The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

主成分分析による方策パラメータの低次元化を用いた直接方策探索の提案

Proposal of direct policy search using reduction of policy parameters by principal component analysis

村田 悠稀 Yuuki Murata　宮下 恵 Megumi Miyashita　矢野 史朗 Shiro Yano　近藤 敏之 Toshiyuki Kondo

東京農工大学
Tokyo University of Agriculture and Technology

In the sampling based direct policy search in reinforcement learning, higher dimensional decision variables causes the deterioration of optimal value and the slowing down of the learning speed. We clarified that the variance of the sampling probability distribution affects both for the optimal value and the learning speed. Especially, there exists the tradeoff between the optimal value and the learning speed. In this paper, we propose two trick to improve the learning speed without deteriorating the optimal value. First trick is to employ the small variance sampling distribution for improving the optimal value; It causes slower convergence as a side effect. As the second trick, we employed the dimensionality reduction of the decision variable for improving the learning speed.

1. はじめに

近年, 公共機関, 医療, 軍事的など, 様々な場面で活躍するロボットが開発される時代になってきており, 今では, 自律・自己学習型ロボットも開発されつつある.

強化学習とは, エージェントが試行錯誤を繰り返すことで, 環境からの累積報酬を最大にするような最適な行動指針（方策）を見つけだす機械学習の枠組みである. 最近では, 目的関数が未知の状態で決定変数を一つ決めると, 値が一つ求まるという条件のもとで, 値を最小化するような決定変数を見つける問題クラスであるブラックボックス最適化問題の考えになぞらえて強化学習を理解することが重要であると言われている [4]. ブラックボックス最適化問題における決定変数を方策とみなし, 出力を累積報酬だと見なせば, ブラックボックス最適化の流れにそって強化学習を理解することができる.

本研究では, 強化学習の枠組みの一つであるサンプリングベースの直接方策探索に着目して研究を行った. 直接方策探索とは, 方策を数理モデルで表現し, モデルのパラメータを最適化するものであり, この方策を更新する際に, ある確率分布からのサンプリングによって方策を更新する. G-MDS[1] や TRPO[7] などがこれに相当し, 一般的に確率分布は正規分布であることが多い. このサンプリングベースの直接方策探索には, 2 つの問題点がある. 1 つ目は, 方策を特徴づける方策パラメータを, 累積報酬を最大化することを指標として更新していくことになるが, この方策パラメータの次元が高次元になると, 学習速度や最適値が悪化することである. Funny[5] らが行った実験では, ロボットアームで対象物を掴むことを強化学習させたとき, 主成分分析 [6] を用いて低次元化した方が, 学習速度と最適値が共に向上したという結果を示している. 2 つ目は, サンプリングベースの直接方策探索では, サンプリングするための確率分布の分散によって, 最適値と学習速度に影響を及ぼしてしまうことである (詳細は 2.4 節を参照). 例えば, 分散を大きくすると, 学習速度は速くなるが最適値は悪化する. 分散を小さくすると, 逆の現象が起こる. すなわち, 分散によって, 最適値と学習速度がトレードオフになる.

本研究では, トレードオフをできるだけ解消して, 最適値を悪化させずに学習速度を向上させることを目的とし, 以下の 2 つの操作を行う.

1. 方策パラメータ高次元化による最適値悪化を解消する確率分布への操作
2. 1 の操作を行うことにより悪化した学習速度を向上させるために, 主成分分析による低次元空間を用いて決定変数の削減

今回はサンプリングベースの直接方策探索である G-MDS に提案手法を適用して従来手法との比較実験を行い, 提案手法の有用性を評価した.

