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

深層強化学習による推薦システム

Deep Reinforcement Learning for Recommendation System

川島 崇 *1 Takashi Kawashima
川本 峻頌 *1 Syunyo Kawamoto
積田 大介 *1 Daisuke Tsumita
下山 翔 *1 Syo Shimoyama
宗政 一舟 *1 Isshu Munemasa
友松 祐太 *1 Yuta Tomomatsu
林 邦興 *2 Kunioki Hayashi
高木 友博 *1 Tomohiro Takagi

*1明治大学大学院理工学研究科情報科学専攻 Department of Computer Science, Meiji University
*2DesignOne Japan DesignOne Japan,Inc

Services that introduce stores to users on the Internet are increasing in recent years. Each service conducts thorough analyses in order to display stores matching each user's preferences. When sufficient data cannot be obtained, a multi-armed bandit algorithm is applied. Bandit algorithms advance learning by testing each of a variety of options sufficiently and obtaining rewards (i.e. feedback). It is practically impossible to learn everything when the number of items to be learned periodically increases. In order to solve this problem, we propose a recommender system based on deep reinforcement learning. In deep reinforcement learning, a multilayer neural network is used to update the value function.

1. はじめに

近年, インターネット上でユーザに対して店舗の紹介を行うサービスが増えてきている. 各サービスでは同時に, ユーザの嗜好に合った店舗を表示させる分析が幅広く行われてきている. 推薦の分野ではユーザのクリック情報が十分に存在する時には協調フィルタリングが高い性能を誇る. 一般的にユーザ×アイテムの行列を作成した際, データスパースの問題が発生するので新規ユーザに対応することが難しい. また十分にデータが得られなかった場合, バンディットアルゴリズムなどを応用しているケースが見られる. バンディットアルゴリズムは各アームを十分に試行してそれぞれから報酬を得ることで学習を進めていくため, アイテム数が多くなった場合に全てを学習するのは実質的に不可能である. 新たなユーザが出てきた時に十分にデータを集める必要性は協調フィルタリングと同様の問題がある. 上記の問題を解決すべく本稿では強化学習の価値関数の更新に多層ニューラルネットを用いた深層強化学習による推薦システムの提案を行う.

2. 関連研究

バンディットアルゴリズムではランダムで広告の配信を行う探索と期待値の高い広告を配信する活用の 2 つのフェーズが存在する. 探索, 活用のフェーズどちらにおいてもクリックされた場合 1, クリックされなかった場合 0 というフィードバックを受け取る. これを一般的に報酬と呼ぶ. バンディットアルゴリズムにおいて有名な手法として ϵ-greedy 法, softmax 法 [Richard 98], UCB 法 [P. Auer 02], そして Thompson Sampling[William 33] がある (Non Contextual-Bandit). Contextual なバンディットとして linUCB[Lihong 10] も提案されている.

バンディットアルゴリズムは強化学習の枠組みで議論することができる. 強化学習ではエージェントと環境を定義し, ある状態においてエージェントが行動したことによるフィードバック (報酬) と次の状態を環境から受け取る. また行動価値を定義し, この価値を最大化するようにエージェントの学習を進める (Q-learning)[Christopher 92]. 行動価値は即時報酬ではなく, 将来もらえる報酬を考慮して行動を選択する. バンディットアルゴリズムでは環境は行動によって変化せず, 広告表示後のフィードバックを直ぐに反映させるような即時報酬を最大化する. 対して Q-learning では一連の行動を行った結果得られた累積報酬を最大化する.

近年, 価値関数を多層ニューラルネットワークで表現する深層強化学習の研究が盛んに行われている.Deep Q Net(DQN)[Volodymyr 13] は深層強化学習でも有名な手法であるが, 行動空間と状態空間が連続値である場合に使用することが難しい. Deep Deterministic Policy Gradient(DDPG)[Timothy 16] はエージェントの状態空間や行動空間が連続値の場合でも高い性能を誇る手法であり本研究の推薦システムに用いた.

