Toward Approximate Adaptive Learning

James F. Peters

Department of Electrical and Computer Engineering,
University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
jfpeters@ee.umanitoba.ca

Abstract. The problem considered in this paper is how the classification of observed behaviour of organisms can be used to influence adaptive learning, beneficially. The solution to this problem hearkens back to the pioneering work during the 1980s by Zdzisław Pawlak and others on classification of objects and approximation spaces, where elementary sets of equivalent objects a framework for perceptions concerning observed behaviours. The seminal work by Oliver Selfridge and Chris J.C.H. Watkins on delay rewards and adaptive learning, also during the 1980s, combined with more recent work on reinforcement learning provide a basis for the forms of adaptive learning introduced in this article. In addition, recent work on approximation spaces has led to what is known as approximate adaptive learning. This article presents two forms of run-and-twiddle (RT) adaptive learning, each using the Watkins' stopping time strategy to mark the end of an episode. Twiddling amounts to adjusting what one does to achieve a better result. This becomes more apparent in approximate RT adaptive learning introduced in this article, where a record of observed behaviour patterns during each episode recorded in an ethogram makes it possible to define a pattern-based learning rate in the context of approximation spaces. Both forms of adaptive learning are actor-critic methods. The contribution of this article is the introduction of two forms of adaptive learning with Watkins' stopping time strategy with differential discount on returns in both cases and differential learning rate for adaptive learning in the context of approximation spaces.

Keywords: Actor-critic, adaptive learning, approximation space, behaviour pattern, perception, stopping time.

> *An approximation space ... serves as a formal counterpart of perception ability or observation.*
> – Ewa Orłowska, March, 1982.

1 Introduction

The problem considered in this paper is how the classification of observed behaviour of organisms can be used to influence adaptive learning, beneficially. The term *organism*, in general, is understood in Whitehead's sense as something that

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 57–68, 2007.

emerges from (belongs to) the world [43]. The solution to this problem hearkens back to the pioneering work by Zdzisław Pawlak and others on classification of objects and approximation spaces (see, *e.g.*, [4,13,9,15,16,22,33,34,39]), work on delayed rewards and adaptive learning by Oliver Selfridge and C.J.C.H. Watkins also during the 1980s (see, *e.g.*, [31,40]), extensive work on reinforcement learning (see, *e.g.*, [2, 5, 38, 29, 42]), and recent work on reinforcement learning and intelligent systems in the context of approximation spaces (see,*e.g.*, [24, 25, 23, 17, 18, 20, 21, 33, 34]). This article presents two forms of run-and-twiddle (RT) adaptive learning, each using the Watkins' stopping time strategy to mark the end of an episode. Twiddling amounts to adjusting what one does to achieve a better result. This becomes more apparent in approximate RT adaptive learning introduced in this article, where a record of observed behaviour patterns tabulated in an ethogram [24, 26] during each episode makes it possible to consider a pattern-based learning rate defined in the context of an approximation space. Both forms of adaptive learning introduced in this article are variant actor-critic methods, where action discounting as well as learning rate are defined relative to temporal differences. The contribution of this article is the introduction of two forms of adaptive learning that construct a semi-martingale with Watkins' stopping time strategy with differential discount on returns in both cases and differential learning rate for adaptive learning in the context of approximation spaces.

This article is organized as follows. An approach to RT adaptive learning is presented in Sect. 2. A refinement of the generalized approximation space model is given in Sect. 3. Approximate RT adaptive learning is introduced in Sect. 4.

2 Adaptive Learning

Watson [40] suggests using the value of a state $V(s)$ as the basis for an adaptive control strategy used by an organism to determine what to do next. This strategy can be summarized intuitively as follows.

1. **Estimate.** *If things are expected to improve or stay the same, then carry on with the same action.*
2. **Twiddle.** *If things are expected to get worse, then search for a more promising action.*
3. **End of Episode.** *If things are expected to get worse, regardless which possible action we choose, then that marks the end of an episode.* This is analogous to a situation faced by a gambler who either withdraws from the game because the expected return is not favorable or bets based on luck and stands a chance of losing [3]. The form of adaptive learning in this paper implicitly constructs a semi-martingale [3], where an episode continues as long as $V(s) \leq V(s')$, *i.e.*, $E[R_a] \leq E[R_{a'}]$ based on Monte Carlo estimates [7, 30] of $V(s), V(s')$ for actions a, a' in states s, s', respectively[1].

