Deterministic Policy Gradient Algorithms

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014, June). In ICML.

2015/8/20

D1 金子 貴輝

選考理由

- ICML 2014の論文
- 高次元入出力に対応できる強化学習
- DeepMindの中の人のグループの論文
- （証明のAppendixが見つからず）

内容

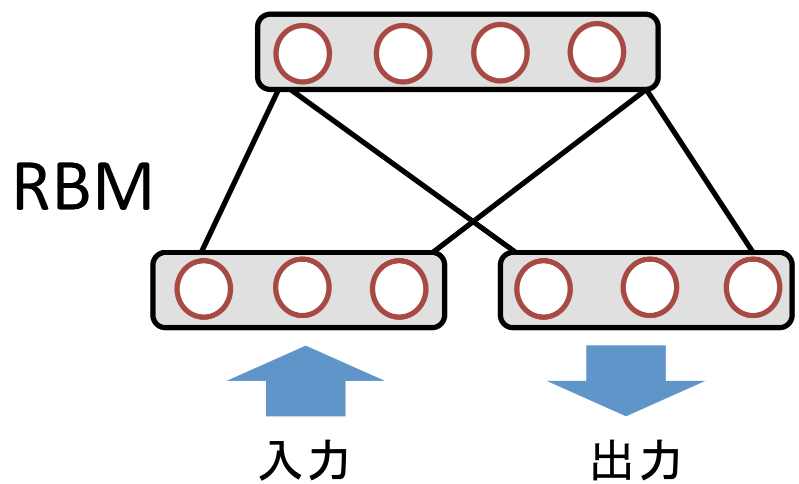
- 背景・課題
  - 高次元入出力の強化学習の課題
- 着想
  - 確率方策＋方策オン → 確定方策＋方策オフ
- 提案手法
  - 確定方策勾配法
  - 方策オフ型確定方策勾配法
  - 互換方策オフ型確定的アクタークリティック(COPDAC)
- 実験
  - 確定方策の有効性の検証タスク
  - 標準的な強化学習タスク
  - 高次元入出力タスク（Octopus Arm）
- 結論

内容

- 背景･課題
  - 高次元入出力の強化学習の課題
- 着想
  - 確率方策＋方策オン → 確定方策＋方策オフ
- 提案手法
  - 確定方策勾配法
  - 方策オフ型確定方策勾配法
  - 互換方策オフ型確定的アクタークリティック(COPDAC)
- 実験
  - 確定方策の有効性の検証タスク
  - 標準的な強化学習タスク
  - 高次元入出力タスク（Octopus Arm）
- 結論

著者の先行研究

- ２つの強化学習アルゴリズム(ENATDAC,EQNAC)
  - **方策勾配法**の一つであるNatural Actor-Critic法を使用
  - 方策分布にRBMを使用し，アルゴリズムをさらに単純化
  - RBMなので高次元入出力にも対応できる
    - 実験では入力次元34，出力次元14のタコ腕タスクも解けた



方策勾配法

- 勾配法の一種
- 期待報酬の勾配を方策分布の対数勾配から求める
- おおまかに言えば，価値の高いところで行動が出やすくなるように方策分布を修正する

報酬関数

$$J(\pi_\theta) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[r(s,a)\right]$$

状態の確率分布

行動価値関数（報酬の見積もり）

$$\underline{\nabla_\theta J(\pi_\theta)} = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi_\theta(a|s) Q^\pi(s,a) \mathrm{d}a\mathrm{d}s$$

期待報酬勾配

$$= \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[\underline{\nabla_\theta \log \pi_\theta(a|s)} Q^\pi(s,a)\right]$$

方策分布の対数勾配

先行研究の問題点

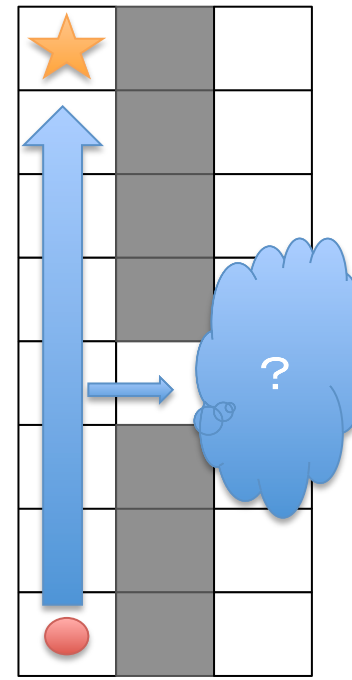
従来の方策勾配法では,

学習が進むに連れて

方策が確定的になり,

方策勾配の推定が困難になる

(最適でない行動時の予測が, サンプル数の不足で, 不正確になるためだと思われる)