連絡先: 村田悠稀，東京農工大学, murata@livingsyslab.org

2. 先行研究

2.1 MDS

強化学習においてコスト最小化問題を考えたとき, 目的関数 $\mathcal{J}$ を次の式 (1) で表せる.

$$\mathcal{J} = \sum_{j=1}^{M} p\left(\boldsymbol{\theta}_j\right) J\left(\boldsymbol{\theta}_j\right) \tag{1}$$

なお, $\boldsymbol{p} = [p(\theta_1), p(\theta_2), ..., p(\theta_M)]$ は方策パラメータ θ が $\theta = \theta_j$ をとる確率分布, $J(\boldsymbol{\theta}_j)$ は方策パラメータ θ_j の報酬であり, 条件は以下に示すとおりである.

$$\boldsymbol{p} \in \mathcal{P} := \left\{ x | x \in \mathbb{R}^M, \sum_{i}^{M} x_i = 1, x_i \geq 0 \right\} \tag{2}$$

ここで, 生起確率 $\boldsymbol{p}$ を決定変数とみなし, $\boldsymbol{p}$ を更新して最適な方策を獲得することを考えると, ブラックボックス最適化の観点から,

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p} \in \mathcal{P}} \mathcal{J} \tag{3}$$

というような最小化問題となる.
また, $\boldsymbol{p}$ の更新方法として, 最適化手法の一つである鏡像降下法を用いる. 一般的な鏡像降下法 [8] の式が次の式 (4) である.

$$\beta_t = \arg\min_{\beta \in \mathcal{B}} \left\{ \langle g_t, \beta \rangle + \psi\left(\beta\right) + \eta_t B_\phi\left(\beta || \beta_{t-1}\right) \right\} \tag{4}$$

ここで, $\mathcal{B}$ はパラメータ空間, β はパラメータ, $g_t \in \delta(\beta_{t-1})$, η_t は更新幅を調節するハイパーパラメータである. 式 (4) の β_t を $\beta_t = p_t$ とすると,

$$p_t = \arg\min_{\boldsymbol{p}\in\mathcal{B}} \{\langle g_t, p(\boldsymbol{\theta})\rangle + \eta_t B_\phi\left(p(\boldsymbol{\theta})||p_{t-1}\right)\} \tag{5}$$

となり, $g_t = \nabla_p \mathcal{J} = J(\theta)$ であるため, 元の目的関数を微分しなくても更新を行うことが可能になる. これを MDS(Mirror Desent Search) と呼ぶ.

2.2 G-MDS

2.1 節の MDS をさらに発展させたのが, G-MDS(Gaussian-MDS)[1] である. まず, 式 (4) の鏡像降下法を発展させる. 具体的には, 式 (4) のブレグマン距離を KL 距離とする. つまり, 式 (4) の B_ϕ における ϕ を, $\phi(x_{t,j}) = \sum_{j=1}^{N} x_{t,j}\log(x_{t,j})\left(\boldsymbol{x}\in\mathbb{R}^N, \beta_{t,j} > 0\right)$ とすると, 更新式として式 6 の指数勾配降下法が求まる.

$$p_t(\theta_i) = \frac{\exp\left(-\eta_t g_{t,i}\right) p_{t-1}(\theta_i)}{\sum_{j=1}^{N}\exp\left(-\eta_t g_{t,j}\right) p_{t-1}(\theta_j)} \tag{6}$$

また, ここでは $p_t(\boldsymbol{\theta})$ の確率分布をガウス分布として仮定しており, $p_t(\boldsymbol{\theta})$ は,

$$p_t(\theta) = \mathcal{N}\left(\theta \mid \mu_{t-1}, \Sigma_{\varepsilon_{t-1,i}}\right) \tag{7}$$

となる. さらに, G-MDS では, $p(\theta)$ の平均値の更新についてのみ考えており, 平均値を式 (6) から計算することができる.

$$\mu_t = \sum_{i=1}^{N}\theta_i p_t(\theta_i) = \frac{\mathbb{E}_{p_{t-1}}[\theta_i \exp(-\eta_t g_{t,i})]}{\mathbb{E}_{p_{t-1}}[\exp(-\eta_t g_{t,i})]} \tag{8}$$

式 (8) にモンテカルロ積分を適用して, 推定される μ_t を $\hat{\mu_t}$ とすると, 以下の式となる.

$$\hat{\mu}_t = \frac{\frac{1}{N}\sum_{i=1}^{N}\theta_i \exp(-\eta_t g_{t,i})}{\frac{1}{N}\sum_{i=1}^{N}[\exp(-\eta_t g_{t,i})]} \tag{9}$$

