TD学習

・おさらい

- 一般化方策反復

TD、MC、DPに共通する考え方

$V^{\pi}(s)$を求める ⇆ πを更新する

・どうやって状態価値$V^{\pi}(s)$を求めるかが3つの違い

(πの更新の方は、方策オン型オフ型など、TDやMCの中でさらに複数の手法がある)

･DP(動的計画法)

要請

- 環境が行動によってどう変化するかのモデルがすべて判明していること

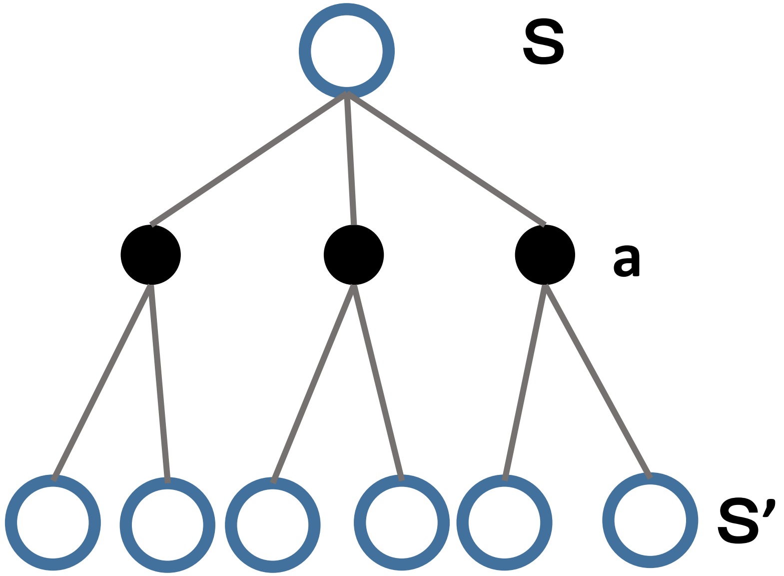
Vの計算方法(更新式)

- $V_{k+1}(s) = \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{s\,s'}[\mathcal{R}^a_{s\,s'} + \gamma V_k(s')]$

欠点

･Vを求める計算量がクソ

バックアップ線図



・MC(モンテカルロ法)

要請

- タスクが有限回で終了すること

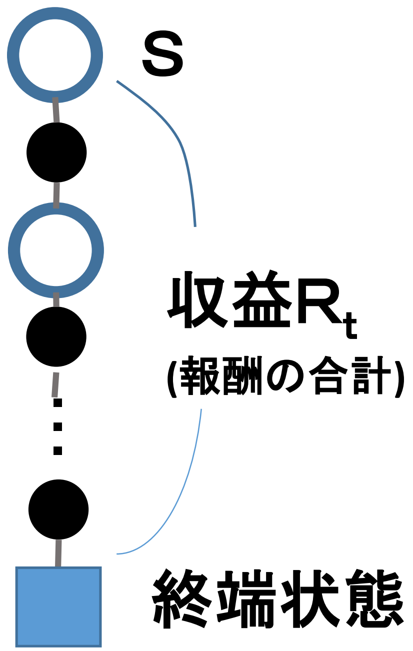
Vの計算方法(更新式)

