2020･09･03　ゼミ資料

Pythonで学ぶ強化学習

1章 強化学習の位置づけを知る

1116 17 9036
山口真哉

講談社『Pythonで学ぶ強化学習』

例となるコードがかなり豊富なのが特徴で,
コードを動かしながら読んでいくとさらに理解が深まると思うので,
実際に各自コードを実行して欲しい.

発表時の注意点:

わからないこと, 聞きたいことがあれば
発表中に質問してください.
質問してくれた方が発表しやすいです.

機械学習

- 機械(model)を与えられたデータに合うように学習させる手法.
  - 学習手法に教師あり学習, 教師なし学習, 強化学習がある.

- 教師あり学習

  データとラベルを与えて, データが与えられたら正解が出力されるように学習させる.
  - 画像分類など

- 教師なし学習

  データのみを与えて，データの特徴を抽出できるように学習させる.
  - オートエンコーダなど

- 強化学習

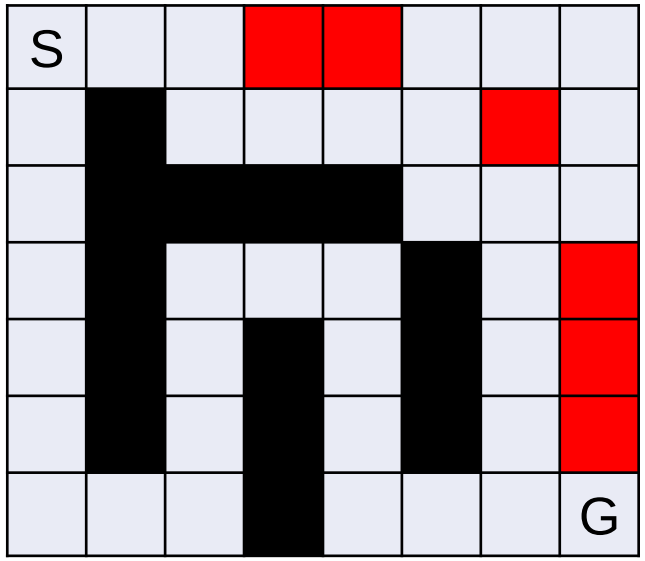
  行動より報酬が与えられる環境を与えて,
  各状態で報酬につながる行動が出力されるように学習させる.
  - 車の自動運転, 将棋AIなど

強化学習

- 強化学習の他2つと大きく異なる点は「データ」を与えるのではなく「環境」を与える点.
- 環境とは 行動 と行動に応じた 状態 の変化が定義されており,
  ある状態へ到達すると 報酬 が与えられる空間のこと.
- 強化学習では環境が与えられて報酬を最大化するようにパラメータを調整する.
- モデルは状態を受け取り行動を出力する関数である.

例 迷路

- 状態：S:スタート, G:ゴール, 赤マス:通ると死ぬマス, 黒:通れないマス
- 行動：上 下 左 右 (ただし黒マスと壁は除く, 行動しないという選択肢はない)
- 報酬：1歩進むごとに-4, 赤マスに到達すると-100, Gに到達すると100得られる
- 注意：マップ全体で風が吹いていて20%の確率で操作がうまくいかない.
  中間地点に正の報酬を置いたりする.,



強化学習のメリット・デメリット

- 強化学習は報酬(≒正解)があるという点で教師あり学習と似ている.
- 教師あり学習は単体で最適化を行うのに対し,強化学習は全体で行う.
- 下の例だと教師ありでは単体の行動評価をするため下の選択肢を選ばない.
  対して, 強化学習だと全体で評価するため, いずれ下を選ぶようになる.

～ルール～

- 毎日1000円あげるよ.
- 3日我慢すれば10000円あげるよ.