3. 問題設定

店舗情報サイト「エキテン」では, ユーザがエリアとジャンルをキーワードとして検索を行うと, キーワードにマッチする店舗が表示される. ユーザは表示される店舗の紹介文, 口コミ, 写真や評価を基に訪れたい店舗に問い合わせや予約ができる. 本研究ではエキテン内で検索が行われた際, ユーザに最適な店舗を表示する. その際, 表示した店舗がユーザの嗜好に近づくようにモデルの更新を行う. ユーザから得られる入力情報としてはエリアとジャンルであるが, ユーザが探しているエリアから離れすぎた店舗を推薦することは現実的ではない. よって推薦対象となるのは, エリアの周辺に存在する店舗である. 店舗の推薦は過去の店舗情報の詳細ページへのクリックログを基に深層強化学習の枠組みで学習が進められる. 本研究では DDPG を用い状態空間及び行動空間が連続の問題に対応する. エージェントの出力する店舗の学習が進むにつれてユーザの嗜好に近づいていくかを本研究では確認する.

連絡先: 川島崇, 明治大学大学院理工学研究科情報科学専攻, 〒 214-8571 神奈川県川崎市多摩区東三田 1-1-1, t_kawashima@cs.meiji.ac.jp

4. 深層強化学習による店舗推薦

店舗推薦をするために 深層強化学習を用いた. 深層強化学習では Q learning などの強化学習と同様, 行動, 状態, 報酬, エピソードの定義をする必要がある. エキテンが扱う店舗のレビューをコーパスとして LDA を用いて分散表現にする. エージェントの学習は行動ベクトルとして店舗の分散表現を出力するように進められる. 以下, 本節では店舗ベクトルの表現方法 (4.1 節), 深層強化学習に必要な要素の定義 (4.2 節), 本研究で用いる深層強化学習のアルゴリズム (4.3 節) を順に記述する.

4.1 店舗ベクトルの分散表現

店舗ベクトルを分散表現にするために Latent Dirichlet Allocation(LDA) を用いる. LDA は文書の確率的言語モデルとして, 単語の共起性を統計モデルとして数理的に扱うために提案された. 文書中の単語の順序は無視し, Bag of Words(BoW) と呼ばれる単語と出現頻度のペアの集合をモデル化する. LDA では 1 つの文書が N 個のトピックから成ることが仮定されているため, 各トピックの値が確率値になっている. よって各トピックの和は 1.0 になる. 本研究では店舗ごとのレビュー群を 1 文書として LDA を行い, 文書を推論した際の各トピックの確率値を要素とするような N 次元の分散表現を店舗ベクトル $\boldsymbol{item}$ とする.

4.2 強化学習の構成要素

本節では強化学習の構成要素であるエージェント, 環境そしてエピソードをどのように定義したかの説明を記述する.

4.2.1 エージェント

エージェントは時刻 t に環境から状態 $\boldsymbol{s}_t$ を受け取った時に N 次元の店舗ベクトルを行動 $\boldsymbol{a}_t$ を出力し, フィードバックとして報酬 $\boldsymbol{r}_t$ を受け取りモデルの更新を行う. $\boldsymbol{a}_t$ は行動空間 $\boldsymbol{A} \in [0,1]^N$ 上の要素である. 状態 $\boldsymbol{s}_t$ と報酬 $\boldsymbol{r}_t$ については後述する. 時刻 t に店舗を推薦する際, ユーザが入力したエリアとジャンルを基に推薦対象になる店舗集合 $\boldsymbol{Item}_t$ が決まる. エージェントが出力した行動 $\boldsymbol{a}_t$ と推薦対象になる店舗 $\boldsymbol{item}$ 間の類似度を算出する. 類似度計算には以下の方法を用いる.

$$\mathrm{sim}(\boldsymbol{a}_t, \boldsymbol{item}) = \frac{\sum_{n=1}^{N} a_{t,n} item_n}{\sqrt{\sum_{n=1}^{N} a_{t,n}^2}\sqrt{\sum_{n=1}^{N} item_n^2}} \tag{1}$$

推薦対象の店舗集合 $\boldsymbol{Item}_t$ をそれぞれ行動ベクトル $\boldsymbol{a}_t$ と式 (1) で定義した類似度を算出して, 類似度が最大となるような店舗 $\boldsymbol{recommend\_item}$ をユーザに対して表示する.

$$\boldsymbol{recommend\_item} = \underset{\boldsymbol{item} \in \boldsymbol{Item}_t}{\arg\max}\ \mathrm{sim}(\boldsymbol{a}, \boldsymbol{item}) \tag{2}$$

4.2.2 環境

環境はエージェントから受け取った店舗をユーザに対して表示した後, クリックしたか否かを基に計算した報酬 r_t と次の状態 $\boldsymbol{s}_{t+1}$ をエージェントに渡す.

