Scale invariant value computation for reinforcement learning in continuous time

Zoran Tiganj, Karthik H. Shankar and Marc W. Howard
Department of Psychological and Brain Sciences
Boston University

Abstract

Natural learners must compute an estimate of future outcomes that follow from a stimulus in continuous time. Critically, the learner cannot in general know *a priori* the relevant time scale over which meaningful relationships will be observed. Widely used reinforcement learning algorithms discretize continuous time and use the Bellman equation to estimate exponentially-discounted future reward. However, exponential discounting introduces a time scale to the computation of value, implying that the relative values of various states depend on how time is discretized. This is a serious problem in continuous time as successful learning requires prior knowledge of the solution. 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.

Introduction

In reinforcement learning, an agent learns how to optimize its actions from interacting with the environment, aiming to maximize temporally-discounted future reward. In order to navigate the environment, the agent perceives stimuli that define different states. The stimuli are experienced embedded in continuous time with temporal relationships that the agent needs to learn in order to learn the optimal action policy. Temporal discounting is well justified by numerous behavioral experiments on humans and animals (see e.g. Kurth-Nelson, Bickel, and Redish (2012)) and it is useful in numerous practical applications (see e.g. Mnih et al. (2015)). If the value of a state is defined as expected future reward discounted with an exponential function of future time, value can be updated in a recursive fashion, following the Bellman equation (Bellman, 1957). The Bellman equation is a foundation of highly successful and widely used modern reinforcement learning approaches such as dynamic programming and temporal difference (TD) learning (Sutton and Barto, 1998).

Exponential temporal discounting is not scale-invariant

When using the Bellman equation (or exponential discounting in general), values assigned to the states will depend on the chosen discretization of the temporal axis in a non-linear fashion. Consequently the ratio of the values attributed to the states changes as a function of the chosen temporal resolution and the base of the exponential function. To illustrate this let us define the value of a state s observed at time t as a sum of expected rewards r discounted with an exponential function:

$$V_{s_t} = E\left\{\sum_{i=1}^{n} \gamma^{i-1} r_{t+i}\right\}, \tag{1}$$

with $0 \leq \gamma \leq 1$ and n is the last state (either a terminal state in episodic tasks or absorbing state in continuing tasks). $E\{\}$ denotes the expectation value over many experiences with the environment. Equation (1) can be written in a recursive form, known as the Bellman equation:

$$V_{s_t} = E\left\{r_{t+1} + \gamma V_{s_{t+1}}\right\}. \tag{2}$$

The number of states between two points in continuous time depends on the discretization. Let Δ refer to the temporal difference between adjacent states. Let us suppose that non-rewarding states A and B always precede rewarding state R by times $t = t_A$ and $t = t_B$ respectively. With exponential discounting values attributed to states A and B depend on the choice of temporal resolution Δ:

$$V_A(\Delta)/V_B(\Delta) = \gamma^{\frac{t_A - t_B}{\Delta}}. \tag{3}$$

The above relation illustrates that the values computed using the Bellman equation are not scale-invariant. If Δ is big relative to the difference $t_A - t_B$, then the values of the two stimuli are effectively identical. Conversely if Δ is much less than the difference, then, for $\gamma < 0$ the difference in value is effectively infinite, with the falloff depending on the choice of γ. Choosing γ and Δ effectively requires us to already know the value of $t_A - t_B$. Moreover, the number of trials needed to converge to the true values also depends on the chosen discretization. This implies that when using the Bellman equation, prior knowledge about the temporal structure of the environment is necessary in order to choose an appropriate temporal scale. Exponential discounting therefore limits the applicability of learning methods based on the Bellman equation in autonomous learning systems that need to operate in a dynamical, ever-changing world with unknown temporal scales.

Power-law temporal discounting is scale-invariant

With power-law discounting value of a state s at time t can be expressed as:

$$V_{s_t} = E\left\{\sum_{i=1}^{n} \frac{1}{i} r_{t+i}\right\}. \tag{4}$$

Using the same example as above we compare relative values attributed to states A and B:

$$V_A(\Delta)/V_B(\Delta) = \frac{t_A}{t_B}. \tag{5}$$

Unlike with exponential discounting, the ratio of the values does not depend on the choice of Δ demonstrating scale-invariance.