強化学習のデメリット

- 人が正解を「ラベル」で与えないため，行動の評価がモデル任せになる．
  - 人の感覚とは違う評価が獲得され意図しない行動をする．

- 教師なしとのデメリットと似ている．

- 教師あり学習はデータが増えれば増えるほど精度が上がるため，
  教師あり学習が可能な場合は先に教師あり学習を試す方が好ましい．

強化学習における問題設定

- 与えられた環境は「遷移先の状態は直前の状態とその行動にのみ依存する」ことを仮定する.
- この性質を マルコフ性 という. (講義でもやったね)
- マルコフ性を持つ環境を マルコフ(決定)過程 (MDP, Markov Decision Process)という.

MDPの構成要素

- s：状態 (State).
- a：行動 (Action)
- $T(s,\ a)$：状態遷移確率 (遷移関数 / Transition function)
  状態と行動を引数に遷移先(次の状態)と遷移確率を出力する関数
- $R(s_t,\ s_{t+1},\ a = None)$：即時報酬 (報酬関数 / Reward function)

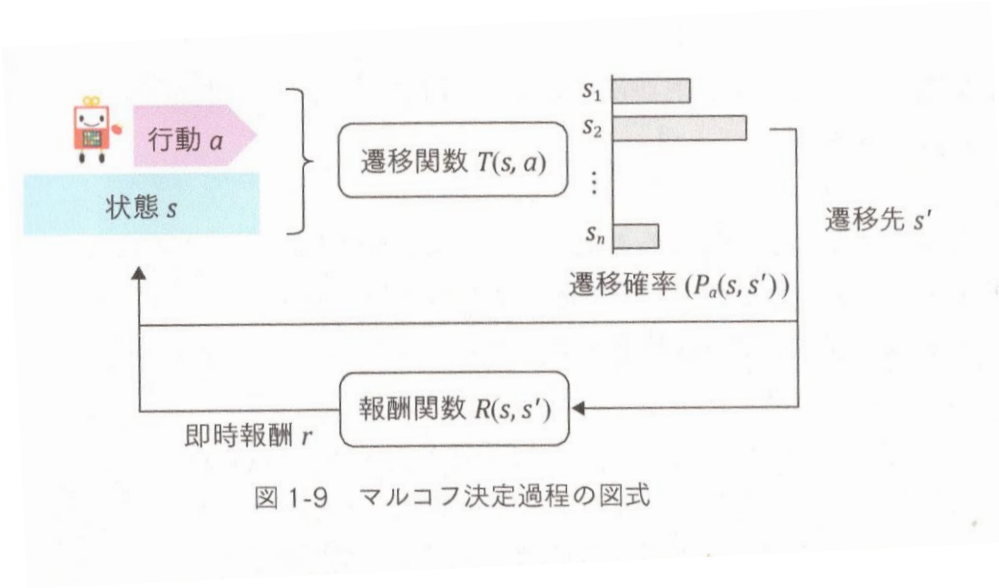


図 1-9　マルコフ決定過程の図式

言葉の説明

- ロボットは状態を受け取り行動を出力する関数とみなせる.
- この関数を 戦略 (Policy) π と呼ぶ.　戦略が強化学習における「モデル」となる.
- 戦略のパラメータを調整し, 状態に応じて適切な行動を出力できるようにすることが強化学習でいう「学習」となる.
- 戦略に従って動く主体 (今回だとロボット) をエージェント (Agent) と呼ぶ.

報酬

- MDPにおける報酬 (r)は直前の状態と遷移先に依存する.
- この報酬を即時報酬 (Immediate reward)と呼ぶ.
- 強化学習の目的は即時報酬を最大化することではなく
  即時報酬の総和 G_t を最大化することであった.

$$G_t := \sum_{i=t+1}^{T} r_i$$

報酬

- 将来の即時報酬はわからないため, G_t はエピソード終了まで計算できない.
- 報酬の総和を最大化することが目的なので, 行動する前に報酬の総和を予測することを考える.
- 見積もりは不確かな値であるため割引率 γ $(0 \leq \gamma < 1)$ を使って 割引現在価値 G_t を定義する.

$$G_t := \sum_{i=t+1}^{T} \gamma^{t+1-i} r_i = \sum_{i=0}^{T-t-1} \gamma^i r_{t+i+1}$$

- G_tは再帰的な構造を持っている.