報酬 r_t はエージェントが出力した行動ベクトル $\boldsymbol{a}_t$ と推薦対象になる店舗集合 $\boldsymbol{Item}_t$ の中で実際にクリックされた店舗 $\boldsymbol{clicked\_item}_t$ の類似度とする.

$$r_t = \frac{\sum_{n=1}^{N} a_{t,n} clicked\_item_{t,n}}{\sqrt{\sum_{n=1}^{N} a_{t,n}^2}\sqrt{\sum_{n=1}^{N} clicked\_item_{t,n}^2}} - 0.5 \tag{3}$$

上記のように報酬を定義する. 式 (3) の前半部分はコサイン類似度である. 行動空間は $\boldsymbol{A} \in [0,1]^N$ であるためコサイン類似度は 0 から 1 の間の値を取る. クリックされた店舗との類似度が低い推薦をした場合, 負の報酬を与えたい. そこでコサイン類似度から 0.5 を差し引くことで報酬 r_t は $r_t \in [-0.5, 0.5]$ になる. 本研究では報酬の与え方として 2 通りある. 1 つめは DDPG が出力した N 次元のベクトルとクリックされた店舗の類似度を報酬として与える方法である (r_{action_vec}). 2 つめは式 (3) における行動ベクトルを式 (2) で求めた $recommend\_item$ に置き換えて計算したものである (r_{item_vec}).

状態はユーザが時刻 $t-1$ までに閲覧した店舗と時刻 t に入力したエリアとジャンルを基に表現する. エリアは緯度経度の情報 $\boldsymbol{area}_t$ の 2 次元のベクトルで表現をする. 本研究ではグルメジャンルを対象に推薦を行う. グルメジャンルには 40 種類の小ジャンルがあり, ジャンルについては 40 次元の one-hot ベクトル $\boldsymbol{genre}_t$ とする. 過去の閲覧店舗ベクトルを $\boldsymbol{history}_t$ として, ユーザが直近で閲覧した店舗ベクトル j 件を加算して表現する. ただし, 直前に閲覧した店舗とそれよりも前に閲覧した店舗を比べた時に直前の方がユーザのリアルタイムな嗜好が反映されていると考えることができるため割引和 $\gamma \in [0,1]$ を採用する. この時, ユーザの閲覧履歴 $\boldsymbol{history}_t$ を以下のように定義する.

$$\boldsymbol{history}_t = \sum_{i=1}^{j} \gamma^i \boldsymbol{clicked\_item}_{\boldsymbol{t-i}} \tag{4}$$

店舗ベクトルは確率値で表現される. 閲覧履歴も確率値で表現することにする. 以下の式のように $\boldsymbol{history}_t$ を正規化する.

$$\begin{aligned}\mathrm{Normalized}(\boldsymbol{history}_t) &= \boldsymbol{normalized\_history}_t \\ &= \frac{\boldsymbol{history}_t}{\|\boldsymbol{history}_t\|}\end{aligned} \tag{5}$$

上記で定義した $\boldsymbol{area}_t$, $\boldsymbol{genre}_t$, $\boldsymbol{normalized\_history}_t$ を用いて状態ベクトル $\boldsymbol{s}_t$ を以下のように定義する.

$$\boldsymbol{s}_t = \mathrm{concat}\big(\boldsymbol{area}_t, \boldsymbol{genre}_t, \boldsymbol{normalized\_history}_t\big) \tag{6}$$

concat はベクトルを連結させるような関数とする.

4.2.3 エピソード

エピソードはユーザが最初にグルメジャンルの店舗が表示されるページに訪れた時に開始する. エピソード内の各ステップすなわち状態遷移はユーザがグルメジャンルの店舗が表示されるページに訪れる度に発生する. ユーザのページの遷移が 30 分間なかったらエピソード終了とする. その後, 同ユーザが新たに訪問した場合, 新たなエピソードとして開始させる. このようにエピソードを定義したのは同ユーザでも時間が空いた後の検索は目的や嗜好が異なる可能性が考えられるためである.

