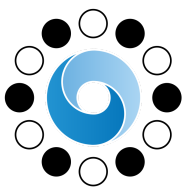
Robust probabilistic target-oriented exploration with reliability approximation

Tokyo Denki University

Moto Shinriki, Yu Kono, Tatsuji Takahashi

背景

強化学習は囲碁やチェスにおいて
超人的なレベルに達している



AlphaGo ＞

タスクの複雑性

環境の複雑性という観点では
実世界のタスクの方が高い

>

強化学習と人間の学習

【課題】強化学習は未だ実運用するにはハイコストである

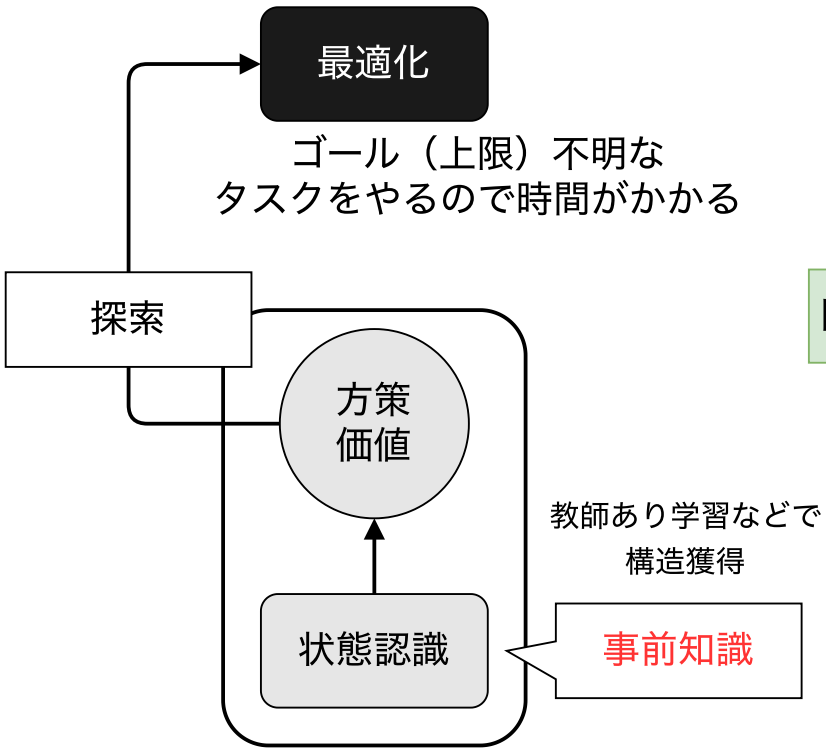
- 最適化のためのサンプリング時間が現実的な時間内に収まらない
- 最適化のための探索時間が現実的な時間内に収まらない

【発想】人間の学習方法に倣うことで解決できるのではないか？

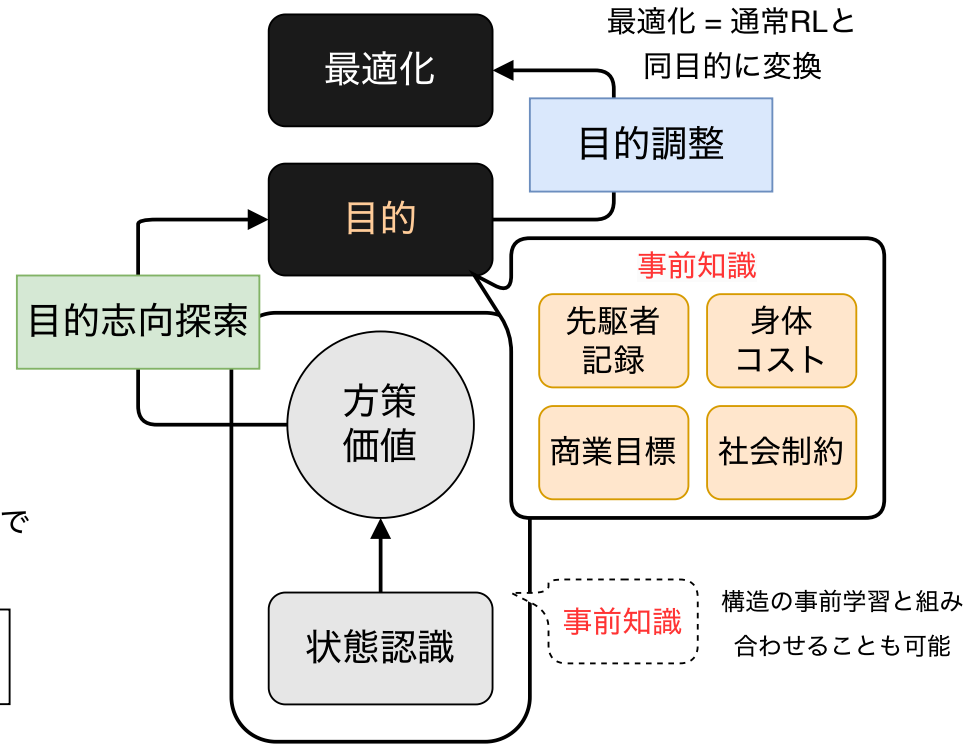
- 人間の学習傾向である **満足化** を参考にする
- 強化学習に希求水準を導入する
- 強化学習の目指すところを最適化ではなく満足化とする（しかし、最適化とすることも可能）