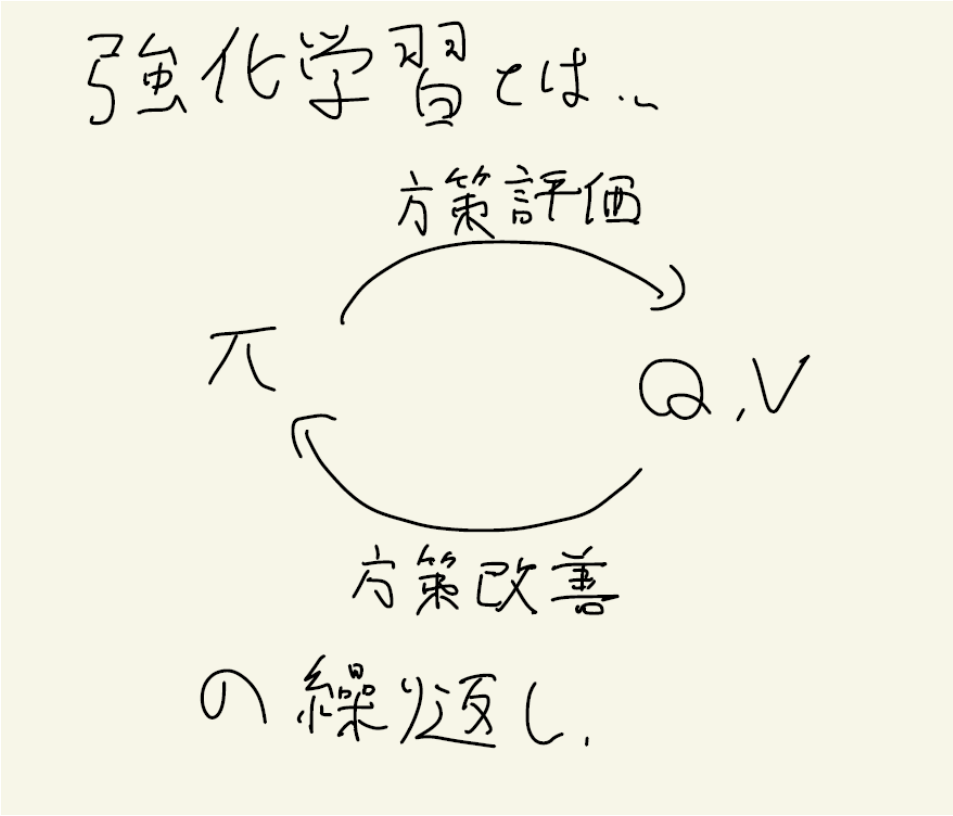
$$G_t = \sum_{i=t+1}^{T} \gamma^{t+1-i} r_i = r_{t+1} + \gamma \sum_{i=t+2}^{T} \gamma^{t+1-i}$$

$$= r_{t+1} + \gamma G_{t+1}$$

報酬

- 割引報酬和で書かれた G_t を 期待報酬 または 価値 (Value) という.
- 価値を算出することを 価値評価 と呼ぶ.
- この価値評価が強化学習が学習する2つのことの1点目である「行動の評価方法」である.
  - 価値評価や戦略を学習する具体的な方法はchapter 2 で見る.

