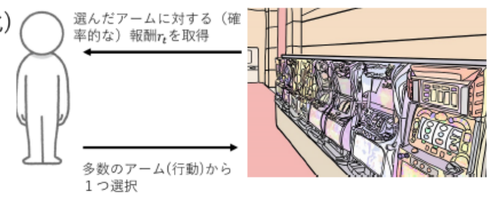
利得がアームを引く間隔に依存する場合のバンディットアルゴリズム

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)4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



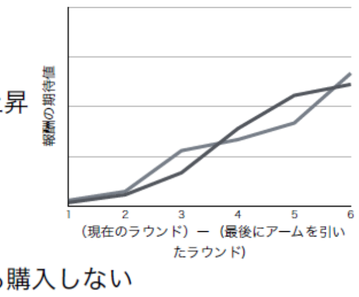
- アームが定常（報酬の期待値がroundに関わらず一定）

  ➡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

State の導入

- State = 最後にアームiを引いた時点からの経過ラウンド数
  - s_t^i: アームiのラウンドtにおけるState
    - **s_t^iが大きいほど（引いたときの）報酬の期待値が大きい**$(r_i(s_t^i) \le r_i(s_t^i+1))$
      - 経過ラウンド数が一定以上(s_{max})になるとそれ以上は定常
      - 初期値：全てのアームでs_{max}
    - $r_i(s_t^i)$: アームiを引いた時の報酬
      - **報酬はアームiのstateのみに依存（アームごとの独立性）**
- 通常のQ学習の問題点
  - Stateの組み合わせ：$|s_{max}|^K$　（K：アームの総数）
  - s_{max}とKが両方小さくないと推定するQ関数が多すぎる

アームごとに分離したQ-learning

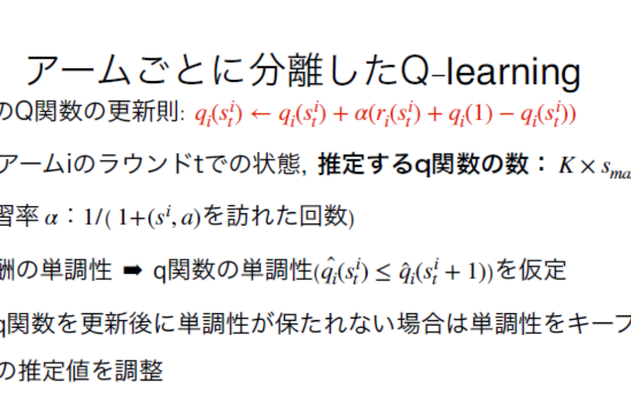
- アームiのQ関数の更新則: $q_i(s_t^i) \leftarrow q_i(s_t^i) + \alpha(r_i(s_t^i) + q_i(1) - q_i(s_t^i))$
  - s_t^i :アームiのラウンドtでの状態, **推定するq関数の数：** $K \times s_{max}$
  - 学習率 α：$1/(1+(s^i, a)$を訪れた回数$)$
  - 報酬の単調性 ➡ q関数の単調性$(\hat{q_i}(s_t^i) \le \hat{q_i}(s_t^i+1))$を仮定
    - q関数を更新後に単調性が保たれない場合は単調性をキープするようにq_iの推定値を調整
- Policy (焼きまなしϵ-greedy)
  - $\pi_t = \text{argmax}_{i \in [K]} q_i(s_t^i)$ w. p. $1-\epsilon$ random otherwise
  - ϵの初期値：1.0,　1ラウンドごとにϵを0.9999倍

シミュレーション

- シミュレーションのセッティング
  - 利得：ベルヌーイ分布（クリック率を想定）
  - ラウンド数：10^5、シミュレーション回数：100
- 比較するアルゴリズム
  - (状態の組み合わせを考慮しない)**Q-learning**
    - Stateの組み合わせ：$|S_{max}|^A$
  - **ϵ-greedy**(state を無視):
    - 確率ϵでアームをランダムに選択, $1-\epsilon$で最も利得の大きいアームを選択
  - **KS-armed bandit** : 各(s, a)のペアを独立なアームとみなす
  - **Two-ahead**: 2期間の利得の合計の最も大きいアームの組み合わせを選択

ランダムに期待値の伸びを変えて実験

Stateが一つ上昇するたびに$(0.05 \times Uni(0,1))$期待値が上昇



アームの期待値の推移（期待値がたまに下がる）

Stateが一つ上昇するたびに$0.05 \times Uni(0,1) - 0.025 \times Uni(0,1)$期待値が上昇