[1] $V(s)$ (value of the current state s) is defined in terms of $E[R_a]$, the expected value of return R_a for an action a. $V(s')$ denotes the value of next state s' following s.

This control strategy was originally suggested by Oliver Selfridge in 1984 [31] and elaborated in the context of the value of a state and Monte Carlo methods by Chris J.C.H. Watkins in 1989 [40]. Selfridge called this a *run-and-twiddle* (RT) strategy, which he based on observations of the behavior of E. coli bacteria, male silk moths, and ants.

The notion of a stochastic process and what known as semi-martingales are important in RT adaptive learning introduced in this article.

Definition 1. Stochastic Process
A stochastic process is any family of random variables $\{X_t, t \in T\}$ [3]. In practice, X_t is an observation at time t. A random variable (r.v.) X_t is a real-valued function $X : \Omega \rightarrow \Re$ defined on $(\Omega, \mathcal{F})$, where $\Omega, \mathcal{F}$ is sample space and family of events, respectively [8, 44].

It can be shown that during each episode of RT adaptive learning, what is known as a semi-martingale is constructed. Semi-martingales were introduced by Doob during the early 1950s [3] and elaborated by many others (see, *e.g.*, [8, 44]).

Definition 2. Semi-Martingale
A semi-martingale is a stochastic process $\{X_t, t \in T\}$ such that

$$E[X_t] \leq E[X_{t+1}],$$

where $E[|X_t|] < \infty$.

The form of semi-martingale we have in mind is $\{R_t, t \in T\},\ E[R_t] \leq E[R_{t+1}]$, where R_t is the return on a sequence of actions at time t during an episode.

2.1 Toward RT Adaptive Learning

The basic framework for an approach to a run-and-twiddle (RT) form adaptive learning is shown in Fig. 1, where the conventional framework for actor-critic learning has been changed. Instead of the usual temporal difference (TD) δ term [38, 41], a TD γ is source of input to a critic in evaluating observed action-rewards[2]. The policy structure enforced during adaptive learning is an actor, since the selection of an action a in each state s is determined by a policy $\pi(s, a)$. The estimated value function $V(s)$ serves a critic during adaptive learning. Twiddling begins at the end of each episode[3], where the actions within an episode are discontinued as a result of some halting condition being satisfied.

An elaborate form of twiddling is possible by recording observed behaviours during an episode and constructing what is known as a rough ethogram. A *rough ethogram* is a decision table that records acceptable as well as unacceptable behaviour patterns of organisms [26]. It will become apparent that an ethogram represents a decision system, where each possible behaviour leading from the current state to a new state is evaluated relative to an action-selection policy.

[2] TD γ denotes the rate of change of action rewards.

[3] *i.e.*, an episode is constituted by a sequence of actions that ends in a terminal state.

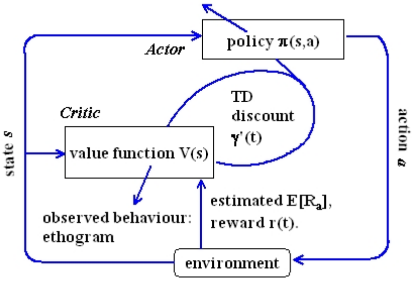

Fig. 1. Basic Framework for Adaptive Learning

That is, among all of the possible actions in a state, an action a that has been selected represents a perceptual judgement *accept* a based on a perception that the performance of a conforms to a standard more than the other possible actions, which is explained in the sequel. By the same token, an ethogram provides a record of each action b deemed unacceptable and a corresponding perceptual judgement *reject* b.

