
Power-law temporal discounting over a logarithmically compressed timeline for scale invariant reinforcement learning

Zoran Tiganj
ztiganj@bu.edu
Center for Memory and Brain
Boston University

Karthik H. Shankar
shankark@bu.edu
Center for Memory and Brain
Boston University

Marc W. Howard
marc777@bu.edu
Center for Memory and Brain
Boston University

Abstract

We discuss a recent computational hypothesis, developed based on work in psychology and neuroscience, for computing a scale-invariant timeline of future events. This hypothesis efficiently computes a model for future time on a logarithmically-compressed scale. Here we show that this model for future prediction can be used to generate a scale-invariant power-law-discounted estimate of expected future reward. The scale-invariant timeline could provide the centerpiece of a neurocognitive framework for reinforcement learning in continuous time.

1 Introduction

In Reinforcement Learning (RL), an agent learns how to optimize its actions from interacting with the environment, aiming to maximize temporally-discounted future reward. The standard approach to RL is to consider each different configuration of the environment as a different state, assuming the Markov property [Sutton and Barto, 1998]. The objective is then to learn a value of temporally discounted expected future reward for each state. This approach turned out to be tremendously useful and efficient in numerous practical applications (see e.g. Mnih et al. [2015]).

However, this approach has several limitations when applied in real-world environments and it differs from how humans and other mammals accomplish RL. Specifically: 1) Representing time with discrete states requires choice of a time scale. 2) RL algorithms based on Bellman equation require choosing a time scale for exponential discounting. 3) Representing value with a scalar obscures temporal information. 4) Markov assumption that the future depends only on the current state is not in general property of a realistic world.

Here we propose an approach that constructs a representation that can be useful for learning (including RL) and that has some desired properties: treating time as a continuous log-compressed dimension with stimuli (no states), capturing temporal relationships among the stimuli beyond the Markov property and power-law temporal discounting. In addition, this representation provides scannable log-compressed future timeline that for each time step in the future gives a probability distribution over all the stimuli (similar distributional models in classical RL, but with stimuli rather than states and with log-compressed time). One can compute the expected discounted future reward by simply integrating over such timeline. Moreover, the timeline can be used in a more model-based way as the available information is a function over internal time, rather than a scalar equivalent to value.

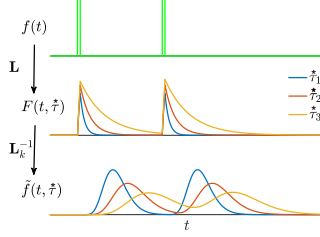


Figure 1: Constructing a scale-invariant compressed memory representation through an integral transform and its inverse. A transient input stimulus $f(t)$ (top row) is presented twice and feeds into a layer of leaky integrators $F(t, \tau^*)$ with a spectrum of time constants τ^* constituting a discrete approximation of an integral transform (middle row). The transform is denoted as \mathbf{L} since it is equivalent to the real part of the Laplace transform. Only three nodes in $F(t, \tau^*)$ are shown. Each leaky integrator is characterized with its time constant, τ^* . \mathbf{F} projects onto $\tilde{f}(t, \tau^*)$ through a set of weights defined with the operator denoted as \mathbf{L}_k^{-1} which implements an approximation of the inverse of the Laplace transform. Nodes in $\tilde{f}(t, \tau^*)$ activate sequentially following the stimulus presentation creating a memory representation. The width of the activation of each node scales with the peak time determined by the corresponding τ^* , making the memory scale-invariant. Logarithmic spacing of the τ^* assures that the memory representation is compressed.

2 Constructing log-compressed timeline of the future

One approach to estimating discounted future values would be to directly estimate future events and then evaluate their value. This “model-based” approach has been extensively studied as an alternative to recursive value estimation such as TD learning. The disadvantage of model-based approaches is that it is typically assumed that estimation of distant time points is computationally costly, with the number of steps requiring estimation of events N time points in the future going up linearly in N . Moreover, because power-law discounting implies a long tail for many choices of exponent, explicit estimation of the future seems to be prohibitively expensive computationally. Here we propose that an efficient estimation of future events using a recent proposal [Shankar et al., 2016] inspired by results from psychology and neuroscience enables a solution to this problem by computing estimates of the future along a logarithmically-compressed timeline. Using this approach, the estimate of events N time points in the future goes up like $\log N$. This approach will result in scale-invariant power-law discounting. Moreover, logarithmically-compressed timeline of expected future events is a particularly powerful representation that can be used for rapid scanning through imaginary future trajectories. It expands the scalar representation of expected future value into a functional representation over internal estimate of future time. The construction of this representation critically depends on having an analogous memory representation: we use log-compressed memory representation [Shankar and Howard, 2013] to construct symmetric log-compressed future estimation.