希求水準への到達（満足すること）を目指し学習を行う**目的志向探索**の実装

既存の強化学習



目的ベース強化学習



先行研究

Risk-sensitive Satisficing (RS)

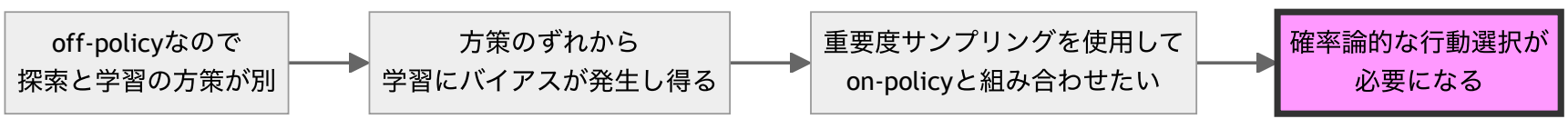
- 概要：強化学習に希求水準を組み込んだ目的志向型強化学習
- 性質：off-policy, 決定論的な行動選択
- 特徴：バンディット問題、強化学習問題において他の手法よりも良い性能
  - Takahashi et al., 2016
  - Tamatsukiri et al., 2019

先行研究

Regional Linear RS (RegLinRS)

- 概要：状態を識別できるRS手法の1つ
- 性質：off-policy, 決定論的な行動選択
- 特徴：文脈付きバンディット問題においてLinUCBやLinTSよりも良い性能
  - Minami et al., 2022

深層強化学習に拡張することを考えると、行動選択は確率論的な方が好ましい



本研究の目的

【大目的】 RSの深層強化学習への適用

【小目的】 RegLinRSの確率論的な行動選択への一般化

- 拡張した手法が性能劣化なく確率論的に一般化できていることを示す
- LinUCB, LinTSと同等以上の性能を示すことを目指す

Contextual Bandit Problems

- 本研究で使用するのは線形文脈付きバンディット問題
- 文脈 $\mathbf{x}_t$ と重みによって各行動 $A = \{a_1, a_2, \cdots, a_K\}$ の報酬期待値 $P_t = \{p_{t,1}, p_{t,2}, \cdots, p_{t,K}\}$ が決定
- エージェントは時刻 t において文脈 $\mathbf{x}_t$ を観測し、行動 a_i を選択
  - その結果エージェントは報酬 r_t を観測（本研究では報酬期待値 $p_{t,i}$ とする）
- $p_{t,i}$ の算出方法は下記の通り
  - θ_i は報酬期待値のパラメータ
  - ϵ_t は期待値0の誤差項

$$p_{t,i} = \mathbf{x}_t^T \theta_i + \epsilon_t$$

文脈付きバンディット問題においていい成績を残している手法

- LinUCB (Li et al., 2010)

Regret

- 本研究では評価指標として **regret** を使用する
  - 期待される報酬の損失

$$\text{regret} = \sum_{t=1}^{T} \left(p_{\max} - p_{t,\,\text{chosen}} \right)$$

    - $p_{\max}$: 最も高い報酬期待値
    - $p_{t,\,\text{chosen}}$: t 回目で選択された行動の報酬期待値

- regret の性質
  - regret はエージェントの期待損失の値であり、広義単調増加関数である
  - regret の最小値は0 (最適な行動を選び続けた場合)

目的志向探索の実装: 主観的な regret

- 目的志向探索であれば regret ではなく主観的な指標を使用することができる
- 主観的な regret (SR) とする

$$I_i^{\mathrm{SR}} = \sum_{t=1}^{T} (\aleph - p_t^{\mathrm{select}})$$

  - p_t^{select}: 選択した行動の報酬期待値
  - $\aleph$: 希求水準

- SR の性質
  - 獲得した報酬が $\aleph$ 以上（つまり十分である）の場合、I_i^{SR} は減少する
  - 獲得した報酬が $\aleph$ 未満（つまり不十分である）の場合、I_i^{SR} は増加する

