原著論文

Discounted UCB1 - tuned の Q 学習への適用†

野津 亮*・本多 克宏*

本論文ではQ学習に、重み付き価値推定値と重み付き分散を用いるDiscounted UCB1-tuned を行動選択政策として組み込むためのアルゴリズムを提案し、その性能を実験によって評価した。Discounted UCB1-tuned は価値推定値の探索と開拓を最適化した手法の一つであり、一般的によく用いられる ϵ -greedy 手法などよりも性能が良いことで知られている。まず、初期値と学習率に関するいくつかの非定常N本腕バンディット問題の予備実験を通してその性能を比較し、その性能を確認する。そして、価値推定値の更新がない行動、すべて更新している場合は価値推定値の更新した部分で最もUCB値の高い行動を選択するというアプローチを提案法とし、連続空間ゴール探索問題を題材とした従来法と提案手法の数値実験結果を比較、検討する。

キーワード：強化学習, Discounted UCB1-tuned, Q学習, 知的エージェント

1. はじめに

近年、A/Bテストなど、政策決定手法がwebマーケティングの分野で盛んに活用、研究され、様々な応用されてきている。オバマ大統領の選挙キャンペーンでwebページのデザイン選択に用いられたりしたことは有名であるし[1]、人工知能の分野では急速に発展するコンピュータ囲碁などにおいて、Zenなどの人工知能は、UCB (Upper Condence Bound) アルゴリズム[2]をベースとしたUCT探索[3, 4]の要素を持ち、その強さでよく知られるようになってきた[5]。このように、UCBのような政策決定問題(多腕バンディット問題)に対するアルゴリズムは探索(exploration)と開拓(exploitation)のバランスを対象とする様々な方法に応用でき、数ある選択肢の中でどれを効率よく選択すべきかを検討するのに適した手法である[6-8]。

Q学習[9-13]に代表される状態行動空間における強化学習においても、行動選択における探索と開拓のバランスをどのようにとるかは非常に大きな問題である[14]。UCBに代表される手法をこのような強化学習に応用することを考えると、通常、UCB手法は静的な状況を想定しているため、変化する環境では探索の効率が落ちることが予想される。価値推定値が動的に変化していく環境に適用させるための工夫が必要となる。一方で、UCB1-tuned[2]は価値推定値の分散も

利用しながら最大損失を推定しているため、UCBよりは動的な環境に対する適応可能性を持つ。さらに、動的な環境に対する学習方法として、Discounted UCBなどでは価値推定値の重み付き平均をとることによって、直近の経験を重視し、変化に対応可能としている[15]。また、多腕バンディット問題で、一部の環境がスワップするような問題に特化したメタな学習機構を備えたAdapt-EvEなどの方法も提案されている[16]。これらは環境の変化に強いモデルとなっている。

本論文では、Q学習で行動選択手法の探索と開拓のバランスを最適化させるために、Discounted UCB1-tuned を行動選択政策として組み込むためのアルゴリズムを提案し、その性能を評価する。重み付き価値推定値とその分散の計算手法を整理し、より現時点のデータに近い価値推定値とその分散をUCB値として計算することで学習性能をどの程度改善できるのか検証する。UCB手法を適用するにあたり、問題となる点は、Q学習ではQ値の初期値を高めに設定しておく、意図的に探索させたり、更新できない状態と行動が選択されにくいように初期値に0に設定しておくなど、設計者の意図を反映できるが、UCBでは初期値を与えることは特に検討されていないという点である。今回は3.1節に示す簡単な予備実験を通して、初期値を与えない方がUCB手法には適していることがわかったので、こちらをベースに検討する。また、UCBを採用する場合、価値推定値を更新できない状態と行動にも対応しなければならない。UCB手法では未更新の行動が選択されつづけてしまうからである。これに関しては今回は未更新の行動を優先して選択し、更新されるか一定回数行動が選択された後に

† Discounted UCB1 - tuned for Q - Learning
Akira NOTSU and Katsuhiro HONDA

* 大阪府立大学大学院 工学研究科 電気・情報系専攻 知能情報工学分野
Graduate School of Engineering, Electrical Engineering and Information Science, Osaka Prefecture University

はUCB 値の最も高い行動を選択するというアプローチを採用した。

まず、従来の行動選択手法とその学習方法を示し、いくつかの非定常 N 本腕バンディット問題の予備実験を通してその性能を比較する。次に提案手法の連続空間ゴール探索問題を題材とした数値実験結果を示し、検討する。

2. 基礎理論

2.1 N 本腕バンディット問題と政策

N 本腕バンディット問題では、目の前に N 本の選択可能なレバーがあるとし、レバーを選択すると当たりか外れとなり、当たると報酬をもらえる。レバーによって当たる確率に違いがあり、どのように選択すれば得られる報酬が最も大きくなるかという問題である。以下のようなアプローチがよく用いられる。

