宇宙物理・数理科学研究室　Astrophysics & Mathematical Sciences

Q-learningを使った迷路の最短距離学習

卒業研究中間報告　B17-028　三田大晃

Q-learningを使った迷路の最短経路学習

目的　　強化学習の一種であるQ-learningを用いて迷路の最短経路を見つける

問題設定

右図のような10×10の迷路を用いて，s(start)とg(goal)を設定する．

ルール
1. 1度通った場所は通れない．
2. 上下左右のいずれかに必ず進む．
3. 上下左右のいずれにも進むことが不可能な場合，またg(goal)にたどり着いた場合，1試行を終了する．

```
# # # # # s # # # #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# # g # # # # # # #
```

図1 迷路図

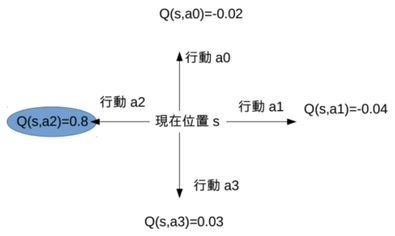
Q-learningとは

Q-learningは強化学習の一種で，右図のようなQtableを用いる．Qtableは現在位置sから行動aをしたときの行動価値Q(s,a)を示す表である．

	s0	s1	s2	s3	.	.
a0	Q00	Q10	Q20	Q30	.	.
a1	Q01	Q11	.	.	.	.
a2	Q02					
a3	.					
.	.					
.	.					

表1 Qtable

例として右図の場合，Q(s,a2)の値が最も大きいので行動a2が最も行動価値の高い行動と言える．



Q(s,a)とは？
1. Q(s,a)とはどちらに進むかの判断に使う値である（行動価値）．
2. 現在位置sでどちらに行動aするべきかの表をQtable(表1 参照)という．
3. はじめはすべてのQ(s,a)=0である．
4. ゴールまでの最短距離が定まればQ(s,a)の各要素の合計は収束する．

エージェントの設定

エージェントとは，どちらに進むかの判断の基準であり，現在位置sを受け取って行動aを出力する．

エージェントの行動政策は以下のとおりである．

・70%の確率で最もQ(s,a)の高い行動aを選択する．
・残りの30%の確率でランダムに行動する．

エージェントは，学習しながらQtableを更新する．

Q-learningでよりよい学習を行うため，以下の検証を行った．

1. 何回学習すればよいか．
2. 迷路のサイズを変えてみる．
3. Qtableに従うときと，無視するときの割合を変えてみる．
4. 報酬の値を変えてみる．
5. 学習率(1歩進んだときのQ値の優先度)を変えてみる．

Q値更新式

Q(s,a)の値は次式を使い，エージェントが行動aを選択するたびに更新していく．

$$Q(s_t,a_t)=Q(s_t,a_t)+\alpha(r+\gamma \max Q(s_{t+1},a_{t+1})-Q(s_t,a_t))$$

状態
行動
学習率$(0<\alpha<1)$
割引率$(0<\gamma<1)$

t:時間(行動するたびに+1する)

maxQ(s,a):状態sにおいてQが最大となる行動aにおいてのQ(s,a).
aはQ(s,a)のとsにより決まる．

何回学習すればよいのか?

迷路の最短距離を見つけるのに必要な学習回数を調べるため次式(Q値の総量)を用いた．

$$\text{Q値の総量}=\sum_{s,a} Q(s,a)$$

結論

右図が，10万回と2万回の学習によるQ値の総量の推移である．
以上の結果から約1万回ほどで学習が収束しているため，約1万回学習すれば十分なことがわかった．

```
# # # # # s # # # #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# 0 0 0 0 0 0 0 0 #
# # g # # # # # # #
```