さらに, θ_i を $\varepsilon_{t,i} \sim \mathcal{N}(0, \Sigma_{\varepsilon_{t-1,i}})$ のガウスノイズを用いて表記すると,

$$\theta = \mu_{t-1} + \varepsilon_{t-1,i} \tag{10}$$

と表すことができる. ちなみに, $\Sigma_{\varepsilon_{t-1,i}}$ は初期標準偏差である. 式 (9),(10) より,

$$\begin{aligned}\hat{\mu_t} &= \frac{\sum_{i=1}^{N}(\hat{\mu}_{t-1} + \varepsilon_{t-1,i})\exp(-\eta_t g_{t,i})}{\sum_{i=1}^{N}[\exp(-\eta_t g_{t,i})]} \\ &= \hat{\mu}_{t-1} + \frac{\sum_{i=1}^{N}\varepsilon_{t-1,i}\exp(-\eta_t g_{t,i})}{\sum_{i=1}^{N}[\exp(-\eta_t g_{t,i})]}\end{aligned} \tag{11}$$

となるので, ガウシアンの平均値をガウスノイズの重み付けによって更新が可能になる.

2.3 χ^2 分布に依存する最適値

確率変数 $X_1, ..., X_n$ が独立に標準正規分布に従うとき, その二乗和の確率変数 $X = X_1^2 + \cdots + X_n^2$ は自由度 n の χ^2分布 に従う [2]. G-MDS では, 生起確率 $p(\boldsymbol{\theta})$ の平均値を最適解に向けて更新していく. そして, $p(\boldsymbol{\theta})$ の平均値が最適解に収束するとき, $\mu_{t-1} = \mu_t = \theta^*$ となる. このとき, 方策パラメータ θ の推定二乗誤差を考えると, 式 (10) から

$$(\theta^* - \theta)^2 = \{\mu_t - (\mu_{t-1} + \varepsilon_{t-1,i})\}^2 = \|\varepsilon_{t-1,i}\|^2 \tag{12}$$

となることがわかる. ところで, 方策パラメータ θ は正規分布からのサンプリングなので, $\varepsilon_{t-1,i}$ も正規分布からのサンプリングであり, ノイズの次元は決定変数 $\theta \in \mathbb{R}^n$ に依存している. このことから $\varepsilon_{t-1,i}$ は自由度 n の χ^2 分布に従うことがいえる. χ^2 分布に従う確率変数の期待値は n(自由度 n) であることが知られているため,

$$\mathbf{E}\left[\|\boldsymbol{\varepsilon}_{t-1}\|^2\right] = dim(\boldsymbol{\varepsilon}) = n \tag{13}$$

となる. このことから, 最適化の収束時に決定変数の次元に比例したノイズの残差成分が上乗せされてしまうため, 決定変数の次元に応じて収束コストが悪化すると考えられる.

2.4 サンプリングの確率分布

G-MDS における更新則は式 (6) で表される式を用いており, 事前分布が次の式で表せるものとする.

$$p^t(\theta) \propto \frac{\exp\left(-\lambda^{-2}(\theta-\theta_t)\Sigma^{-1}(\theta-\theta_t)\right)}{Z} \tag{14}$$

ここで, λ^{-2} はガウス分布の分散に相当するものである. ここからさらに, $p^{t+1}(\theta)$ の更新式を求めると,

$$p^{t+1}(\theta) = \frac{\exp(-\eta J(\theta) - \lambda^{-2}(\theta-\theta_t)\Sigma^{-1}(\theta-\theta_t))}{Z} \tag{15}$$

となる. この式 (15) の λ の値が大きい (分散が大きい) と, $\eta J(\theta)$ の影響が強くなるので分布の更新幅が大きくなる. 逆に, λ の値が小さい (分散が小さい) と, 事前分布の影響が強くなるので分布の更新幅が小さくなる. まとめると, 分散が大きい場合は更新幅が大きくなるので, 学習速度が速くなり, 分散が小さい場合は更新幅が小さくなるので, 学習速度が速くなる.

