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

線形可解マルコフ決定過程のためのバッチ強化学習

Batch Reinforcement Learning for Linearly Solvable MDP

西智樹 *1 Tomoki Nishi　大滝啓介 *1 Keisuke Otaki　吉村貴克 *1 Takayoshi Yoshimura

*1豊田中央研究所
Toyota Central R&D Labs., Inc.

Linearly solvable Markov decision process (L-MDP) is an essential subclass of MDP to find a better policy efficiently. We first develop a novel batch reinforcement learning algorithm for L-MDP. The algorithm simultaneously learns a state value function and a predictor of state values at next step by using pre-collected data. We evaluate our method on traffic signal control domain in a single intersection with the traffic simulator SUMO. Our experiment demonstrates that our method finds the policy on the domain efficiently.

1. はじめに

線形可解マルコフ決定過程 (L-MDP) は Todorov によって提案されたマルコフ決定過程 (MDP) の一種である [Todorov 06]．L-MDP はベルマン方程式を，規定の行動を実施時のある状態と次状態に関する線形差分方程式に変形することで，ダイナミクスが既知の場合には最適政策を効率的に求められる．また文献 [Todorov 06] ではさらに，ダイナミクスが未知の場合の強化学習として Z-learning が提案され，Q-learning [Watkins 92] よりも効率的に政策を学習できることが報告されている．

一方で，Fitted Q-iteration [Riedmiller 05] に代表されるバッチ強化学習は，予め収集したデータに基づき，オフラインでより良い行動を学習することができる強力な枠組みの一つである．自動運転・交通制御など学習中に事故や渋滞を引き起こす可能性がある問題では，有用な学習手法である [Riedmiller 07]．筆者は，先に述べた L-MDP に対し，連続状態行動空間におけるエージェントのダイナミクスが既知の場合のバッチ強化学習 passive Actor-Critic を提案し，自動運転の混雑時の高速道合流において有用であることを示した [Nishi 17]．しかしながら，離散行動空間における L-MDP のためのバッチ強化学習はこれまで提案されておらず，交通信号機など行動が離散的に表現される場合には利用することができなかった．

そこで本稿では，離散行動空間における L-MDP のためのバッチ強化学習 Fitted Z-learning を提案する．本手法は，線形化されたベルマン方程式に基づく状態価値の学習と，ある状態である行動を選択した際の遷移先の状態価値の学習とを同時に行うことで政策を学習する．また提案手法の有効性を検証するために，シミュレーションを用いた 1 交差点での信号機制御で評価を行う．

2. 準備

2.1 マルコフ決定過程と Q-learning

MDP は $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ の 5 つの要素の組で表現できる．$\mathcal{S}$ は状態空間，$\mathcal{A}$ は行動空間，$T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ は状態遷移モデル，$R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ はコストモデル，γ はコストの 1 ステップ毎の割引率を表す．また政策 $\pi : \mathcal{S} \to \mathcal{A}$ は状態 $s \in \mathcal{S}$ の時に行動 $a \in \mathcal{A}$ を出力する関数である．マルコフ決定過程における基本的な問題設定は，状態 s が与えられた時に，目的関数 J を最小化する政策 $\pi^* : \mathcal{S} \to \mathcal{A}$ を発見することである．その時の目的関数は，

$$J \coloneqq \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_k \right],$$

で表される無限ホライゾンにおけるコストの累積和の期待値がよく用いられる．ここで，$\mathbb{E}_\pi[\cdot]$ は政策 π の下での期待値，r_k は時刻 k での即時コストを表す．政策 π の下での状態価値関数 V^π，及び行動価値関数 Q^π を下記のように定義すると，

$$V^\pi(s) \coloneqq \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right] \quad s \in \mathcal{S},$$

$$Q^\pi(s, a) \coloneqq \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a \right] \quad s \in \mathcal{S}, a \in \mathcal{A},$$

時刻 t における最適な政策 $\pi^*(s_t)$ 及び，その下での行動価値関数 $Q^*(s_t, a_t)$，状態価値関数 $V^*(s_t)$ はベルマン方程式

$$Q^*(s_t, a_t) = r(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, \pi^*(s_t))} \left[\gamma V^*(s_{t+1}) \right], \tag{1}$$

$$V^*(s_t) = \min_{a_t \in \mathcal{A}} Q^*(s_t, a_t),$$

$$\pi^*(s_t) = \arg\min_{a_t \in \mathcal{A}} Q^*(s_t, a_t), \tag{2}$$

を満たす．

モデルフリー強化学習の最もよく知られた方法である Q-learning は，ベルマン方程式に基づき計算される TD 誤差 $\mathcal{E}$ を減少させるように，行動価値関数 Q の推定値を更新する手法である．

$$\mathcal{E}_t = Q(s_t, a_t) - \left(r(s_t, a_t) + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a) \right).$$