- この指標はリスク考慮した価値関数として解釈することもできる

目的志向探索の実装: Risk-sensitive Satisficing (RS)

- 目的志向探索の核となる指標の数式
- $I_i^{\mathrm{RS}} := -I_i^{\mathrm{SR}}$ として、RS の価値関数を定義する
- I_i^{RS} から argmax を取ることで行動選択を行う

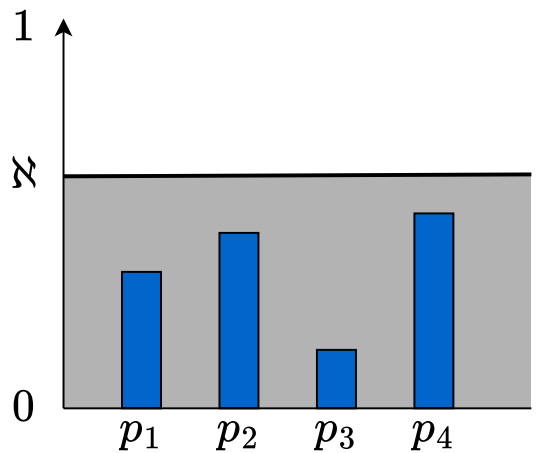
$$I_i^{\mathrm{RS}} = \frac{n_i}{N}(p_i - \aleph) = \frac{n_i}{N}\delta_i$$

- $\aleph$: 希求水準
- p_i: 行動 a_i の報酬期待値
- n_i: 行動 a_i を選択した回数
- N: 現在までの行動選択回数
- n_i/N: 信頼度、試行割合

非達成状況と達成状況

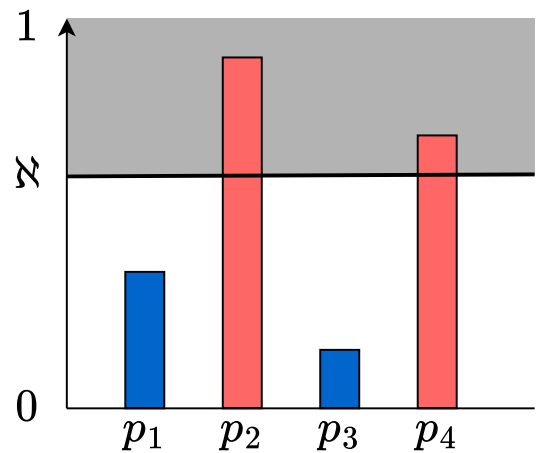
非達成状況

全ての報酬期待値が $\aleph$ 未満である状態



達成状況

いずれかの行動の報酬期待値が $\aleph$ 以上である状態



探索領域

信頼度の反射効果

達成状況と非達成状況の選択傾向の違い

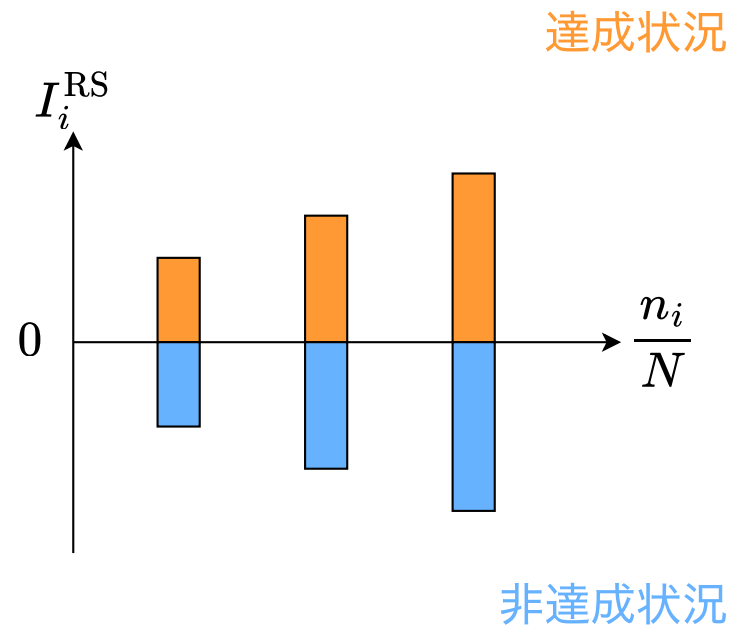
$$I_i^{\mathrm{RS}} = \frac{n_i}{N}(p_i - \aleph)$$