4.3 深層強化学習アルゴリズム

本研究では DDPG を用いて店舗推薦システムを作成した.DDPG は DQN と Deterministic Policy Gradient(DPG)[David 14] を組合せた手法である. DQN からは experience replay でサンプル間の相関を最小にし, ターゲット関数を用いることで学習を安定させる方法を取り入れている. DPG からは状態の価値を評価するクリティック関数と状態に応じて確率的に行動を選択するアクター関数を用いることで行動空間が連続値な場合でも扱える方法を取り入れ

ている．DDPG では時刻 t で replay buffer に格納されている現在の状態, 行動, 報酬, 次の状態を含むタプル集合から B_t 個のミニバッチをランダムにサンプリングしてきたものを使ってクリティック関数とアクター関数の更新を行う．B_t 個のサンプルに含まれるタプル集合の要素を $(s_{t,i}, a_{t,i}, r_{t,i}, s_{t,i+1})$ とする．また行動価値関数を $Q(s_{t,i}, a_{t,i})$ とするとクリティック関数は以下に示す損失関数 L を最小化するように更新される．

$$L = \frac{1}{B_t}\sum_i (y_{t,i} - Q(s_{t,i}, a_{t,i}|\theta^Q))^2 \tag{7}$$

ここでクリティックとアクターのターゲットネットワーク $\theta^{Q'}, \theta^{\mu'}$ より算出される行動価値と報酬 $r_{t,i}$ を用いて教師信号 $y_{t,i}$ を以下のように表す．γ は割引率で $\gamma \in [0, 1]$ である．

$$y_{t,i} = r_{t,i} + \gamma Q'(s_{t,i}, \mu'(s_{t,i+1}|\theta^{\mu'})|\theta^{Q'}) \tag{8}$$

アクター関数については以下の式で表される勾配を基に更新される．

$$\nabla_{\theta^\mu} J \approx \frac{1}{B_t}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_{t,i}, a=\mu(s_{t,i})} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_{t,i}} \tag{9}$$

バッチ内では先程記述した 2 つのターゲットネットワークを固定し, クリティックネットワーク θ^Q とアクターネットワーク θ^μ を常に更新するパラメータにして学習を進めることで安定させている．

5. 実験

本節では実験に関する記述を行う．実験ではエキテンで過去10ヶ月分の配信データを用いる．具体的には, ユーザが検索をして店舗が表示された際, クリックが発生したログを対象とする．クリックが発生しなかったログについては実験の対象外とする．5.1 では DDPG の実験設定の説明を行う．5.2 では比較手法の種類と実験設定の説明を行う．5.3 では評価方法について説明を行い, 5.4 で実験結果を示し, 5.5 で考察を行う．また, 本実験では店舗の分散表現の次元数を $N = 50$ としている．

5.1 提案手法

DDPG を使用する． 報酬の与え方を 4.2.2 で説明した r_{action_vec} と r_{item_vec} の 2 パターンで実験を行う．

5.2 比較手法

提案手法の優位性を検証するために推薦を Random に行う方法とバンディットアルゴリズムによる方法と比較する．

コンテキスト情報を用いないバンディットアルゴリズムとして ϵ-greedy($\epsilon = 0.1, 0.3$), UCB, thompson Sampling, softmax($\tau = 0.05, 0.2$) アルゴリズム, コンテキスト情報を使うバンディットアルゴリズムとして LinUCB($\alpha = 1.0, 1.4$) を採用した．報酬は比較手法では店舗を直接推薦するため,r_{item_vec} で与える．

5.3 評価方法

推薦アルゴリズムの評価に Mean Reciprocal Rank(MRR) と Recall@k を用いる．MRR は各ステップでクリックされたアイテムの逆順位を平均したものである．Recall@k は各ステップでクリックされたアイテムを上位 k 件以内に推薦できた割合である．実験では,k の値を 3 として評価する．

表 1: DDPG における報酬変更時の MRR と Recall@3 のスコア

method	reward_type	MRR	Recall@3
ddpg	r_{action_vec}	0.302	0.327
	r_{item_vec}	0.295	0.317

