla_\theta \int_{\mathcal{S}} d^\pi(s) \int_{\mathcal{A}} \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \mathrm{d}a \, \mathrm{d}s \\
&\propto \int_{\mathcal{S}} d^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \mathrm{d}a \, \mathrm{d}s \\
&\overset{(6.2.1)}{=} \int_{\mathcal{S}} d^\pi(s) \int_{\mathcal{A}} \nabla_\theta \ln \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \pi_\theta(a|s) \mathrm{d}a \, \mathrm{d}s \\
&= \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{(s_t,a_t) \in \tau} \nabla_\theta \ln \pi_\theta(a_t|s_t) \cdot Q^{\pi_\theta}(s,a)].
\end{aligned} \tag{6.2.2}$$

式6.2.2の方策の変数 (s_t, a_t) と価値関数の変数 (s, a) は別物であることを留意していただきたいが，時刻 t における方策の行動がそれ以前の価値関数に影響を与えているため，

$$\widehat{g_\theta} := \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{(s_t,a_t) \in \tau} \nabla_\theta \ln \pi_\theta(a_t|s_t) \cdot Q^{\pi_\theta}(s_t,a_t)], \tag{6.2.3}$$

と考えるほうが自然である．ただし，(6.2.2) ≈(6.2.3) である．目的関数を $J^{\mathrm{ent}}(\pi_\theta)$ と置いた時，勾配は

$$\widehat{g_\theta} := \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{(s_t,a_t) \in \tau} \nabla_\theta \ln \pi_\theta(a_t|s_t) \cdot Q^{\pi_\theta}(s_t,a_t) - \kappa \nabla_\theta D_{\mathrm{KL}}[\pi_\theta \| \bar{\pi}](s_t)], \tag{6.2.4}$$

となる．

(6.2.3) の特徴として偏り (bias) がない代わりに高分散 (high variance) であることは広く知られている．この問題を緩和するために分散削減の技法としてベースライン関数が使われるようになったので，(6.2.3) を直接的に推定するのではなく

$$\begin{aligned}
\widehat{g_\theta} &:= \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{(s_t,a_t) \in \tau} \nabla_\theta \ln \pi_\theta(a_t|s_t) \cdot (Q^{\pi_\theta}(s_t,a_t) - V^{\pi_\theta}(s_t))] \\
&\overset{(3.2.8)}{=} \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{(s_t,a_t) \in \tau} \nabla_\theta \ln \pi_\theta(a_t|s_t) \cdot A^{\pi_\theta}(s_t,a_t)],
\end{aligned} \tag{6.2.5}$$

を求める．方策勾配の推定ができれば，確率的勾配降下法

$$\theta_{k+1} = \theta_k + \alpha_k \widehat{g_\theta}_k, \tag{6.2.6}$$

を用いて，ニューラルネットワークの重みを更新する．このとき，α は学習率である．

6.2.2 Advantage Actor-Critic

A2C は複数のエージェントを並列にサンプルを集めさせ，平均の勾配を求め，グローバルネットワークの重みを更新するアルゴリズムである．並列で学習することによってネットワークの更新が PG に比べて安定的である．純

粋な PG との違いは損失関数の定義にあり，結果から言えば上記したような方策の勾配だけではなく価値関数の勾配も求めることになる．α を学習率としたとき，

$$\begin{aligned}\mathcal{L}^{a2c}(\theta) &:= \nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{1}{2}\|(G - Q^{\pi_\theta}(s,a))\|^2\right] \\ &= \mathbb{E}_{\tau\sim\pi_\theta}\left[\alpha\nabla_\theta(-\log\pi_\theta(a|s)(G - V^{\pi_\theta}(s))) + \nabla_\theta\frac{1}{2}\|(G - V^{\pi_\theta}(s))\|^2\right].\end{aligned} \tag{6.2.7}$$

損失関数 $\mathcal{L}^{a2c}$ は行動価値関数の推定がリターンに近似するように定義されており，教師あり学習に近い形式になっている．

6.2.3 Proximal Policy Optimization

信頼領域方策最適化 (Trust Region Policy Optimization; TRPO と呼ぶ) は，近接方策最適化 (Proximal Policy Optimization; PPO と呼ぶ) の基礎となるアルゴリズムであるので基本的な動機づけについて書く．これまで，方策パラメータ θ に関する勾配の良い推定を抽出することが注目され続けたが，リターンを直接最適化することが困難であることもわかった．それゆえに，近年の方策勾配ベースのアルゴリズムは代理報酬 (surrogate reward) を最適化する問題に変わった．真の報酬の局所近似

$$\mathcal{L}^{sur}(\theta_k,\theta) := \mathbb{E}_{\tau\sim\pi_{\theta_k}}\left[\sum_{(s_t,a_t)\in\tau}\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}\widehat{A}^{\pi_{\theta_k}}(s_t,a_t)\right], \tag{6.2.8}$$

の最大化に注目されるようになった．このとき，$\widehat{A}^{\pi_{\theta_k}}$ は $A^{\pi_{\theta_k}}$ を正規化

$$\widehat{A}^{\pi_{\theta_k}} = \frac{A^{\pi_{\theta_k}} - \mu(A^{\pi_{\theta_k}})}{\sigma(A^{\pi_{\theta_k}})} \tag{6.2.9}$$