迷路問題での
確定的な動作例

着想

問題点を2つに分ける

- 行動が確定的なときの積分の近似(サンプリング)が不正確
  - →確定的な方策なら，積分の必要性が生じない

- 行動が学習とともに確定的になっていく
  - →方策オフ型にして，学習の進みと方策の確定さを切り離す

内容

- 背景・課題
  - 高次元入出力の強化学習の課題
- 着想
  - 確率方策＋方策オン → 確定方策＋方策オフ
- 提案手法
  - 確定方策勾配法
  - 方策オフ型確定方策勾配法
  - 互換方策オフ型確定的アクタークリティック(COPDAC)
- 実験
  - 確定方策の有効性の検証タスク
  - 標準的な強化学習タスク
  - 高次元入出力タスク（Octopus Arm）
- 結論

確定的な方策での方策勾配法

- 従来，決定方策勾配は存在しないか，環境モデルを使うときに限られると考えられてきた[Peters, 2010]
- 状態の関数で確定的な方策 $\mu_\theta(s)$ を定義
- この方策で方策勾配法を再定式化

$$
\begin{aligned}
J(\mu_\theta) &= \int_{\mathcal{S}} \rho^\mu(s) r(s, \mu_\theta(s)) \mathrm{d}s \\
&= \mathbb{E}_{s \sim \rho^\mu} \left[r(s, \mu_\theta(s)) \right]
\end{aligned}
$$

Deterministic Policy Gradient Theorem

$$
\boxed{
\begin{aligned}
\nabla_\theta J(\mu_\theta) &= \int_{\mathcal{S}} \rho^\mu(s) \nabla_\theta \mu_\theta(s) \left. \nabla_a Q^\mu(s, a) \right|_{a=\mu_\theta(s)} \mathrm{d}s \\
&= \mathbb{E}_{s \sim \rho^\mu} \left[\nabla_\theta \mu_\theta(s) \left. \nabla_a Q^\mu(s, a) \right|_{a=\mu_\theta(s)} \right]
\end{aligned}
}
$$

方策オフ型確定方策勾配法

- 報酬を最大化する方策μと，探索のための方策βを分けて，期待報酬を定義
- 単純に期待値が置き換わる

$$J_\beta(\mu_\theta) = \int_S \rho^\beta(s) V^\mu(s) \mathrm{d}s$$
$$= \int_S \rho^\beta(s) Q^\mu(s, \mu_\theta(s)) \mathrm{d}s$$

$$V^\pi(s) = \mathbb{E}\left[r_1^\gamma | S_1 = s; \pi\right]$$
$$Q^\pi(s,a) = \mathbb{E}\left[r_1^\gamma | S_1 = s, A_1 = a; \pi\right]$$

Off-policy Deterministic Policy Gradient Theorem

$$\nabla_\theta J_\beta(\mu_\theta) \approx \int_S \rho^\beta(s) \nabla_\theta \mu_\theta(a|s) Q^\mu(s,a) \mathrm{d}s$$
$$= \underline{\mathbb{E}_{s\sim\rho^\beta}} \left[\nabla_\theta \mu_\theta(s) \, \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right]$$

提案手法：COPDAC (COPDAC-Q)

- 方策オフ型確定的方策勾配法に基づく
- アクタークリティック法で価値関数を近似

$$Q^\pi(s,a) \approx Q^w(s,a) = \phi(s,a)^\top w$$

- 基底関数 ϕ は方策と互換になるように選ぶ（近似精度のため）

$$\phi(s,a) = a^\top \nabla_\theta \mu_\theta(s)$$

- 方策と価値関数のパラメータは以下のように更新

$$\begin{aligned}
\delta_t &= r_t + \gamma Q^w(s_{t+1}, \mu_\theta(s_{t+1})) - Q^w(s_t, a_t) \\
\theta_{t+1} &= \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t) \left(\nabla_\theta \mu_\theta(s_t)^\top w_t\right) \\
w_{t+1} &= w_t + \alpha_w \delta_t \phi(s_t, a_t) \\
v_{t+1} &= v_t + \alpha_v \delta_t \phi(s_t)
\end{aligned}$$