2.2 線形可解マルコフ決定過程と Z-learning

L-MDP は $\langle \mathcal{S}, \mathcal{U}, T, R, \gamma \rangle$ の 5 つの要素の組で表現される．但し $\mathcal{U}$ は L-MDP における行動空間であり，他の要素は通常の MDP と同様である．L-MDP は通常の MDP に下記の 2 つの仮定を追加した MDP である．

連絡先: 西智樹，豊田中央研究所，愛知県長久手市横道 41-1，nishi@mosk.tytlabs.co.jp

1. $p(s_{t+1}|s_t, \mathbf{u}_{s_t}) \coloneqq p(s_{t+1}|s_t, \pi_0)\exp(\mathbf{u}_{s_t})$,
2. $r(s_t, \mathbf{u}_{s_t}) \coloneqq q(s_t) + \lambda \mathrm{KL}(p(s_{t+1}|s_t, \pi_0)||p(s_{t+1}|s_t, \mathbf{u}_{s_t}))$,

但し, $\mathbf{u}_{s_t} \in \mathbb{R}^{|\mathcal{S}|}$ は状態 s_t における行動ベクトル, $p(s_{t+1}|s_t, \pi_0)$ 及び $p(s_{t+1}|s_t, \mathbf{u}_{s_t})$ はそれぞれ予め設計者が決めた政策 π_0 に基づいて行動した時の状態遷移の確率分布，行動 $\mathbf{u}_{s_t}$ をとった時の状態遷移の確率分布，$q(s_t)$ は状態にのみ依存した状態コスト，$\mathrm{KL}(\cdot||\cdot)$ は KL-divergence．また λ は，状態コスト $q(s_t)$ に対する KL-divergence で表される行動コストの，相対的な大きさを調整するためのパラメータである．1 つめの仮定は，L-MDP における行動 $\mathbf{u}_{s_t}$ は，MDP の行動とは異なり，状態 s からある状態 s' への状態遷移確率をどれくらい変化させるかであり，その大きさが $\exp(\mathbf{u}_{s_t})$ で表されるとする仮定である．また 2 つめの仮定は，L-MDP の即時コスト関数 $r(s_t, \mathbf{u}_{s_t})$ は，状態コスト $q(s_t)$ と，$p(s_{t+1}|s_t, \pi_0)$ と $p(s_{t+1}|s_t, \mathbf{u}_{s_t})$ の KL-divergence で表される行動コスト，の 2 つに分解できるとする仮定である．

上記の仮定から，最適な政策 π_L^* は，

$$Z^*(s_t) = \exp(-q(s_t)/\lambda)\mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,\pi_0)}\left[(Z^*(s_{t+1}))^\gamma\right], \tag{3}$$
$$Z^*(s_t) \coloneqq \exp(-V^*(s_t)/\lambda),$$

で表される線形化ベルマン方程式を満たすことが導かれる (詳細は [Todorov 06] を参照のこと)．その時最適な政策 $\pi_L^*(s_t, s_{t+1})$ 及びその政策の下での状態遷移は，

$$\pi_L^*(s_t, s_{t+1}) = \left(V^*(s_t) - q(s_t) - \gamma V^*(s_{t+1})\right)/\lambda, \tag{4}$$
$$p(s_{t+1}|s_t, \pi_L^*) = \frac{p(s_{t+1}|s_t, \pi_0)Z^*(s_{t+1})}{\exp(q(s_t)/\lambda)Z^*(s_t)}, \tag{5}$$

となる．

政策 π_0 に従う際の状態遷移確率 $p(s_{t+1}|s_t, \pi_0)$ と状態コスト $q(s_t)$ が既知である場合には，離散状態行動空間における L-MDP の状態コスト V と最適政策 π_L^* は，式 (3) の Z^* についての固有値問題を解くことで求められる．また連続状態行動空間の場合においても，Z 関数 $Z(s_t)$ を基底関数の線形和により近似することで，効率的に学習できる [Todorov 09]．

