Vol.2015-GI-34 No.4
2015/7/4

プロ棋士の棋譜データベースを用いない局面評価関数の学習法についての考察

五十嵐治一[†1]　森岡祐一　山本一将[†2]

本論文では，コンピュータ将棋において，プロ棋士の棋譜データベースを用いることなく，コンピュータが自己または他者との対局のみを通じて局面評価関数を学習し，棋力向上を図る方法について考察した．その結果，学習エージェント自身との自己対局や他者との対局を行い，勝敗や主観的評価，探索における最善応手手順や自分の対局譜から，強化学習や教師付学習を用いて局面評価関数を学習し，棋力を向上させる方法を提案する．

Considerations of Learning a Positional Evaluation Function without Using Game Records between Professional Shogi Players

HARUKAZU IGARASHI[†1]　YUICHI MORIOKA
KAZUMASA YAMAMOTO[†2]

This paper considers a learning method of a positional evaluation function in shogi only through self-play and actual games of a learning agent itself without using a large amount of game records between professional shogi players. As a result, it proposes a learning method based on reinforcement learning and supervised learning with the results and records of actual games that the learning agent plays, subjective evaluations given to the games, and the principal variations in search trees.

1. はじめに

近年，コンピュータ将棋の棋力向上がめざましい．プロ棋士と将棋プログラムとの対戦である「将棋電王戦」[1]も，第 1 回(2012 年)から第 4 回(2015 年)までの結果を合わせると 10 勝 5 敗 1 分けとコンピュータ将棋側が大きく勝ち越している．この大きな一因として機械学習による局面評価関数の構築が成功している点が挙げられる．この代表例が，「Bonanza メソッド」[2][3]と称される教師付き学習法である．ここで用いられる訓練データは，プロ棋士間の対局を主とする人間同士の対局棋譜であり，人間側の知識の結晶をコンピュータ将棋側は最大限に活用している．例えば，Bonanza の文献[3] では，48,566 局の棋譜データベース[a]を用いて学習を行った例が報告されている．

しかし，「人間 vs コンピュータ」という対決の観点からは，コンピュータ側が人間側の作り上げた貴重な財産を一方的にうまく利用した結果であるという見方もできる．そこで我々は，プロ棋士の棋譜データベースを全く用いないでコンピュータが自ら正確な局面評価関数を構築し，ついにはプロ棋士に匹敵する棋力に到達できれば，本当にコンピュータが人間に勝利したと言えるのではないかと考えた．また，このような学習法の研究は，人間が将棋を学ぶ上で効率的な上達法を発見する手がかりとなるであろうし，人間のプロが存在しないような AI ゲームにおける局面評価関数の有効な学習法になることも期待できる．

そこで，本論文ではプロ棋士の棋譜データベースを全く用いないで，コンピュータが自己または他者との対局のみを通じて精度の高い局面評価関数を学習し，棋力向上を図る方法について考察する．この目的を実現するために，まず，対局からの学習について基本的な問題点を考察する（2 章）．ここでは，局面評価関数の学習のために，「対局の何から学ぶのか？」，「だれと対局するのか？」，「いかに学習するのか？」という 3 点を論ずる．次に，本研究における学習の基本方針を述べる（3 章）．本研究では学習法として主として強化学習を用いるが，教師付学習も行う．ただし，Bonanza メソッドとは異なり，方策勾配を利用した確率的方策の教師付学習[4]を用いる．強化学習としては価値ベースの学習法と方策ベースの学習法の両方を用いるが，学習の目的関数の最大化という観点から VAPS アルゴリズムに従って両者を統一的に取扱う．さらに，学習則の導出など学習法の詳細を述べ（4 章），最後にプロ棋士の棋力へ到達するまでの学習スケジュールについて考察する（5 章）．

2. 対局からの学習について

2.1 対局の何から学ぶか？

対局において最も確実な客観的情報は勝敗である．勝つか負けるか，あるいは引き分けたかは対局終了時に一意的に定まる．この確定情報を学習に用いるのは当然である．

次に，客観的な情報ではないが，対局内容に対する人間やコンピュータの主観的評価も学習に用いることができる．例えば，戦法や陣形など個人の好み（棋風）や，手筋などの一連の手順を教えたい際などには，主観的評価を積極的

†1 芝浦工業大学工学部情報工学科
Shibaura Institute of Technology
†2 (株) コスモ・ウェブ
Cosmoweb Co., Ltd.

a)このうちに 3 万局以上はプロ棋士同士の棋譜であるが，それ以外に一手 30 秒の早指し将棋やアマチュア高段者の棋譜も含んでいる．

に報酬の形で与えて強化学習を行うことも考えられる．しかし，主観的評価の場合は評価を与える主体によって学習結果が依存する．場合によっては誤った評価を与えてしまい，学習に支障をきたす可能性もある．

さらに，学習するプログラム（以下，学習エージェント）が対局中に作成した探索木の情報も学習に用いることができる．例えば，探索により得られた最善応手手順（PV：Principal Variation）や，その末端局面（Principal leaf）と評価値などの情報である．Principal leaf の評価値を利用した学習法としては，RootStrap 法や TreeStrap 法などの Bootstrapping 的な学習アプローチが考案されており，実際にチェスに適用されている．チェスではこの方法が成功し，局面評価関数の重みを全くランダムな初期値から学習を始めて自己対局だけでマスターレベルの棋力への強化に成功している[5]．さらに，対局も一種の探索木上の PV と見なせば，学習エージェントの対局棋譜を用いた同様の Bootstrapping 的な学習アプローチも可能である．見方によれば TD 学習もこの一種と見なすことができる[5]．

また，自己と比べて対局相手の棋力レベルが非常に高いと分かっている場合，あるいは対局により分かった場合は，学習エージェントは対局相手の手番局面と実際の指し手をそのまま訓練データとして教師付学習に利用することも考えられる．相手以上に強くなるかどうかはわからないが，少なくとも相手レベルに追いつくという点ではこのような「対局相手の指し手に学ぶ」学習も有効であろう．

2.2 だれと対局するか？

プロ棋士の棋譜データベースを用いない，さらには，将棋に関する人間の知識を極力利用しないという立場からは，最も望ましいと思われる対局相手は学習エージェント自身である．すなわち，自己対局を通して局面評価関数を学習して棋力を高めるということである．自分自身との対局の最大の利点は，常に棋力が同程度の相手と対局できる点である．強化学習では勝利して報酬を得る体験が必要であるが，対局相手が強すぎて常に報酬がゼロでは，自分の方策を積極的に強化してゴールに到達するための方向性を定めることは難しいと考えられる．