ϵ -greedy 手法 一番大きい価値推定値 $\bar{X}_i$ を持つ行動を決定的に選択することを、greedy な行動をとる手法と呼び、 ϵ ($0 \leq \epsilon \leq 1$) の確率でランダムに選択し、確率 $1-\epsilon$ で greedy な行動をとることを ϵ -greedy 手法と呼ぶ。パラメータとなる ϵ の意味が分かりやすく、実装しやすい。プレイの数を増やし、極限に至るならば、すべての行動が無限回試されるため、真の価値に収束することが保証されている。しかし、探索が無作為であるため、最悪の行動を選択する可能性が残りがちである。価値の大小関係と選択確率の大小関係を対応させた Soft Max 手法などもあるが、大小の基準点の定義が難しいという問題がある。

UCB 価値推定値 $\bar{X}_i$ の代わりに、UCB 値と呼ばれる以下の値の一番高いものを選択する手法である [2]。最適戦略ではそれぞれのマシンを $O(\log n)$ 回プレイする [17] という条件を満たしている。この手法では有意水準を $1/n$ で見積もったときの区間推定における上側信頼限界を比較している。分散はどのレバーも一定のものとし、初回は必ず未選択のレバーを選択する。

$$UCB_i = \bar{X}_i + C \sqrt{\frac{\ln n}{n_i}}$$

i はレバーの番号、 n は今までの総プレイ回数、 n_i はレバー i が選択された回数を意味する ($n = \sum_i n_i$)。また、 $\bar{X}_i$ はレバー i のその時点までの報酬の期待値である。 C は探索の傾向を定める定数である。第一項は価値推定値、第二項は探索をしなければ相対的に大きくなる値となっている。

総プレイ回数を小さく保ちつつ高い報酬を期待することができるが、基本的に期待値の高いところにリソースが集中するため、時間変化を伴うような動的環境には適していない。

UCB1-tuned UCB を報酬のばらつきを考慮し改良したもので、以下の値を UCB1-tuned の値として、その値の一番高いものを選択する [2]。性質は UCB と似ているが、報酬の分散 V_i を考慮した式にすることでより性能が良くなっている。

$$UCB1t_i = \bar{X}_i + C \sqrt{\frac{\ln n}{n_i} \min \left\{ \frac{1}{4}, V_i + \sqrt{\frac{2 \ln n}{n_i}} \right\}}$$

2.2 Q 学習

Q 学習は定められた政策に従いつつ状態と行動の組に対して価値推定値を与えていく強化学習のひとつの手法である [9-11]。Q 学習のひとつの強みは、環境モデルを必要とせずに、政策によらない価値推定値である Q 値を計算していく点にある。

価値推定値 Q は、エージェントの現在の状態 s 、エージェントに選択された行動 a 、行動 a を取ったあとに到達した次状態 s' 、そしてエージェントの取った行動により得られる報酬 R に依存して Q 値を更新する。更新は下記の式により行われる。

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[R + \gamma \max_{a' \in A(s')} Q(s', a') - Q(s, a) \right]$$

ここで α は学習率、 γ は割引率を示している。

3. Discounted UCB と提案方法

価値推定値に関して、Discounted UCB と従来法には、初期値を設定するかどうか、学習率を変動させるかという 2 点の大きな違いがある。探索させたい価値推定値の初期値を大きくしてやればそこを優先的に探索するようになるし、また、学習率を固定することで初期値の影響を残しつつシンプルな学習を実現することができる。一方で、Discounted UCB では初期値を用いない後述する重み付き平均が用いられている。ここでは「いくつかの大きさの初期値を設定することと初期値を設定せずに重み付き平均を用いることの性能の違い」と、学習率を変動させて「価値推定値に過去の経験の平均を用いる場合 (UCB) と重み付き平均を用いる場合 (Discounted UCB) の性能の違いとパラメータ C の関係」を調査し、提案法を検討する。

Q 学習などの一般的な強化学習では学習率を 0.1 などの値に設定し、価値推定値を更新していくことによって過去のデータの重みを徐々に減少させていくことが多い。しかしながら、この方法では初期値の影響を完全に排除することはできない。たとえば、 n_i 回データを観測し、価値推定値を更新したとき、価値推定値の初期値 $x_{i,0}$ は減衰率 r ($0 < r < 1$) の元で価値推定値 $\bar{X}_{i,n_i}$ のうち、 $x_{i,0} r^{n_i} / \bar{X}_{i,n_i}$ の割合の影響を持つ。この値は 0 に収束するが、観測回数が少ない場合は依然として大きなままである。実験 3.1.1 に示すように、 N 本バンディット問題において初期価値推定値が 1 の時と 0 の時とでは学習プロセスが大きく異なってくる。