3. 提案手法

3.1 最適値改善操作

2.3 節で説明したノイズの残差を打ち消すことで最適値の改善を図る. χ^2 分布に従う確率変数の期待値は決定変数に依存することがわかっているので, 正規分布からサンプリングされるノイズ ε に対して, $\sqrt{\frac{1}{k}}$ を掛け合わせる. このとき, $k = dim(\boldsymbol{\varepsilon})$ である. このような操作を加えることで, χ^2 分布に従うノイズを以下のように表すことができる.

$$\left(\frac{\varepsilon_1}{\sqrt{k}}\right)^2 + \left(\frac{\varepsilon_2}{\sqrt{k}}\right)^2 + \cdots + \left(\frac{\varepsilon_k}{\sqrt{k}}\right)^2 = \frac{1}{k}\left(\|\varepsilon_1\|^2 + \|\varepsilon_2\|^2 + \cdots + \|\varepsilon_k\|^2\right) \tag{16}$$

ここで, $\left(\|\varepsilon_1\|^2 + \|\varepsilon_2\|^2 + \cdots + \|\varepsilon_k\|^2\right) \to k$ となるので, この操作を加えることにより, 決定変数に依存しない値にできる. つまり, 決定変数の高次元化に伴う最適値の悪化を改善することが可能になる.

ところで, ノイズに対して係数倍する操作を行うことは, 確率分布の再生性 [2] よりガウス分布の分散に係数倍を行っていることと等価になる. よって, ノイズを以下で表すことができる.

$$\varepsilon_{t,i} \sim \mathcal{N}\left(0, \frac{1}{\sqrt{k}}\Sigma_{\varepsilon_{t-1,i}}\right) \tag{17}$$

このときの分散をみると, 最適値改善の操作を加えたときの方が分散がかなり小さくなることがわかり, さらにノイズの次元数が高次元になるにつれて, ガウシアンの分散が小さくなっていく. このことから, 最適値は改善されるが, ノイズの決定変数の高次元化により学習速度は大幅に低下することも引き起こされてしまう (2.4 節を参照).

3.2 低次元空間を用いた直接方策探索

3.1 節の操作を加えると残差成分が決定変数の次元に依存しないものになるが, 学習速度がさらに悪化してしまう. そこで, この悪化した学習速度を改善する操作として, 主成分分析によって獲得した低次元空間を用いて, 方策パラメータを計算する操作を行う.

あらかじめ収集した分析用データに対して主成分分析を行い, 最適方策パラメータについての主成分を算出する. そして, この主成分の上位主成分だけを抽出して低次元空間とする. この低次元空間の主成分とこの主成分をどの程度の比率で用いるのかを決定する決定変数 $\boldsymbol{\omega}$ を混合することにより, 方策パラメータを計算する. これを具体的に示すと以下のようになる.

$$(\boldsymbol{p_1}, \boldsymbol{p_2}, \cdots, \boldsymbol{p_k}) \cdot \boldsymbol{\omega} = \boldsymbol{\theta} \tag{18}$$

ここで, $\boldsymbol{p_j} = (p_{1,j}, p_{2,j}, \cdots p_{k,j})^{\mathrm{T}}$, $\boldsymbol{\omega} = (\omega_1, \omega_2, \cdots \omega_k)^{\mathrm{T}}$, $k = dim(\boldsymbol{\omega})$ であり, 従来の手法 $(G-MDS)$ では方策パラメータ $\boldsymbol{\theta}$ そのものが決定変数であるのに対し, 提案手法では方策パラメータを構成するための混合比 $\boldsymbol{\omega}$ が決定変数であることに注意されたい. 提案手法により, 決定変数の次元を低次元化することで, 学習速度の低下が緩和されることになる.