しかしながら，自己対局では対局相手が自分の読み通りの手を指すことが多いので，出現する局面が限定されやすく，学習に用いられる局面に偏りが生じてしまう可能性がある．したがって，学習で得られた局面評価関数の一般性という点で問題を生ずる可能性がある．また，対局相手（＝自分）を弱くすると勝つ可能性が高くなるので，勝利を報酬とする強化学習では学習が一種の局所解に陥って進まなくなる可能性もある．したがって，自分以外の将棋ソフトや人間との対局も必要であろう．人間や将棋プログラムと自動対局ができる floodgate[6]の利用も考えられる．

次に，自己対局ではない場合の対局相手の棋力であるが，自分よりも極端に弱い相手では棋力向上にはあまり寄与しないことが予想される．逆に，極端に強い相手との対局は強化学習を行う際には効果が少ないが，相手に学ぶという教師付学習においては相手が強い方が学習の効果は大きい．

2.3 いかに学習するか？

本研究はプロ棋士の棋譜データベースを利用しないことを基本方針としている．したがって，プロ棋士の棋譜データベースを利用した教師付学習は行わない．そこで，実際に学習プログラムが対局を行い，その対局結果や内容に対して与えられた報酬信号を手がかりとする強化学習を本研究における学習法の基礎とする．ただし，対局相手の指し手や自分の探索木中の情報を教師データとする教師付学習は許されるものとする．本研究では Bonanza メソッドと呼ばれる教師付学習法ではなく，方策勾配法と同様に方策勾配を用いた確率的方策の教師付学習法を用いる．

以下，本研究における学習の基本方針は 3 章で，学習則の導出など数理的な定式化の詳細については 4 章で述べる．

3. 学習の基本方針

3.1 教師付学習と強化学習

コンピュータ将棋の局面評価関数の学習における代表的な例として，Bonanza による教師付学習がある．一般に，ゲームにおける教師付学習では学習訓練用データとして，局面の状態 s とそのときの正解行動 a のペア(s,a)のデータを大量に用意する．次に入力 s と出力 a の対応関係を関数で近似し，訓練用データにおける出力誤差を最小化するように近似関数中の重みパラメータの値を勾配法などにより求めるのが一般的な教師付学習法である．将棋の場合，このときの訓練用データとしてはプロ棋士の棋譜データベースがよく利用されてきた．

一方，強化学習は学習エージェントに正解行動を与える必要はない．学習システムが行動後の結果の良し悪しを評価し，それを報酬信号の形でフィードバックすることにより学習エージェントが行動決定方法（方策）を修正する．

この強化学習は価値ベースの強化学習法（value-based algorithm）と方策ベースの強化学習法（policy-based algorithm）の２つの方式に分けることができる．前者は，Q 学習のように行動価値関数を通して，あるいは TD 法のように状態価値関数を通して，間接的に方策を学習する[7]．一方，後者は，方策中にパラメータを入れておき，パラメータ空間内で勾配法や GA などの探索法によりパラメータを更新する．方策を if-then ルール集合で表し，ルールの各重みを一定の規則に基づき強化する利益共有法(Profit Sharing)も後者に分類され，部分観測マルコフ決定過程(Partially Observable Markov Decision Processes, POMDPs)における強化学習の問題にも適用されている．

勾配法を用いる方策ベースの強化学習法は方策勾配法(Policy gradient reinforcement learning)と呼ばれている．方策の表現範囲が広く，数学的な定式化や収束性の点でも理論

的に扱いやすい．さらに，価値ベースと方策ベースの両方式を学習の目的関数の最大化という観点から統一的に取り扱った方策勾配法としてBairdとMoore のVAPS (Value and Policy Search)アルゴリズム[8]が提案されている．本研究でもこれに従い，価値ベースと方策ベースの両方の強化学習法を統一的に取り扱うことを基本方針としたい．

3.2 方策勾配法

方策勾配法の代表的な例としては，Williams の REINFORCE アルゴリズム[9]や，POMDP 環境における木村らの確率的傾斜法[10]などがある．また，Q 値を用いて期待報酬の勾配を表現する方式[11][12][13]や，自然勾配を利用する方式[14]も考案されている．方策勾配法の理論と適用例は，例えば，Peters らの文献[15]中にまとめられている．

本研究では，五十嵐らが提案している方策勾配法[16]を用いる．この方式は，Williams のエピソード単位の学習方式（episodic REINFORCE algorithm）[9]をベースとし，環境モデル（状態遷移確率と報酬）と方策に関する単純マルコフ性を必要としない．また，Williams の原著論文[9]ではエピソード長は固定で，最終状態を評価して一度だけ報酬を与えていた．しかし，本方式では可変長のエピソードを取り扱い，報酬もエピソード全体の状態・行動列を評価して計算する非マルコフ的な関数として与えることができる．

さらに，一般的な方策勾配法では単位時間あたりの報酬の極大化を目的とするが，本方式はエピソード長に関係なくエピソードあたりの報酬を極大化することを特徴としている．したがって，将棋のように対局の長さが定まっておらず，一手ごとではなく一局全体の局面・指し手列を評価する場合の学習に向いている．なお，本方式はこれまでに追跡問題や粒子群を用いた最適化手法である PSO (Particle Swarm Optimization)等へ適用されている[17][18].

また，方策として用いられる関数として Williams の原著論文[9]では階層型ニューラルネットワークモデルが用いられていたが，本研究では探索により得られた指し手の評価値をエネルギー関数（目的関数）とする Boltzmann 分布による確率的方策を使用することを念頭に置いている．例えば，指し手の評価値として，探索木の Principal leaf の局面評価値を用いる「PGLeaf 法」[19]や，探索木の全 leaf の局面評価値とその局面への選択確率から計算される期待値を用いる「PG 期待値法」[20]である[b].

3.3 学習の目的関数

3.1 でも述べたように，本研究では価値ベースと方策ベースの両方の強化学習法を統一的に取り扱うことを基本方針とする．そのために 3.5 で述べる VAPS アルゴリズムのように，強化学習の目的をある関数の最大化（または最小化）に帰着させる．すなわち，本論文では局面 s におけるパラメータ ω を含む局面評価関数 $E_s(s;\omega)$ の学習のために次の目的関数 $U(\omega)$ を定義する．

$$U(\omega) = E[U_\sigma(\sigma;\omega)] \quad (1)$$

$$\equiv \sum_\sigma P(\sigma;\omega)U_\sigma(\sigma;\omega) \quad (2)$$

ただし，$U_\sigma(\sigma;\omega)$ はエピソード σ における学習の目的関数の値であり，$E[\cdot]$ はエピソード σ の出現確率 $P(\sigma;\omega)$ による期待値操作を表している．ここで，$U_\sigma(\sigma;\omega)$ を

$$U_\sigma(\sigma;\omega) \equiv \alpha R(\sigma) - \beta\delta(\sigma;\omega) \quad (3)$$