達成状況

- 信頼度が高ければ高いほど I_i^{RS} は大きくなる
- 達成状況では悲観的な行動選択を行う

非達成状況

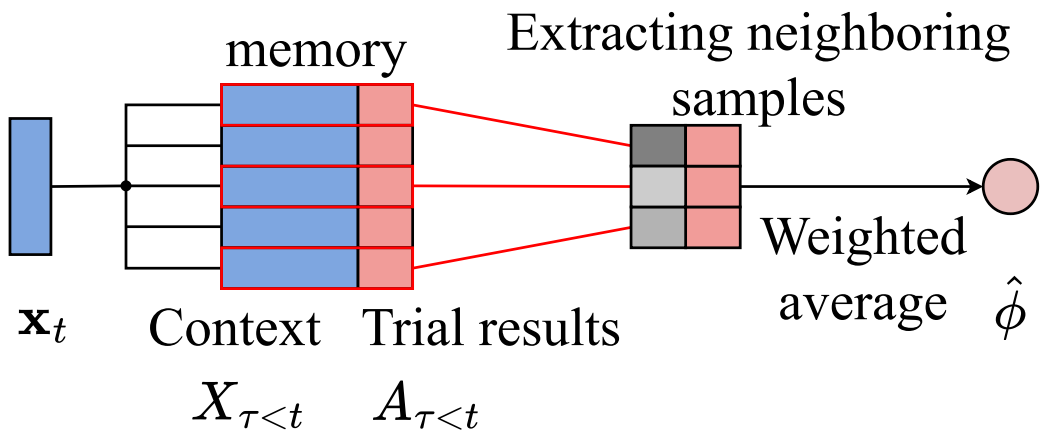
- 信頼度が高ければ高いほど I_i^{RS} は小さくなる
- 非達成状況では楽観的な行動選択を行う



→ 達成状況では必ずしも報酬期待値が最大の行動を選択するわけではない

信頼度の近似推定

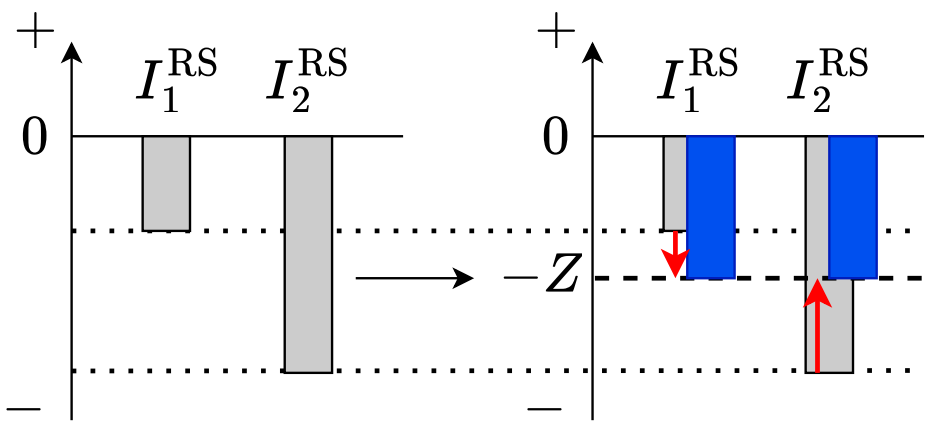
- episodic memoryとk近傍を用いて信頼度を近似推定する
- **Regional Linear RS** (RegLinRS)
  - Minami et al., 2022で提案された手法



RS の選択分布を逆算

- 非達成時においてRSは内部選択比率が推定可能である
- 推定した信頼度と実際の信頼度の差分から確率分布を生成
  - RS平衡値 $-Z$ を用いて推定信頼度 ρ_i^z を生成する
- **Stochastic RS** (SRS)

RS平衡値 $-Z$



$$I_i^{\mathrm{RS}} = -Z$$

$$\rho_i = n_i/N = Z/(\aleph - p_i)$$

$$\sum_{i=1}^{K} \rho_i = \sum_{i=1}^{K} Z/(\aleph - p_i) = 1$$

$$Z = 1/\sum_{i=1}^{K} \frac{1}{\aleph - p_i}$$

SRS の方策算出

非達成状況

- 非達成状況は $-Z$ が算出可能
- 下記の式から方策を算出することができる

$$
\begin{aligned}
b_i &= \frac{n_i}{\rho_i^z} - N + \epsilon \\
I_i^{\mathrm{SRS}} &= (N + \max_i(b_i))\rho_i^z - n_i > 0 \\
\pi_i &= I_i^{\mathrm{SRS}} / \sum_{i=1}^{K} I^{\mathrm{SRS}_i}
\end{aligned}
$$