- $V_{k+1}(s) = V_k(s) + \alpha[R_t - V_k(s)]$

欠点

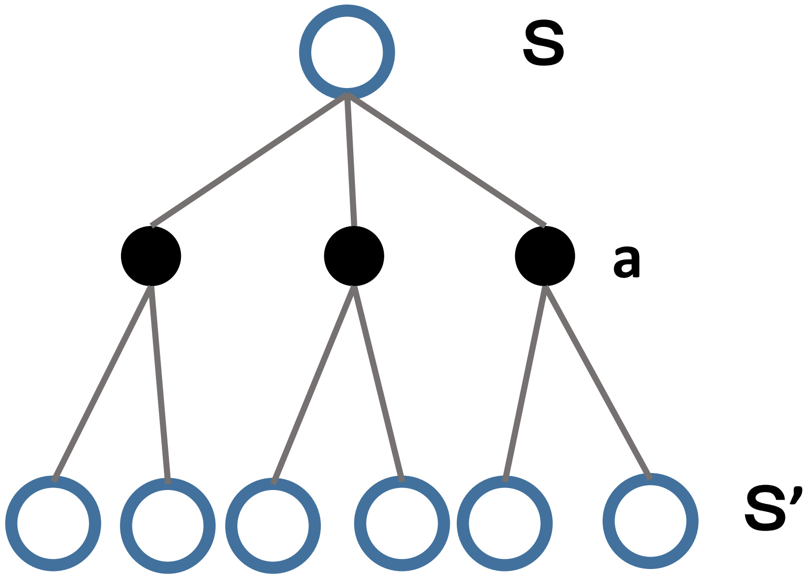
・タスクが終わらない限り更新されない

バックアップ線図



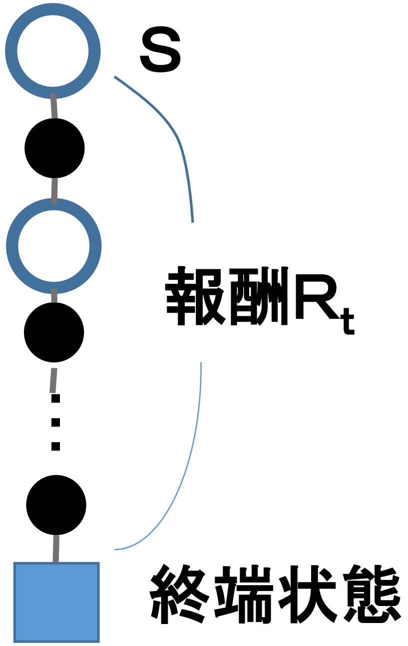
TD学習

動的計画法



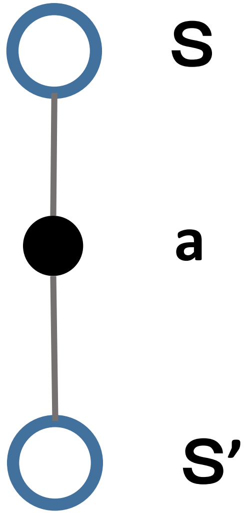
枝分かれを全チェックするには
環境がどう変化するか知らないと無理

モンテカルロ法



報酬Rで更新するなら
タスクが終わらないと無理

TD学習



・TD学習

要請

- ほぼなし！

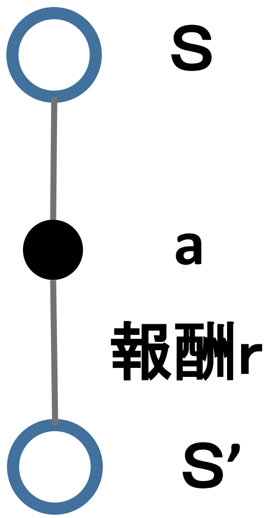
Vの計算方法(更新式)

- $V_{k+1}(s)$
  $=V_k(s)+\alpha[r+\gamma V_k(s') - V_k(s)]$

$$= (1-\alpha)V_k(s)+\alpha[r+\gamma V_k(s')]$$

αが小さければVπに収束することは保証されている

バックアップ線図



TD(0)のアルゴリズム

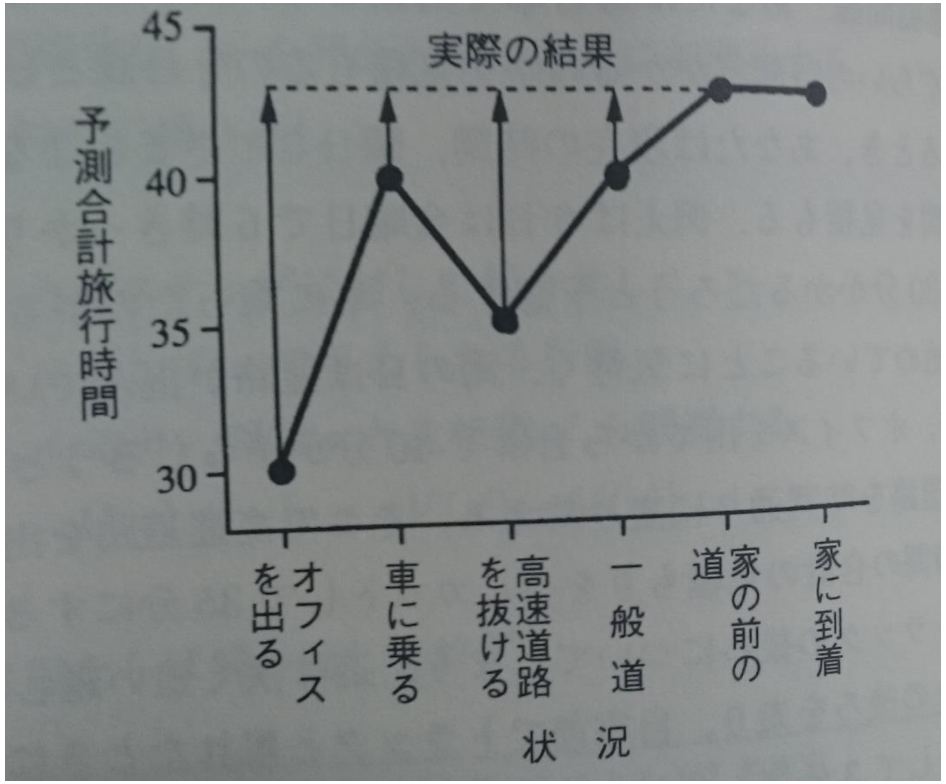
V(s)を適当に初期化　πを評価対象の方策で初期化
各エピソードに対して繰り返し:
　　s初期化
　　エピソードの各ステップに対して繰り返し:
　　　　a←sに対してπで得られる行動
　　　　行動aをとり、報酬rと状態s’を得る
　　　　$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
　　　　$s \leftarrow s'$
　　sが終端状態なら繰り返しを終了