と定義する．(3)の $R(\sigma)$ はエピソード σ に対する報酬の総和であり，「エピソード収益」と呼ぶ．通常の強化学習で用いられる収益や割引収益もこれの一部に含めることができる．$\delta(\sigma;\omega)$ は状態価値関数などが満たすべき条件に関する誤差，例えば，TD 誤差のような 2 乗 Bellman 誤差(squared Bellman residual)[8]，あるいは，価値関数を関数近似したときの近似誤差である[c]． $\delta(\sigma;\omega)$ を本論文では「エピソード誤差」と呼ぶ．α,β は 2 つの項の重みである．VAPS アルゴリズムでは，(3)の第 1 項と第 2 項の両方の項を考える．

VAPS 以前の REINFORCE アルゴリズムなどの方策勾配法では，$U_\sigma(\sigma;\omega)$ として(3)の第 1 項のエピソード収益 $R(\sigma)$ だけを考えていた（$\beta = 0$）．将棋において我々がこれまでに提案してきた PGLeaf 法[19]や PG 期待値法[20]でも(3)の第 2 項のエピソード誤差は取り扱って来なかった．逆に，第 2 項だけを考慮する（$\alpha = 0$）のが TD 学習などの価値ベースの強化学習である．ただし，将棋の場合は状態空間が巨大なので状態価値関数には近似が必要である．この関数近似誤差をエピソード誤差とする．

なお，(2)において，棋譜データベースを用いた学習であれば $P(\sigma;\omega)$ は ω に依らない．しかし，実際の対局を用いた学習であれば学習エージェントの方策に，したがって ω に依存し，ω による微分を考える際には考慮する必要がある．

3.4 環境モデルと方策に関するマルコフ性について

一般に強化学習における環境モデルとは，報酬信号と状態遷移確率である．コンピュータ将棋の場合，報酬信号は勝敗情報などであり，状態遷移確率は対局相手の方策である．通常，勝敗や対局相手の方策についてのマルコフ性を仮定して学習を行うことは妥当であると考えられる．しかし，本論文では勝敗だけではなくエピソード全体を評価してエピソード収益 $R(\sigma)$ を与える場合も考慮したいので，環境モデルの内，報酬信号についてはマルコフ性を必ずしも仮定しないこととする[d].

b) どちらの方策も温度パラメータ T→0 の極限で決定論的な Min-Max 探索に一致する．また，文献[20]では後者を「PG 行動期待値法」と呼んでいる．

c) VAPS の原著論文[8]のように価値関数がテーブル形式の場合は前者のタイプの $\delta(\sigma;\omega)$ が用いられる．一方，状態や行動の空間が大きい場合には後者のタイプが用いられる．

d) 後述するように，エピソード収益をマルコフ性のあるステップ報酬と，マルコフ性のないエピソード報酬とに分離する．

また，学習エージェントの方策に関しては，通常，将棋の場合は現局面だけを元に指し手を決定するので，マルコフ性を仮定する方が自然である．

3.5 VAPS アルゴリズムの学習則

VAPS アルゴリズムでは(1)の目的関数 $U(\omega)$をωで微分し，その勾配方向へωを更新する．更新ベクトルは，Baird と Moore の原著論文[8]における形式とは異なるが，

$$\Delta\omega = \varepsilon\nabla_\omega U(\omega) = \varepsilon\nabla_\omega E[U_\sigma(\sigma;\omega)] \quad (4)$$

$$= \varepsilon E[\nabla_\omega U_\sigma(\sigma;\omega) + U_\sigma(\sigma;\omega)\textstyle\sum_t e_\omega(t)] \quad (5)$$

と表される．ただし，t は離散時刻（手番）を表し，

$$e_\omega(t) \equiv \nabla_\omega \ln \pi(a_t; s_t, \omega) \quad (6)$$

である．原著論文[8]では環境モデルが MDP であることと方策のマルコフ性を仮定していたが，上記(5)の関係は環境モデルが MDP でなくとも，また，方策にマルコフ性がない場合でも成立する[16]．したがって，報酬にマルコフ性が成り立たない場合には，VAPS アルゴリズムのような価値ベースと方策ベースの折衷型の強化学習の方が適していると考えられる．Baird と Moore の原著論文[8]では行動ステップごとの学習則が与えられているが，本論文ではエピソード単位の学習を考える．この場合，(5)の右辺の期待値操作 E[･]を外した次の学習則，

$$\Delta\omega = \varepsilon[\nabla_\omega U_\sigma(\sigma;\omega) + U_\sigma(\sigma;\omega)\textstyle\sum_t e_\omega(t)] \quad (7)$$

により，エピソード終了時にωを更新する． さらに，$U_\sigma(\sigma;\omega)$として(3)を(7)へ代入する．

$$\Delta\omega = \varepsilon[-\beta\nabla_\omega\delta(\sigma;\omega) + (\alpha R(\sigma) - \beta\delta(\sigma;\omega))\textstyle\sum_t e_\omega(t)] \quad (8)$$

(7)の右辺の第 1 項は，エピソードごとの学習の目的関数$U_\sigma(\sigma;\omega)$を直接増大させる勾配方向なので，本論文では「勾配項」と呼ぶ． (8)からわかるように勾配ベクトル$\nabla_\omega U_\sigma(\sigma;\omega)$はエピソード誤差を直接減少させる勾配方向であり，価値関数の整合性や関数近似の誤差に関する拘束条件を満足するための更新方向である．一方，(7)の第 2 項は$U_\sigma(\sigma;\omega)$の値そのものを増大させる方向ではなく，$U_\sigma(\sigma;\omega)$の値が大きいエピソードの生成確率，すなわち，エピソード中に選択した各行動の選択確率を高める方向を表している．以下では「生成項」と呼ぶ．

一般に，価値ベースの学習は，MDP が成り立っている場合には報酬が遅延しても価値関数を通して因果関係（Bellman 方程式）を逆にたどることにより，報酬を時間的に逆伝搬させる．こうした因果関係は拘束条件として利用できるので，求解のために有用な知識の一種と考えることができる．方策勾配法などの方策ベースの学習法はこうした知識を用いないので学習速度の点で劣る可能性がある．

しかし，方策ベースの学習は，MDP を前提としていないので方策や報酬の設計において大きな自由度がある．例えば，行動決定に何らかのモデルを使用したい場合や，状態行動履歴に依存した非マルコフ的な報酬を与える場合などに向いていると考えられる．文献[8]には，(3)のような折衷型の目的関数を用いて，価値ベースの強化学習法である Q 学習と方策ベースの強化学習法である REINFORCE アルゴリズムとの融合方式の方が，単独の学習アルゴリズムよりも高速に最適方策が得られた例が報告されている．