図4 迷路学習の結果

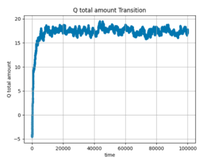


図2 10万回の学習

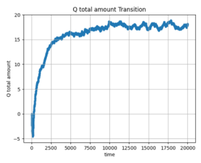


図3 2万回の学習

迷路のサイズを変えてみる

5×5の迷路で学習した結果，約5000回ほどで学習が収束していることがわかった．また，12×12は1万回ほどで収束したがstart近くで無駄な行動が見られた．

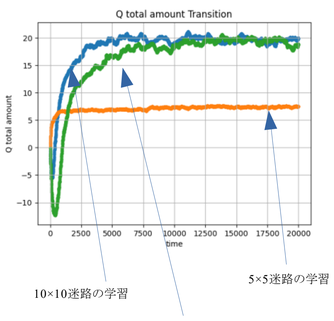


図4 迷路サイズを変えたときのQ値総量の推移

結論

1. 迷路のますの数を減らすと，必要な学習回数は減る．
2. 迷路のますの数を増やしすぎると，start付近で無駄な行動をする．

```
# # # s #
# 0 0 0 #
# 0 0 0 #
# 0 0 0 #
# # g # #
```

図5 迷路の学習結果

Qtableに従うときと，無視するときの割合を変えてみる

100%でQ値に従うとき，0%でQ値に従うとき，50%でQ値に従うときの3つを試した．

結論

1. 100%Q値に従うとき，一度ゴールしてしまうと，Q値が固まり変化しなくなる．
2. 0%Q値に従うとき，Q値の総量が大きくなっている．これは無駄な行動を多数行い学習したためだと思われる．
3. 50%Q値に従うとき，70%のときより無駄な行動が増えたが収束が早くなった．

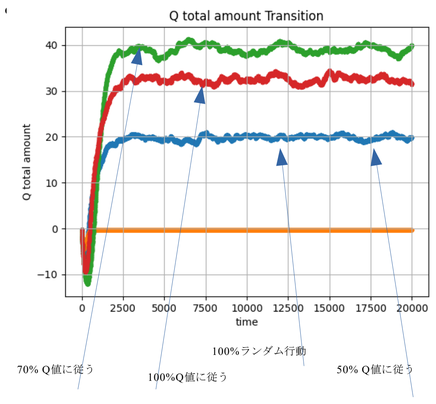


図6 Qtableに従う確率を変えたときのQ値総量の推移

報酬の値を変えてみる．

報酬rはゴールするとき1,それ以外で-0.04と設定していたが，それ以外を-0.04から0に変更した．

結論

1. 報酬が0以上だけになったためQ値が全体的に大きくなると考えられる．
2. 報酬-0.04のときよりQ値の収束は早くなった．

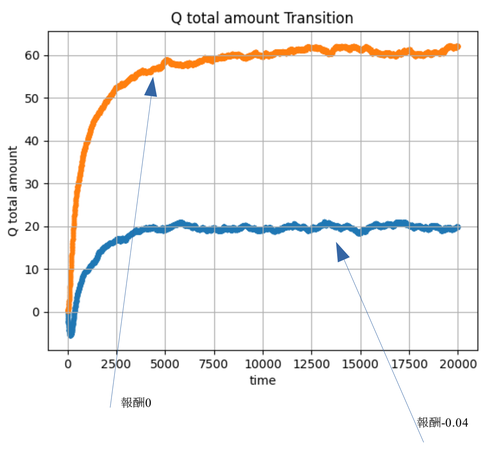


図7 報酬を変えたときのQ値総量の推移

学習率を変えてみる．

学習率の値を0.1から0.9に変えてみた．

結論

学習スピードは0.1のときよりとても早くなったが，Q値が収束してからも振れ幅が激しく不安定である．

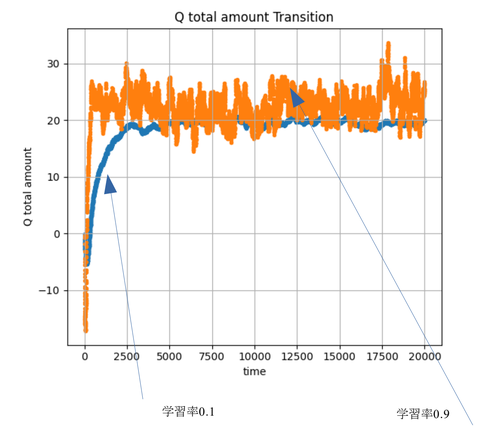


図8 学習率を変えたときのQ値総量の推移