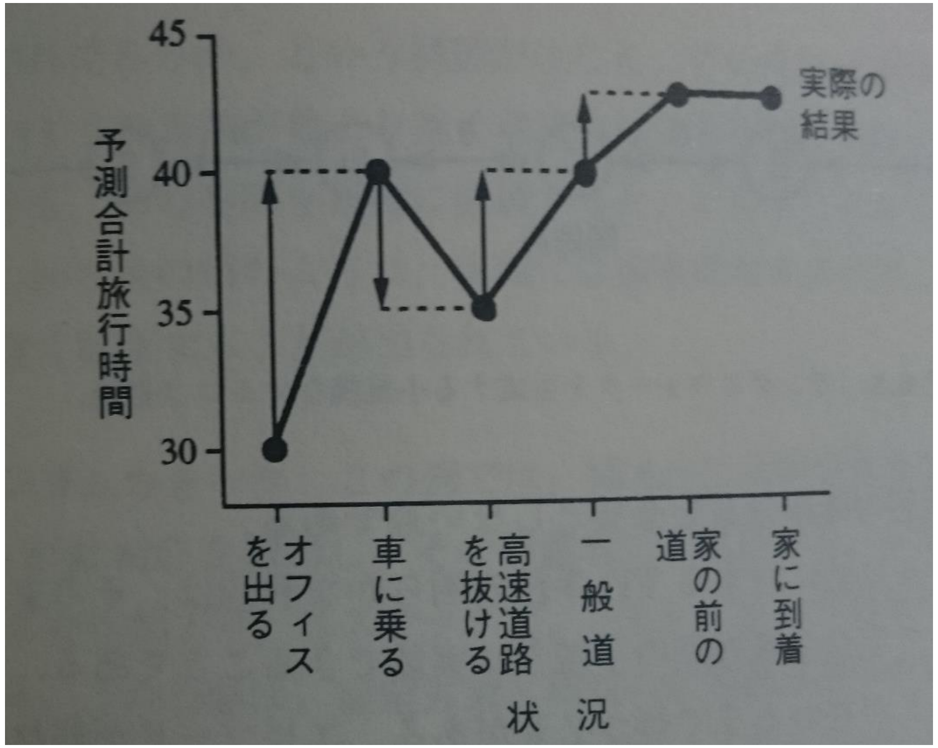
MCとTDの比較をしていこー

- 報酬は各工程の経過時間とする

状態	所要時間 (分)	予想残り時間	予想合計時間
オフィスを出る．金曜日午後 6 時	0	30	30
車に乗る．雨	5	35	40
高速道路を抜ける	20	15	35
一般道．トラックの後ろ	30	10	40
家の前の道に入る	40	3	43
家に到着	43	0	43