This approach requires three key components, a compressed memory representation, an associative memory between the compressed memory representation and ability to do time-local temporal translation. We first briefly describe the overall mechanism of the model and then in the sequel focus on more thorough description of the key components.

Let us start by assuming that we are presented with a vector-valued input $\mathbf{f}(t)$, where \mathbf{f} spans the space of all possible stimuli. At each time point we maintain the real-value Laplace transform of the input up to the current time step, $\mathbf{F}(t, \tau^*) = \mathbf{L}\mathbf{f}(t' < t)$. An approximation of the Laplace transform is implemented as a set of leaky integrators with a spectrum of time constants. The compressed memory representation is a fuzzy estimate (better resolution for more recent than for more distant events) of the input function $\mathbf{f}(t)$ computed by a linear operator $\tilde{\mathbf{f}}(t, \tau^*) \equiv \mathbf{L}_k^{-1}\mathbf{F}(t, \tau^*)$. The variable τ^* is in one-to-one relationship with the Laplace domain variable and is chosen such that $\tilde{\mathbf{f}}(t, \tau^*)$ provides a fuzzy estimate of the value of \mathbf{f} a time τ^* in the past at time t . Figure 1 provides a graphical summary of the properties of \mathbf{F} and $\tilde{\mathbf{f}}$. The inverse operator \mathbf{L}_k^{-1} will be described in detail below. Briefly, it is a linear operator that can be implemented as a one-layer feedforward network. An associative memory \mathbf{M} associates the states of the compressed memory representation $\tilde{\mathbf{f}}(t, \tau^*)$ to the current state of the stimulus $\mathbf{f}(t)$. In this way $\tilde{\mathbf{f}}(t, \tau^*)$ functions like a temporal context for $\mathbf{f}(t)$ [Howard and Kahana,

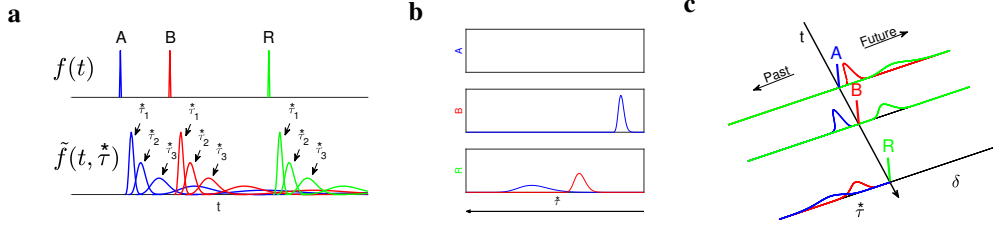


Figure 2: Constructing a scale-invariant future using an associative memory. **a**. Scale-invariant dynamical compressed memory representation. Stimuli A , B (non-rewarding) and R (rewarding) presented at different times (top row) induce a logarithmically spaced spreading sequential activation (bottom row) corresponding to $\tilde{f}(t, \tau^*)$. Activation corresponding to different stimuli are represented with different colors. For clarity, only a handful of τ^* nodes are displayed. **b**. For each value of τ^* temporal associations across stimuli are stored in the compressed associative memory \mathbf{M} . The state in \mathbf{M} predicting stimulus A is shown in blue, the state predicting stimulus B is shown in red. In this example, A was followed by B and R resulting in strong, temporally-precise association for B at a small value of τ^* and weaker, less temporally precise association with R at a larger value of τ^* (blue trace at the middle and at the bottom plot). Similarly, B was followed by R resulting in association between the two (red trace at the bottom plot). **c**. Memory of the recent past, $\tilde{f}(t, \tau^*)$, and the estimated near future, $\mathbf{p}(t, \delta)$. We consider again the sequence $[A, B, R]$ and for simplicity assume that the agent has previously experienced this and only this sequence. When the sequence is repeated again, after stimulus A has been presented (blue vertical bar) the agent predicts B and R in the future. The peak of the prediction corresponding to B (red trace) is closer to the origin, larger and more narrow than the peak of the prediction corresponding to R (green trace). This is because the time interval between A and B is shorter than the time interval between A and R . The memory trace of A is equal to zero since both B and R occur after A , so there are no stimuli in the recent past of A . After B is presented (red vertical bar), R is predicted and A is in the memory trace. After R is presented (green vertical bar), both A and B are in the memory trace, such that B is closer to the origin and represented with a larger and more narrow peak than A .