3.6 ステップ報酬とエピソード報酬

本節では(3)のエピソード収益$R(\sigma)$について考える．通常，価値ベースの強化学習法ではマルコフ性のある報酬を考えるが，方策ベースの強化学習法ではその必然性はない．そこで，本論文では次のようなエピソード収益を提案する．

$$R(\sigma) = R_{t=0} + R_\sigma \quad (9)$$

(9)のR_tは，エピソード長を L，割引率を$\gamma(\in [0,1])$，時刻 t における報酬信号をr_tとおくと，時刻 t 以降の割引収益

$$R_t \equiv \textstyle\sum_{k=0}^{L-t-1}\gamma^k r_{t+k+1} \quad (10)$$

である．ただし，報酬信号r_tはマルコフ性を持っていると仮定する．すなわち，

$$E[r_{t+1}] = E[r_{t+1}\,|s_t, a_t, s_{t+1}] \quad (11)$$

と表され，過去の履歴に依存しないとする．勝敗に対する報酬は(11)の特別な例として表すことができる．

一方，(9)の右辺のR_σは，エピソードの状態・行動履歴に依存したマルコフ性を持たない報酬であり，以下では「エピソード報酬」と呼ぶ．将棋では，序盤／中盤での内容に基づいて傾斜配分された報酬，攻め／守りを重視した報酬，一連の指し手手順や戦法・陣形・駒の選好等の棋風などに対する報酬を表すのに用いる．なお，(9)の第 1 項$R_{t=0}$は離散時刻 t（時刻ステップ，将棋では手番）ごとに与えられる報酬信号r_tの割引和なので，本論文では「ステップ報酬」と呼ぶ．オリジナルの VAPS アルゴリズム [8]はマルコフ性のある報酬を仮定していたので，(9)においてステップ報酬だけを考えた場合に相当する．

次章では(3)の右辺のエピソード収益$R(\sigma)$とエピソード誤差$\delta(\sigma;\omega)$として具体的な関数を与え，その場合の学習則を実際に導出する．すなわち，4.1 ではエピソード収益としてステップ報酬だけを与え，エピソード誤差として状態価値関数の関数近似誤差を考えた場合の一般論を述べる．強化学習の代表的手法である TD(λ)法は勾配項だけを考慮した場合に相当する．4.2 では 4.1 の特別な場合として(10)の右辺の報酬信号r_tが勝敗情報である場合を論ずる．これまでに将棋やチェスへ TD(λ)法や TDLeaf(λ)法が適用されてきたが，そこで用いられた学習法はこの場合に属する．4.3 ではエピソード収益としてステップ報酬の他にエピソード報酬R_σを加える．さらに，4.4 では方策勾配を利用した確率的方策の教師付学習法を述べる．教師付学習も VAPS アルゴリズムのような目的関数の最大化という統一的な枠組みの中で取り扱うことが出来ることを示す．

4. 学習法の詳細

4.1 関数近似誤差の最小化

本節では，エピソード収益としてステップ報酬$R_{t=0}$だけを考える．状態価値関数$V^{\pi}(s) \equiv E[R_t|s_t = s]$[e]の近似関数を V_ω (*s*;ω)で表し，(3)の右辺の$\delta(\sigma;\omega)$として，次の関数近似誤差$\delta_{MSE}(\sigma;\omega)$を用いる．

$$\delta(\sigma;\omega) = \delta_{MSE}(\sigma;\omega) \equiv \frac{1}{2}\sum_{s\in\sigma}[V^{\pi}(s) - V_\omega(s;\omega)]^2 \quad (12)$$

$\delta_{MSE}(\sigma;\omega)$ は近似関数 V_ω (*s*;ω)の 2 乗誤差(mean square error)を表している．(12)の勾配ベクトルを計算すると，

$$\nabla_\omega\delta(\sigma;\omega) = \nabla_\omega\delta_{MSE}(\sigma;\omega) \quad (13)$$
$$= -\sum_{s\in\sigma}[V^{\pi}(s) - V_\omega(s;\omega)]\nabla_\omega V_\omega(s;\omega) \quad (14)$$

と表される．

(14)では，手番局面 *s* で状態価値関数$V^{\pi}(s)$の値を与える必要がある．そのための代表的な方法として，①実測による方法と，②λ収益で置き換える方法の２つがある[7]．①の方法は「モンテカルロ法」と呼ばれ，方策 π を用いて局面 *s* から対局を多数回行い，得られた報酬値の平均値により $V^{\pi}(s)$を近似する方法である．この期待値計算には時間がかかり，学習もエピソード単位の学習に限定される．②は「TD(λ)法」と呼ばれ，以下で簡単に説明する．なお，②でλ=1 と置くと①のモンテカルロ法に帰着する．

λ収益R_t^{λ}を次の式で定義する．

$$R_t^{\lambda} \equiv (1-\lambda)\sum_{n=1}^{L-t-1}\lambda^{n-1}R_t^{(n)} + \lambda^{L-t-1}R_t \quad (15)$$

ただし，R_tは時刻 *t* 以降の割引収益，$R_t^{(n)}$は *n* ステップ収益で， *n* ステップ先の状態価値とそれまでの割引報酬の和，

$$R_t^{(n)} \equiv r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1}r_{t+n} + \gamma^n V_\omega(s_{t+n};\omega) \quad (16)$$
$$= \sum_{k=1}^{n}\gamma^{k-1}r_{t+k} + \gamma^n V_\omega(s_{t+n};\omega) \quad (17)$$

と定義されている． $V^{\pi}(s)$を R_t^{λ}で近似すると，(14)は

$$\nabla_\omega\delta(\sigma;\omega) \approx -\sum_{s\in\sigma}\left[R_t^{\lambda} - V_\omega(s;\omega)\right]\nabla_\omega V_\omega(s;\omega) \quad (18)$$

と表される．(18)は「前方観測的な見方」（Forward view または Theoretical view）と呼ばれており[7]，各時刻の更新量の計算には将来の報酬と状態を知ることが必要である．したがって，エピソード単位の学習であれば適用可能である．この場合，学習則は(8)と(18)で与えられる．すなわち，

$$\Delta\omega = \varepsilon[-\beta\nabla_\omega\delta(\sigma;\omega) + (\alpha R_{t=0} - \beta\delta(\sigma;\omega))\textstyle\sum_t e_\omega(t)] \quad (19)$$
$$e_\omega(t) = \nabla_\omega \ln\pi(a_t;s_t,\omega) \quad (20)$$
$$\nabla_\omega\delta(\sigma;\omega) = -\sum_{s\in\sigma}\left[R_t^{\lambda} - V_\omega(s;\omega)\right]\nabla_\omega V_\omega(s;\omega) \quad (21)$$