MCとTDの比較をしていこー



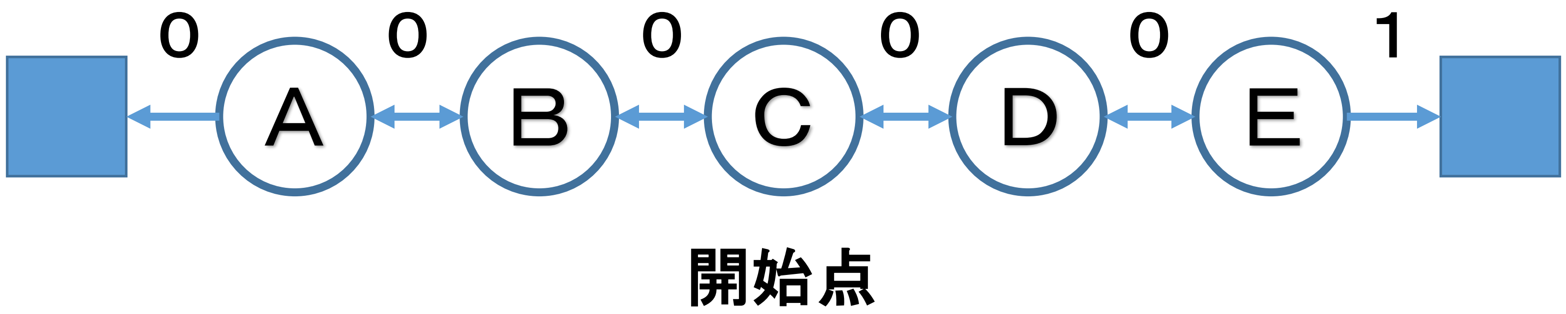


モンテカルロ法
実際にかかった時間を目標に学習

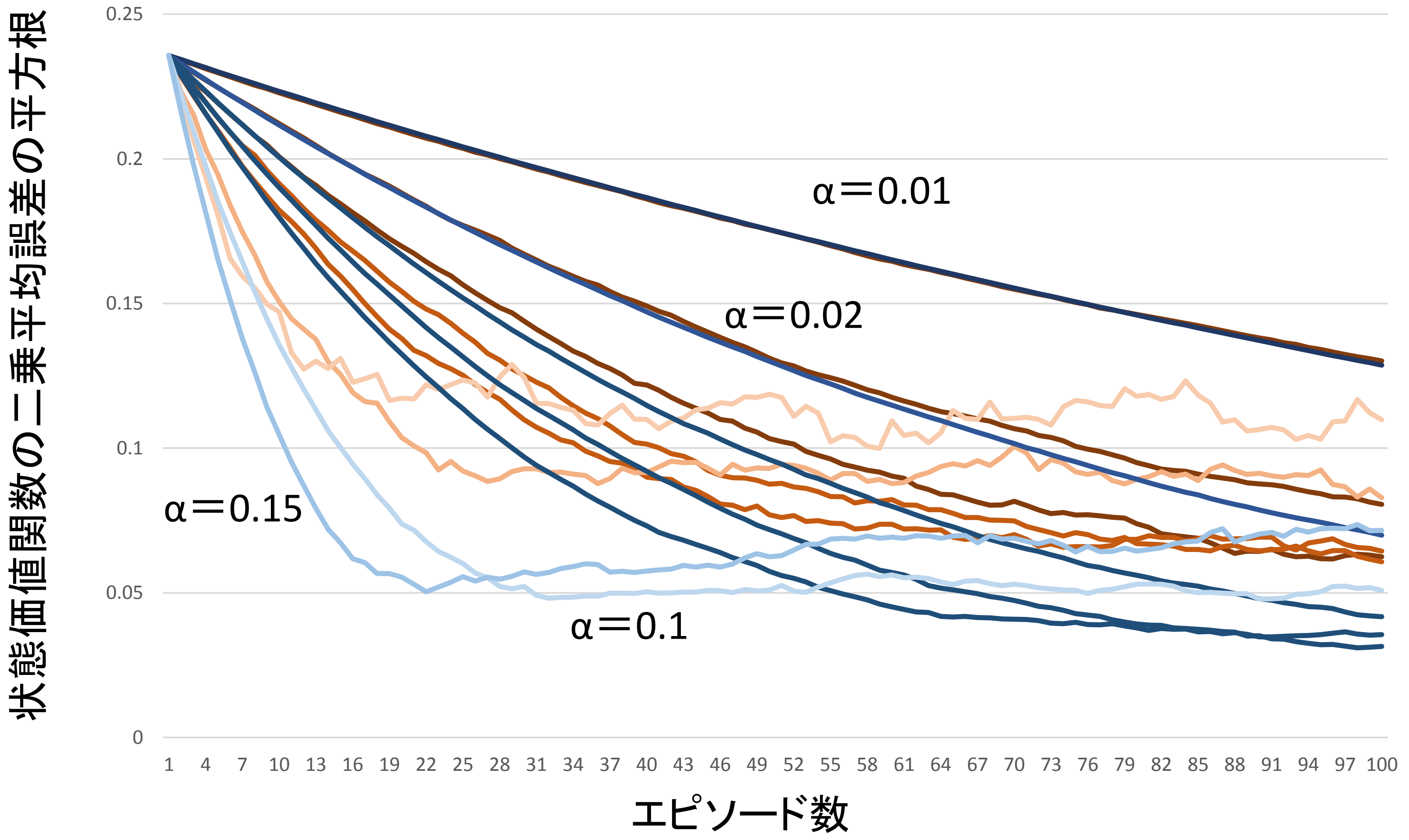
TD学習
直後の状況においての合計旅行時間の推定値を目標に学習

TDとMCのどっちが収束がはやいの？

- ランダムウォークをするマルコフ決定過程で状態価値関数Vをそれぞれの手法で求めさせる。



シミュレーション結果



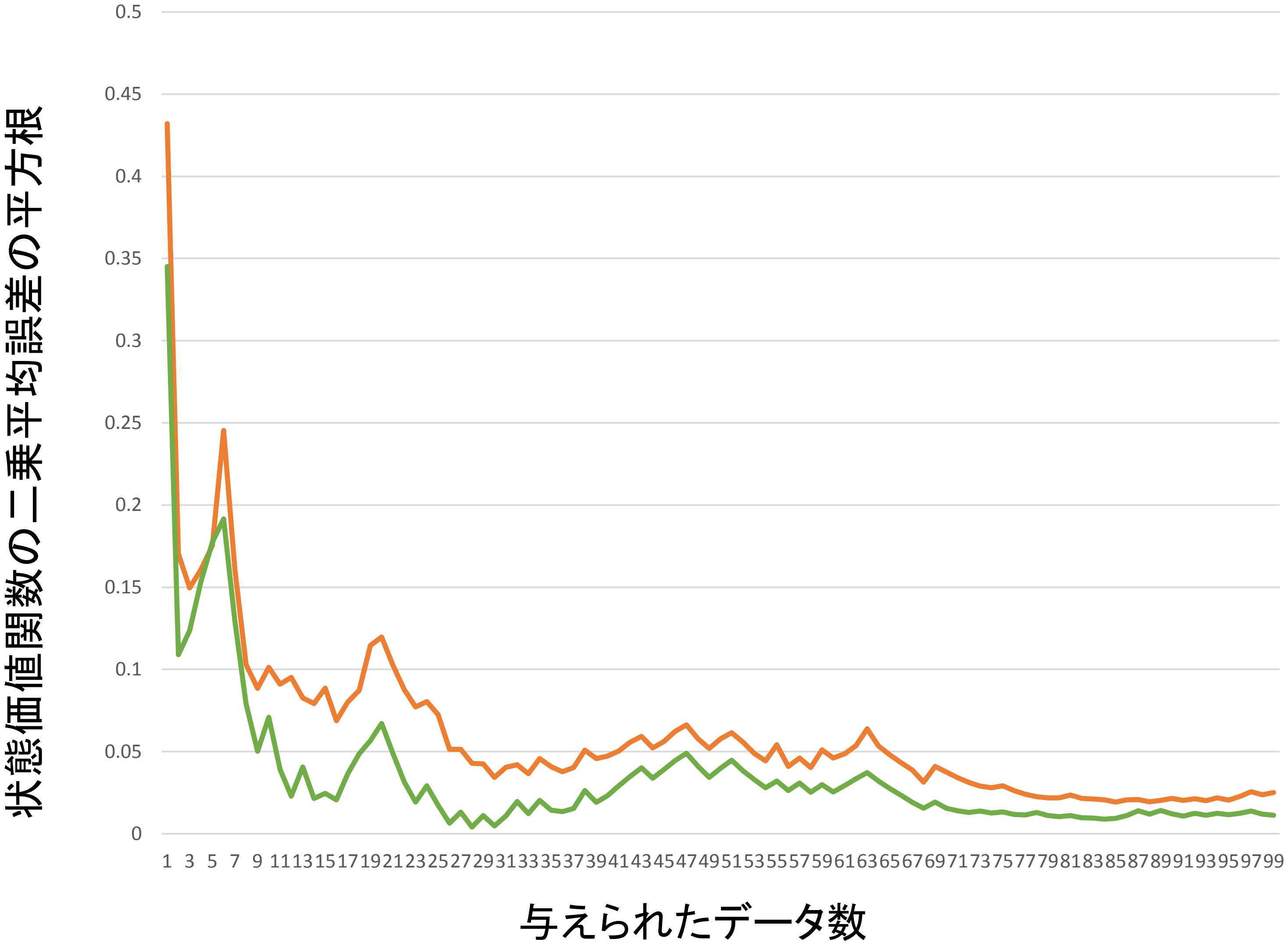
青がTD
赤がMC(初回訪問MC)
αはそれぞれ
0.01　0.02　0.03
0.04　0.05　0.1　0.15
の7通り

αは最初は大きく、
学習が進むにつれて
小さくするのが望ましい

バッチ更新

- 例えば10エピソードや100ステップなど、限られた経験しか与えられておらず、さらに経験を増やすことはできないものとする。
- この限られた経験のみを元に学習をすることをバッチ更新という。
- αを十分小さくすればTDもMCも収束することは知られているが、
  その収束値が異なる
- その理由を見ていく

シミュレーション結果(バッチ更新)