また，状態遷移確率 $p(s_{t+1}|s_t, \mathbf{u}_{s_t})$ と即時コスト関数 $q(s_t)$ が未知である場合は，予め決められた政策 π_0 に従って行動した時のデータに基づいて，Z 関数を学習する Z-learning が適用できる．Z-learning は，線形化ベルマン方程式に基づき計算される TD 誤差 $\mathcal{E}_L$ を減少させるように，Z 関数の推定値を更新する．

$$\mathcal{E}_{L,t} = Z(s_t) - \exp(-q(s_t)/\lambda)(Z(s_{t+1}))^\gamma.$$

3. 提案手法 Fitted Z-learning

L-MDP のためのバッチ強化学習として，Fitted Z-learning を提案する．Z-learning は，既定の政策 π_0 に従って行動した際の状態遷移のデータに基づき，Z 関数を学習することができる．また，状態コストを既知とすると，L-MDP における政策 π_L は，式 (4) により学習した Z 関数を用いて求められる．さらに $p(s_{t+1}|s_t, \pi_0)$ を既知とすると，政策の下での状態遷移は，式 (5) により算出できる．しかしながら，L-MDP における行動は状態遷移に与える影響の強さであり，エージェントが具体的に取り得るどの行動（MDP の意味での行動）を取れば，式 (5) から算出される望ましい状態遷移を実現することができるかは自明ではない．

行動価値関数 $Q^*(s_t, a_t)$ が学習出来れば，MDP における政策 π^* は式 (2) により求められる．ここで，即時コスト $r(s_t)$ が状態コスト $q(s_t)$ で近似できると仮定する．これは，λ を小さく設定することで常に実現できる．

$$r(s_t, a_t) = q(s_t) + \lambda \mathrm{KL}(p(s_{t+1}|s_t, \pi_0)||p(s_{t+1}|s_t, \mathbf{u}_{s_t})) \approx q(s_t). \tag{6}$$

この時式 (6) の仮定により行動価値関数は，

$$\begin{aligned} Q^*(s_t, a_t) &= r(s_t, a_t) + \mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,\pi^*)}\left[\gamma V^*(s_{t+1})\right], \\ &\approx q(s) + \mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,\pi^*)}\left[\gamma V^*(s_{t+1})\right], \end{aligned}$$

と近似できる．また近似した行動価値関数に基づく政策 $\hat{\pi}$ は，

$$\begin{aligned} \hat{\pi}(s_t) &= \arg\min_{a_t\in\mathcal{A}}\left(q(s_t) + \mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,\hat{\pi})}\left[\gamma V(s_{t+1})\right]\right), \\ &= \arg\min_{a_t\in\mathcal{A}}\mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,\hat{\pi})}\left[V(s_{t+1})\right], \\ &= \arg\max_{a_t\in\mathcal{A}}\mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,\hat{\pi})}\left[Z(s_{t+1})\right], \end{aligned}$$

と書き直すことができる．この政策は，状態 s_t において，次状態 s_{t+1} での Z 関数の期待値が最大となる行動を選択することを意味している．つまり，即時コスト $r(s_t)$ が状態コスト $q(s_t)$ で近似できる場合には，状態 s_t においてある行動 a_t をとった時の次状態 s_{t+1} での Z 値を予測することで，行動を決定することができる．

提案する Fitted Z-learning は，2 つの関数，

- 現状態 s_t の Z 値を推定する関数 $\hat{Z}(s_t; \boldsymbol{\theta})$,
- 行動 a_t を選択した時の次状態 s_{t+1} での Z 値を予測する関数 $\hat{Z}(s_t; a_t, \boldsymbol{\phi})$,

の学習を通して政策 π を学習する．ただし，$\boldsymbol{\theta}$ 及び $\boldsymbol{\phi}$ は共に関数近似のためのパラメータである．Z-learning と同様に，$\boldsymbol{\theta}$ は，L-MDP におけるベルマン方程式から算出される TD 誤差を小さくするよう学習する．$\boldsymbol{\phi}$ は，時刻 t で行動 a_t を選択した時の次状態 s_{t+1} の $\hat{Z}(s_{t+1}; \boldsymbol{\theta})$ の値と $\hat{Z}(s_t; a_t, \boldsymbol{\phi})$ との最小 2 乗誤差を小さくするように学習する．本稿ではさらに multi-step Q-learning [Peng 96] を参考に，連続した T ステップの状態遷移を考慮してパラメータの更新を行う multi-step Fitted Z-learning を用いた．

