2023年5月26日　統計数理研究所　オープンハウス

報酬が過去の履歴に依存する場合のバンディットアルゴリズム

谷本　悠斗　　総合研究大学院大学統計科学専攻博士課程(5年一貫制)5年

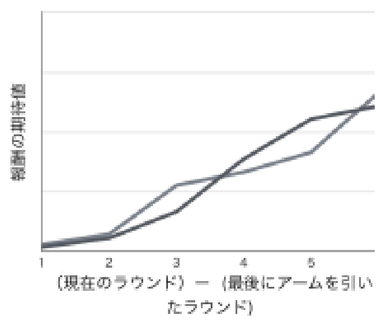
多腕バンディット問題

- 目標：リグレット最小化(<=>累積報酬最大化)
  - リグレット $= \underbrace{E[\sum_{t=1}^{T} r_t^*]}_{Oracle\ policy} - \underbrace{E[\sum_{t=1}^{T} r_t]}_{Learner}$
- 探索と活用のトレードオフ
  - 探索：各アームの報酬の情報収集
  - 活用：探索での情報を基に累積報酬を最大化
- アームが定常（報酬の期待値がroundに関わらず一定）

  ➡活用時に同じアームばかり引く
  - 満たさない例：推薦システム（同じ商品ばかり勧めてしまう）

問題設定

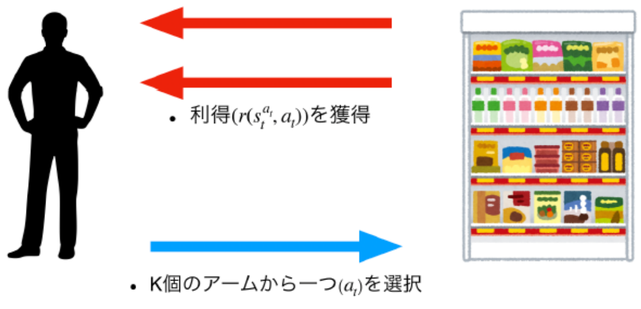
- （非定常な）報酬の構造
  - **アームiの報酬の期待値は最後にアームiを引いた時点から経過したラウンド数が長いほど大きい**
    - アームを引く ➡ そのアームの期待値は低下
    - アームを引かない ➡ そのアームの期待値は上昇



- 応用例：多様な商品の推薦
  - 商品を購入 → 続けて同じ商品を勧められても購入しない
  - 推薦せずに時間が経過 → 再び商品を購入する可能性が上昇

状態と利得の構造

- K次元の状態(state) $\mathbf{s_t} = (s_t^1, s_t^2, \cdots, s_t^K)$ (K：アーム数)
  - s_t^k: **最後にアームkを引いた時点からの経過ラウンド数** $(s_t^k = 1,2,\cdots, s_{max})(\forall k)$
  - **sの組み合わせ：** $|\mathbf{s}| = s_{max}^K$
- $r(s^k, a)$: **アームiを引いた時の報酬**
  - **報酬はアームkの状態のみに依存**
    - 報酬の単調性：$r(s^a, a) \leq r(s^a + 1, a)$
    - 経過ラウンド数が一定以上(s_{max})になるとそれ以上は利得は一定



各状態の次元ごとのQ関数の導入

- s_{max} **or** K**のいずれかが大きいとsの組み合わせ爆発によりQ学習ではうまくいかない**

  → Q関数の組み合わせを減らすアルゴリズムの必要性
- k**次元目(アームk)のみの状態のアーム**a**の関するQ関数**:

$$Q_\pi(s^k, a) := E_\pi[\sum_{t=0}^{T} \gamma^t r(s_t^{a_t}, a_t) \mid s_0^k = s_k, a_0 = a] = E_\pi[r(s^a, a) + \gamma V_\pi((s^k)') \mid s_0^k = s_k, a_0 = a]$$

  - π :policy, $\gamma \in (0,1]$:割引率, T: 期間, $V_\pi(s^k) = E_\pi[\sum_{t=0}^{T} \gamma^t r(s_t^{a_t}, a_t) \mid s_0^k = s_k]$ (k次元目の状態の価値関数)
  - **Q関数の組み合わせ：** $K^2 \times s_{max}$
  - 通常の(K次元の状態をもつ)Q関数$(Q_\pi(\mathbf{s}, a) := E_\pi[\sum_{t=0}^{T} \gamma^t r(s_t^{a_t}, a_t) \mid \mathbf{s_0} = \mathbf{s}, a_0 = a])$は$K$個の$Q_\pi(s^k, a)$ $(k = 1,2,\cdots, K)$の平均と等しい (i. e. $Q_\pi(\mathbf{s}, a) = \frac{1}{K}\sum_{k=1}^{K} E_\pi[Q_\pi(s_t^k, a_t) \mid \mathbf{s_0} = \mathbf{s}, a_0 = a]$)