Discounted UCB では減衰率を r ($0 < r < 1$) としたときに、 n_i 回目の価値推定値の更新で、 $\bar{X}_{i,n_i} = (\sum_{j=1}^{n_i} x_{i,j} r^{n_i-j}) / (\sum_{j=1}^{n_i} r^{n_i-j})$ という重み付き平均を価値推定値としている。 $x_{i,j}$ はレバー i を選択したときの j 回目の観測データである。このとき、 n_i 回目の学習率は $(1-r) / (1-r^{n_i})$ と自動的に決まる(付録参照)。逆に言えば、学習率をこのように設定することで初期値の影響をなくしつつ、過去のデータを減衰させていくことが可能になる。価値推定値を重みなしの通常のアVERAGEとしたときは学習率を $1/n_i$ とすることで可能となる(付録参照)。また、価値推定値の標準分散についても上記と同様の考察ができ、重み付き標準分散を

$$V_{i,n_i} = \frac{\sum_{j=1}^{n_i} (x_{i,j}^2 - \bar{X}_{i,n_i}^2) r^{n_i-j}}{\sum_{j=1}^{n_i} r^{n_i-j}}$$

と定義すれば、

$$V_{i,n_i+1} = \frac{r(1-r^{n_i}) / (1-r) (V_{i,n_i} + \bar{X}_{i,n_i}^2) + x_{i,n_i+1}^2}{(1-r^{n_i+1}) / (1-r)} - \bar{X}_{i,n_i+1}^2$$

と逐次更新していくことができる(付録参照)。

3.1 初期値と学習率に関する基礎実験

Q 学習中では価値推定値が学習進度によって変化していくことを想定し、この場合に各手法がどの程度の性能を持つかに関して調査した。通常の Q 学習のように初期値を設定し、学習率を固定させて価値推定値を計算する方が良いのか、あるいは UCB のように平均値や Discounted UCB のように重み付き平均を価値推定値に用いる方が良いのかを検討する。

簡単な N 本腕バンディット問題(非定常)実験を行う。この実験では、当選確率が変化する仮想的なスロットマシンを 2 つ用意し、50 回トライする間の価値期待値最大化を目指す。これは非定常 2 本腕バンディット

問題であり、1 状態、2 行動の Q 学習とも捉えることができる。

スロットマシンの組としては、当たる確率が途中で急に変化する不連続な当選確率を持つ 2 台のスロットマシンパターン A と、当たる確率が徐々に変化する 2 台のスロットマシンパターン B を用意した。これは強化学習中の価値推定値が新しいルートの発見によって急に変化する場面と通常の緩やかな価値推定値の更新という場面に暗に表現したものである。 n を選択回数として前者(パターン A)は一方が当たる確率 0.9 から 0.1 へ ($P_1(n)$)、もう一方は逆に 0.1 から 0.9 へ ($P_2(n)$) トータル挑戦回数 26 回目で変化する。後者(パターン B)は、一方が当たる確率が 0 から 1 へ、さらに 0 に変化するように、 $\sin$ 関数 (0 から π) に基づき定義されていて ($P_3(n)$)、もう一つのスロットマシンは逆に $1 - \sin$ (0 から π) に基づき定義されている ($P_4(n)$)。式と図で示すと以下ようになる。

スロットマシンパターン A

$$P_1(n) = \begin{cases} 0.9 & \text{if } n \leq 25 \\ 0.1 & \text{else} \end{cases}$$

$$P_2(n) = \begin{cases} 0.1 & \text{if } n \leq 25 \\ 0.9 & \text{else} \end{cases}$$

図 1 SLOT1: $P_1(n)$

図 2 SLOT2: $P_2(n)$

スロットマシンパターン B

$$P_3(n) = \sin((n-1)\pi/49)$$

$$P_4(n) = 1 - \sin((n-1)\pi/49)$$

図 3 SLOT3: $P_3(n)$

図 4 SLOT4: $P_4(n)$

3.1.1 初期値問題

第 1 に、 ϵ -greedy 政策と UCB1-tuned 政策で、初期価値推定値を与え、学習率を固定した場合について実験した。以下の図 5-12 は縦軸が平均累積勝率、横軸が選択回数 n で、1000 回の平均値である。図の上

図5 パターン A: ϵ -greedy 初期値の影響 図6 パターン B: ϵ -greedy 初期値の影響

図7 パターン A: UCB1-tuned 初期値の影響

図8 パターン B: UCB1-tuned 初期値の影響

部にプロットされるほど成績が良いということを意味する。図5-8は学習率 $\alpha = 0.1$ 、各価値推定値の初期値が0, 1/4, 1/2, 3/4, 1のときの学習プロセスの変化を示す。価値推定値は以下のように更新する。

$$\bar{X}_n = (1 - \alpha)\bar{X}_{n-1} + \alpha x_n$$

ϵ -greedy 政策, $\epsilon = 0.1$ の場合(図5, 6), 特に初期価値推定値が低い場合($x_0 = 0$), 性能が悪いことがわかる。たとえ0.1という低い価値推定値を持つ選択肢でも, 初期値よりも高いので選ばれやすくなる。このことが非常に大きな影響力を持っていることが明らかになった。