4. 実験

4.1 実験条件

本実験は, 多自由度アームの最適軌道学習を用いて評価実験を行なった. [3] で述べられている実験内容を引用しているが, シミュレータ内容を簡単に説明する. xy 座標上に描かれたアームに対して, 任意の数の関節を設定することが可能であり, 関節はアーム上に等間隔に設定され, エンドエフェクタは xy 座標上を自由に動くことが可能である. シミュレーションでは, 中継点と呼んでいる赤い点を通過したのち, ゴールに向かうように設定されており, 中継点の通過時間はスタートしてから 300ms 後, ゴール到達時間はスタートしてから 500ms と設定されている. 最終的に, 条件を満たすようなエンドエフェクタの軌道を学習することが目的となる. また, このシミュレータの報酬関数は以下の式で表される.

$$r_t = \frac{\sum_{i=1}^{d}(d-1-i)f^2}{\sum_{i=1}^{d}(d-1-i)} \tag{19}$$

$$r_{300ms} = 10^8((x_{viapoint} - x_{t_{300ms}})^2 + (y_{viapoint} - y_{t_{300ms}})^2) \tag{20}$$

ここで, d はアームの関節数, $(x_{viapoit}, y_{viapoint})$ は中継点の座標, f は各関節の加速度である. また, r_{300ms} は 300ms の時のエンドエフェクタと中継点の二乗誤差に係数倍したペナルティである.

なお, 主成分分析のための分析用データは, 中継点 (0.5, 0.5) を重心とする一片の長さが 0.2 の正方形の範囲からランダムに 100 個の中継点を選択して, それぞれの最適軌道の方策パラメータをあらかじめ学習し, まとめたデータを分析用データとした. そして, この分析用データに主成分分析を適用して獲得された低次元空間を元に再度軌道学習を行った. 実験パラメータは以下の通りである.

表 1: 実験パラメータ

中継点座標	(0.54, 0.44)
更新回数	10 万
ロールアウト	15
初期標準偏差	1

4.2 実験結果

決定変数 1000 次元問題を, 提案手法で決定変数を 500 次元, 100 次元に削減したときの収束速度の比較を行った. 比較したのが以下の図である.

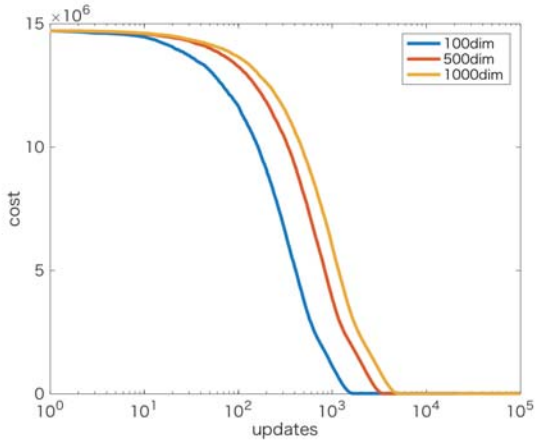

図 1: 学習速度の比較

1000 次元は G-MDS, 500 次元, 100 次元は提案手法を用いている. 学習速度を比較してみると, 1000 次元よりも 500 次元, 500 次元よりも 100 次元の方が学習速度が速くなっていることがわかる.

次に最適値の改善についてである. 最適値を比較したのが次の図である.

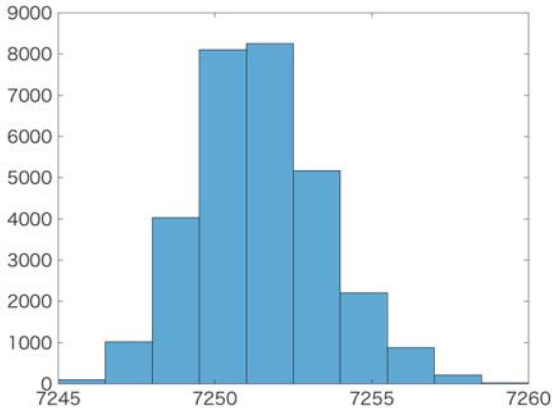

図 2: 終盤コストのヒストグラム (100 次元)