2002]. In this framework, we can estimate future states of \mathbf{f} , $\mathbf{f}(t + \delta)$ by probing the associative memory with $\tilde{f}(t + \delta, \tau^*)$. This is accomplished by implementing a translation operator in the Laplace domain operating on $\mathbf{F}(t, \tau^*)$ to obtain an estimate of $\mathbf{F}(t + \delta, \tau^*)$. Figure 2 summarizes the properties of the associative memory and translation operator graphically.

We describe these mathematical operations in more detail below. The subsection ‘‘Computing value . . .’’ will demonstrate that this approach implements power-law temporal discounting. Here we note several important points before describing the mathematical details. First, the temporal resolution of \tilde{f} is on a logarithmic scale [Shankar and Howard, 2013]; \tilde{f} is a scale-invariant estimate of the past values of \mathbf{f} . Second, because of the properties of \mathbf{L}_k^{-1} , the translation operator can be understood as a continuous change in the values of the weights of \mathbf{L}_k^{-1} . This means that one can translate δ steps in whatever period of time is necessary to fix the weights in \mathbf{L}_k^{-1} . Detailed consideration of a mechanistic neural network model inspired by findings from neurobiology suggests that scale-invariant translation is implemented in the brain as a logarithmically-compressed sweep through successive values of δ [Shankar et al., 2016]. Identifying this sweep through the future with hippocampal theta oscillations provides a concise account of numerous findings from the place cell literature and suggests that the brain can sweep through a future trajectory in a few hundred milliseconds, implementing a fast estimate of the future.

2.1 Constructing dynamical compressed memory representation of the recent past

We first define an input vector \mathbf{f} consisting of N elements such that each its element corresponds to a unique stimulus. Thus observing stimulus A makes an element in \mathbf{f} that corresponds to stimulus A , f_A , equal to one for the time A is presented and zero otherwise. Each element of the input vector \mathbf{f} has a dynamical compressed memory representation which is constructed as a two layer feedforward neural network with fixed, analytically derived weights (see Shankar and Howard [2012, 2013] for a detailed derivation). The first layer of the network implements an approximation of an integral transform of the input (Laplace transform, but as a function of a real rather than a complex variable). This means that nodes in the first layer, $\mathbf{F}(t, \tau^*)$, act as leaky integrators (first order low-pass filters)

with a spectrum of time constant k/τ^* , where k is positive integer (Figure 1):

$$\frac{\mathbf{F}(t, \tau^*)}{dt} = -\frac{k}{\tau^*} \mathbf{F}(t, \tau^*) + \mathbf{f}(t). \quad (1)$$

Leaky integrators project to the second layer, $\tilde{\mathbf{f}}$, through fixed weights that implement an approximation of the inverse of the transform by applying a k^{th} order derivative with respect to k/τ^* , denoted as $\mathbf{F}^{(k)}(t, \tau^*)$. The inverse is derived based on Post's inversion formula [Post, 1930]:

$$\tilde{\mathbf{f}}(t, \tau^*) = C_k \left(\frac{k}{\tau^*}\right)^{k+1} \mathbf{F}^{(k)}(t, \tau^*), \quad (2)$$

where C_k is a constant that depends only on k . The cells in the second layer constitute a dynamical memory representation of the input signal. To understand the properties of the memory representation we consider an impulse response of a cell in $\tilde{\mathbf{f}}$. For $f_A(\tau) = \delta(\tau = 0)$ the corresponding activation of the cells in the second layer is:

$$\tilde{f}_A(t, \tau^*) = C_k \frac{1}{\tau^*} \left(\frac{t}{\tau^*}\right)^k e^{-k\frac{t}{\tau^*}}, \quad (3)$$