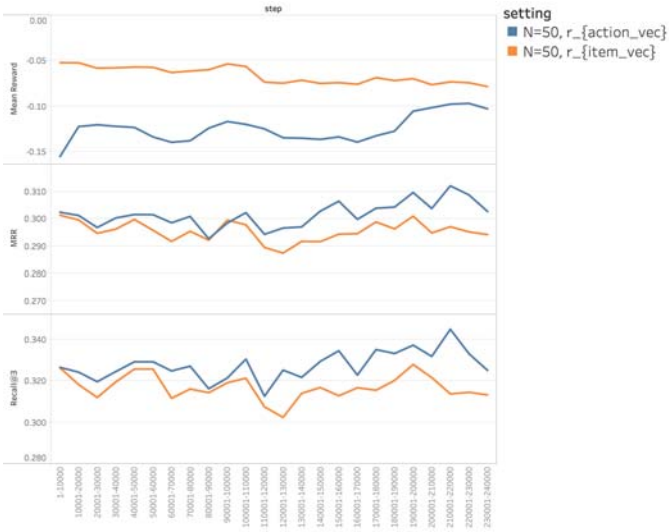


図 1: DDPG における報酬を変えたときの 10,000 ステップごとの平均報酬,MRR,Recall@3

5.4 実験結果

5.4.1 報酬の与え方による性能比較

DDPG に適した報酬の与え方を求めるための事前実験を行う． 報酬を r_{action_vec} と r_{item_vec} とした場合の比較実験を行った．表 1 では 2 種類の報酬で実験を行ったときの MRR と Recall@3 のスコアを示す．MRR と Recall@3 のスコアはどちらも r_{action_vec} が r_{item_vec} を上回っている．図 1 では DDPG アルゴリズムにおいて報酬を変えて実験を行ったときの 10,000 ステップごとの平均報酬, MRR, Recall@3 である．平均報酬に関して r_{action_vec} では増加傾向, r_{item_vec} では減少傾向が見られる．以上のことから報酬は r_{action_vec} を与えることによって学習が安定し, 各ステップで MRR,Recall@3 ともに高いスコアとなることがわかる．

5.4.2 既存手法との性能比較

5.4.1 節では DDPG の報酬設計の比較を行った．ここでは, DDPG と従来手法を比較することによって DDPG の優位性を示す．表 2 は各手法の全実験設定の結果である．提案手法の DDPG(r_{action_vec}) は MRR=0.302,Recall@3=0.327 でありバンディットアルゴリズムとランダム推薦の全ての条件の中で最も良いスコアであった．バンディットアルゴリズムで最もスコアが高かったのは softmax($\tau = 0.2$) であり, MRR=0.292, Recall@3=0.311 であった．図 2 では DDPG が他の手法に比べて学習初期段階から高い MRR 値と Recall@3 を示している．また, DDPG は学習が進んでもほぼ全てのステップにおいて MRR, Recall@3 ともに最大のスコアである．

5.5 考察

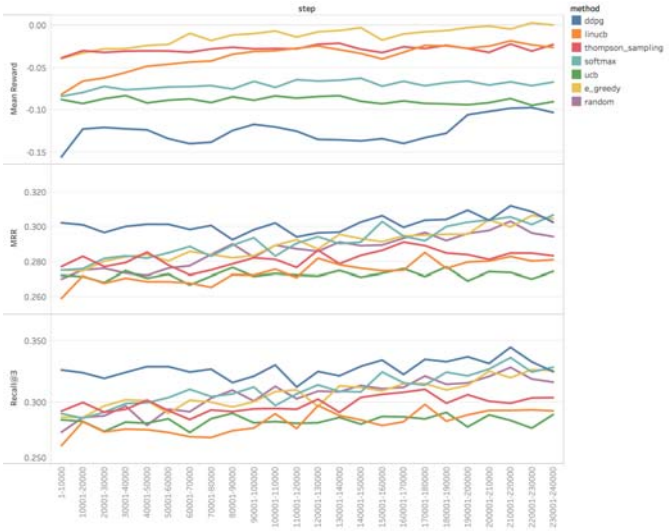
実験の結果から考察を行う．

まず, DDPG とバンディットアルゴリズムを比較すると, DDPG では学習初期段階から高い MRR と Recall@3 であった．DDPG では行動として店舗ベクトルを分散表現で出力している．その為, ある店舗を推薦した際に得られる報酬を他の店舗の推薦の際も活用できる．対して, バンディットアルゴリ