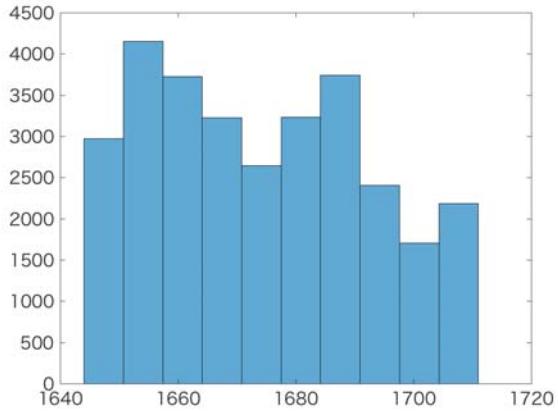

図 3: 終盤コストのヒストグラム (500 次元)

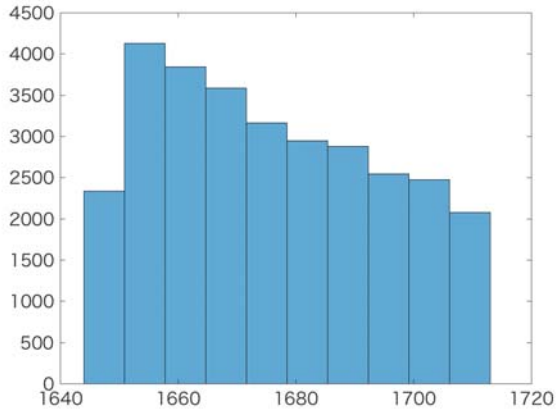

図 4: 終盤コストのヒストグラム (1000 次元)

各学習の更新回数の 7 万回〜最終までを集計してヒストグラムとして比較した. 決定変数 100 次元の場合 (図 2), 1000 次元 (従来手法)(図 4) と比べて明らかに最適値が悪くなってしまっているのがわかる. また, 決定変数 500 次元 (図 3) の場合でも, 従来手法と最適値がほとんど変わらないという結果が示された.

4.3 考察

従来手法で 1000 次元問題を解くのと, 提案手法で全主成分を使用して解くのは解釈がほぼ同じになる. このことから, 用いる主成分を削減した低次元では, 従来手法よりも情報量が少なくなり, 収束できる値に制限ができてしまうために, 最適値が悪化してしまったと考えられる. しかし, 今回の条件では, 決定変数を半分に減らしても最適値が従来手法とほぼ同じになるという結果も得ることができた. 最適値がほぼ同じであるならば, 学習速度が速い方が好ましい. 低次元空間を用いることの有用性に期待がもてる結果である.

5. まとめ

2 つの提案手法を従来手法に適用し, 評価実験を通して, 最適値と学習速度のトレードオフを従来よりも解消することに成功し, 最適値を悪化させずに学習速度を向上させることができた.

今後は, 強化学習と関係のある手法を MDS の観点からまとめ, 本研究の提案手法の適用範囲を広げていく. 低次元空間の拡張であるオンライン PCA などを提案手法に適用することで, 提案手法自体の拡張も行っていきたい.

謝辞

本研究は JSPS 科研費 JP17K12737, JP26120005, JP16H03219 の助成を受けたものです.

参考文献

[1] M. Miyashita, S. Yano, and T. Kondo, “Mirror descent search and acceleration, ” arXiv:1709.02535, 2017.

[2] 藤澤 洋徳 (2016), ”確率と統計.” 朝倉書店

[3] E. Theodorou, J. Buchli, S. Schaal, Reinforcement learning of motor skills in high dimensions: A path integral approach, in: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, 2010, pp. 2397-2403.

[4] Stulp, Freek, and Olivier Sigaud. ”Policy improvement methods: Between black-box optimization and episodic reinforcement learning.” (2012).

[5] Ficuciello, Fanny, Damiano Zaccara, and Bruno Siciliano. ”Synergy-based policy improvement with path integrals for anthropomorphic hands.” Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016.

[6] Bishop, Christopher M. ”パターン認識と機械学習 下.” ベイズ理論による統計的予測 (2008).

[7] Deisenroth, Marc Peter, Gerhard Neumann, and Jan Peters. ”A survey on policy search for robotics.” Foundations and Trends in Robotics 2.1-2 (2013): 1-142.

[8] 鈴木 大慈 (2015), ”確率的最適化.” 講談社