1. **Reward signal:** Define action a in terms of a reward $r(t)$ as a function representing a signal observed at time t_i, which results from interaction with the environment as a result of performing some action a at time t_{i-1}[4]. Then associate with each action $a(t)$ a discounted reward $r(t)$ at time t, namely, $\gamma'(t)r(t)$, where $\gamma(t)$ is a discount function and $\gamma'(t)$ denotes the differential of $r(t)$. It is important to define a reward function $r(t)$ that reflects the form of the signal produced by each action.
2. **Discount γ:** Either choose fixed $\gamma(t)$, *e.g.*, $\gamma(t) = 1$, or put $\gamma(t) = r(t)$ and obtain the differential

$$\gamma'(t) = \frac{d(r(t))}{dt}\bigg|_{t \leftarrow t_i} \approx \frac{|r(t_i) - r(t_{i-1})|}{|t_i - t_{i-1}|}.$$

   In other words, let the value of γ vary over time instead of using a fixed value of γ that diminishes (*i.e.*, monotonically decreases) over time[5]. The critic in Fig. 1 is influenced by a Temporal Difference (TD) discount γ, which replaces the usual TD δ term (see, *e.g.*, [38,42]). The discount factor reflects the rate of change of the signal $r(t)$ coming from the environment at time t.
3. **Return:** Let $E[R_t], r_a, t_i$ denote expected return at time t, reward for action a, elapsed time at step i during an episode, respectively. Define $V(s_t) = E[R_t] \approx \frac{1}{n}\sum_{i=1}^{n} \gamma'(t_i) \cdot r_a(t_i)$, where value of state $V(s)$ is estimated over n time steps for each action a in state s. The assumption made here is that a reward r_t is a r.v. and, as a consequence, return R_t is a r.v. and $\{R_t, t \in T\}$

[4] t can be viewed as the elapsed time since the start of an episode.

[5] The form of discount factor introduced in this paper differs from what was originally suggested by Watkins [40] in estimating return R_t at time t, where $R_t = r_1 + \gamma r_2 + \cdots + \gamma^{t-1} r_t, \gamma \in [0, 1]$.

is a stochastic process, where R_t is the return computed at time t for each state during an episode and T is a set of episode times. It is also the case that the $Pr(R_t = \omega)$ is unknown for $\omega \in \Omega$. For this reason, $E[R_t]$ is estimated using a Monte Carlo method [7, 30] (for a detailed explanation, see [25]).

Theorem 1. Adaptive Learning Semi-martingale.
The RT form of adaptive learning constructs a semi-martingale.

Proof. An episode continues as long as $V(s) \leq V(s')$. Let t_i denote elapsed time t at the start of i^{th} state during an episode and let s' denote the state immediately following state s. Each time an episode continues after finding that the condition $V(s') > V(s)$ is satisfied at time t_n, another term is added to a sequence of estimates of $V(s)$ at time t_n, namely, $V(s') \approx E[X_{t_{n+1}}]$, namely,

$$E[X_{t_1}] \leq E[X_{t_2}], \dots, \leq E[X_{t_n}] \leq E[X_{t_{n+1}}]. \qquad \square$$

An important problem to consider in constructing semi-martingales is a stopping time, *i.e.*, a time $\mathcal{T}$ when a semi-martingale ends. The notion of a stopping time can be explained in general.

Definition 1. Stopping Time. *A stopping time results from a strategy for determining when to stop a sequence based only on the outcomes seen so far [8].*

Axiom 1. Discount Rate. *During each episode, $\gamma'(t) < \varepsilon$ for any given threshold $\varepsilon > 0$ and for sufficiently large t. This means that $|r(t_{i+1}) - r(t_i)| < \varepsilon|t_{i+1} - t_i|$ for sufficiently large i, e.g., $i > n_{large}$.*

Theorem 2. Adaptive Learning Semi-martingale with Stopping Time.
In RT adaptive learning, (1) a semi-martingale constructed during each episode has a stopping time, and (2) $E[R_{t_n}] > E[R_{t_{n+1}}]$ occurs at some time t_n, (3) each adaptive learning episode has finite duration and each semi-martingale has a finite number of terms.

