実習２０２５

強化学習編

作成：原　武史（岐阜大学）＋ChatGPT5
作成日：2025/8/30

目　　次

この書類の目的

いわゆる数理最適化問題は，Operations Research分野で十分に議論されてきた.
ここでは，そのような内容はすっとばして，機械学習（特に強化学習：Reinforcement Learning, RL）を使って解法や順最適解を探索するためのヒントを記す.

配布内容

ReinforcementLearning.zip

にサンプルプログラムをまとめる.

動作環境の構築（別途資料あり）

１．Python環境の構築
　　venvを利用してPythonの仮想環境を構築.

２．仮想環境へのライブラリのインストール
　　以下のPythonライブラリが必要です.
　　仮想環境へpip/pip3でインストールしてください.
　　数値計算系：numpy, scipy,
　　グラフ・表示系：matplotlib, scikit-image, pillow
　　表の処理系：pandas
　　機械学習系：scikit-learn
　　内部処理系：joblib, tqdm
　　強化学習関連：gymnasium

３．同封のプログラムを実行するとバージョンが表示される.
　　以下は原が実行している環境である.

```
$ python3 check_versions.py
numpy       : 1.26.4
scipy       : 1.15.3
matplotlib  : 3.10.3
skimage     : 0.25.2
PIL         : 10.2.0
pandas      : 2.3.0
sklearn     : 1.6.1
joblib      : 1.5.1
tqdm        : 4.67.1
gymnasium   : 1.2.0
```

４．jupyter notebookの起動

（注意）
なし.

迷路の解法探索を強化学習で行う

ファイル名：　　maze.ipynb

目的：強化学習の概念を理解する

方法：迷路の解法探索を強化学習の問題に落とし込む

（概念）
強化学習の基本は MDP（Markov Decision Process，マルコフ決定過程） です.
迷路は自然にこの形に当てはまります：
・状態 (State)：今どのマスにいるか
・行動 (Action)：上下左右の移動
・遷移 (Transition)：行動をとると次のマスに移動（壁があれば移動できない）
・報酬 (Reward)：
　　ゴールに着いたら +1
　　それ以外は 0（または −0.01 で「早くたどり着け」を促す）

（動作原理）
最初はランダムに動くので「迷子」になります.
しかし，ゴールに到達したときだけ報酬を受け取る
すなわち，「この経路がよかった」と経験が残る.
Q学習は，「その状態でその行動を選んだときの期待報酬」を表す「Q値」を更新する.
だんだん「どのマスでどちらに動けば良いか」がわかる.
その結果，最短経路を選べるようになる.

（他の方法との違い）
代表的なダイクストラ法や幅優先探索とは違う.
迷路の最短経路は，グラフ探索アルゴリズム（Dijkstra, BFS） で簡単に解ける.

（なぜ強化学習を使うのか？）
環境のルールがわからなくても学習できる.
探索アルゴリズムは地図や遷移モデルが与えられる必要がある.
強化学習は「行動してみて，結果と報酬を観察するだけ」で方策を学ぶ.
だから「未知の環境」「確率的に動く環境」でも適応可能と考えられる.
つまり，ゴールまでの道を試行錯誤で学ぶ　行動を学習で行なっている.

在庫管理問題をDP／強化学習で行う

ファイル名：　DP.ipynb　線形計画法（Dynamic Programming）
　　RL.ipynb　強化学習

目的：最適化問題を２つの方法で取り組み比較する.

そもそも，在庫管理問題とは？
・在庫がないと売るものがない
・しかし，在庫が多いと管理費用がかかる
・いっぽう，在庫を発注すると費用がかかる
おおよそ
・在庫が少なければ「たくさん発注する」方が良い
・在庫が多ければ「発注しない」方が良い
の方策になるが，ではどのくらい発注したらよいか最適な値があるか？

ー＞簡単な問題は最適化問題で解かれている.
DPを解くことで「その場で一番良い発注数」＝ optimal order a* が決まる.