また, Discounted UCBにおける価値推定を行い, ϵ -greedy 法で行動選択した場合についても調査した。つまり, 初期値の影響を受けないように減衰率を考慮した学習率 $(1-r)/(1-r^n)$, $r = 1 - \alpha = 0.9$ を適用した(図中の $r = 0.9$)手法である。初回は未経験のレ

バーを選択する。

ϵ -greedy 手法で, この推定方法は初期値が3/4 や 1のときと比べ期待通りには働かなかった。定常状態の時の学習では問題がないが, 初期値を考慮しない分, 初期値を「探索可能性を残しておくこと」などの効果に使うことができず, 学習があまりうまくいっていない。もし, スロットマシンの当たる確率の平均値や変動があらかじめわかっている場合, 初期値を高めに設定しておき, 探索可能性を大きめに残しておくことができる。

このように ϵ -greedy 手法でも, 初期値を適切に設定してやれば数回の環境の変化に対応が可能であり, この実験のような学習初期の挙動だけを考えるのであれば十分な性能を持つということがわかった。しかしながら, 条件によって適切な値が異なるため, より詳しい議論が必要である。未知の選択肢については大きめの価値推定値を設定し, 経験後は初期値を考慮しない更新方法によるのが一般的な問題に適しているよう

図9 パターン A: UCB1-tuned

図10 パターン B: UCB1-tuned

図11 パターン A: Discounted UCB1-tuned

図12 パターン B: Discounted UCB1-tuned

に思われる。

UCB1-tuned 手法で、価値推定値の学習率を0.1に固定し、初期値の影響を調査した(図7, 8)。図はパラメータ $C = 1$ の時、初期価値推定値を0, 1/4, 1/2, 3/4, 1 と変えたときの結果と初期値の影響を受けない通常のDiscounted UCB1-tuned (図中の $r=0.9$)の結果を示している。前実験とは異なり、この場合は重み付き平均をとって初期値の影響を受けないにした方が成績が良く、わずかではあるが初期値ありの、 ϵ -greedy 政策よりも高い平均累積勝率を示した。探索可能性がUCB 値によって最適化されているために、重み付き分散の初期値の影響がUCB 値に悪影響を及ぼし、適切な行動選択ができないためと思われる。

3.1.2 重み付き価値推定値の効果

第2に重み付き価値推定値の非定常状態に対する学習効果を確認するために、価値推定値として、平均と

分散を用いた通常のUCB1-tuned の場合(図9, 10)と重み付き平均と重み付き分散をUCB1-tuned に加えた場合(=Discounted UCB1-tuned) (図11, 12)についても実験した。予想通り、状況が変わった場合の変化に対して、Discounted UCB1-tuned の方が若干ではあるがより追従できていることが図からわかる。パターンB では、パラメータによっては差がほとんどないものもあるが、同様の結果が観察された。全体を通して、Discounted UCB1-tuned ($C = 1, 2$)が良い成績であった。

3.2 Discounted UCBQ Algorithm

Q 学習において初期値を利用することには2つの意味があると考えられる。上記の3.1の実験のように価値推定値の収束を早めたり、探索可能性を残すことと、もう一つは、初期設定値を低くすることで更新不能状態と行動が選択されにくくするというものである。前

者については初期値を設定しないDiscounted UCB1- 目的達成には、一定回数選択しても価値更新がない行
tuned が成績が良かったため、これを採用し、後者の 動を選択しないことで対応することを考える。取り扱