$\phi(s_t)$ は状態の特徴量

※ 状態価値関数のパラメータvの扱いはわかりませんでした

内容

- 背景・課題
  - 高次元入出力の強化学習の課題
- 着想
  - 確率方策＋方策オン → 確定方策＋方策オフ
- 提案手法
  - 確定方策勾配法
  - 方策オフ型確定方策勾配法
  - 互換方策オフ型確定的アクタークリティック(COPDAC)
- 実験
  - 確定方策の有効性の検証タスク
  - 標準的な強化学習タスク
  - 高次元入出力タスク（Octopus Arm）
- 結論

実験ー連続値バンディット

目的:確率方策勾配と確定方策勾配の比較

環境:

- バンディット:状態を持たずすぐに報酬が得られるタスク
- 報酬は正しい出力値とのマハラノビス距離x(-1)
- パフォーマンスはステップごとの平均報酬
- 出力次元は 10,25,50

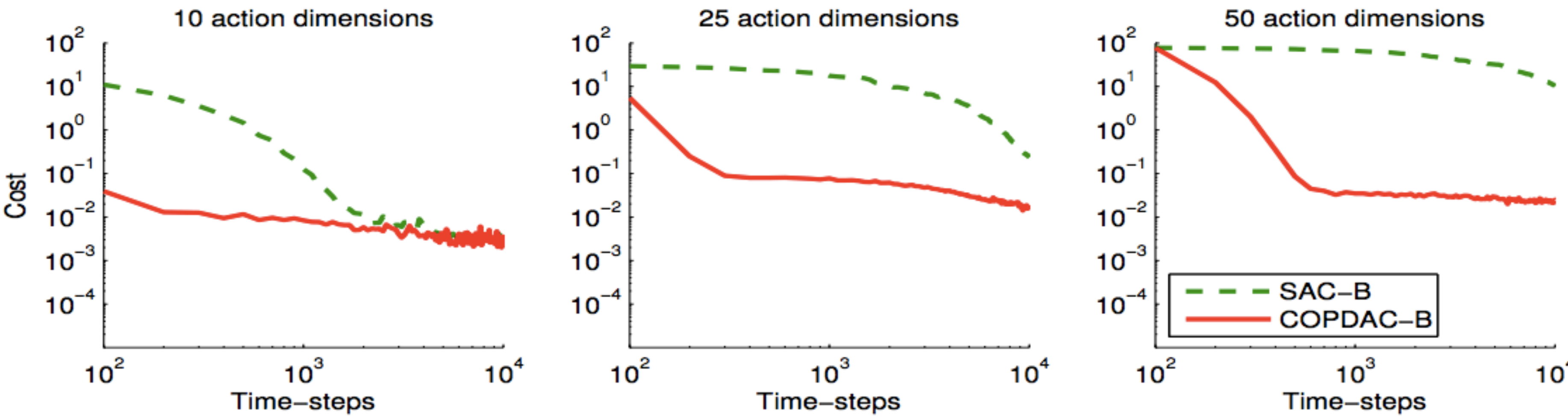
アルゴリズム:

- ガウス分布の方策を使用
- 互換な行動価値関数近似をする
- 方策オン/オフ型による方策の違いは以下

	確率方策(SAC-B)	確定方策(COPDAC-B)
ガウス分布の方策	平均と分散が学習される	分散固定で探索，平均が学習される

実験ー連続値バンディット

- 最適なパラメータ，ステップサイズでコストを比較
  - 5試行での平均
- 次元が増えるほど，確定方策のほうが良くなる



学習が進むほど勾配推定が困難になる確率方策より，確定方策なら速く収束する

実験ー連続値強化学習

目的：一般的なタスクでの比較

環境：

- 山登りカー，振り子，沼地，のベンチマーク
- エピソードは5000ステップ制限
- 割引率0.99(山登りカーと振り子)0.999(沼地)
- 領域外への移動は制限される