where C_k here is a different constant that depends only on k . The activity of each node in $\tilde{f}_A(t, \tau^*)$ is the product of an increasing power term $\left(\frac{t}{\tau^*}\right)^k$ and a decreasing exponential term $e^{-k\frac{t}{\tau^*}}$. Consequently, each node in $\tilde{f}_A(t, \tau^*)$ has a peak that corresponds to the τ^* value of that node: $\frac{d\tilde{f}_A(t, \tau^*)}{dt} = 0 \Rightarrow t = \tau^*$. Thus, following a transient input, cells in \tilde{f}_A activate sequentially in time constituting a dynamical memory representation of the input A . This memory representation has perfect accuracy in the limit when $k \rightarrow \infty$ (when it effectively becomes a shift register). In our implementation where k is finite and τ^* is a discrete variable supported with a limited number of nodes, the memory representation becomes an approximation of the past. The approximation is scale-invariant since the width of the activation of each node scales with the peak time (this is scale-invariant since rescaling the temporal axis rescales the width of the activation by the same amount). In other words, the accuracy of the memory representation decreases with the elapse of time since the stimulus presentation. With appropriately distributed τ^* the representation can be made logarithmically compressed.

Figure 2a shows the sequential, spreading activation with logarithmically spaced τ^* for three different transient stimuli. This implementation is neurally plausible as described in Howard et al. [2014]. See Tiganj et al. [2015] and Tiganj et al. [2013] for arguments on biological plausibility of leaky integrators with a spectrum of time constants. Neurons with firing properties resembling those in Figure 2a have been found in several brain regions, including hippocampus [MacDonald et al., 2011, Salz et al., 2016], prefrontal cortex [Tiganj et al., 2016] and striatum [Mello et al., 2015].

2.2 Constructing compressed associative memory

At each time step t , an associative memory tensor $\mathbf{M}(t)$ is updated with the outer product of the input vector \mathbf{f} and the dynamical compressed memory representation $\tilde{\mathbf{f}}$. Hence $\mathbf{M}(t)$ is a three-tensor that stores the temporal relationships compressed along the τ^* axis across the space of all the stimuli (Fig. 2b):

$$\Delta\mathbf{M}(t; \tau^*) = |\mathbf{f}(t)\rangle \langle \tilde{\mathbf{f}}(t, \tau^*)|. \quad (4)$$

The dimensions of the three-tensor $\mathbf{M}(t)$ are $N \times N \times L$, where L is the number of τ^* nodes. With the three-tensor constructed in such way and in the limiting case when $k \rightarrow \infty$, an element in i^{th} row, j^{th} column of $\mathbf{M}(t; \tau^*)$ will be proportional to the number of times the stimulus i followed stimulus j after a time τ^* . In the limiting case when $\tilde{\mathbf{f}}$ takes the form of a shift register, and column-normalized $\mathbf{M}(t; \tau^*)$, which we shall denote by $\bar{\mathbf{M}}(t; \tau^*)$, will give an exact measure of the conditional probability of observing stimulus i given that the stimulus j was observed τ^* time ago.

2.3 Estimating a future timeline: Method 1 operating on a state with a multi-scale associative matrix

Given $\bar{\mathbf{M}}(t; \tau^*)$ we can at time t readily estimate a prediction of future outcomes as a function of future time $\delta = -\tau^* > 0$ as

$$\mathbf{p}(t; \delta) = \bar{\mathbf{M}}(t; \tau^*) |\mathbf{f}(t)\rangle \quad (5)$$

We can understand $\mathbf{p}(t; \delta)$ as a set of units indexed by $\delta = -\tau^*$. Because Eq. 3 is scale invariant, this prediction will also be scale invariant in time. Note that the amount of time needed to compute a future time δ does not depend on the value of δ . Also note that the prediction is a sum over future outcomes.