Proof. During adaptive learning, construction of a semi-martingale ends whenever Watkins' condition $V(s') > V(s)$ is not satisfied. Hence, (1) holds, *i.e.*, from Def. 1, Watkins' condition provides a stopping time strategy. (2) From Ax. 1, $\gamma'(t) \rightarrow 0$ during each episode. Hence, $E[R_{t_n}] > E[R_{t_{n+1}}]$ occurs at some time t_n, since the estimated value of $E[R_{t_{n+1}}]$ gets smaller than $E[R_{t_n}]$ for $\lim\limits_{i \rightarrow n_{large}} \gamma'(t_i) < \varepsilon$ for sufficiently large i. (3) *sunset* → 0 during each episode in Alg 1 and Alg. 2, which guarantees that each episode has finite duration. Hence, each semi-martingale constructed during an adaptive learning episode has finite length. □

2.2 Adaptive Control Algorithm

The run-and-twiddle control strategy is given a more formal representation by Watkins [40], p. 67. Let $a(x_t), s, s'$ denote action of object x at time t in state s, current state and next state, respectively. A representation of the adaptive

Algorithm 1. RT Adaptive Learning

```
Input : States $s \in S$, Actions $a \in A$, Objects $x \in U$, $V(s)$.
Output: Semi-martingale, i.e., $\{R_t, t \in T\}$.
while True do
    Begin episode;
    Initialize policy $\pi(s,a), s, V(s), sunset \leftarrow maxTime, episode \leftarrow true$;
    Estimate $V(s') = E[R_t]$ for every $a$ leading from $s$ to $s'$;
    while $V(s) \leq V(s')$ do
        $V(s) \leftarrow V(s')$ ;
        Perform action $a$, observe $r(t)$ signal, compute $\gamma'(t)$;
        Update $a(x_t) \leftarrow \gamma'(t) \cdot r(t)$;
        Choose new $a$ from new $s$ according to policy $\pi(s,a)$ ;
        Estimate new $V(s') = E[R_t]$;
        $sunset \leftarrow sunset - 1$;
        if $sunset > 1$ then
            if $V(s') > V(s)$ then
                episode continues ;
            end
        else
            $episode \leftarrow false$ {publish $\{R_t, t \in T\}$} ;
        end
    end
end
```

learning method suggested by Selfridge is represented by Alg. 1. This algorithm reflects Selfridge's run-and-twiddle (RT) adaptive control strategy. In its simplest form, RT is a greedy method that works by steepest ascent hill-climbing, where an attempt is made to maximize return R over time by choosing the most promising action in each state. The *most promising action* a means that action a has the highest estimated expected return R_t at time t. Alg. 1 looks one step ahead in each state during an episode and takes the best pick among all possible actions for the next step.

3 Approximation Spaces

The original generalized approximation space (GAS) model [32] has recently been extended as a result of recent work on nearness of objects (see, *e.g.*, [6, 17, 18, 20, 21, 33, 34]). A nearness approximation space (NAS) is a tuple

$$NAS = (U, A, N_r, \nu_B),$$

where U is a universe of objects, A, a set of probe functions, N_r, a family of neighbourhoods and ν_B is an overlap function defined by

$$\nu_B : \mathcal{P}(U) \times \mathcal{P}(U) \longrightarrow [0, 1],$$

where $\mathcal{P}(U)$ is the powerset of U. The overlap function ν_B maps a pair of sets to a number in $[0, 1]$ representing the degree of overlap between the sets of objects with features defined by $B \subseteq A$ and $\mathcal{P}(U)$ is the powerset of U [35]. For each subset $B \subseteq A$ of probe functions, define the binary relation $\sim_B = \{(x, x') \in U \times U : \forall f \in B, f(x) = f(x')\}$. Since each $\sim_B$ is, in fact, the usual Ind_B (indiscernibility) relation, for $B \subset F$ and $x \in U$, let $[x]_B$ denote the equivalence class containing x, *i.e.*,

$$[x]_B = \{x' \in U : \forall f \in B, f(x') = f(x)\} \subseteq U.$$

If $(x, x') \in \sim_B$ (also written $x \sim_B x'$), then x and x' are said to be *indiscernible* with respect to all feature probe functions in B, or simply, B-*indiscernible*. Then define a family of neighborhoods $N_r(A)$, where

$$N_r(A) = \bigcup_{B_r \subseteq P_r(A)} [x]_{B_r},$$

where $P_r(A) = \{B \subseteq A \mid |B| = r\}$ for any r such that $1 \leq r \leq |A|$. That is, r denotes the number of features used to construct families of neighborhoods. For the sake of clarity, we sometimes write $[x]_{B_r}$ to specify that the equivalence class represents a neighborhood formed using r features from B. Families of neighborhoods are constructed for each combination of probe functions in B using $\binom{|B|}{r}$, *i.e.*, $|B|$ probe functions taken r at a time. Information about a sample $X \subseteq U$ can be approximated from information contained in B by constructing a $N_r(B)$-lower approximation