である．

これに対し，(14)の「後方観測的な見方」（Backward view または Mechanistic view）による学習則が知られており[7]，各時刻での更新量はその時刻での TD 誤差と過去の状態・行動履歴から計算できる．この見方を適用すると次の学習則が得られる[7]．ただし，詳細な導出は付録 A に記した．

$$\Delta\omega = \varepsilon[-\beta\nabla_\omega\delta(\sigma;\omega) + (\alpha R_{t=0} - \beta\delta(\sigma;\omega))\textstyle\sum_t e_\omega(t)] \quad (22)$$
$$e_\omega(t) = \nabla_\omega \ln\pi(a_t;s_t,\omega) \quad (23)$$
$$\nabla_\omega\delta(\sigma;\omega) = -\textstyle\sum_{t=0}^{L-1} e^{\lambda}(t)\delta_t \quad (24)$$
$$e^{\lambda}(t) = \gamma\lambda e^{\lambda}(t-1) + \nabla_\omega V_\omega(s_t;\omega) \quad (25)$$
$$\delta_t \equiv r_{t+1} + \gamma V_\omega(s_{t+1};\omega) - V_\omega(s_t;\omega) \quad (26)$$

一般的な価値ベースの TD(λ)法では, (22)の右辺の[ ]内の第 1 項（勾配項）のみを考え，かつ，エピソード単位ではなく， (24)の時刻 *t* についての和を取らずに各時刻でωを更新するオンライン的な学習がよく用いられる[f].

それに対して VAPS アルゴリズムでは，(22)の右辺第２項（生成項）や勝利報酬以外の一般的なエピソード報酬も用いた学習も可能である．しかし，いずれにしても近似関数 V_ω(*s*;ω)を，局面評価関数を用いていかに設計するかという問題は残されている．

4.2 勝率の最大化

終局時における勝敗を勝敗変数 *z*（勝てば *z*=1，負ければ *z*=0）で表す．時刻 *t* で与える報酬信号r_tを，*t*=*L*（終局時）において$r_t = z$，それ以外の時刻では$r_t = 0$とする．この報酬の与え方を「勝利報酬 R_z」と定義する．勝利報酬 R_z は$\gamma = 1, t = 0$とおいたときの割引収益R_tに一致する．

今，勝率を最大化するために，エピソード収益$R(\sigma)$として勝利報酬 R_z を用いる．すなわち，

$$R(\sigma) = R_z \quad (27)$$

の場合を考える．ここで，局面 *s* において学習プログラムが現在の方策πと環境の下で勝利する確率の予測値を「予測勝利確率」と称し，$P^{\pi}(s) \equiv E[z|s]$で定義する．(27)の下では，予測勝利確率は状態価値関数$V^{\pi}(s) \equiv E[R_t|s_t = s] = E[R_z|s]$と等価である[g]．このとき，(8)の学習則は，

$$\Delta\omega = \varepsilon[-\beta\nabla_\omega\delta(\sigma;\omega) + (\alpha R_z - \beta\delta(\sigma;\omega))\textstyle\sum_t e_\omega(t)] \quad (28)$$

となる．ここで，$\delta(\sigma;\omega)$として予測勝利確率$P^{\pi}(s)$の関数近似誤差だけを考えることにする．すなわち，

$$\delta(\sigma;\omega) = \frac{1}{2}\sum_{s\in\sigma}[P^{\pi}(s) - P_\omega(s;\omega)]^2 \quad (29)$$

と定義すると，

$$\nabla_\omega\delta(\sigma;\omega) = -\sum_{s\in\sigma}[P^{\pi}(s) - P_\omega(s;\omega)]\nabla_\omega P_\omega(s;\omega) \quad (30)$$

となる．さらに，$P^{\pi}(s)$の近似関数 $P_\omega(s;\omega)$として，Beal らのように局面評価関数 E_s(*s*;ω)を含んだシグモイド関数

$$P_\omega(s;\omega) = 1/\left[1 + e^{-E_s(s;\omega)/\tau}\right] \quad (31)$$

を用いる[21]と，

e) $E[R_t + R_\sigma|s_t = s]$でないことに注意する．

f) Beal と Smith は将棋に TD(λ)法を適用し，Baxter らはチェスに TDLeaf (λ)法を適用したが，いずれも(24)のような一局ごとのエピソード単位の学習であった．一方，薄井らが将棋に TD(λ)法を適用した際には一手ごとの学習であった．そこでは，(24)を *t* についての和は取らずに用いた．

g) (27)ならば， $^{\forall}t(0 \le t \le L), R_t = R_z$ である．

$$\nabla_\omega P_\omega(s;\omega) = (1/\tau)[1 - P_\omega(s;\omega)]P_\omega(s;\omega)\nabla_\omega E_s(s;\omega) \quad (32)$$

と表される．(28)～(32)はエピソード単位の学習であり，かつ，予測勝利確率$P^\pi(s)$の真の値を実測などの手段で与える必要がある．しかし，4.1 で述べた TD(λ)法を用いると，$P^\pi(s)$をλ収益で近似することや，一手ごとの学習を行うことも可能になる．これまでに，(28)の第 1 項（勾配項）のみを考慮して，Beal ら[21]や薄井ら[22]は将棋に TD(λ)法を適用し，Baxter らはチェスに TDLeaf (λ)法を適用した[23][h].

それに対して VAPS アルゴリズムでは(28)の第 2 項（生成項）も考慮する．この場合，(28)～(32)の学習法は，対局終了時に勝敗結果 z を 0 または 1 の二値でエピソード収益 $R(\sigma)$（ここでは勝利報酬 R_z）を通して与えておけば，(31)のシグモイド関数$P_\omega(s)$がその局面での勝利予測確率を正確に与える方向へ学習が進み，かつ，生成項も考慮すれば勝率を増加させる方向へも進むことが期待される[i]．すなわち，予測勝利確率の精度と棋力との両方を同時に向上させる学習と言える．ただし，エピソード報酬が勝利報酬に限定され，予測勝利確率$P^\pi(s)$の近似関数$P_\omega(s;\omega)$が局面評価関数$E_\omega(s;\omega)$を用いて適切に設計されている必要がある．

4.3 部分報酬の最大化

強化学習では，ゴールに到達するために一種のサブゴールを設定し，それに到達すれば「部分報酬」(partial reward) を与えて，学習の速度を高める場合がある．そこで，将棋に関する経験的知識を用いて部分報酬を与える方法が考えられる．例えば，序盤における駒組みを学習させたいときには，序盤の早い段階で大きな報酬を与えたり，逆に大きな負の報酬を与えてエピソードを打ち切るなどの方法が考えられる．この他，戦法や陣形などの棋風を学習させる場合にも部分報酬を与えることが考えられる．