2.4 Estimating a future timeline: Method 2 time-local temporal translation

Another approach to constructing a future timeline follows from the observation that Eq. 1 is the Laplace transform of the past history. This means that future states of the memory representation (under the assumption of no new inputs) can be obtained by applying a linear translation operator $\mathbf{R}^\delta \equiv e^{-k\delta/\tau^*}$ on the leaky integrators, where δ is the time by which the memory representation is translated [Shankar et al., 2016].

$$\mathbf{p}(t; \delta) = \bar{\mathbf{M}}(t; \tau^*) \mathbf{L}_k^{-1} \mathbf{R}^\delta |\mathbf{f}(t)\rangle \quad (6)$$

In this approach, one could understand δ as being set successively and swept through a range of values from 0 to some large number. Previous work [Shankar et al., 2016] has argued that in order to preserve scale-invariance this sweep should be compressed such that the time taken to sweep should accelerate exponentially as δ increases from zero. Shankar et al. [2016] further argued that results from neurophysiology experiments suggest that translation corresponds to hippocampal phase precession [O’Keefe and Recce, 1993], suggesting that the entire sweep is completed in the brain within about 200 ms.

2.5 Properties of the timeline

Although the two methods described above are somewhat different in their mechanism, they both produce a similar outcome. Figure 3 illustrates the basic properties. Because the representation of the past is scale-invariant, the model naturally produces a scale-invariant estimation of the future. Note that events further in the future are predicted with lower temporal resolution. It can be shown that this function over future time is scale-invariant; rescaling time results in a prediction of the same functional form. Moreover, this approach does not simulate a single path leading forward from the future. It rather results in an estimate of the future trajectory of events, averaged across paths. However, unlike methods that sum over time points (in particular, by computing expected future value using the Bellman Equation) this method preserves the relative temporal order of future events. This is quite powerful in that one can imagine a decision-maker using this future timeline flexibly in a way that would not be possible simply from a cached value.

As such, these methods provide an intriguing complement to traditional model-based methods. The amount of time to compute an estimate of a point in the future does not depend on the distance in the future using Method 1; for Method 2 the amount of computational time goes like the log of the point in the future. Moreover, if one accepts the argument that translation corresponds to theta phase precession [Shankar et al., 2016], translation can be accomplished quickly on a cognitive time scale in the brain. However, unlike traditional model-based methods for simulating the future, these methods cannot provide separate information about separate paths, providing only information about the expected future.

2.6 Computing value by probing the association memory with the time-translated input

One utility of having a compressed timeline of the expected future events is the ability to rapidly (since the future time is log-compressed) integrate over the timeline and compute a scalar value of expected future reward. Value can be given either to individual stimuli by probing the temporal associations stored in $\bar{\mathbf{M}}$ this time with a time-translated stimulus representation (Fig. 2c):

$$v_A(t, \delta) = \langle \mathbf{r} | \bar{\mathbf{M}}(t; \tau^*) \mathbf{R}^\delta | F_A(t, \tau^*) \rangle, \quad (7)$$

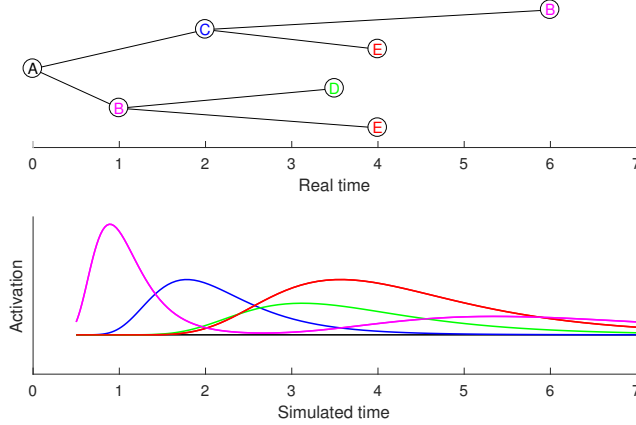


Figure 3: Compressed future timeline constructed for a continuous world with complex temporal dynamics. Top: a set of contingencies in which a variety of states follow from state A at various temporal lags. Bottom: An estimate of the representation that comes from translating \vec{f}_A differing values of δ (here labeled as “Simulated time”). The model produces an estimate of the outcomes that will arrive in internal future time. The estimate sums over branches of the tree but does not sum over future time. Note that the prediction of E at time 4 is large even though it is reached from different branches.

where $v_A(t, \delta)$ is the prediction of reward at time $t + \delta$ computed at time t after probing with the stimulus A . \mathbf{r} is a vector of length N that denotes the amount of reward given by each stimulus. The value of a stimulus is computed as an integral of the prediction over the future time:

$$V_A(t) = \int v_A(t, \delta) g(\delta) d\delta, \quad (8)$$

where $V_A(t)$ is a value of the stimulus A presented at time t . The prediction is scaled with a function $g(\delta)$. We set $g(\delta) = 1/\delta^\alpha$ with $\alpha \geq 1$ to accomplish power-law scaling (Figure 4a). $V_A(t)$ provides a power-law discounted estimate of the future reward (Figure 4b and Figure 4c).