$$N_r(B)_* X = \bigcup_{x : [x]_{B_r} \subseteq X} [x]_{B_r},$$

and a $N_r(B)$-upper approximation

$$N_r(B)^* X = \bigcup_{x : [x]_{B_r} \cap X \neq \emptyset} [x]_{B_r}.$$

Then $N_r(B)_* X \subseteq N_r(B)^* X$ and the boundary region $BND_{N_r(B)}(X)$ between upper and lower approximations of a set X is defined to be the complement of $N_r(B)_* X$, *i.e.*

$$BND_{N_r(B)}(X) = N_r(B)^* X \backslash N_r(B)_* X = \{x \in N_r(B)^* X \mid x \notin N_r(B)_* X\}.$$

A set X is termed a "near set" relative to a chosen family of neighborhoods $N_r(B)$ *iff* $|BND_{N_r(B)}(X)| \geq 0$. This means that, relative to B, every rough set is a near set but not every near set is a rough set. Object recognition and the problem of the nearness of objects have motivated the introduction of near sets (see, *e.g.*, [17, 20]).

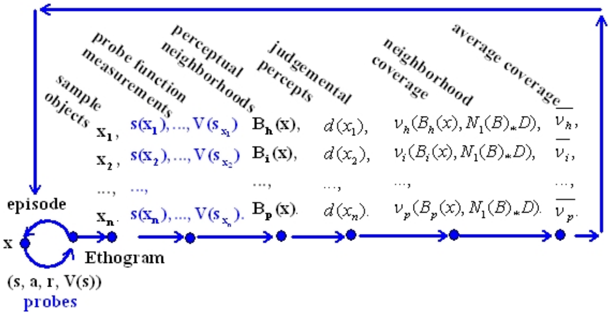
sample objects
probe function measurements
perceptual neighborhoods
judgemental percepts
neighborhood coverage
average coverage
x1, s(x1),..., V(sx1) Bh(x), d(x1), νh(Bh(x), N1(B)*D), ν̄h,
x2, s(x2),..., V(sx2) Bi(x), d(x2), νi(Bi(x), N1(B)*D), ν̄i,
..., ..., ..., ..., ..., ...,
xn. s(xn),..., V(sxn). Bp(x). d(xn). νp(Bp(x), N1(B)*D). ν̄p.
episode
x
Ethogram
(s, a, r, V(s))
probes

Fig. 2. Approximate Adaptive Learning Cycle

3.1 Percepts and Perception

The set $N_r(B)$ contains a set of percepts. A *percept* is a byproduct of perception, *i.e.*, something that has been observed [10]. For example, a member of $N_r(B)$ represents *what has been perceived about objects belonging to a neighborhood, i.e.*, observed objects with matching probe function values. Collectively, $N_r(B)$ represents a *perception*, a product of perceiving. Perception is defined as the extraction and use of information about one's environment [1]. This basic idea is represented in the *sample objects*, *probe function measurements*, *perceptual neighborhoods* and *judgemental percepts* columns in Fig. 2[6]. In this article, we focus on the perception of acceptable objects.

3.2 Sensing, Classifying, and Perceptual Judgement

Sensing provides a basis for probe function measurements commonly associated with features such as colour, contour, shape, arrangement, entropy, and so on [12,22]. A probe function can be thought of as a model for a sensor. Classification combines evaluation of a disposition of sensor measurements with judgement (apprehending the significance of a vector of probe function measurements for an observed object). The result is a higher level percept, which has been traditionally called a decision. In the context of percepts, the term *judgement* means a conclusion about an object's measurements rather than an abstract idea. This form of judgement is considered *perceptual.* Perceptual judgements provide a basis for the formulation of abstract ideas (models of perception, rules) about a class (type) of objects. Let D denote a feature called *decision* with a probe $d_B : X \times B \longrightarrow \{0, 1\}$, where X denotes a set of sample objects; B, a set of probe functions; 0, "reject perceived object" and 1, "accept perceived object". A set of objects d with matching perceptual judgements (*e.g.*, $d_B(x) = 1, x \in X$ for an acceptable object) is a mathematical model representing the abstract notion *acceptable.*

For each possible feature value j of a and $x \in U$, put $B_j(x) = [x]_B$ if, and only if, $a(x) = j$, and call $B_j(x)$ an *action block.* Put $\mathcal{B} = \{B_j(x) : a(x) = j, x \in U\}$,

[6] Subscripts h, i, p denote probe function values for a single feature, *i.e.*, where r = 1.