- b は負の割合を防ぐための調整パラメータ
- ϵ はゼロ除算を防ぐための極僅かな値

達成状況

- 達成状況は $-Z$ が算出不可能
- $\aleph$ を超える報酬期待値を持つ行動があるのでそれを選択するような確率を算出する

$$
\begin{aligned}
I_i^{\mathrm{RS}'} &= \begin{cases} I_i^{\mathrm{RS}} + \epsilon, & \text{if } p_i \geq \aleph \\ 0, & \text{if } p_i < \aleph \end{cases} \\
\pi_i &= I_i^{\mathrm{RS}'} / \sum_{j=1}^{K} I_j^{\mathrm{RS}'}
\end{aligned}
$$

Regional Linear SRS

- **提案手法**
- RegLinRS の信頼度推定部分と SRS を組み合わせた手法
  - SRS の式には n_i と N が含まれている
  - n_i/N は近似できるが、n_i と N は近似できない

SRS の式変形

$$\frac{b_i}{N_x} = \frac{1}{\rho_i^z} \cdot \frac{n_i}{N_x} - 1 + \epsilon = \frac{\rho_i}{\rho_i^z} - 1 + \epsilon$$

$$\begin{aligned}\frac{I_i^{\mathrm{SRS}}}{N_x} &= \left(\frac{N_x + \max_i(b_i)}{N_x}\right) \\ &= \left\{\max_i\left(\frac{\rho_i}{\rho_i^z}\right) + \epsilon\right\}\rho_i^z - \rho_i\end{aligned}$$

$$\frac{I_i^{\mathrm{SRS}}}{N_x} = \left\{\max\left(\frac{\hat{\phi}_i}{\rho_i^z}\right) + \epsilon\right\}\rho_i^z - \hat{\phi}_i$$

$$\pi_i = \frac{I_i^{\mathrm{SRS}}}{N_x} \Big/ \sum_{j=1}^{K} \frac{I_j^{\mathrm{SRS}}}{N_x} = I_i^{\mathrm{SRS}} \Big/ \sum_{j=1}^{K} I_j^{\mathrm{SRS}}$$

→ SRS を RegLinSRS に拡張することができる

人工データセット

- 希求水準 $\aleph$ が常に一定である人工データセットを作成
  - RegLinRS,RegLinSRSと他の手法との比較を同じ評価指標で行うため

- Minami et al., 2022 のデータセットと同じデータセットを使用した
- 探索と活用のバランスを適切に評価するために、最適な行動が偏らないようにデータセットを設計した

設定項目	設定値
特徴ベクトル次元数 d	128
行動数 K	8
最適希求水準 $\aleph_{\mathrm{opt}}$	0.7
データサイズ N	100,000