まとめ



- 教師あり学習, 教師なし学習, 強化学習の違いを見てきた.
- 強化学習の概要やメリット, デメリットをみてきた.
- 強化学習ではMDPを仮定して, 最大化の対象は価値 G_t であることがわかった.
- ここからPythonでMDPに従う環境(迷路)の実装をするが長いので各自見てもらいたい.
  https://github.com/masa-aa/Reinforcement-Learning-with-Python/tree/master/chap1

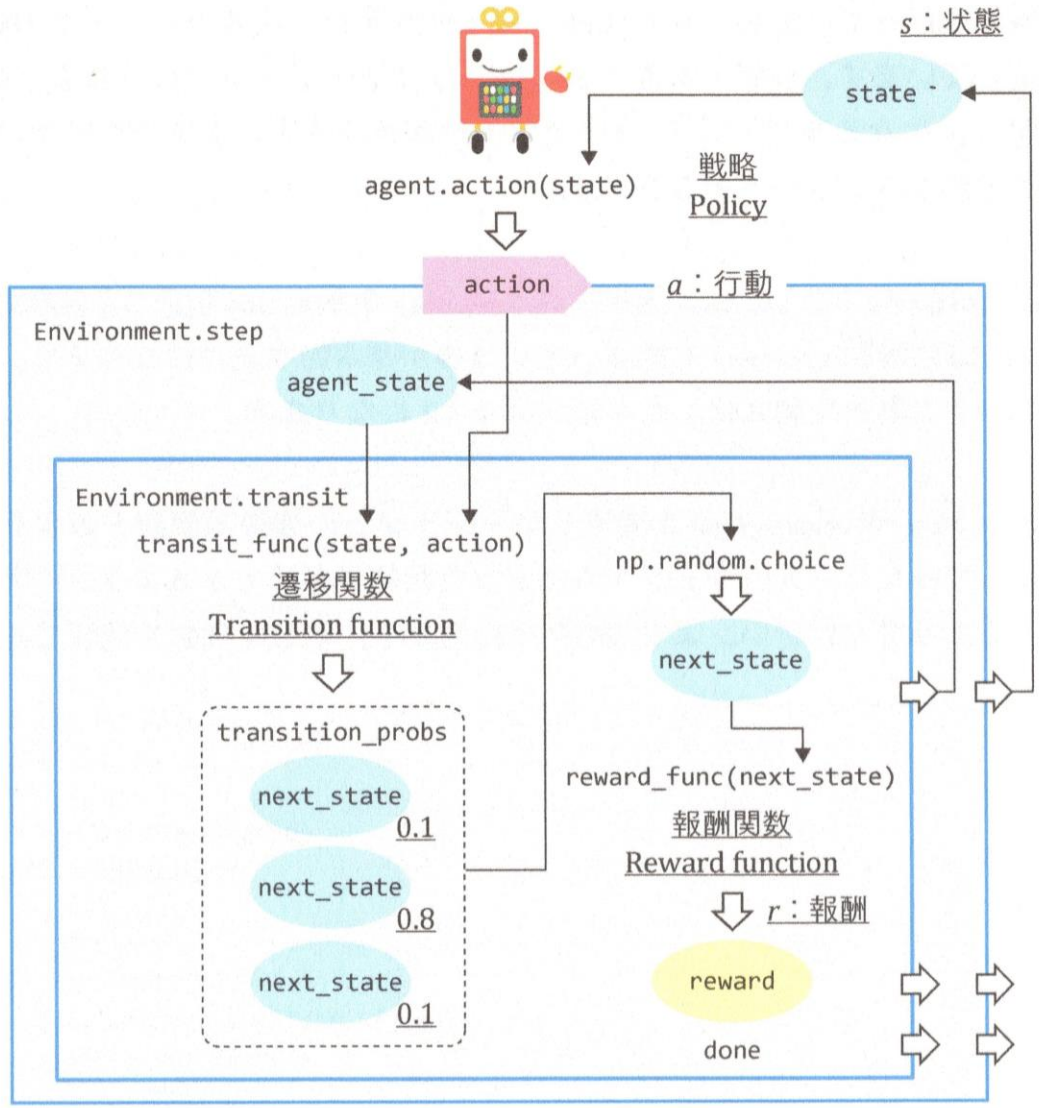


図 1-12　実装コードと対応づけた、マルコフ決定過程の図

何か質問はありますか？