そこで，ステップ報酬を考慮しないで，エピソードの内容（状態・行動列）を評価して与えるマルコフ性のない部分報酬を(9)のエピソード報酬 R_σ として与える．すなわち，

$$R(\sigma) = R_\sigma \quad (33)$$

の場合を考える．さらに，マルコフ性を仮定しているエピソード誤差の項を除くことにする($\beta = 0$)と，(8)の学習則は，

$$\Delta\omega = \varepsilon[\alpha R_\sigma \textstyle\sum_t e_\omega(t)] \quad (34)$$

と表され，通常の方策勾配法の学習則に帰着する．

以上，4.2 ではエピソード収益がステップ報酬$R_{t=0}$（の例である勝利報酬）で与えられる場合の学習則を，4.3 ではエピソード収益がエピソード報酬 R_σ（の例である部分報酬）で与えられる場合の学習則を示した．(9)のようにエピソード収益が両者の和で与えられた場合は，(28)と(34)の更新ベクトル$\Delta\omega$を合成すれば，両者の目的を同時に達成するような方向へ局面評価関数のパラメータωを更新していくことが可能である．しかし，どのように合成してどこへ誘導するかは学習システム設計者が決定しなければならない．

4.4 方策勾配を用いた確率的方策の教師付学習

1.で述べたように本論文の目的はプロ棋士の棋譜データベースを用いないで局面評価関数を学習することである．したがって，4.1～4.3 では教師付学習ではなく，強化学習をベースとした学習法を論じてきた．しかし，本論文では 2.1 で述べたように，学習エージェント自身が対局中に作成した探索木や対局相手の指し手から学習することも考えたい．そこで，本節では VAPS の枠組みの中に教師付学習を取り込む方法を提案する．

(3)の$\delta(\sigma;\omega)$として誤差関数$\delta_{KLD}(\sigma;\pi^*,\pi)$を考える[4].

$$\delta(\sigma;\omega) = \delta_{KLD}(\sigma;\pi^*,\pi) \equiv \textstyle\sum_{s\in\sigma}\sum_{a\in A(s)} \pi^*(a|s)\ln[\pi^*(a|s)/\pi(a|s;\omega)] \quad (35)$$

ただし，教師となる着手決定方策を π*，学習システムの着手決定方策を π とする．両者ともに確率分布関数として与えられているとする．(35)の$\delta_{KLD}(\sigma;\pi^*,\pi)$(≥0)は，正解の方策 π*と学習システムの方策 π との距離を表すカルバック・ライブラー情報量（Kullback–Leibler divergence）である．

(35)の$\delta_{KLD}(\sigma;\pi^*,\pi)$の勾配，

$$\nabla_\omega\delta_{KLD}(\sigma;\pi^*,\pi) = -\textstyle\sum_{s\in\sigma}\sum_{a\in A(s)} \pi^*(a|s)\nabla_\omega \ln\pi(a|s;\omega) \quad (36)$$

を(8)の学習則へ代入すると，

$$\Delta\omega = \varepsilon[-\beta\nabla_\omega\delta_{KLD}(\sigma;\pi^*,\pi) + \big(\alpha R(\sigma) - \beta\delta_{KLD}(\sigma;\pi^*,\pi)\big)\textstyle\sum_t e_\omega(t)] \quad (37)$$

$$= \varepsilon[\beta\textstyle\sum_{s\in\sigma}\sum_{a\in A(s)} \pi^*(a|s)\nabla_\omega \ln\pi(a|s;\omega) + (\alpha R(\sigma) - \beta\sum_{s\in\sigma}\sum_{a\in A(s)} \pi^*(a|s)\ln[\pi^*(a|s)/\pi(a|s;\omega)]\big)\textstyle\sum_t e_\omega(t)] \quad (38)$$

と表される．$e_\omega(t)$ は時刻 t での方策の対数微分値(6)である．この学習は教師付学習であるが，学習則は VAPS と同様に学習の目的関数の最大化により方策勾配を用いて導出された．(37)や(38)において，もし，報酬R(σ)を与えないで教師付学習だけを行うのであれば$\alpha = 0$とおけばよい．また，学習させたい局面 s を学習対象となる方策$\pi(a|s;\omega)$で生成しないのであれば，局面を出現させる方策は学習しないで固定するので$e_\omega(t) = 0$とおけばよい．すなわち，(37)や(38)の[・]内の第 2 項（生成項）を無視すればよい[4].

方策勾配法を将棋に適用した PGLeaf 法[19]では，次のように方策$\pi(a|s;\omega)$を，局面 s で手 a を指した後の最善応手手順の leaf 局面 s^*の評価値$E_s(s^*|a,s;\omega)$を用いた Boltzmann 分布で表す．

$$\pi(a|s;\omega) = exp(E_s(s^*|a,s;\omega)/T)/Z \quad (39)$$

$$Z \equiv \textstyle\sum_{x\in A(s)} exp(E_s(s^*|x,s;\omega)/T) \quad (40)$$

T は温度と呼ばれるパラメータである．この場合，(38)の生成項を無視すると，学習則は，

h) TDLeaf (λ)法は(25),(26)において出現局面 s_t ではなく，s_t をルートノードとする最善応手手順の leaf 局面 s_t^*を用いる点が TD(λ)法と異なる．
i) 生成項がなく，勾配項だけでも方策改善がうまくいけばエピソード収益は増加する．したがって，(27)であれば勝率が向上する．

$$\Delta\omega = \varepsilon\beta \sum_{s\in\sigma}\sum_{a\in A(s)} \pi^*(a|s)\nabla_\omega \ln\pi(a|s;\omega) \quad (41)$$

$$= \frac{\varepsilon\beta}{T}\sum_{s\in\sigma}\sum_{a\in A(s)} \pi^*(a|s)\cdot$$

$$\left[\nabla_\omega E_s(s^*|a,s;\omega) - \sum_{x\in A(s)} \pi(x|s;\omega)\nabla_\omega E_s(s^*|x,s;\omega)\right] \quad (42)$$

となる．これは次のように表される．

$$\Delta\omega = \frac{\varepsilon\beta}{T}\sum_{s\in\sigma}\sum_{a\in A(s)} \pi^*(a|s)\cdot$$

$$\left[(1-\pi(a|s;\omega))\nabla_\omega E_s(s^*|a,s;\omega) - \sum_{x\neq a} \pi(x|s;\omega)\nabla_\omega E_s(s^*|x,s;\omega)\right] \quad (43)$$