詳細なアルゴリズムを Algorithm 1 に示す．

4. 数値実験

本節では 1 交差点における信号機制御タスクにより Fitted Z-learning の有効性を検証する．

4.1 ニューラルネットワークの構成

Fitted Z-learning の $\hat{Z}(s; \boldsymbol{\theta})$ 及び $\hat{Z}(s; a, \boldsymbol{\phi})$ をニューラルネットワークでモデリングし，学習した．ニューラルネットワークの構成を図 1 に示す．$\hat{Z}(s; \boldsymbol{\theta})$ と $\hat{Z}(s; a, \boldsymbol{\phi})$ は，状態 s の Z 値と，状態 s で行動 a を選択した時の次状態の Z 値の推定値であり，この 2 つの Z 値は相互に関連するため，ニューラルネットワークの多くのパラメータは共有出来ると予想される．そこで Dueling Network Architectures [Wang 15] を参考に，図 1 のように下位層のネットワークを共有したフォーク

Algorithm 1 multi-step Fitted Z-learning

procedure 予め収集したデータセット $\mathcal{D}$ から政策を学習
  パラメータ $\boldsymbol{\theta}$ 及び $\boldsymbol{\phi}$ を初期化
  for エピソード $i = 1$ to N **do**
    $\mathcal{D}$ から T ステップ連続したデータをランダムに抽出:
      $\{(i, s_1, a_1, s_2, q_1), ..., (i, s_T, a_T, s_{T+1}, q_T)\} \sim \mathcal{D}$
    パラメータの勾配を初期化: $d\boldsymbol{\theta} \leftarrow \mathbf{0}$
    $R \leftarrow \hat{Z}(s_{T+1}; \boldsymbol{\theta})$
    for ステップ $t = T$ to 1 **do**
      $R \leftarrow \exp(-q_t/\lambda)R^{\gamma}$
      TD 誤差 $\mathcal{E}_{L,t}$ を計算: $\mathcal{E}_{L,t} \leftarrow \left(R - \hat{Z}(s_t; \boldsymbol{\theta})\right)$
      パラメータの勾配 $d\boldsymbol{\theta}$ を更新: $d\boldsymbol{\theta} \leftarrow d\boldsymbol{\theta} + \partial \mathcal{E}_{L,t}^2 / \partial \boldsymbol{\theta}$
      誤差 δ_t を計算:
        $\delta_t \leftarrow \left(\hat{Z}(s_{t+1}; \boldsymbol{\theta}) - \hat{Z}(s_t; a_t, \boldsymbol{\phi})\right)$
      パラメータの勾配 $d\boldsymbol{\phi}$ を更新: $d\boldsymbol{\phi} \leftarrow d\boldsymbol{\phi} + \partial \delta_t^2 / \partial \boldsymbol{\phi}$
    end for
    パラメータ $\boldsymbol{\theta}, \boldsymbol{\phi}$ を $d\boldsymbol{\theta}/T, d\boldsymbol{\phi}/T$ に基づき更新
  end for
end procedure

型のネットワークを用いた．つまり，下位層のパラメータは，$\boldsymbol{\theta}$ 及び $\boldsymbol{\phi}$ で同じ値を用い，$\boldsymbol{\theta}$ 及び $\boldsymbol{\phi}$ の更新時に更新される．また，全ての中間層は，全ノードが結合された全結合層 (Fully Connected Layer, FCL) を用い，最終層以外の活性化関数は全て ReLU$(\cdot) \coloneqq \max(0, \cdot)$ を用いた．出力層の活性化関数は，$Z(s) > 0, \forall s \in \mathcal{S}$ であるため，$\hat{Z}(s; \boldsymbol{\theta})$ 及び $\hat{Z}(s; a, \boldsymbol{\phi})$ 共に，Softplus$(\cdot) \coloneqq \log(1 + \exp(\cdot))$ を用いた．

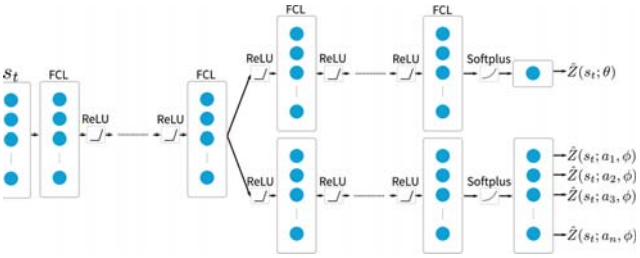

図 1: Fitted Z-learning のためのネットワークの構成

4.2 実験設定

従来手法 Q-learning と提案手法 Fitted Z-learning に対し，学習の安定性，速度，性能の 3 つの観点での比較評価を行った．評価タスクとしては，上下左右に伸びた 4 道路各方向 1 車線ずつ，計 8 車線が結合している道路（図 2）において，交差点の信号機を制御し，全道路の交通量を最大化するタスクを用いた．以下で，評価タスクの詳細を述べる．