In a probabilistic world where we transition through different states obeying the Markov property, the above mechanism for computing the value of a stimulus based on future reward will exactly converge to the statistically expected future reward value. The convergence of the proposed approach is analogous to the convergence of TD learning as proved for TD(0) in Sutton [1988] and for TD(λ) in Dayan [1992]. This is because when computing $V_A(t)$ in Equation (8) we are summing the conditional probability of observing all rewarding states over all time lags (see appendix where this is explicitly shown).

2.7 Methods for non-Markov statistics

Because they predict a trajectory from only the currently available state, the computational approaches above are suitable for worlds with Markov statistics. However, one can imagine worlds with much more complex temporal relationships. Although the number of possible generating functions is in general quite complex, one can generalize this approach to a simple extension. Suppose the world provides a number of stimuli. (Unknown) subsets of these stimuli predict one another according to a Markov process, but the time lag between these observations is not known.

If the Markov processes are independent of one another and the time scales between them are sufficiently large, one could effectively model this world by maintaining a scale-invariant representation of the entire history. In this case, one can construct an analog of Method 2 by time-translating the entire compressed memory representation:

$$\mathbf{p}(t, \delta) = \bar{\mathbf{M}}(t; \tau^*) \mathbf{L}_k^{-1} \mathbf{R}^\delta |\mathbf{F}(t, s)\rangle, \quad (9)$$

where $\mathbf{p}(t, \delta)$ is a prediction vector across the stimulus space computed at time t of the stimuli expected to occur at time $t + \delta$. Notice that the entire content of the present state of the compressed

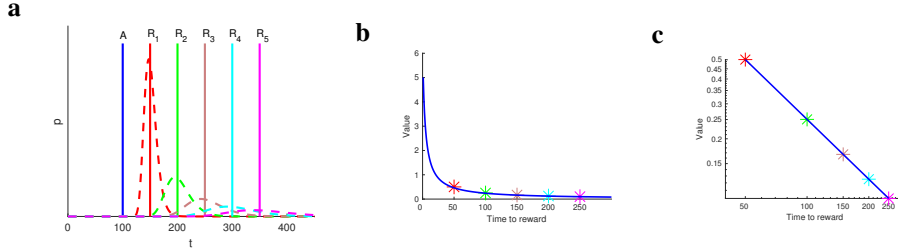


Figure 4: Scale-invariant value computation. **a.** Constructing prediction of the future reward. An agent observed a temporal sequence consisting of a non-rewarding stimulus A , followed by five rewarding stimuli, R_1 to R_5 (solid lines). After observing the sequence, the next time A is presented, the agent computes a prediction for each of the five rewarding stimuli (dashed lines with colors corresponding to the rewarding stimuli). The prediction is less accurate for rewards that come later in time: 1) amplitude of the prediction decreases in a power-law fashion, 2) the width of the prediction increases with time such that if the time axis was plotted with logarithmic spacing the half-width of the prediction would be the same for all five rewarding stimuli. **b.** and **c.** Power-law discounted value on lin-lin axes (plot *b*) and log-log axes (plot *c*). Value is computed as an integral of the prediction. To illustrate the power-law discounting, we computed the integral separately for each of the five rewarding stimuli (surface under the dashed lines on plot *a*). Stars represent the exact value of the integral for the rewarding stimulus of corresponding color and the blue line is a power-law function with the exponent of -1 .

memory representation is used to construct the estimate of the future. Thus the temporal structure of the history will affect this estimate of the future.¹

The impulse response of the memory nodes translated in time by δ , which we can refer to as $\tilde{\mathbf{f}}(t + \delta, \tau) \equiv \mathbf{L}_k^{-1} \mathbf{R}^\delta \mathbf{F}(s; t)$ has the following form:

$$\tilde{\mathbf{f}}(t + \delta, \tau) = C_k \frac{1}{\tau} \left(\frac{t + \delta}{\tau} \right)^k e^{-k \frac{t + \delta}{\tau}}. \quad (10)$$

Here C_k is the same as in Equation 3.