ここで，[ ]内の 2 つの項の符号を考える．第 1 項では，$1-\pi(a|s;\omega) > 0$なのでエピソード中に学習エージェントが実際に指した手 a の価値$E_s(s^*|a,s;\omega)$を高める方向にωは更新される．第 2 項では，$-\pi(x|s;\omega) < 0 (x \neq a)$なので a の兄弟手 x の価値を低下させる方向にωは更新されることが分かる．特に，棋譜からの学習のように正解手が 1 つに限定される場合には，正解手の価値を高め，兄弟手の価値を低下させる方向にωが更新されることを表している．

また，正解手が 1 つでなく，確率分布で与えられる場合には，(39)を次のように変形すると意味がわかりやすい．

$$\Delta\omega = \frac{\varepsilon\beta}{T}\sum_{s\in\sigma}\sum_{a\in A(s)}[\pi^*(a|s) - \pi(a|s;\omega)]\,\nabla_\omega E_s(s^*|a,s;\omega) \quad (44)$$

(44)の右辺の[ ]内の符号に注意すると，局面 s において正解分布の選択確率値よりも学習中の方策の選択確率値が小さい場合にはその指し手 a の価値を高め，逆の場合は過大評価なので価値を低下させようとしているのが分かる．

さらに，探索木に対する Bootstapping 的な学習アプローチとして，対局中の探索木の PV 上に出現する各局面において，最善応手を正解手とする教師付学習を行うことも可能である．また，対局手順も一種の探索木中の PV と見なせば，学習エージェントの対局棋譜を用いた同様の Bootstapping 的な教師付学習も可能である．この場合，(39)の$\pi(a|s;\omega)$には実現局面 s から指し手 a を指した後の浅い探索局面 s'とその評価値$E_s(s'|a,s;\omega)$を用い，正解手としては実現手を用いる．

以上の議論の中で，(37)や(38)においては生成項を考慮しなかった．もし，学習エージェントが方策$\pi(a|s;\omega)$を用いた対局後に実現局面に対する教師分布$\pi^*(a|s)$が与えられたとする[j]．その場合には生成項も考慮した方法や，エピソード報酬$R(\sigma)$を与えて強化する学習方法も考えられる．すなわち，対局局面での教師付学習とエピソード報酬$R(\sigma)$による方策ベースの強化学習の同時学習が可能である．さらに，4.1 の$\delta_{MSE}(\sigma;\omega)$を考慮すると価値ベースの強化学習も組み合わせることも可能である．

j) 人間同士の対局では対局後に「感想戦」と称して，出現局面における最善手を検討することがある．

5. 将棋における学習スケジュール

本章では，将棋の局面評価関数の学習スケジュールについて考察する．局面評価関数中のパラメータに関しては，駒割（駒の重み）が最も重要と考えられる．したがって，最初に駒割だけの評価関数を考えて，駒割だけを学習する．その後に駒割以外のパラメータを学習する．

次に，①自己対局による学習，②他者との対局による学習，③ユーザが指定する棋風などの学習の順で学習を行うのが妥当と考えられる．①では，PV 上での RootStrap 法[5]や 4.4 の方策勾配を用いた確率的方策の教師付学習，4.3 の PGLeaf 法，4.2 の TDLeaf(λ)法を用いる．可能ならば生成項も考慮する．

しかし，自己対局だけでは学習自体が局所解に陥る可能性もあるので②も行う．4.2 の強化学習の他，レーティングや過去の対局内容から見て実力がかなり上の相手については，4.4 の教師付学習を適用する．その結果，いろいろな基本的な戦法や陣形を相手から学ぶことができる．

①，②の学習で一通りの棋力が付いたところで，③のユーザの好みの棋風を学習するために，4.3 の部分報酬を最大化する学習法を行う．将棋に関する知識を初期の段階で与えた方が学習の効率が良いという考え方もあるが，おかしな癖が付いてしまう可能性もある．また，感想戦の後で，指し手をピンポイントで教師付学習することも考えられる．いずれにせよ，知識や教師となる指し手を与える際には，その的確さやコストも考慮する必要がある．

6. おわりに

本論文では，コンピュータ将棋において，プロ棋士の棋譜データベースを全く用いないで，コンピュータが自己または他者との対局のみを通じて局面評価関数を学習し，棋力向上を図る方法について考察した．その結果，学習エージェント自身との自己対局や他者との対局を行い，勝敗や主観的評価，探索における最善応手手順や自分の対局譜から，強化学習や教師付学習を用いて局面評価関数を学習し，棋力を向上させるなどの方法を提案した．特に，この中で，①マルコフ性を仮定することなくエピソード単位の VAPS アルゴリズムを適用すること，②VAPS アルゴリズムにおける目的関数をエピソード収益とエピソード誤差の線形和とすることを提案し，③エピソード収益を勝敗情報のようにマルコフ性のある報酬（ステップ報酬）と，戦法や陣形，駒の選好などの棋風に基づく主観的評価を表現したマルコフ性のない部分報酬（エピソード報酬）とに分離して与えた場合の学習則を導出し，④これらの枠組みで TD(λ)法や方策勾配法の学習則が導出でき，これまでに将棋に適用されてきた強化学習の殆どの学習則が導出できることを示し，さらに，⑤同じ枠組みで教師付学習法の学習則を導出し，この際に VAPS アルゴリズムの生成項も考慮することを提案した．最後に，これらの強化学習，教師付学習を用いて

プロ棋士の棋譜データベースを使用することなく棋力がプロ棋士程度にまで向上するための一つの方針を示した．

本論文で述べた個々の学習法をどういう順番でどの程度行うかは，学習システムの設計者に依存する．これは読みの力を固定したときの将棋の上達法を設計する問題と深く関連する．今後，様々な学習方式モデルを立てて，その有効性を比較検討して行きたいと考えている．