したものである．$\mathcal{L}^{sur}(\theta_k,\theta)$ は過去の方策 π_{θ_k} がサンプルしたデータを用いて方策 π_θ と π_{θ_k} の相対的な能力を測っていると云える．

代理報酬の勾配は局所的に方策勾配を推定しているが，パラメータ空間 Π_θ ではなく，方策空間 $\Pi := \{\pi : \pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}, \sum_a \pi_{s,a} = 1, \pi_{s,a} \geq 0\}$ で方策 π_θ の周りの信頼領域を制約がなければ，学習の結果はよくならないことがわかった．TRPO は $\mathcal{L}^{sur}(\theta_k,\theta)$ の最大化に信頼領域の制約

$$\mathbb{E}_{s\sim\pi_{\theta_k}}\left[D_{\mathrm{KL}}[\pi_\theta\|\pi_{\theta_k}](s)\right] \leq \delta \tag{6.2.10}$$

を加えたものである．

この TRPO は非常に良い精度を誇るものの，$\mathcal{L}^{sur}(\theta_k,\theta), \mathbb{E}_{s\sim\pi_{\theta_k}}\left[D_{\mathrm{KL}}[\pi_\theta\|\pi_{\theta_k}](s)\right]$ の推定はそれぞれ非線形共役勾配法や，ヘシアンの二次形式の計算などが必要になり，計算コストが大きすぎることが問題として挙げられるようになった．PPO は KL 制約をクリッピングするような目的関数とする．$r(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$ としたとき，

$$\mathcal{L}^{PPO}(\theta_k,\theta) := \mathbb{E}_{\tau\sim\pi_{\theta_k}}\left[\sum_{(s_t,a_t)\in\tau}\min(\mathrm{clip}(r(\theta),1-\epsilon,1+\epsilon)\widehat{A}^{\pi_{\theta_k}}(s_t,a_t), r(\theta)\widehat{A}^{\pi_{\theta_k}}(s_t,a_t))\right] \tag{6.2.11}$$

というような計算が平易になった．結果的に，PPO は TRPO に比べて計算速度とサンプル効率の両方共上回る性能が出るようになった．

6.3 生成モデル

6.3.1 Variational Autoencoder

強化学習のアルゴリズムとは別で，本研究で Variational Autoencoder(VAE と呼ぶ) と呼ばれる生成モデルを用いるため，ここではその説明をする．このアルゴリズムは入力をある確率分布 p_θ に写像する．入力を x とし潜在ベクトルを z とするとき，$p_\theta(z)$ を事前分布，$p_\theta(x|z)$ を尤度，$p_\theta(z|x)$ を事後分布と定義する．最適なパラメータ θ^* は真のデータサンプルを生成する最大確率だとすると，

$$\theta^* = \arg\max_\theta \prod_{i=1}^{N} p_\theta(x_i) \tag{6.3.1}$$

$$= \arg\max_\theta \prod_{i=1}^{N} \int p_\theta(x_i|z) p_\theta(z) \mathrm{d}z, \tag{6.3.2}$$

となる．一方，$p_\theta(x_i)$ を計算するには z に対する積分をするため，計算量が増大してしまう．潜在空間の探索を容易にするため，入力 x が与えられたときに制約を与えられたコードを出力する確率分布 $q_\phi(z|x)$ を新しく定義する．

現在行いたいことは，真のデータを生成する尤度の最大化と真の事後分布 $p_\theta(z|x)$ と推定の事後分布 $q_\phi(z|x)$ の距離の最小化である．よって損失関数は

$$\mathcal{L}^{vae}(\theta, \phi) := -\log p_\theta(x) + D_{\mathrm{KL}}[q_\phi(z|x) \| p_\theta(z|x)], \tag{6.3.3}$$

と定義できる．(6.3.3) は式変形によって

$$\mathcal{L}^{vae}(\theta, \phi) = -\mathbb{E}_{q_\phi(z|x)}[p_\theta(x|z)] + D_{\mathrm{KL}}[q_\phi(z|x) \| p_\theta(z)], \tag{6.3.4}$$

となる．これにより計算機により現実的に計算が可能となった．この損失関数の最小化は

$$-\mathcal{L}^{vae}(\theta, \phi) = \log p_\theta(x) - D_{\mathrm{KL}}[q_\phi(z|x) \| p_\theta(z|x)] \leq \log p_\theta(x) \tag{6.3.5}$$

と，常に $\log p_\theta(x)$ の下限となるので，エビデンス下限とも呼ばれる．

Bibliography

[1] Lei Tai, Jingwei Zhang, Ming Liu, Joschka Boedecker, and Wolfram Burgard. A survey of deep network solutions for learning control in robotics : From reinforcement to imitation. 2018.

[2] Martin A. Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Van de Wiele, Volodymyr Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing - solving sparse reward tasks from scratch. In *ICML*, 2018.

[3] Carl Doersch. Tutorial on variational autoencoders. *CoRR*, Vol. abs/1606.05908, , 2016.

[4] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *NIPS*, 2016.

[5] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *ICML*, 2017.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, pp. 529–533, 2015.

[7] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.