Algorithm 2. Approximate Adaptive Learning

```
Input  : States $s \in S$, Actions $a \in A$, Objects $x \in U$.
Output: Ethogram resulting from policy $\pi(s, a)$.
while True do
    Begin episode;
    Initialize $\bar{\nu_a}'(t)$, policy $\pi(s, a), s, V(s), sunset \leftarrow maxTime, episode \leftarrow true$;
    Insert experimental $(x, s, a, r, V(s), d(x))$ rows in ethogram, then continue ;
    Estimate $V(s') \leftarrow E[R_t]$;
    while $V(s) \leq V(s')$ do
        Perform action $a$ based on $\pi(s, a)$, observe $r(t)$ signal, compute $\gamma'(t)$;
        Update $a(x_t) \leftarrow \gamma'(t) \cdot r(t)$;
        Choose new $a$ from new $s$ according to policy $\pi(s, a)$ ;
        Estimate $V(s') \leftarrow E[R_t]$;
        $V(s) \longleftarrow V(s) + \bar{\nu_a}\ '(t) \cdot [r + max_a\{V(s')\} - V(s)]$;
        $sunset \leftarrow sunset - 1$;
        if $sunset > 1$ then
            if $V(s') > V(s)$ then
                Episode continues ;
                Add $(x, s, a, r, V(s), d(x))$ to ethogram ;
            end
        else
            $episode \leftarrow false$ {publish constructed ethogram} ;
            Compute learning rate $\bar{\nu_a}\ '(t)$ using ethogram, (1), & (2);
        end
    end
end
```

a set of blocks that "represent" action $a(x) = j$. Define $\bar{\nu}_a(t)$ (average rough coverage)[7] with respect to an action $a(x) = j$ at time t in (1).

$$\bar{\nu}_a(t) = \frac{1}{|\mathcal{B}|} \sum_{B_j(x) \in \mathcal{B}} \nu\left(B_j(x), N_r(B)_* D\right). \tag{1}$$

4 Approximate RT Adaptive Learning

Based on the introduction of families of neighbourhoods, there are different forms of adaptive learning that is influenced by the perceived behaviours recorded in episode ethograms. A behaviour is defined by the tuple

$$(s, a, r, V(s)),$$

where $V(s)$ is the estimated value of expectation $E[R_t]$. A Monte Carlo method [7,30] is used to estimate $E[R_t]$, which, in its simplest form, is a running average of the rewards received up to the current state.

[7] $\bar{\nu}_a(t)$ is computed at the end of each episode using an ethogram that is part of the adaptive learning cycle shown in Fig. 2.

The differential $\bar{\nu_a}\ '(t)$ of $\bar{\nu_a}(t)$[8] takes the place of learning rate α in Q-learning [40], where $\bar{\nu_a}\ '(t)$ reflects the rate of change of average action acceptability across adjacent episodes. Starting with the end of the second episode during approximate adaptive learning, it is possible to define a learning rate $\bar{\nu_a}\ '(t)$ shown in (2).

$$\bar{\nu}'(t) = \frac{d(\bar{\nu_a}(t))}{dt}\bigg|_{t \leftarrow t_i} \approx \frac{|\bar{\nu_a}(t_i) - \bar{\nu_a}(t_{i-1})|}{|t_i - t_{i-1}|}, \tag{2}$$

where $\bar{\nu_a}(t_i), \bar{\nu_a}(t_{i-1})$ is the average action coverage at times t_i, t_{i-1} at the end of the current and the previous episodes, respectively. In other words, at the end of each episode, $\bar{\nu}'(t)$ is refreshed to reflect a varying learning rate (see Alg. 2). Other forms of Alg. 2 are possible, if we consider combinations of features in addition to the single-feature case, where multiple-feature families of neighborhoods are used to estimate average coverage. At present, a number of fairly intensive experiments with approximate adaptive learning in colonies of organisms (*e.g.*, fish and ants) and in computer vision, are being carried out [18, 19].