謝辞 本研究は JSPS 科研費 26330419 の助成を受けました．ここに感謝の意を表します．

参考文献

1) 将棋電王戦のホームページ http://ex.nicovideo.jp/denou/
2) 保木邦仁：局面評価の学習を目指した探索結果の最適制御，第 11 回ゲーム・プログラミングワークショップ，pp.78-83 (2006).
3) Hoki, K. and Kaneko, T.: Large-Scale Optimization for Evaluation Functions with Minimax Search, Vol. 49, pp. 527-568 (2014).
4) 五十嵐治一，森岡祐一，山本一将：方策勾配法による局面評価関数とシミュレーション方策の学習，情報処理学会研究報告，Vol. 2013-GI-30, No. 6, pp.1-8 (2013).
5) Veness, J., Silver, D., Uther, W. and Blair, A.: Bootstrapping from Game Tree Search, Advances in Neural Information Processing Systems 22 (NIPS 2009).
6) floodgate のホームページ
http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html
7) Sutton, R. S. and Barto A. G. : Reinforcement Learning, The MIT Press, Massachusetts (1998).
8) Baird, L. and Moore, A.: Gradient Descent for General Reinforcement Learning, Advances in Neural Information Processing Systems 11 (NIPS'98), pp.968-974 (1999).
9) Williams, R. J. : Simple Statistical Gradient- Following Algorithms for Connectionist Reinforcement Learning, Machine Learning, Vol.8, pp.229-256 (1992).
10) 木村元，山村雅幸，小林重信：部分観測マルコフ決定過程下での強化学習-確率的傾斜法による接近，人工知能学会誌，Vol.11, No.5，pp761-768 (1996).
11) Sutton, R.S., McAllester, D., Singh, S. and Mansour, Y. : Policy Gradient Methods for Reinforcement Learning with Function Approximation, Advances in Neural Information Processing Systems 12 (NIPS'99), pp.1057-1063 (2000).
12) Konda, V. R. and Tsitsiklis, J. N.: Actor-Critic Algorithms, Advances in Neural Information Processing Systems 12 (NIPS '99), pp.1008-1014 (2000).
13) 阿部健一：強化学習―価値関数推定と政策探索"，計測と制御，第 41 巻，第 9 号，pp.680-685 (2002).
14) Kakade, S.: A natural policy gradient, Advances in Neural Information Processing Systems 14 (NIPS'01), pp.1531- 1538 (2002).
15) Peters, J., and Schaal, S.: Policy Gradient Methods for Robotics, Proc.of the IEEE International Conference on Intelligent Robotics Systems (IROS 2006), pp.2219-2225 (2006).
16) 五十嵐治一，石原聖司，木村昌臣：非マルコフ決定過程における強化学習―特徴的適正度の統計的性質―，電子情報通信学会論文誌 D, Vol.J90-D, No.9, pp.2271-2280 (2007).
17) 石原聖司，五十嵐治一：マルチエージェント系における行動学習への方策こう配法の適用-追跡問題-，電子情報通信学会論文誌 D-I, Vol.J87-D1, No.3, pp.390-397 (2004).
18) 五十嵐 治一，半田 雅人，石原 聖司，篠埜 功：マルチエージェントシステムにおける行動制御―PSO における重み係数の強化学習―，電子情報通信学会論文誌 D, Vol. J94-D, No. 10, pp. 1612-1621 (2011).
19) 森岡祐一，五十嵐治一：方策勾配法とα β探索を組み合わせた強化学習アルゴリズムの提案，第 17 回ゲーム・プログラミングワークショップ, pp.122-125 (2012).
20) 五十嵐治一，森岡祐一，山本一将：方策勾配法による静的局面評価関数の強化学習についての一考察"，第 17 回ゲーム・プログラミングワークショップ, pp.118-121(2012).
21) Beal, D. F. and Smith, M. C.: Temporal difference learning applied to game playing and the results of application to shogi, Theoretical Computer Science, Vol. 252, pp.105-119 (2001).
22) 薄井克俊，鈴木豪，小谷善行：TD 法を用いた評価関数の学習，第 4 回ゲーム・プログラミングワークショップ, pp.31-38 (1999).
23) Baxter, J., Tridgell, A., and Weaver, L., : KnightCap: A chess program that learns by combining TD(λ) with game-tree search, Proceedings of the Fifteenth International Conference (ICML '98), pp.28-36 (1998).

付録 A

(22)～(26)の導出を以下に記す．$V^{\pi}(s)$を R_t^{λ} で近似すると，

$$V^{\pi}(s_t) - V_{\omega}(s_t;\omega) \approx R_t^{\lambda} - V_{\omega}(s_t;\omega) \quad (A.1)$$
$$= \sum_{k=t}^{L-1}(\gamma\lambda)^{k-t}\delta_k \quad (A.2)$$

と表される[k]．δ_k は時刻 k における TD 誤差で，

$$\delta_k \equiv r_{k+1} + \gamma V_{\omega}(s_{k+1};\omega) - V_{\omega}(s_k;\omega) \quad (A.3)$$

で定義されている．これを用いると，(14)の右辺は，後方観測的な見方として，以下のように過去の情報だけで書き表すことができる．

$$\sum_{s\in\sigma}[V^{\pi}(s) - V_{\omega}(s;\omega)]\nabla_{\omega}V_{\omega}(s;\omega)$$
$$= \sum_{t=0}^{L}[V^{\pi}(s_t) - V_{\omega}(s_t;\omega)]\nabla_{\omega}V_{\omega}(s_t;\omega) \quad (A.4)$$
$$\approx \sum_{t=0}^{L}\left(\sum_{k=t}^{L-1}(\gamma\lambda)^{k-t}\delta_k\right)\nabla_{\omega}V_{\omega}(s_t;\omega) \quad (A.5)$$
$$= \sum_{k=0}^{L-1}\left(\sum_{t=0}^{k}(\gamma\lambda)^{k-t}\nabla_{\omega}V_{\omega}(s_t;\omega)\right)\delta_k \quad (A.6)$$
$$= \sum_{k=0}^{L-1} e^{\lambda}(k)\delta_k \quad (A.7)$$
$$= \sum_{t=0}^{L-1} e^{\lambda}(t)\delta_t \quad (A.8)$$

ただし，$e^{\lambda}(t)$は適格度トレース(eligibility trace)と呼ばれ，

$$e^{\lambda}(t) \equiv \sum_{k=0}^{t}(\gamma\lambda)^{t-k}\nabla_{\omega}V_{\omega}(s_k;\omega) \quad (A.9)$$

で定義されるが，次のように逐次的に計算できる．

$$e^{\lambda}(t) \equiv \sum_{k=0}^{t-1}(\gamma\lambda)^{k-t}\nabla_{\omega}V_{\omega}(s_k;\omega) + \nabla_{\omega}V_{\omega}(s_t;\omega) \quad (A.10)$$
$$= \gamma\lambda e^{\lambda}(t-1) + \nabla_{\omega}V_{\omega}(s_t;\omega) \quad (A.11)$$

したがって, (14)は, (A.8)を用いると次のように表される．

$$\nabla_{\omega}\delta(\sigma;\omega) = -\sum_{s\in\sigma}[V^{\pi}(s) - V_{\omega}(s;\omega)]\nabla_{\omega}V_{\omega}(s;\omega)$$
$$\approx -\sum_{t=0}^{L-1} e^{\lambda}(t)\delta_t \quad (A.12)$$

なお, (A.5)は Baxter ら[23]がチェスで用いた学習則である．(A.12)は将棋で Beal ら[21]と薄井ら[22]が用いた学習則である．ただし，薄井らの学習は一手ごとの学習なので，(A.12)の時刻 t についての和は取らないで用いていた．

k) 文献[7]の第 7 章 7.4 節参照．