赤がMC、緑がTD

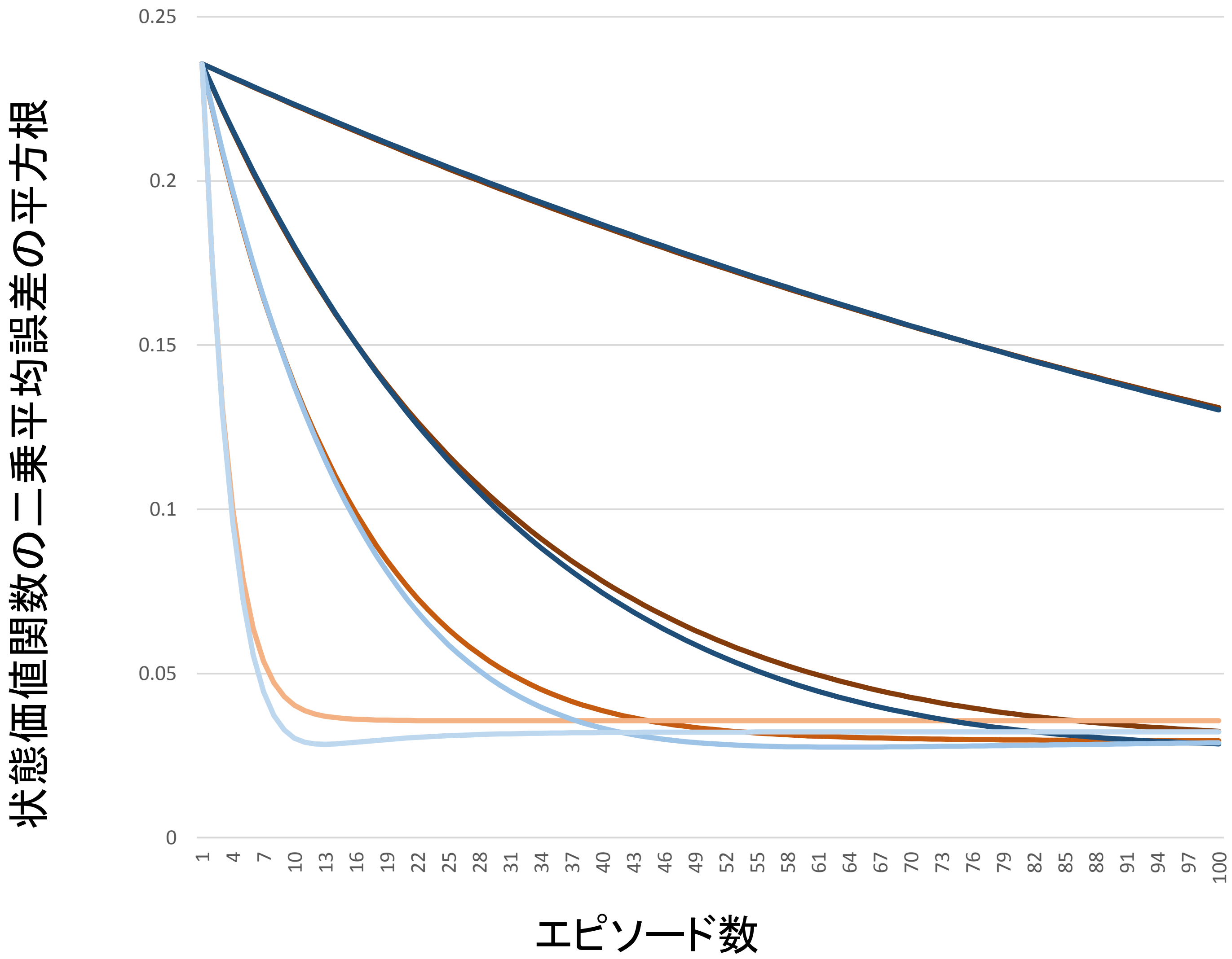
同じ経験データを元に収束するまで学習させた。

横軸は与えられたデータ数

縦軸は状態価値関数の誤差

(教科書のグラフとはまた別)

シミュレーション結果(バッチ更新)



赤がMC、青がTD

同じ経験データを元にn周り学習させた

学習率は上から

0.0001　　0.0005

0.001　　0.005

の4つ。

なんで収束先が違いうるのか？

以下の学習データを考える

A,0,B,0	B,1
B,1	B,1
B,1	B,1
B,1	B,0

Bからは確率3/4で収益１で終了

確率1/4で収益0で終了

よってV(B) = 3/4

V(A)について

①直後に確率１でＢへ移る

V(B)=3/4より、V(A) =3/4

②Aを通ったデータの報酬は必ず0で終わっている

V(A) = 0

①がTD　②がMC

TDを機械学習に使ってみよう

- 一般化反復法を用いてより良い方策πをみつける

$V^{\pi}(s)$を求める ⇆ πを更新する

- ただし、TDではDPと異なり、環境が行動によってどう変化していくかのモデルが分からない。これではVが求まっても新たな行動の評価ができない。

- そこで、Vではなく行動価値関数Qの方を求める。

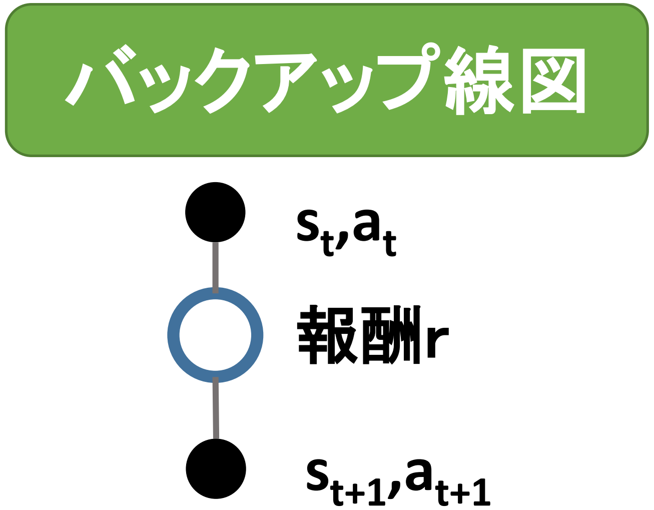
方策オン型TD(Sarsa)