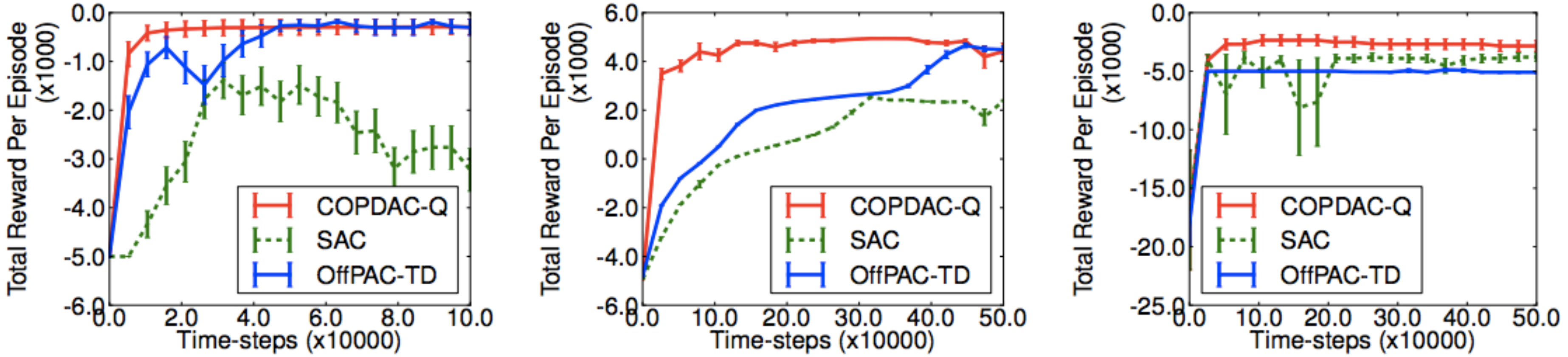
アルゴリズム

- 特徴量φは状態空間のタイルコーディングで計算される

	方策オン 確率方策	方策オフ 確率方策	方策オフ 確定方策
アルゴリズム	SAC(山登りカーでベストな手法(Degris2012a))	OffPAC	COPDAC-Q
方策	$\mathcal{N}(\theta^\top \phi(s), \exp(y^\top \phi(s)))$	学習方策：同左 探索方策：同右	学習方策： $\theta^\top \phi(s)$ 探索方策： $\mathcal{N}(\theta^\top \phi(s), \sigma_\beta^2)$