表1 Discounted UCBQ Algorithm

procedure Discounted UCBQ Algorithm
begin
initialize $Q, V, UCB, N_a, NN_a, \forall s \in S, \forall a \in A, \max R = 1, \min R = 0$
set initial state s_0 and goal state s_n
for cycle:= 1 to MAXCYCLE do
$s := s_0$
while $s \neq s_g$ do
if $\exists a \in A, \text{s.t. } N_a(s, a) = 0 \wedge NN_a(s, a) < \theta$
$a := \text{randomselect } a \text{ s.t. } N_a(s, a) = 0$
else
$a := \text{argmax } UCB(s, :)$
end
$NN_a(s, a) := NN_a(s, a) + 1$
$R_t := \text{GetReward}(s, a)$
if $R_t > 0$
$\max R := \max(R)$
else if $R_t < 0$
$\min R := \min(R)$
end
$NR := (R_t - \min R) / (\max R - \min R)$
$s' := \text{GotoNextState}(s, a)$
if $s \neq s'$
$N_a(s, a) := N_a(s, a) + 1$
$Q'(s, a) := Q(s, a) + \alpha [NR + \gamma \max_{a' \in A(s')} Q(s', a') - Q(s, a)];$
$V(s, a) := \frac{(r(1-r^{N_a(s, a)-1})/(1-r))(V(s, a) + Q(s, a)^2) + (NR + \gamma \max_{a' \in A(s')} Q(s', a'))^2}{(1-r^{N_a(s, a)})/(1-r)} - Q'(s, a)^2$
$Q(s, a) := Q'(s, a)$
for ALL a at s
if $N_a(s, a) > 0$
$UCB(s, a) := Q(s, a) + C \sqrt{\frac{\ln \sum (N_a(s, :))}{N_a(s, a)}} \min \left\{ \frac{1}{4}, V(s, a) + \sqrt{\frac{2 \ln \sum (N_a(s, :))}{N_a(s, a)}} \right\}$
else
$UCB(s, a) := 0$
end
end
$s := s'$
end
end
end

う問題によっては、(たとえば、エージェントが空間を移動する際、壁にぶつかったりするため)現状態と別の状態に到達不可能であるために更新不能な $Q(s_i, a_i)$ 値も存在しうる。UCB では必ず未選択の行動を選択し、価値推定値を獲得することを前提としているため、そのまま適用したのでは次状態に遷移できない行動が選択され続けてしまう。よって、価値推定値を更新していない行動の中で、それまでに閾値回数以下の選択しかしていない行動の中から次行動を選択、そうでないときには、過去に更新したことのある行動の中から最大のUCB 値を持つ行動を選択する、という方法で次状態に到達できない行動が選択され続けるということを防ぐ。アルゴリズム中では、行動選択回数 $NN_a(s, a)$ と価値推定値更新回数 $N_a(s, a)$ からこれを判断している。

また、UCB1-tuned における第2項の中の1/4 という数字は、スロットマシンの当たりを1、はずれを0としたときに、各スロットマシンの分散が最大でもこれを超えないという上限を示している。これを考慮し、Q 学習における価値推定値 Q をUCB1-tuned におけるUCB 値の第1項 $\bar{X}_i$ と捉えれば、受け取る報酬の大きさを区間 $[0, 1]$ に基準化することで、UCB 手法を行動選択手法として適用可能になると予測できる。一つの方法として、報酬 R の最大値を $\max R$ 、最小値を $\min R$ として、 $(R - \min R) / (\max R - \min R)$ を基準化された報酬 NR として Q 値の計算に用いれば良いと考え、このとき、報酬が獲得されるまでは仮の値として $\max R = 1, \min R = 0$ を用いるとする。

以上の議論をまとめ、Q 学習とDiscounted UCB1-tuned を融合したアルゴリズムを表1のように提案する。報酬を $[0, 1]$ に基準化し、Q 学習において Q 値に基づいて行動を選択するという部分をUCB 値が最大である行動を選択するというものに変更する。UCB 値を計算するために必要な分散の値 $V(s, a)$ や Q 値の更新回数 $N_a(s, a)$ も保存させる。更新不能な状態行動対にも対応するため、UCB 値は更新できるものについてのみ更新し、それ以外のは0とする。また、学習率 α を $(1-r) = (1-r^{N_a(s, a)})$ とする。

4. 連続空間経路探索問題への適用

今回、部分観測マルコフ決定過程である、連続空間での学習問題として、強化学習のベンチマーク問題として用いられているIEEE 2007 Car Racing Competition で行われたコンピュータプログラム同士の対戦ゲーム [18-20] を参考に、簡単な連続空間経路探索ゲームを作成し、実験した。

本シミュレーションでは、カーエージェントはゴー

図13 エージェントの行動選択イメージ(8方向のとき)

図14 エージェントの動きの例(赤学習前, 青学習後)

ルするたびにもらえる報酬を最大化するように状態ごとの出力を学習する。ゴールまでを1サイクルとし、次のゴールはランダムにフィールド上に生成される。エージェントは N_d 方向(等間隔に分割)と N_p 出力を持った $N_d \times N_p$ 通りの行動を持ち、状態はゴールへの方向と相対距離の空間をそれぞれ N_{dg} と N_{lg} ずつ分割して、 $N_{dg} \times N_{lg}$ ある。フィールドは x 座標 y 座標ともに $0-1$ の範囲の正方形であり、壁にエージェントがぶつかった場合はそのぶつかった方向の速度を0にした。図13に状態空間の分割イメージ、図14に初期サイクルと200サイクル目のエージェントの動きの例を示す(200サイクル目は199サイクルの終了時の座標からエージェントが動くため、図のようにフィールドの途中から動き始める)。ゴール時に報酬 R が与えられ、それ以外では報酬はない($R=0$)。