- 行動価値関数の更新は次の式で行える
- $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$

この更新式に必要な$(s_t, a_t, r_t, s_{t+1}, a_{t+1})$の頭文字をとりSarsaという

Q^{π}の評価をしながら、方策πをQ^{π}に対してεグリーディとなるよう更新する

具体的なアルゴリズムは次のスライド



方策オン型TD(Sarsa)

Q(s,a)を任意に初期化
各エピソードに対して繰り返し:
　　s初期化
　　Qから導かれる方策(たとえばεグリーディ)を用いてsでとる行動aを選択
　　エピソードの各ステップに対して繰り返し:
　　　　行動aをとり、r,s′を観測する
　　　　Qから導かれる方策を用いてs′でとる行動a′を選択
　　　　$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$
　　　　$s \leftarrow s'$; $a \leftarrow a'$;
　　sが終端状態なら繰り返しを終了

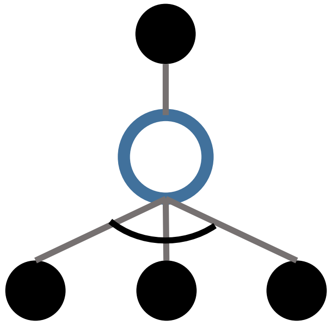
方策オフ型TD(Q学習)

- Q$^{\pi}$を求めるのではなく、最初からQ*(最適行動価値関数)を求める。
- Q*に収束するような更新式は以下

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_{a} Q(s_{t+1},a\ ) - Q(s_t,a_t)]$$

次に選ぶ行動を行動価値関数が最も大きくなるようなものにして更新

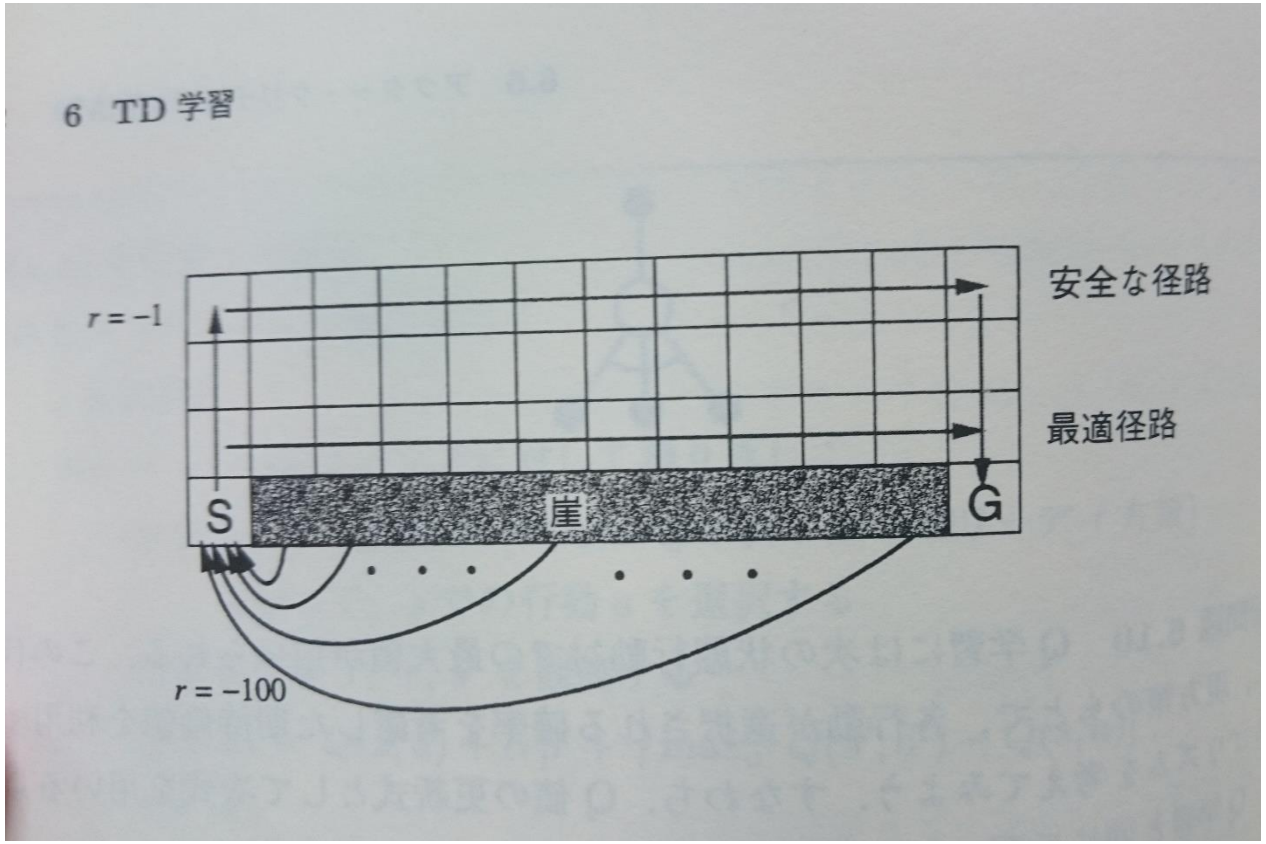
バックアップ線図



方策オフ型TD(Q学習)