表 2: 各実験設定における MRR と Recall@3 のスコア

method_	hyper_parameter	reward_type	MRR	Recall@3
ddpg	-	r_{action_vec}	0.302	0.327
		r_{item_vec}	0.295	0.317
linucb	$\alpha = 1.0$		0.271	0.280
	$\alpha = 1.4$		0.268	0.276
thompson_sampling	-		0.281	0.296
softmax	$\tau = 0.2$		0.292	0.311
	$\tau = 0.05$		0.288	0.304
ucb	-		0.273	0.285
ϵ_greedy	$\epsilon = 0.1$		0.288	0.303
	$\epsilon = 0.3$		0.290	0.306
random	-	-	0.287	0.306

図 2: 各手法で最もスコアの高かった実験設定の 10,000 ステップごとの平均報酬,MRR,Recall@3



ズムでは各店舗に対してそれぞれ探索を行わなければならないため学習の速さに大きな差がある．これはシステムに深層強化学習を適用した効果であると考えられる．

最後に，報酬設計に関して r_{action_vec} が r_{item_vec} よりも安定した学習ができた．報酬設計を r_{item_vec} としたときに学習が安定しなかった理由としては，報酬を求める際に，出力した行動ベクトルを店舗ベクトルに置き換えていることに原因があり，行動ベクトルと店舗ベクトルの乖離が大きい場合に適切な報酬が与えられない可能性がある．特に学習初期段階では行動ベクトルがランダムに出力されるため発生しやすい．そのため 4.2.2 節で，r_{item_vec} によって報酬がうまく与えらず，ステップが増えるに連れて平均報酬が減少していったと考えられる．一方で r_{action_vec} ではステップが増えるに連れて平均報酬が増加し，それに伴い MRR と Recall@3 のスコアも増加した．

6. おわりに

本稿では，深層強化学習を用いた推薦システムのフレームワークを提案し従来手法よりも高い推薦精度を実現することができた．強化学習を用いることで状態遷移を考慮した推薦を行うことができた．さらに，以下の貢献がある．

まず，深層強化学習によって推薦する店舗を N 次元の分散表現のベクトルとして出力することができた．そのため，店舗を直接推薦するバンディットアルゴリズムと比較して効率よく学習を行うことを可能にし，結果として高い MRR 値と Recall 値となった．また本システムでは報酬がクリックされた店舗ベクトルとの類似度で与えられるため，推薦した店舗以外の店舗がクリックされた場合も報酬を与えることができた．そのためオフラインでも学習することが可能となった．3 つ目に店舗ベクトルはレビューから作られているため，新規店舗が出現した場合にもその店舗に対して LDA のベクトルを作ることが出来れば新規店舗も推薦可能である．

本稿では，状態に使う特徴や DDPG の細かいハイパーパラメータ等を検証できていない．将来これらを含めて深層強化学習による推薦手法について検証していく必要がある．

参考文献

[Richard 98] Richard S. Sutton, Andrew G. Brato: Reinforcement Learning, The MIT Press (1998)

[P. Auer 02] P. Auer, N. Cesa-Bianchi, P. Fischer: Finite-time analysis of the multiarmed bandit problem, Machine Learning (2002)

[William 33] William R. Thompson: ON THE LIKELIHOOD THAT ONE UNKNOWN PROBABILITY EXCEEDS ANOTHER IN VIEW OF THE EVIDENCE OF TWO SAMPLES, Oxford University Press on behalf of Biometrika Trust, Vol.25 (1933)

[Lihong 10] Lihong Li, Wei Chu, John Langford, Robert E. Schapire: A Contextual-Bandit Approach to Personalized News Article Recommendation, World Wide Web (2010)

[Christopher 92] Christopher J.C.H. Watkins: Q-Learning, Machine Learning (1992)

[Volodymyr 13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller: Playing Atari with Deep Reinforcement Learning, NIPS Deep Learning Workshop (2013)

[Timothy 16] Timothy P. Lillicrap, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra: CONTINUOUS CONTROL WITH DEEP REINFORCEMENT LEARNING, International Conference on Learning Representations (2016)

[David 14] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, Martin Riedmiller: Deterministic Policy Gradient Algorithms, International Conference on Machine Learning, JMLR (2014)