実験ー連続値強化学習

- 最適なパラメータでの比較
  - 30試行での平均
  - エピソード合計報酬の時間変化をグラフ化
- COPDAC-Qがすべての領域で勝っている

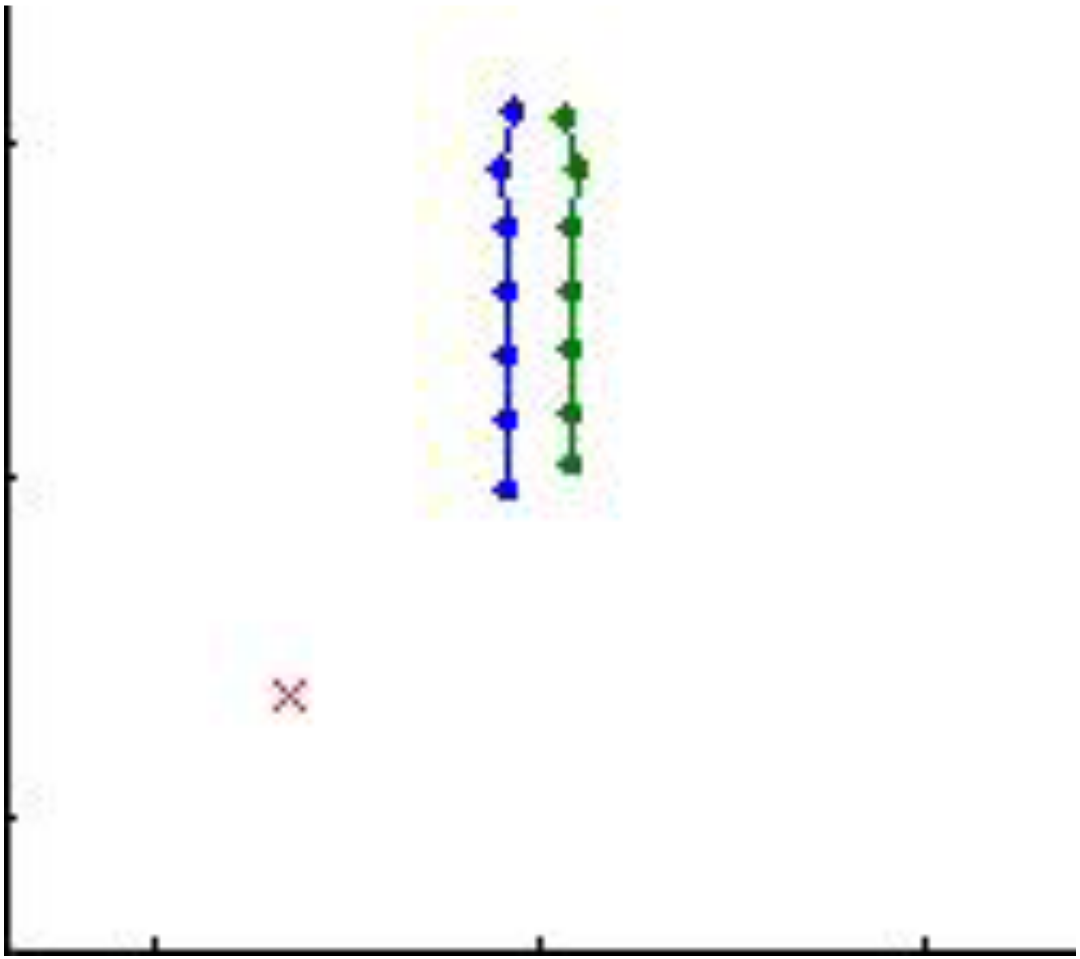


(a) Mountain Car (b) Pendulum (c) 2D Puddle World

実験ータコ腕

- 目標に当たるようシミュレーション上のタコの腕を制御する
  - 6関節に分かれ根本が固定されている
  - 50次元の状態空間
  - 20次元の行動空間(背側/腹側/中央)
  - 報酬は対象と腕の距離で変わる
  - 触れると報酬は+50かつエピソード終了
  - 300ステップで打ち切り
- COPDAC-Qを使用
  - 8ユニットのMLPを方策関数に使用
  - 状態価値関数は40ユニット線形出力のMLPで近似

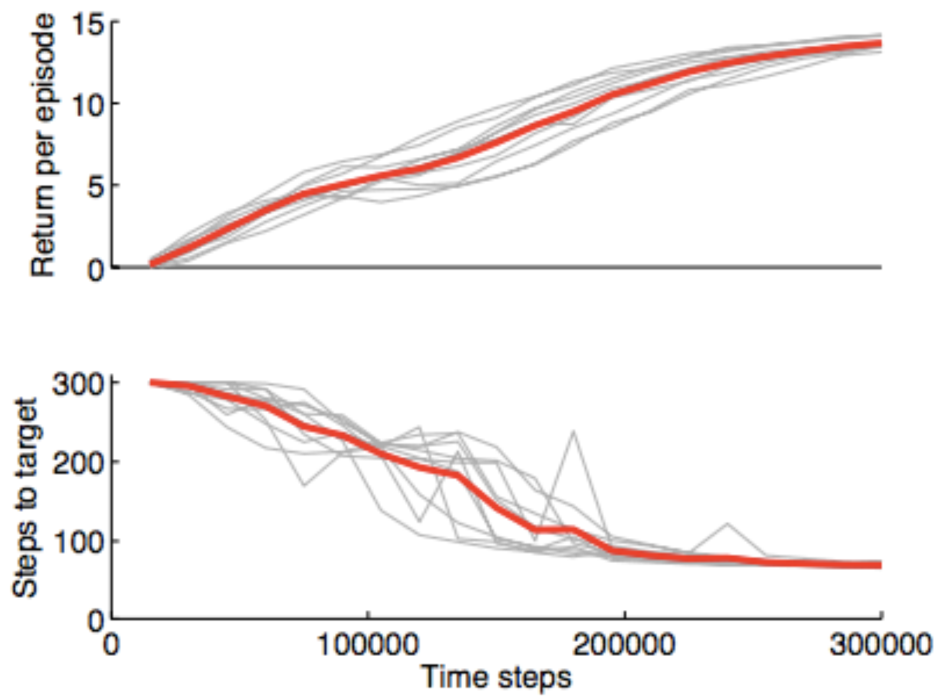
Octopus Arm Task



実験ータコ腕

10回の訓練と平均のグラフ

- すべての場合で制御が学習できている



従来手法では行動のマクロを使って単純化したり,
4関節までの低次元に制限したりしていた

→ 6関節までの学習に成功した

結論

- 確定方策勾配法のフレームワークを提案
- 行動空間の積分を避けて確率方策より効率よく勾配推定できる
- 実際に50次元の連続行動空間でのバンディットで圧倒的に勝利
- 50の状態空間，20の行動空間の難しい課題も解ける