Q(s,a)を任意に初期化
各エピソードに対して繰り返し:
　　sを初期化
　　エピソードの各ステップに対して繰り返し:
　　　　Qから導かれる方策を使って、sでの行動aを選択する
　　　　行動aをとり、r,s'を観測する
　　　　$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$
　　　　s←s';
　　sが終端状態なら繰り返しを終了

シミュレーション

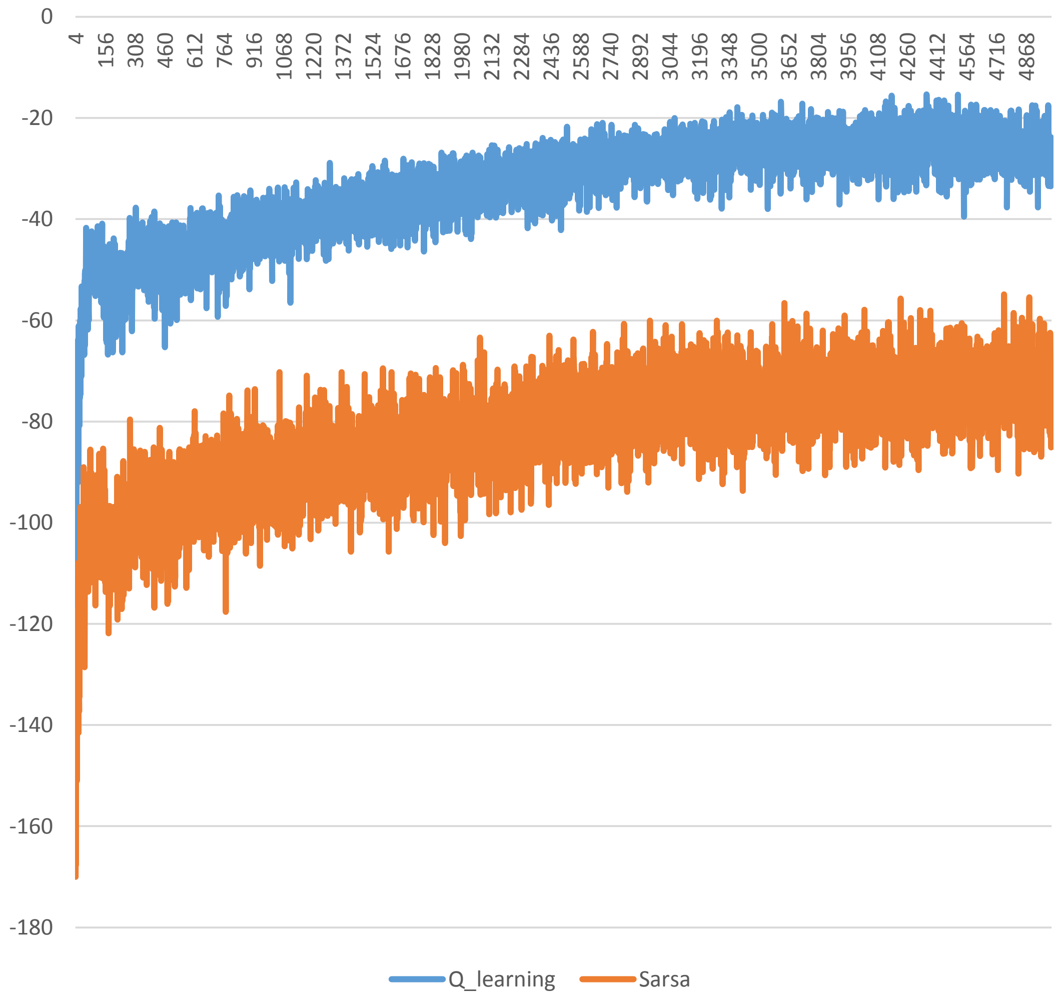


SからGへ移動する最短経路を学習させたい。

一歩進むごとに報酬として-1を与え、崖に落ちると報酬-100かつSへ戻る。