※ $\aleph_{\mathrm{opt}}$ は最適と準最適の間に設定した基準値のこと、詳細は論文参照

Experiment 1

- 比較手法
  - LinUCB
  - LinTS
  - RS系
    - RegLinRS
    - RegLinSRS

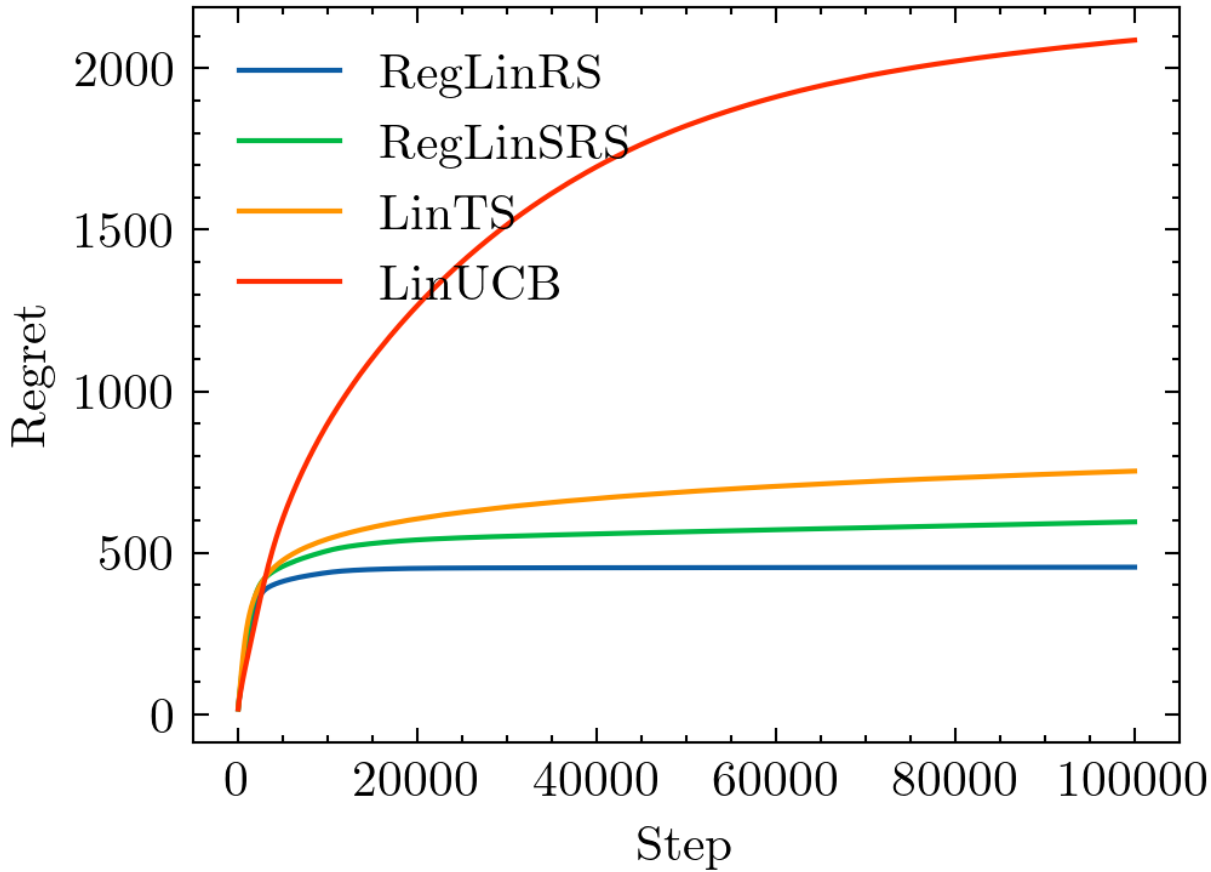
- 100,000ステップを1simとして1,000sim実行
  - 結果は平均値を算出
- 最初は全ての行動を10回ずつ選択する
  - パラメータ初期化のため
- batch sizeは全ての手法で20とする

変数名	設定値・初期値
ϵ	sys.float_info.epsilon in Python
episodic memory size	10,000
k of K-nearest neighbors	50
$\aleph$	0.6
α of LinUCB	0.1
λ of LinTS	0.25
α of LinTS	6
β of LinTS	6
$\mathbf{b}_i$	All 0

Result

Experiment 1

良



Result

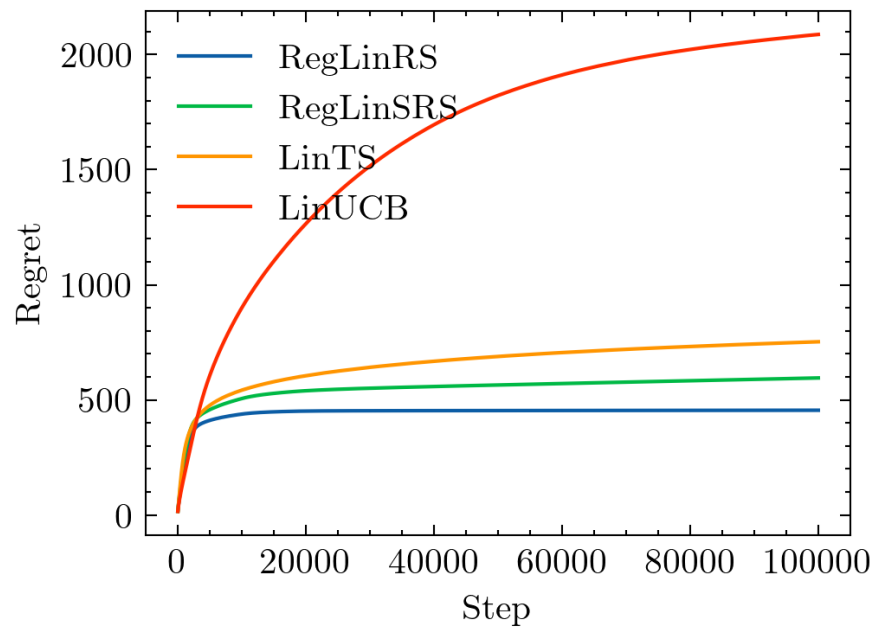
Experiment 1

- RegLinRS, RegLinSRS, LinTS, LinUCBの順で性能が良かった
- LinTS, LinUCBは対数的にregretが増加してしまっている
- RegLinRS, RegLinSRSはほぼregretが収束している