3 Discussion

The Bellman Equation and related approaches [Dayan, 1993] can be understood as an attempt to estimate the outcomes that follow from a particular state. The idea behind this strategy is that it is inefficient or unbiological to simply hypothesize that the future is directly computed. In this paper, we described several methods for directly computing an estimate of the future. These methods provide information about the averaged set of observations that will follow a particular state. They are computationally efficient, especially insofar as they are well-suited for problems in which stimuli are embedded in continuous time. Because the representations of the past and the future are both scale-invariant, it is not necessary to have a strong prior belief about the relevant time scale of the problem one is trying to solve. Although the estimates of the future converge slowly relative to TD learning in a world governed by Markov process at which states come once per time step, this condition does not describe many real-world learning situations. In worlds with continuous time and processes operating simultaneously at multiple scales, the number of states necessary to describe those statistics as a Markov process grows astronomically, so that the advantages of fast convergence *via* TD learning may not actually be of use. Several decades of work on RL approaches have built an impressive array of techniques suitable for a wide range of practical problems. Perhaps scale-invariant representations of future timelines could give rise to a related family of approaches.

¹One could also construct an analog of Method 1 that incorporates the entire temporal history to construct an estimate of the future using a single feedforward projection. However this would require a very high number of connections between units coding for the point of future time and units coding for points of past time. Moreover, one would have to train those connections non-locally. Perhaps this would be possible if translation takes place during learning, perhaps corresponding to offline replay events.

4 Acknowledgments

This work was supported by NIBIB R01EB022864, NIMH R01MH112169 and MURI N00014-16-1-2832.

References

- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- Karthik H Shankar, Inder Singh, and Marc W Howard. Neural mechanism to simulate a scale-invariant future. *Neural Computation*, 28:2594–2627, 2016.
- K. H. Shankar and M. W. Howard. Optimally fuzzy scale-free memory. *Journal of Machine Learning Research*, 14:3753–3780, 2013.
- M. W. Howard and M. J. Kahana. A distributed representation of temporal context. *Journal of Mathematical Psychology*, 46(3):269–299, 2002.
- K. H. Shankar and M. W. Howard. A scale-invariant representation of time. *Neural Computation*, 24:134–193, 2012.
- E. Post. Generalized differentiation. *Transactions of the American Mathematical Society*, 32:723–781, 1930.
- Marc W Howard, Christopher J MacDonald, Zoran Tiganj, Karthik H Shankar, Qian Du, Michael E Hasselmo, and Howard Eichenbaum. A unified mathematical framework for coding time, space, and sequences in the hippocampal region. *Journal of Neuroscience*, 34(13):4692–707, 2014. doi: 10.1523/JNEUROSCI.5808-12.2014.
- Zoran Tiganj, Michael E. Hasselmo, and Marc W. Howard. A simple biophysically plausible model for long time constants in single neurons. *Hippocampus*, 25(1):27–37, 2015.
- Zoran Tiganj, Karthik H Shankar, and Marc W Howard. Encoding the laplace transform of stimulus history using mechanisms for persistent firing. *BMC Neuroscience*, 14(Suppl 1):P356, 2013.
- C. J. MacDonald, K. Q. Lepage, U. T. Eden, and H. Eichenbaum. Hippocampal “time cells” bridge the gap in memory for discontinuous events. *Neuron*, 71:737–749, 2011.
- Daniel M Salz, Zoran Tiganj, Srijesa Khasnabish, Annalyse Kohley, Daniel Sheehan, Marc W Howard, and Howard Eichenbaum. Time cells in hippocampal area CA3. *The Journal of Neuroscience*, 36(28):7476–7484, 2016.
- Z. Tiganj, J. Kim, M. W. Jung, and M. W. Howard. Sequential firing codes for time in rodent mPFC. *Cerebral Cortex*, (1-9), 2016. doi: 10.1093/cercor/bhw336.
- Gustavo B. M. Mello, Sofia Soares, and Joseph J. Paton. A Scalable Population Code for Time in the Striatum. *Current Biology*, 25(9):1113–1122, 2015.
- J. O’Keefe and M. L. Recce. Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus*, 3:317–30, 1993.
- Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.
- Peter Dayan. The convergence of TD (λ) for general λ . *Machine learning*, 8(3-4):341–362, 1992.
- Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.