最短経路は崖すれすれだが、崖から離れた経路の方が崖に落ちにくくなる。

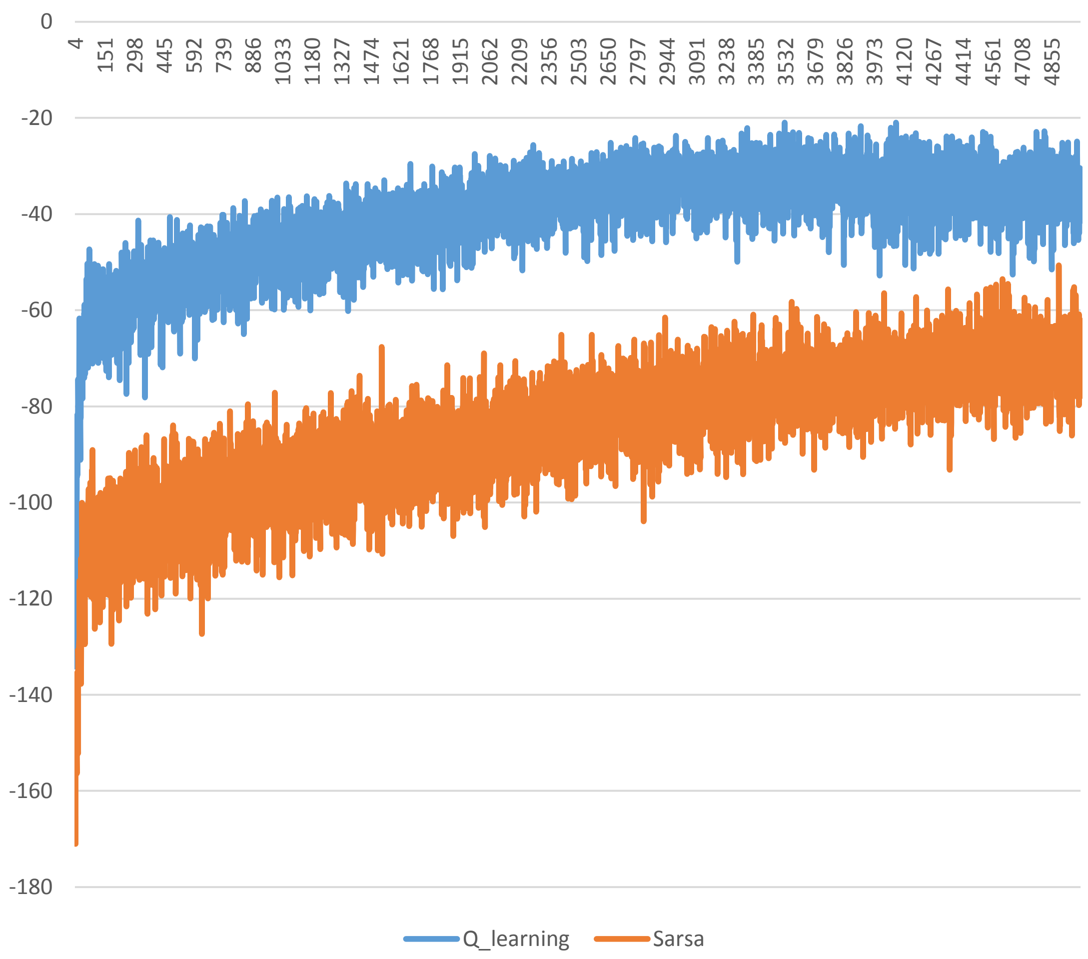
シミュレーション結果



- 赤がSarsa 青がQ学習
- このシミュレーションでは

ε＝1/50とした。

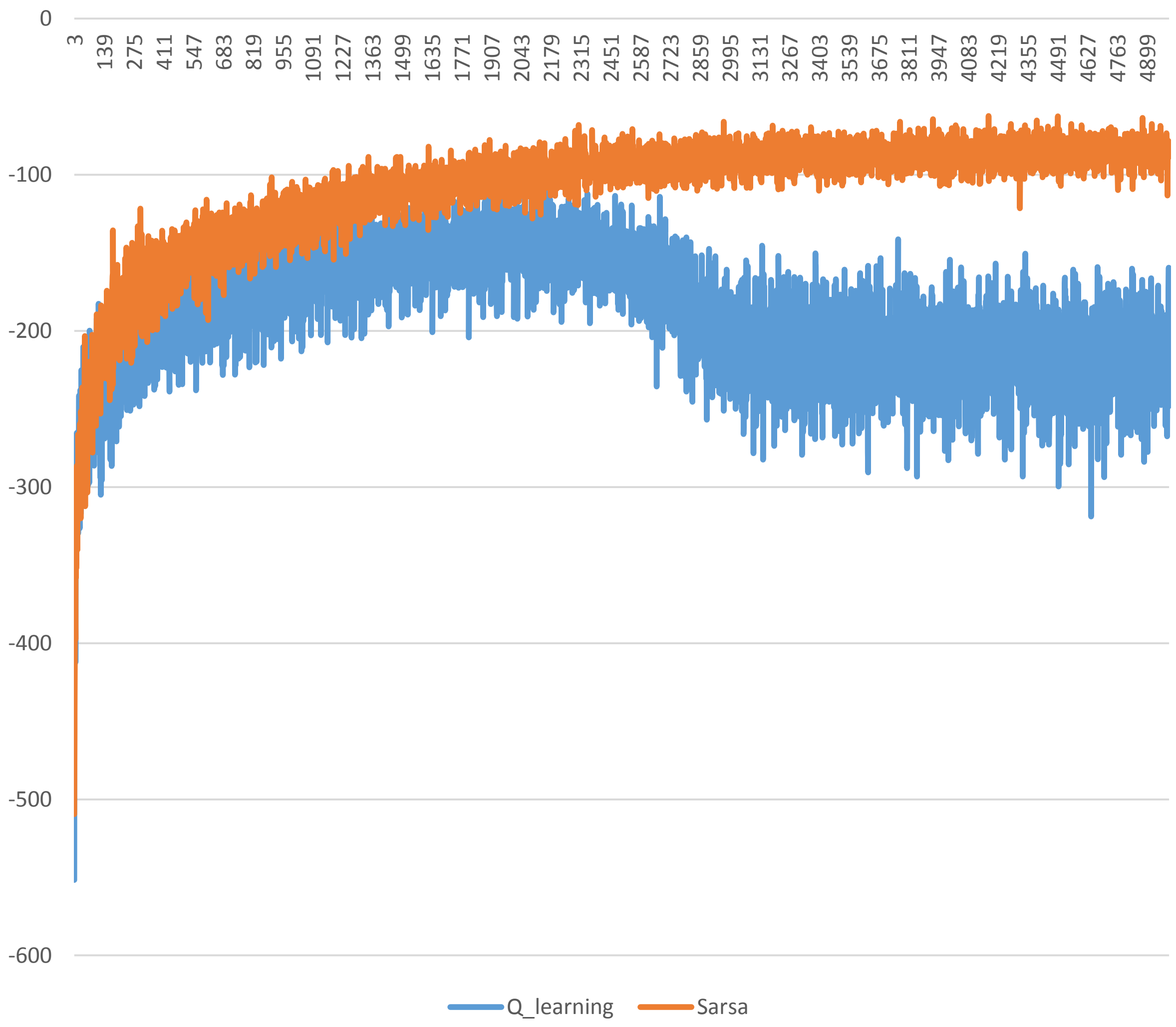
シミュレーション結果



- 赤がSarsa 青がQ学習
- このシミュレーションでは

ε＝1/30とした。

シミュレーション結果



- 赤がSarsa 青がQ学習
- このシミュレーションでは

ε＝1/5とした。