- LinTSは one of the state-of-the-art な手法である(Agrawal et al., 2019)
- RegLinRS, RegLinSRSに初期のregretの立ち上がりも劣っている

↓

RegLinRS,RegLinSRSはLinTSよりも早く学習を打ち切り、まだ正確な近似ができていない状況でも真に最適な行動を選択することができる？

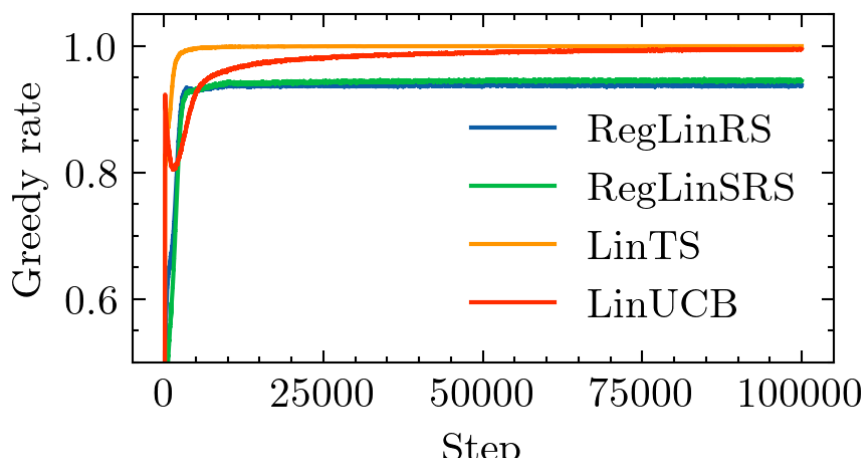


Discussions

Experiment 1

- LinTS, LinUCBは Greedy rate が1.0に達している
- RegLinRS, RegLinSRSは Greedy rate が0.9超で止まっている
  - 10回に1回程度は greedy な選択をしていない