方法：
１．DPで解いてみる：DP.ipynb
２．強化学習で解いてみる：RL.ipynb
３．DPと強化学習を比較する：RL.ipynb（の最後のほう）
４．強化学習の改良を試みる

ねらい：
・古典的には DP（動的計画法）で解ける問題
・でも RL を使うことで「最適解が求めにくい問題」にも応用できる
・強化学習は最適解を知らなくても，経験から近い解を学べることを理解！

ナップサック問題を強化学習で解く

ファイル名：　　Knapsack.ipynb

目的：組み合わせ最適化ができる問題をあえて強化学習で解いてみる.

ナップサック問題とは？
容量制限のあるカバンに価値の高いアイテムを詰めて，合計価値を最大化する問題.

方法：ナップサック問題を強化学習で解く考え方

状態 (state)：
　どのアイテムまで意思決定したか（インデックス i）
　残り容量 cap
行動 (action)：
　そのアイテムを 入れる (1)
　入れない (0)
　（ただし重さが残容量を超えるときは「入れる」は禁止）
報酬 (reward)：
　中間報酬は 0
　エピソード終了時に「選んだアイテムの価値合計」をまとめて報酬にする
方策 (policy)：
　ニューラルネットが「状態→確率分布（入れる／入れない）」を出力
　許されない行動はマスクして選ばせない
学習 (learning)：
　REINFORCE（方策勾配）で「価値が大きかった行動系列」の確率を高める
　移動平均ベースラインを使い，報酬のブレを減らして安定化

ねらい（くりかえし）：
・古典的には DP（動的計画法）で解ける問題
・でも RL を使うことで「最適解が求めにくい大規模ナップサック」にも応用できる
・強化学習は最適解を知らなくても，経験から近い解を学べることを理解！

フローショップスケジューリング問題を強化学習で解く

目的：フローショップスケジューリングを簡単な例から複雑な例へ拡張.

内容：

１．機械２台でJohnson規則で解く
　　機械２台ならば厳密解が存在する．それを確認.
　　利用ファイル：FlowShop_Johnson.ipynb

２．機械２台で強化学習で解いて，Johnson則と比較
　　強化学習で得た機械２台の結果が理想になるか？
　　利用ファイル：FlowShop_RL_2M.ipynb

３．機械３台で強化学習で解く（１）.
　　機械３台になると途端に難しくなる傾向がある．それを解く.
　　利用ファイル：FlowShop_RL_3M.ipynb

４．機械３台で強化学習で解く（２）.
　　ジョブはセット（S），処理（Process: P），取り外し（Remove: R）がある.
　　それらの時間を与えて，強化学習で解く.
　　利用ファイル：FlowShop_RL_3M_SPR.ipynb

５．機械n台で強化学習で解く（３）.
　　機械の台数を増やして解く.
　　利用ファイル：FlowShop_RL_nM_SPR.ipynb

ジョブショップスケジューリング問題を強化学習で解く

目的：ジョブショップスケジューリングを簡単な例から複雑な例へ拡張.

理解：ジョブとフローの違いの理解が必要です.

内容：

１．機械２台で強化学習で解いて，自分の結果と比較.
　　強化学習で得た機械２台の結果に勝てるか？
　　利用ファイル：JobShop_2M.ipynb

２．機械３台で強化学習で解く（１）.
　　ジョブはセット（S），処理（Process: P），取り外し（Remove: R）がある.
　　それらの時間を与えて，強化学習で解く.
　　利用ファイル：JobShop_3M_SPR.ipynb

３．機械３台で強化学習で解く（２）.
　　ガントチャートを見ると同時スタートあり．これは無理.
　　まずはセットにだけ人の制約を加えて解く.
　　利用ファイル：JobShop_3M_SPR2.ipynb

４．機械３台で強化学習で解く（３）.
　　機械によっては複数同時にセットできる場合がある．それも加味する.
　　ただしまずは人の制約は外す.
　　利用ファイル：JobShop_3M_SPR3_multi.ipynb

５．機械３台で強化学習で解く（４）.
　　４に人の条件をつける．ただしセットのときだけ.
　　利用ファイル：JobShop_3M_SPR3_multi2.ipynb

６．機械３台で強化学習で解く（４）.
　　５に人の条件をさらにつける．つまり取り外しも重ならないように制約.
　　利用ファイル：JobShop_3M_SPR3_multi3.ipynb