方向と出力の値に応じて決定される加速度ベクトルが速度ベクトル V に加えられ、座標 X が更新される。加速度を A 、減衰率を ζ としたときに、

$$V(t+1) = \zeta V(t) + A(t)$$

$$X(t+1) = X(t) + V(t+1)$$

となる。

実験環境のパラメータは表2のように設定した。

表2 実験パラメータ

フィールドの大きさ (x, y)	$x, y \in [0, 1]$
出力角度分割数	$N_d = 10$
出力の大きさ	$N_p = 2$ 通り
	$ A \in \{0.005, 0.01\}$
状態距離分割数	$N_{dg} = 4$
状態角度分割数	$N_{lg} = 10$
ゴール直径	$D = 0.01$
速度減衰率	$\zeta = 0.1$
ゴール報酬	$R = 1$
割引率	$\gamma = 0.9$
1 サイクル目スタート地点	$(x, y) = (0, 0)$

この実験は、エージェントは大まかで相対的な座標のみを観測でき、正確な座標や加速度については観測できないという、部分観測マルコフ決定過程である。同状態で同行動をとったとしても必ずしも同じ次状態に移行したり、同じ価値推定値を得るとは限らない。また、価値推定値は報酬を得た状態から順に伝搬してくるので少しずつ変化もする。よって、これは3章での実験を多段階に連ねたものにとらえることができる。

実験結果と考察

図15に、縦軸にサイクルごとの平均移動回数、横軸にサイクル数を各手法(Q学習(ϵ -greedy: $\epsilon = 0.2$, 初期値0), Q学習(ϵ -greedy: $\epsilon = 0.2$, 初期値1), Discounted Q学習(ϵ -greedy: $\epsilon = 0.2$, 初期値1, 学習率可変: $r = 0.9$) (Q学習における価値推定をDiscounted UCBと同じにしたもの), Q値+UCB手法(学習率一定: $\alpha = 0.1$, 初期値0), Q値+UCB手法(学習率一定: $\alpha = 0.1$, 初期値1), Discounted UCBQ(学習率可変: $r = 0.9$))ごとにプロットした。図16には平均移動回数の標準偏差を示す。試行回数1000回の平均、標準偏差をとっている。Q値+UCB手法は提案手法の学習率 α を固定したものとし、Discounted UCBQでの未更新行動の最大選択回数 $\theta = 50$, また、どのUCB手法でも $C = 1$ とした。

学習最初期(1, 2サイクル)では、ほとんどの学習プロセスで1000回移動をしてもゴールにたどり着かず次のサイクルに移る。そのために、どの手法、どの試行でもほぼ移動回数1000回を記録するので、ばらつきである標準偏差が小さくなっている。

大きな特徴として、学習初期(25サイクルくらいまで)では、初期値を1にすることで未経験の行動が選択されやすくなるという仕組み、あるいは初回は未更

図15 平均ゴールターン

図16 ゴールターンの標準偏差

新の行動を選択するというUCB手法が効率的な学習につながっていることがわかる。

さらにその後のサイクルでは、Discounted Q学習やDiscounted UCBQのように、価値推定値を重み付き平均にすることで他の手法に比べ探索を効率化することができた。この2手法を比べるとUCB手法を用いた方がより平均移動回数が少なく、より探索の効率を上げることが可能であると確認できた。初期値0のQ学習の ϵ -greedy手法と比較して、提案手法は4分の1くらいの平均移動回数を記録するサイクルを持つことがわかった。また、その三つ以外の手法に関しては、学習初期以降、ほぼ同じような学習プロセスであった。

第3章での基礎実験では、 ϵ -greedy手法は初期値固定+学習率固定のときの成績がよかったが、ここでは学習率はDiscounted UCB1-tunedのように重み付き平均を価値推定に用いた方が成績が良くなった。これは基礎実験では、期待値の高いスロットマシンが短いスパンで1, 2度入れ替わる環境であるのに対して、この実験ではより長い学習時間の割にそのような

入れ替わりが起こりにくいことと、価値推定値の変動が基礎実験のスロットマシンのように 0 か 1 かのような極端な値よりは小さな範囲で起こることなどが影響しているのではないかと考えられる。

5. おわりに

本論文では政策最適化手法である UCB1-tuned と Q 学習を融合させた手法を提案した。この手法は各状態における行動選択を最適化し、学習中の探索と開拓のバランスを効率よく行うことができる。各状態行動の UCB 値を計算するに当たり、重み付き平均、重み付き分散、スケールにかかるパラメータの調整、計算方法などを整理、提案した。簡単な予備実験にて Discounted UCB1-tuned の性能を確認し、より実践的な部分観測非マルコフ決定過程を含む問題においてもよく機能することを示せた。学習効率が飛躍的に上がり、一般的な手法と比較して、ゴールに至るまでの平均行動数が 4 分の 1 になるサイクルもあるほどであった。

