

2022年度未踏ターゲット事業

量子バンディットアルゴリズムの実験的検証とAPI開発 — 量子計算による「探索と活用のジレンマ」の挑戦 —

小野 宏史

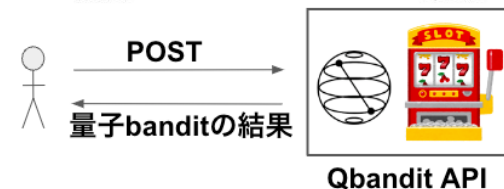
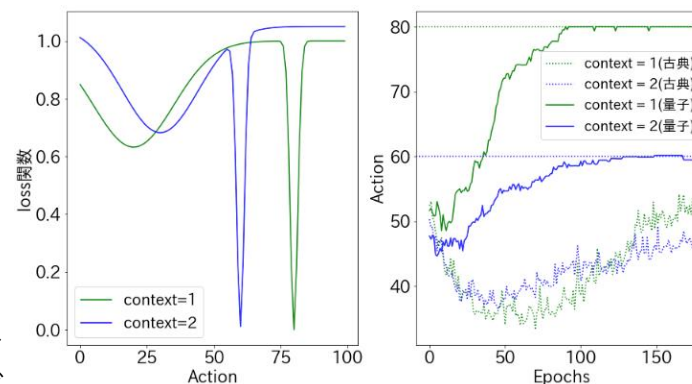
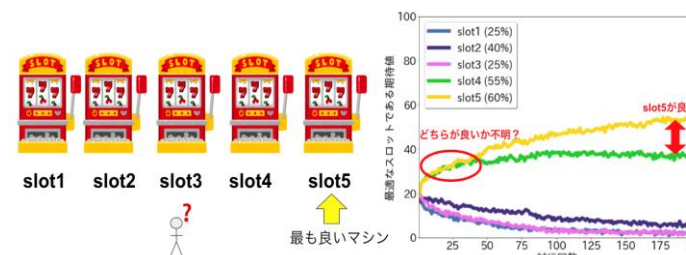
背景及び目的

バンディットアルゴリズムは、複数のスロットマシンの中から最も報酬の高い台を少ない試行回数で見つけるためのアルゴリズムである。本プロジェクトの目的は、バンディットアルゴリズムに量子探索アルゴリズムのGrover探索を適用した量子バンディットアルゴリズムを開発することである。量子回路上で実装を行い、古典アルゴリズムよりも早く最適解を求めるための検証を行う。

検証方法と結果

本プロジェクトで作成したモデルは、選択と活用のバランスをとりながら最適化を可能にする ϵ -greedyと、Grover探索を応用したDHアルゴリズムを組み合わせたものである。

問題設定として非常に狭い領域に最小値があるloss関数の最適化をバンディット問題として扱い検証を行なった。古典的な手法だと局所解に陥ってしまい正しい最適解へ到達することが困難であったが、本検証で実施したDHアルゴリズムを適用した場合、局所解に陥る現象を回避することが可能になり、効率的に最適解に向かっていく様子がわかる。



APIによるアプリケーション開発

本プロジェクトで作成した量子バンディットアルゴリズムのAPI化を行った。本APIはユーザーが量子プログラミングに関する知識がなくても使用できるように設計した。

FastAPI 0.10.0 GA3

```
default
GET /health ヘルスチェック
POST /quantum_bandit_demo Partial Dn
Parameters
No parameters
Request body required application/json
{
  "context": 1,
  "min_value": 0,
  "max_value": 100
}
Execute Clear
Responses
Curl
curl -X POST \
  https://mitou-quantum-bandit-gmswg5gc-uc-4.run.app/quantum_bandit_demo/ \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "context": 1,
    "min_value": 0,
    "max_value": 100
  }'
Request URL
https://mitou-quantum-bandit-gmswg5gc-uc-4.run.app/quantum_bandit_demo
Server response
Code Details
200 Response body
{
  "cost": 0.5232323232323232,
  "action": 1,
  "epoch": 1,
  "context": 1,
  "min_value": 0,
  "max_value": 100
}
```

API仕様書

```
Jupyter 量子バンディットデモ Last Checkpoint: 3分前 (autosaved)
Python 3 (ipykernel)
in [7]: def quantum_strategy(context, y=None):
        data = {
            "function": "cost_func",
            "context": context,
            "min_value": min_value,
            "max_value": max_value,
            "y": y
        }
        r = requests.post(endpoint_url, json=data)
        action, cost, final_count = r.json()
        return action, cost, final_count
in [8]: for epoch in tqdm(range(100)):
        data_dict = {"costs_1": [], "costs_2": [], "costs_1_quantum": [], "costs_2_quantum": [],
                    "actions_1": [], "actions_2": [], "actions_1_quantum": [], "actions_2_quantum": [],
                    "costs_1_dct": {}, "costs_2_dct": {}, "costs_1_dct_quantum": {}, "costs_2_dct_quantum": {}}
        action_quantum_dict = {"context_1": None, "context_2": None, }
        action = None
        for num, i in enumerate(range(400)):
            ...
100%|#####| 100/100 [47:35<00:00, 28.55s/it]
in [18]: ... ax = plt.subplots(ncols=2, figsize=(15, 5))
        ax[0].plot(costs_1_mean, "g", linestyle="dotted", label="context = 1(Classical)")
        ax[0].plot(costs_2_mean, "b", linestyle="dotted", label="context = 2(Classical)")
        ax[0].plot(costs_1_mean_quantum, "g", label="context = 1(Quantum)")
        ax[0].plot(costs_2_mean_quantum, "b", label="context = 2(Quantum)")
        ax[0].set_xlabel("Epochs")
        ax[0].set_ylabel("Cost")
        ax[0].legend()
        ax[1].plot(actions_1_mean, "g", linestyle="dotted", label="context = 1(Classical)")
        ax[1].plot(actions_2_mean, "b", linestyle="dotted", label="context = 2(Classical)")
        ax[1].plot(actions_1_mean_quantum, "g", label="context = 1(Quantum)")
        ax[1].plot(actions_2_mean_quantum, "b", label="context = 2(Quantum)")
        ax[1].set_xlabel("Epochs")
        ax[1].set_ylabel("Action")
        ax[1].axhline(y=80, color="g", linestyle="dotted")
        ax[1].axhline(y=60, color="b", linestyle="dotted")
        ax[1].legend(bbox_to_anchor=(1, 1))
        plt.show()
```

Pythonによる実行例

参考 : <https://github.com/onhrs/mitou-quantum-bandit>