5 Conclusion

This article considers a perception-based approach to adaptive learning. The early work of Zdzisław Pawlak and others on classification of objects and approximation spaces during the 1980s as well as more recent work on approximation spaces by Andrzej Skowron and Jarosław Stepaniuk provide a framework for observing the returns on episodic behaviour during learning. This work has also benefited from the work on adaptive learning by Oliver Selfridge and Chris J.C.H. Watkins, also during the 1980. It was Watkins who suggested a stopping time strategy for episodic behaviour based on the estimated value of state. The work on semi-martingales by Leo Doob introduced during the 1950s has also been helpful in the interpretation of what is happening during what is known as run-and-twiddle (RT) adaptive learning. It has been shown that a semi-martingale is constructed with a stopping time strategy during each adaptive learning episode. Future work will include various families of neighborhoods as a basis for defining a learning rate.

Acknowledgements

The author gratefully acknowledges the suggestions and insights by Andrzej Skowron, Jarosław Stepaniuk, Jay Kraut, Shabnam Shahfar, Christopher Henry, Dan Lockery and David Gunderson concerning topics in this paper. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986.

[8] Average rough coverage $\bar{\nu_a}(t)$ is computed at time t marking the end of episode.

References

1. Audi, R. (ed.): The Cambridge Dictionary of Philosophy. 2nd, Cambridge University Press, UK (1999)
2. Berenji, H.R.: A convergent actor–critic-based FRL algorithm with application to power management of wireless transmitters. IEEE Trans. on Fuzzy Systems 11/4, 478–485 (2003)
3. Doob, J.L.: Stochastic Processes. Chapman & Hall, London (1953)
4. Gomolińska, A.: Approximation spaces based on similarity and dissimilarity. In: Lindemann, G., Schlingloff, H., Burkhard, H.-D., Czaja, L., Penczek, W., Salwicki, A., Skowron, A., Suraj, Z. (eds.): Concurrency, Specification and Programming (CS&P'06). Infomatik-Berichte, Nr. 206, pp. 446–457 (2006)
5. Geramifard, A., Bowling, M., Sutton, R.S.: Incremental least-squares temporal difference learning. In: Proc. 21st National Conf. on AI (AAA06), pp. 356–361 (2006)
6. Henry, C., Peters, J.F.: Image Pattern Recognition Using Approximation Spaces and Near Sets. In: Proc. 2007 Joint Rough Set Symposium (JRS07), Toronto, Canada 14-16 May 2007 (2007)
7. Hammersley, J.M., Handscomb, D.C.: Monte Carlo Methods. Methuen & Co Ltd, London (1964)
8. Mitzenmacher, M., Upfal, E.: Probability and Computing. Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, New York (2005)
9. Orłowska, E.: Semantics of Vague Concepts, Applications of Rough Sets, Institute for Computer Science, Polish Academy of Sciences, Report 469, March 1982 (1982)
10. The Oxford English Dictionary. Oxford University Press, London (1933)
11. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Heidelberg (2004)
12. Pavel, M.: Fundamentals of Pattern Recognition, 2nd edn. Marcel Dekker, Inc, NY (1993)
13. Pawlak, Z.: Classification of Objects by Means of Attributes, Institute for Computer Science, Polish Academy of Sciences, Report 429, March 1981 (1981)
14. Pawlak, Z.: Rough Sets, Institute for Computer Science, Polish Academy of Sciences, Report 431, March 1981 (1981)
15. Pawlak, Z.: Rough sets. International J. Comp. Inform. Science 11, 341–356 (1982)
16. Pawlak, Z., Skowron, A.: Rudiments of rough sets, Information Sciences, ,177 (1), 3–27 (2007) ISSN 0020-0255
17. Peters, J.F.: Near sets. Special theory about nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
18. Peters, J.F.: Near Sets. Toward Approximation Space-Based Object Recognition. In: Proc. 2007 Joint Rough Set Symposium (JRS07), Toronto, Canada 14-16 May, 2007 (2007)
19. Peters, J.F., Borkowski, M., Henry, C., Lockery, D., Gunderson, D., Ramanna, S.: Line-Crawling Bots That Inspect Electric Power Transmission Line Equipment. In: Proc. 3rd Int. Conf. on Autonomous Robots and Agents (ICARA 2006), Palmerston North, NZ, 2006, pp. 39–44 (2006)
20. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness in approximation spaces. Lindemann, G., Schlilngloff, H., et al. (eds.): Proc. Concurrency, Specification & Programming (CS&P'2006). Informatik-Berichte Nr. 206, Humboldt-Universität zu Berlin, 2006, pp. 434–445 (2006)
21. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 77 (in press) (2007)