アルゴリズム

Algorithm 1 State-separated Q-learning

Input T:Horizon, E:Number of Exploration, γ:Discount Rate
Initialize $Q_(s^k, a) \leftarrow 0\ \forall(s, k, a)$, $s^k \leftarrow s_{max}\ \forall k$
for $t = 1, 2, \cdots, T$ **do**
  choose $a \leftarrow \begin{cases} \text{randomly} & (t \leq E) \\ \text{argmax}_{a'} \frac{1}{K}\sum_{k=1}^{K} Q(s^k, a') & (t \geq E) \end{cases}$
  $r \leftarrow r(s^a, a)$
  $Q(s^k, a) \leftarrow Q(s^k, a) + \alpha(r(s^a, a) + \gamma \text{max}_a Q((s^k)', a) - Q(s^k, a))\ \forall k$
  $s^k \leftarrow (s^k)'\ \forall k$
end for

- アームの選択
  - 探索：指定された回数(E)ランダムにアームを選択
  - 活用：各アームのK個のQ関数の平均を比較して最も大きいアームを選択
- 選んだアームに関するK個のQ関数を更新

シミュレーション

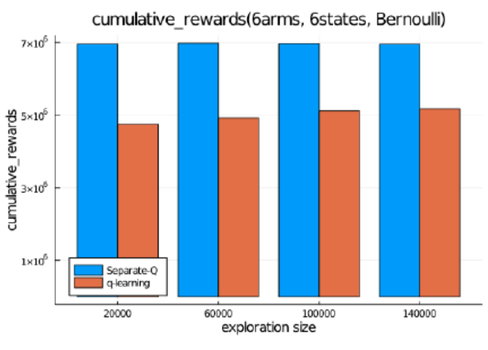
- 利得の分布：ベルヌーイ分布、正規分布（分散 = 1）
- ラウンド数：10^7、シミュレーション回数：100
- 2つの実験のセッティング
  - 状態が1つ上がるたびにランダムに利得の期待値が上昇
  - s_{max}でのアームの利得のみ以下のように期待値を固定

	(アーム数、状態数)	状態が一つ上がるたびに上がる利得の期待値	S_maxの利得の期待値
実験1	(6, 6)	0.07×Uni(0, 1)	全部のアームで0.7
実験2	(10, 5)	0.015×Uni(0, 1)	5つのアームのみ0.7

- 評価の指標（通常のQ学習と比較）
  - 累積利得の平均(cumulative rewards), 最適政策の割合(prob. of best policy)

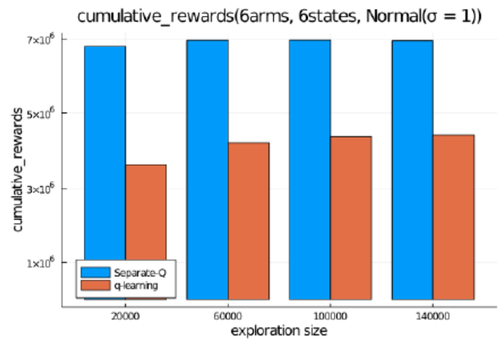
実験1（アーム数 = 6, 状態数 = 6）

- ベルヌーイ分布



prob. of best_policy	20000	60000	100000	140000
Separate-Q	0.98	1.00	1.00	1.00
Q-learning	0.02	0.03	0.06	0.08

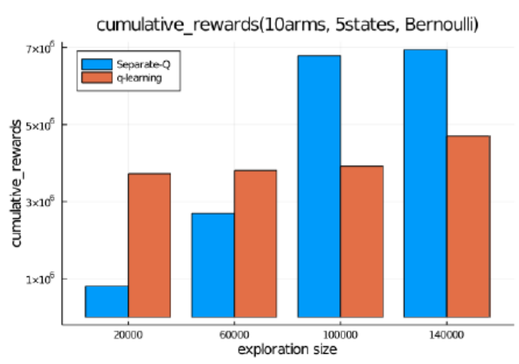
- 正規分布



prob. of best_policy	20000	60000	100000	140000
Separate-Q	0.90	0.99	1.00	1.00
Q-learning	0.01	0.01	0.01	0.03

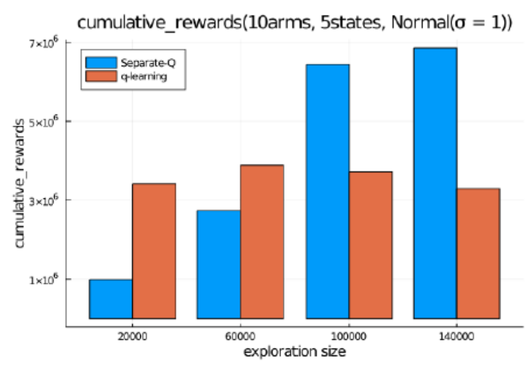
実験2（アーム数 = 10, 状態数 = 5）

- ベルヌーイ分布



prob. of best_policy	20000	60000	100000	140000
Separate-Q	0.00	0.00	0.89	0.99
Q-learning	0.00	0.00	0.00	0.01

- 正規分布



prob. of best_policy	20000	60000	100000	140000
Separate-Q	0.00	0.00	0.70	0.94
Q-learning	0.00	0.00	0.00	0.00