今後の課題としては、計算量、必要メモリ量の削減や手続きの簡略化などが挙げられる。今回の提案法では行動の選択回数を記憶しておき、計算に利用しているが、この情報に前後の文脈を組み込むことで有効活用すればより学習効率を上げることができるのではと考えている。

参 考 文 献

- [1] Kyle Rush homepage, <http://kylerush.net/blog/optimization-at-the-obama-campaign-ab-testing/> (2014/12/19 アクセス)
- [2] P. Auer, N. Cesa-Bianchi, and P. Fischer, Finitetime analysis of the multiarmed bandit problem. Machine Learning, 47, 2-3, pp.235-256, 2002.
- [3] L. Kocsis, C. Szepesvari, Bandit based monte carlo planning, In European Conference on Machine Learning, pp.282-293, 2006.
- [4] N.R. Sturtevant, An Analysis of UCT in Multiplayer Games, Proc. of the 6th international conference on Computers and Games, pp.37-49, 2008.
- [5] 加藤英樹, Zen のアーキテクチャ, 人工知能学会誌, 27, 5, pp.501-504, 2012.
- [6] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, Gambling in a rigged casino: the adversarial multi-armed bandit problem, Proc. of the 36th Annual Symposium on Foundations of Computer Science, pp.322-331, 1995.
- [7] N. Cesa-Bianchi, Prediction, learning, and games, Cambridge University Press, 2006.
- [8] 梁曾漢, 小宮山純平, 大岩秀和, 佐藤一誠, 中川裕志, 多腕バンディットにおけるリグレットの非線形拡張, Proc. of the 28th Annual Conference of the Japanese Society for Artificial Intelligence, 2H1-4, 2014.
- [9] R. S. Sutton, and A. G. Barto, Reinforcement Learning – An Introduction –, The MIT Press, pp.32-33, 1998.
- [10] C. J. C. H. Watkins, and P. Dayan, Technical Note: Q-learning, Machine Learning, 8, 1992.
- [11] T. Jaakkola, M. Jorban, and S. Singh, On the convergence of stochastic iterative dynamic programming algorithms, Neural Computation, 6, 6, 1994.
- [12] 木村元, 宮崎和光, 小林重信, 強化学習システムの設計指針, 計測と制御, 38, 10, pp.618-623, 1999.
- [13] Reinforcement Learning FAQ: Frequently Asked Questions about Reinforcement Learning, Edited by Rich Sutton <http://webdocs.cs.ualberta.ca/~sutton/RLFAQ.html>
- [14] 高玉圭樹, マルチエージェント学習 – 相互作用の謎に迫る –, コロナ社, 2003.
- [15] L. Kocsis, and C. Szepesvari, Discounted UCB, 2nd PASCAL Challenges Workshop, 2006.
- [16] C. Hartland, S. Gelly, N. Baskiotis, O. Teytaud, and M. Sebag, Multi-armed bandit, dynamic environments and meta-bandits, Workshop Online Trading Exploration Exploitation, 2006.
- [17] R. Agrawal, Sample mean based index policies with $o(\log n)$ regret for the multiarmed bandit problem. Advances in Applied Mathematics 27 pp.1054-1078 1995.
- [18] IEEE CEC 2007 Car Racing Competition: <http://julian.togelius.com/cec2007/> (2014/12/19 アクセス)
- [19] S. Fujii, T. Nakashima, and H. Ishibuchi, A study on constructing fuzzy systems for high-level decision making in a car racing game, Proc. of 2008 IEEE International Conference on Fuzzy Systems, pp.2299-2306, 2008.
- [20] 馬野元秀, 岸本文哉, 瀬田和久, カーレースゲームへのファジィ Q 学習の適用, 第 26 回ファジィシステムシンポジウム論文講演集, pp.75-80, 2010. (2014年3月20日 受付) (2014年8月26日 採録)

付録

平均値と学習率 α の関係

$$\begin{aligned}
 \bar{X}_n &= \frac{x_1 + x_2 + \cdots + x_{n-1} + x_n}{n} \\
 \bar{X}_{n+1} &= \frac{x_1 + x_2 + \cdots + x_n + x_{n+1}}{n+1} \\
 &= \frac{n\bar{X}_n + x_{n+1}}{n+1} \\
 &= \frac{n}{n+1}\bar{X}_n + \frac{1}{n+1}x_{n+1} \\
 &= (1-\alpha)\bar{X}_n + \alpha x_{n+1}
 \end{aligned}$$