22. Peters, J.F.: Classification of objects by means of features. In: Kacprzyk, J., Skowron, A.: Proc. Special Session on Rough Sets, IEEE Symposium on Foundations of Computational Intelligence (FOCI07) (2007)
23. Peters, J.F., Henry, C.: Approximation spaces in off-policy Monte Carlo learning, Engineering Applications of Artificial Intelligence (in press) (2007)
24. Peters, J.F.: Rough ethology: Toward a Biologically-Inspired Study of Collective behaviour in Intelligent Systems with Approximation Spaces. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 153–174. Springer, Heidelberg (2005)
25. Peters, J.F., Henry, C.: Reinforcement learning with approximation spaces. Fundamenta Informaticae 71(2-3), 323–349 (2006)
26. Peters, J.F., Henry, C., Ramanna, S.: Rough Ethograms: Study of Intelligent System behaviour. In: Kłopotek, M.A., Wierzchoń, S., Trojanowski, K. (eds.): New Trends in Intelligent Information Processing and Web Mining (IIS05), Gdańsk, Poland, June 13-16, 2005, pp. 117–126 (2005)
27. Polkowski, L.: Rough Sets. Mathematical Foundations. Springer, Heidelberg (2002)
28. Polkowski, L., Skowron, A. (eds.): Rough Sets in Knowledge Discovery 2, Studies in Fuzziness and Soft Computing, vol. 19. Springer, Heidelberg (1998)
29. Precup, D., Sutton, R.S., Paduraru, C., Koop, A., Singh, S.: Off-policy with recognizers. Advances in Neural Information Processing Systems, pp. 1–8 (2006)
30. Rubinstein, R.Y.: Simulation and the Monte Carlo Method. John Wiley & Sons, Toronto (1981)
31. Selfridge, O.G.: Some themes and primitives in ill-defined systems. In: Selfridge, O.G., Rissland, E.L., Arbib, M.A. (eds.) Adaptive Control of Ill-Defined Systems, Plenum Press, London (1984)
32. Skowron, A., Stepaniuk, J.: Generalized approximation spaces. In: Lin, T.Y., Wildberger, A.M (eds.): Soft Computing, Simulation Councils, San Diego, 18–21 (1995)
33. Skowron, A., Swiniarski, R., Synak, P.: Approximation spaces and information granulation. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 175–189. Springer, Heidelberg (2005)
34. Skowron, A., Stepaniuk, J., Peters, J.F., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)
35. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2-3), 245–253 (1996)
36. Skowron, A., Stepaniuk, J.: Information granules and rough-neural computing. In: Pal et al. [11] (2204), pp. 43–84
37. Stepaniuk, J.: Approximation spaces, reducts and representatives. In [28], pp. 109–126
38. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
39. Wolski, M.: Similarity as nearness: Information quanta, approximation spaces and nearness structures. In: Proc. CS&P 2006. Infomatik-Berichte, pp. 424–433 (2006)
40. Watkins, C.J.C.H.: Learning from Delayed Rewards. Ph.D. Thesis, supervisor: Richard Young. King's College, Cambridge University, May 1989 (1989)
41. Watkins, C.J.C.H., Dayan, P.: Reinforcement learning. Encyclopedia of Cognitive Science. Macmillan, UK (2003)
42. Wawrzyński, P.: Intensive Reinforcement Learning, Ph.D. dissertation, supervisor: Andrzej Pacut, Institute of Control and Computational Engineering, Warsaw University of Technology, May 2005 (2005)
43. Whitehead, A.N.: Process and Reality. Macmillan, UK (1929)
44. Williams, D.: Probability with Martingales. Cambridge University Press, UK (1991)