→ RegLinRS, RegLinSRSは行動の過大評価が起こっていても真に最適な行動を選択することができている



Hypothesis

Experiment 1

【事実】RegLinRS,RegLinSRSは近似誤差の影響を部分的に軽減できている

- 近似誤差に過大評価が起こっていても真に最適な行動を選択することができている

【仮説】RegLinRSとRegLinSRSは信頼度によってこれを実現しているのはないか？

- 信頼度の反射効果からRSは必ずしもgreedyな行動選択を行うわけではない
- エージェントが確実に満足な行動を選択することができるという意味で合理的である

→ 報酬期待値に意図的にノイズを加えてこの性質についての検証を実施する

Experiment 2

実験設定

- Experiment 1 の実験を簡単にするために、行動数 $K = 2$ にする

推定報酬期待値へのノイズ

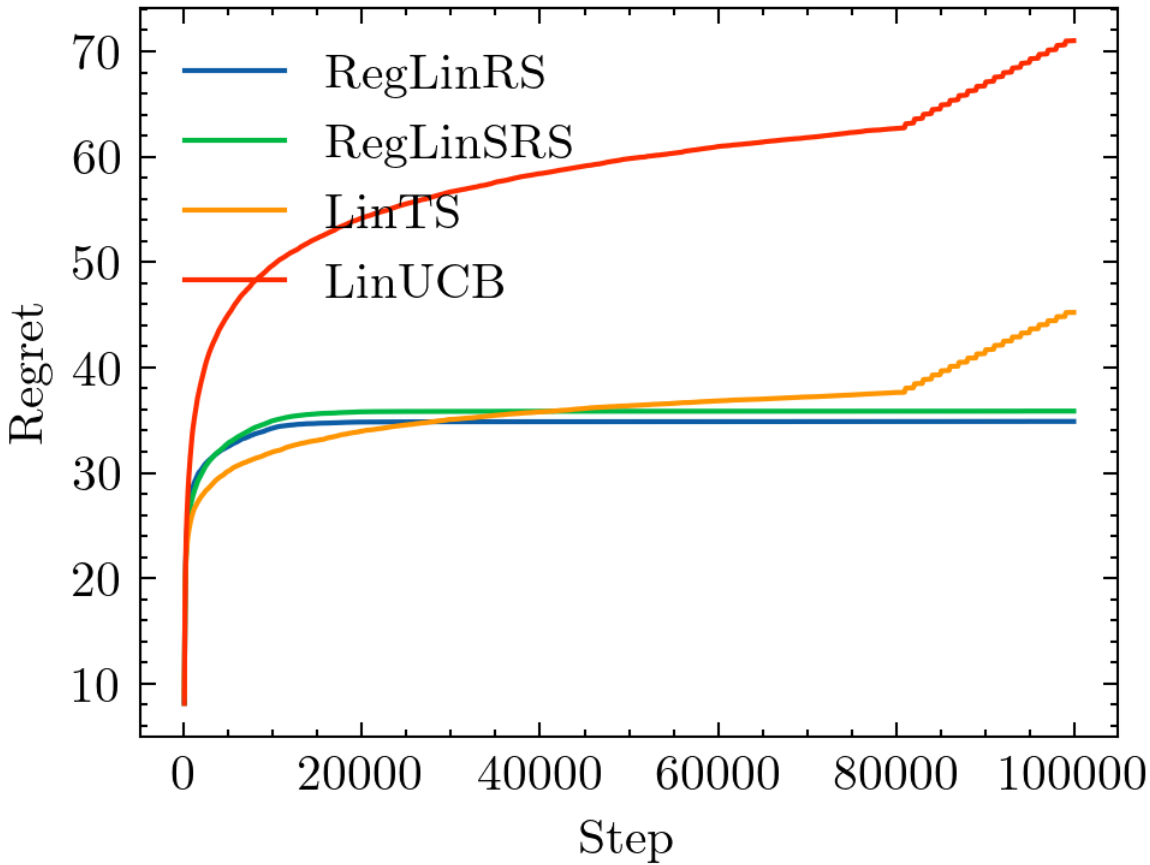
- 80,000 ステップ以降に等間隔で下記のノイズを推定された報酬期待値に加えた
- 大小関係が逆転するようなノイズとなっている
- γ は定数で 0.01 とした
- エージェントに報酬として返す報酬期待値にはノイズは加えない

$$p'_{\mathrm{min}} \leftarrow p_{\mathrm{min}} + \left(p_{\mathrm{max}} - p_{\mathrm{min}}\right) + \gamma$$

Result

Experiment 2

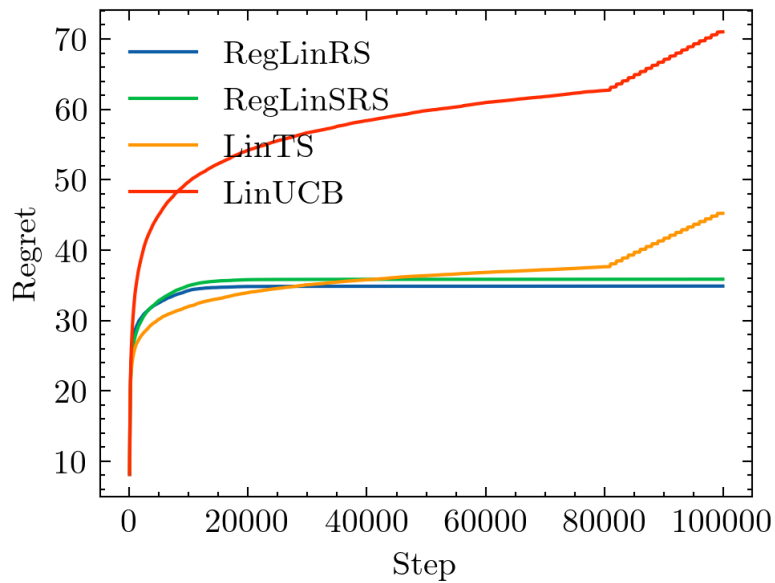
良



Discussions

Experiment 2

- ノイズが乗るステップ以降、LinTS, LinUCBはregretが急激に増加している
- RegLinRS, RegLinSRSはregretが増加していない
  - 信頼度の反射効果によって報酬期待値の誤推定の悪影響を部分的に軽減できている
  - 近似誤差に関して頑健性があることがわかった



Conclusion

- RegLinRSの確率論的な一般化を顕著な性能劣化なく実現できた（RegLinSRS）
- LinTS,LinUCBよりも性能がいいことを示すことができた

- RSの近似誤差への頑健性を示すことができた
  - 将来的に深層化するにあたってこの性質は有用である

→ RSの深層強化学習に適用する準備をすることができた