重み付き平均における減衰率 r と学習率 α の関係

$$\begin{aligned}
\bar{X}_n &= \frac{r^{n-1}x_1 + r^{n-2}x_2 + \cdots + rx_{n-1} + x_n}{r^{n-1} + r^{n-2} + \cdots + r + 1} \\
&= \frac{r^{n-1}x_1 + r^{n-2}x_2 + \cdots + rx_{n-1} + x_n}{\frac{1-r^n}{1-r}} \\
\bar{X}_{n+1} &= \frac{r^n x_1 + r^{n-1}x_2 + \cdots + rx_n + x_{n+1}}{\frac{1-r^{n+1}}{1-r}} \\
&= \frac{r \frac{1-r^n}{1-r} \bar{X}_n + x_{n+1}}{\frac{1-r^{n+1}}{1-r}} \\
&= \frac{r(1-r^n)}{1-r^{n+1}} \bar{X}_n + \frac{1-r}{1-r^{n+1}} x_{n+1} \\
&= (1 - \frac{1-r}{1-r^{n+1}}) \bar{X}_n + \frac{1-r}{1-r^{n+1}} x_{n+1} \\
&= (1-\alpha) \bar{X}_n + \alpha x_{n+1}
\end{aligned}$$

重み付き分散の式変形

$$\begin{aligned}
\bar{V}_n &= \frac{r^{n-1}(x_1^2 - \bar{X}_n^2) + \cdots + (x_n^2 - \bar{X}_n^2)}{r^{n-1} + r^{n-2} + \cdots + r + 1} \\
&= \frac{r^{n-1}x_1^2 + r^{n-2}x_2^2 + \cdots + rx_{n-1}^2 + x_n^2}{r^{n-1} + r^{n-2} + \cdots + r + 1} - \bar{X}_n^2 \\
&= \frac{r^{n-1}x_1^2 + r^{n-2}x_2^2 + \cdots + rx_{n-1}^2 + x_n^2}{\frac{1-r^n}{1-r}} - \bar{X}_n^2 \\
\bar{V}_{n+1} &= \frac{r^n x_1^2 + r^{n-1}x_2^2 + \cdots + rx_n^2 + x_{n+1}^2}{\frac{1-r^{n+1}}{1-r}} - \bar{X}_{n+1}^2 \\
&= \frac{r \frac{1-r^n}{1-r} (\bar{V}_n + \bar{X}_n^2) + x_{n+1}^2}{\frac{1-r^{n+1}}{1-r}} - \bar{X}_{n+1}^2
\end{aligned}$$

[問い合わせ先]

〒599-8531 大阪府堺市中区学園町1-1

大阪府立大学大学院工学研究科

電気・情報系専攻知能情報工学分野

野津 亮

TEL: 072-254-9355

FAX: 072-254-9355

E-mail: notsu@cs.osakafu-u.ac.jp

著者紹介

のつ あきら
野津 亮 [正会員]

2005年3月京都大学大学院情報学研究科システム科学専攻博士後期課程修了。同年4月より大阪府立大学大学院工学研究科助手、2007年同助教、2012年同准教授現在に至る。博士(情報学)。認知モデルの機械学習への応用、マルチエージェントシステムの研究に従事。日本知能情報ファジィ学会、計測自動制御学会、ヒューマンインタフェース学会の会員。

ほん だ かつひろ
本多 寛宏 [正会員]

1999年大阪府立大学大学院工学研究科博士前期課程電気・情報系専攻修了。同年日本電信電話(株)入社。同年大阪府立大学工学部経営工学科助手、2000年同大学院工学研究科助手、2007年同助教、2009年同准教授、2013年同教授、現在に至る。博士(工学)。ファジィクラスタリングによるデータマイニングやニューラルネットワークの研究に従事。IEEE、日本知能情報ファジィ学会、システム制御情報学会、日本経営工学会の会員。

Discounted UCB1-tuned for Q-Learning

by

Akira NOTSU and Katsuhiro HONDA**Abstract :**

In this paper, we integrated Discounted UCB1-tuned, which uses weighted value and weighted variance, into Q-learning agents and experimentally evaluated its performance. Discounted UCB1-tuned is an optimized selection method that balances exploration and exploitation and outperforms other methods, including ϵ -greedy. We conducted experiments on the effect of default values and learning rate in a multi-armed bandit problem. Our algorithm selects actions its value is not updated or with the highest UCB value in updatable state-actions. We show the results of the continuous state spaces shortest path problem followed by a discussion.

Keywords : Reinforcement learning, Discounted UCB1-tuned, Q-learning, Intelligent agent

Contact Address : **Akira NOTSU**

*Division of Electrical Engineering and Information Science, Graduate School of Engineering,
Osaka Prefecture University*

Gakuen-cho 1-1, Naka-ku, Sakai, Osaka, 599-8531, Japan

TEL : +81-72-254-9355

FAX : +81-72-254-9355

E-mail : notsu@cs.osakafu-u.ac.jp