各道路は直進のみができ，右左折できない道路とした．また状態は各車線を走行する全車両の平均速度（8 状態変数）及び各車線で信号待ちしている車両の台数（8 状態変数）の計 16 次元のベクトルで表現し，行動は（縦方向，横方向）の道路の信号機の色がそれぞれ（青，赤）または（赤，青）の 2 つとした．各車両の出発地及び目的地はランダムに各車線の端点を選択した．信号を制御した際の交通流をシミュレートするために交通流シミュレータ SUMO [Krajzewicz 02] を用いた．また Fitted Z-learning は，SUMO により生成された固定サイクル長の信号機で制御した 5,000,000 秒（約 1400 時間）のデータを用いて 40,000 回ニューラルネットワークのパラメータを更新した．本実験では，実際の街中で一般に用いられている固定サイクル長の信号機，Q-learning により学習された信号機

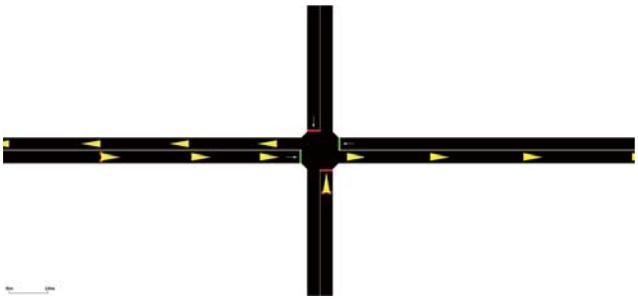

図 2: 1 交差点の信号機制御タスク．上下左右に伸びた 4 道路各方向 1 車線ずつの計 8 車線が結合した交差点の信号機を制御するタスク．三角で示された車両が交差点通過するために待つ時間を最小化するような制御方法を学習する．

と比較実験を行った．Q-learning は Fitted Z-learning で用いたニューラルネットワークと層数が同じになるよう中間層が 4 層のニューラルネットワークを用いた．

4.3 実験結果

4 道路が各方向 1 車線ずつ計 8 車線が結合された交差点の信号機制御を Q-learning, Fitted Z-learning で各 20 回学習した．まず 40,000 ステップ以内に学習初期の政策の平均待ち時間に対して，8 割以下の待ち時間を達成できた場合を学習の成功とし，各手法の成功率を図 3 に示す．この結果から Fitted Z-learning は Q-learning に比べ安定して学習ができることが分かった．これは，Q-learning は状態と行動を入力とする状態行動価値関数を学習する必要があるのに対し，Fitted Z-learning は状態のみを入力とする Z 関数の学習を通して政策を学習するため，学習が容易になり，学習が安定したのではないかと推察される．

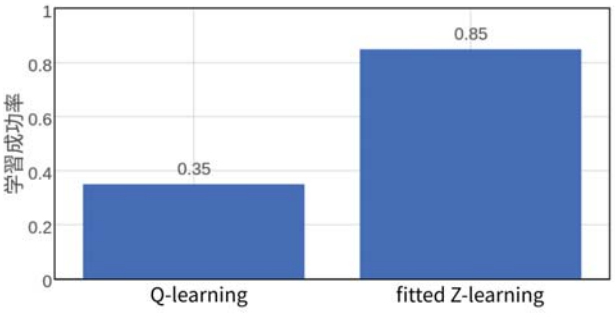

図 3: 学習の成功率

また，学習が成功した時のみの学習曲線の平均，及び最後の繰り返し回数時点での性能の平均値と標準偏差を，それぞれ図 4 に示す．図 4(左図) から Fitted Z-learning は予め収集したデータのみから Q-learning と同等の，固定サイクルに比べ 2 秒以上短い待ち時間を達成する政策を学習できた．また図 4(右図) から，街中で一般に見られる固定サイクルの信号機で制御された交差点のデータを用いることで，Fitted Z-learning は Q-learning に比べ効率的にパラメータを学習できることが分かった．これも上記の学習が安定した理由と同様，Fitted Z-learning の方が学習が容易であったためではないかと推察される．

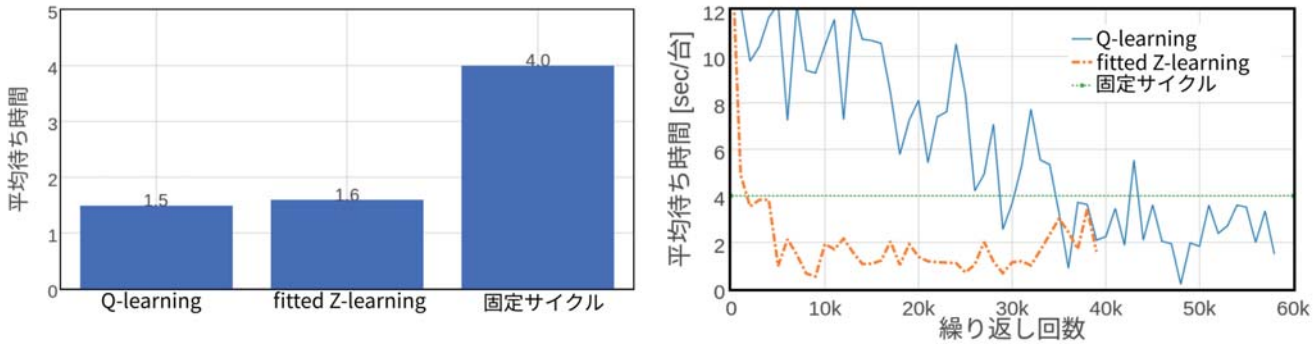

図 4: 1 交差点信号機制御の実験結果．(左図) 学習後の各手法で信号制御した場合の平均待ち時間．Fitted Z-learning は予め収集したデータのみから Q-learning と同等の，固定サイクルに比べ 2 秒以上短い待ち時間を達成する政策を学習できた．(右図)Fitted Z-learning および Q-learning の学習曲線．Fitted Z-learning は Q-learning に比べ少ない繰り返し回数で同等の政策を学習することができた．但し Q-learning は繰り返し回数 4,0000 の時点では収束していなかったため，60,000 まで学習を継続した．

5. 結論

本稿では，離散行動空間における線形可解マルコフ決定過程のためのバッチ強化学習として Fitted Z-learning を提案した．Fitted Z-learning は線形可解マルコフ決定過程から導かれる線形化ベルマン方程式に基づきマルコフ決定過程での状態価値関数に相当する Z 関数と共に，ある行動を選択した時の次状態での Z 関数の値を予測する関数を同時に学習する．提案手法を評価するために 1 交差点の信号機制御で実験を行った．その結果，提案手法はデータのみから固定サイクルに比べ，2 秒以上待ち時間が短い政策を Q-learning より効率的に学習できることが確認できた．

参考文献

[Krajzewicz 02] Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P.: SUMO (Simulation of Urban MObility) An open-source traffic simulation Car-Driver Model, *Proc of the 4th Middle East Symposium on Simulation and Modelling*, pp. 183–187 (2002)

[Nishi 17] Nishi, T., Doshi, P., James, M., and Danil, P.: Actor Critic for Linearly-Solvable Continuous MDP with Partially Known Dynamics, *arXiv preprint:1706.01077* (2017)

[Peng 96] Peng, J. and Williams, R. J.: Incremental multi-step Q-learning (1996)

[Riedmiller 05] Riedmiller, M.: Neural fitted Q iteration - First experiences with a data efficient neural Reinforcement Learning method, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2005)

[Riedmiller 07] Riedmiller, M., Montemerlo, M., and Dahlkamp, H.: Learning to Drive a Real Car in 20 Minutes, in *2007 Frontiers in the Convergence of Bioscience and Information Technologies*, pp. 645–650, IEEE (2007)

[Todorov 06] Todorov, E.: Linearly-solvable Markov decision problems, *Advances in neural information processing systems*, pp. 1369–1376 (2006)

[Todorov 09] Todorov, E.: Eigenfunction approximation methods for linearly-solvable optimal control problems, in *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL 2009 - Proceedings*, pp. 161–168 (2009)

[Wang 15] Wang, Z., Schaul, T., Hessel, M., Hasselt, van H., Lanctot, M., and Freitas, de N.: Dueling Network Architectures for Deep Reinforcement Learning, No. 9 (2015)

[Watkins 92] Watkins, C. J. C. H. and Dayan, P.: Q-learning, *Machine learning*, Vol. 8, No. 3-4, pp. 279–292 (1992)