Furthermore, if we scale the temporal axis by β, the value scales accordingly; the functional form of the value is unaffected by the choice of β:

$$V_A(\beta\Delta) = \frac{\beta\Delta}{t_A} = \beta V_A(\Delta). \tag{6}$$

Notice that the same can not be said for exponential discounting—the equation of analogous form to Equation (6) can not be written for exponential discounting.

For completeness we mention special cases in which power-law discounting can be implemented in a recursive fashion through the Bellman equation. The recursive implementation with hyperbolic discounting (power-law discounting can be seen as a special case of hyperbolic discounting) is possible when the environment contains at most one reward of known amount (Alexander and Brown, 2010). However, for an arbitrary distribution of rewarding states and reward amounts, a recursive form for a scalar value does not exist. This can be seen by noting that the existence of such a solution would require a function of future rewarding states to be mapped to a scalar estimate of discounted value. One can implement power law discounting by hypothesizing multiple exponentially discounting functions with a power law distribution of time constants (Kurth-Nelson, Bickel, and Redish, 2012; Sutton, 1995). With such approach scale-invariance could be achieved by computing the value separately for all the scales and integrating over the scales. In this case the number of steps to converge using the recursive form would still depend on the discretication.

Power-law temporal discounting from direct prediction of the future

Another 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. 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 seem to be prohibitively expensive computationally. Here we propose that an efficient estimation of future events using a recent proposal (Shankar, Singh, and Howard, 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.

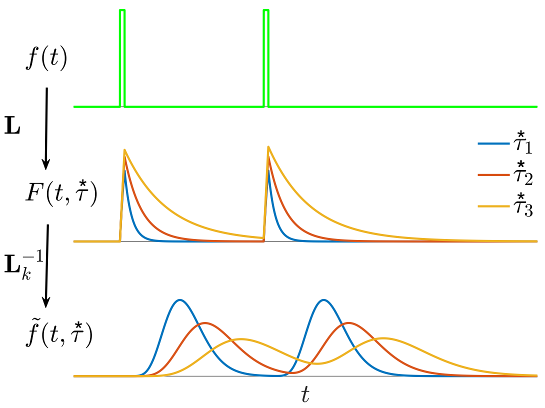

Figure 1: Constructing a scale-invariant compressed memory representation through an integral transform and its inverse. A transient input stimulus $\mathbf{f}(t)$ (top row) is presented twice and feeds into a layer of leaky integrators $\mathbf{F}(t,\overset{*}{\tau})$ with a spectrum of time constants $\overset{*}{\tau}$ 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 $\mathbf{F}(t,\overset{*}{\tau})$ are shown. Each leaky integrator is characterized with its time constant, $\overset{*}{\tau}$. $\mathbf{F}$ projects onto $\tilde{\mathbf{f}}(t,\overset{*}{\tau})$ 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 $\tilde{\mathbf{f}}(t,\overset{*}{\tau})$ 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 $\overset{*}{\tau}$, making the memory scale-invariant. Logarithmic spacing of the $\overset{*}{\tau}$ assures that the memory representation is compressed.

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. If we are presented with a vector-valued input $\mathbf{f}(t)$, we maintain at each time point the Laplace transform of the input up to the current time step, $\mathbf{F}(t,\overset{*}{\tau}) = \mathbf{L}\mathbf{f}(t' < t)$. The compressed memory representation is a fuzzy estimate of the input function computed by a linear operator $\tilde{\mathbf{f}}(t,\overset{*}{\tau}) \equiv \mathbf{L}_k^{-1}\mathbf{F}(t,\overset{*}{\tau})$. The variable $\overset{*}{\tau}$ is in one-to-one relationship with the Laplace domain variable and is chosen such that $\tilde{\mathbf{f}}(t, \overset{*}{\tau})$ provides a fuzzy estimate of the value of $\mathbf{f}$ a time $\overset{*}{\tau}$ 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 $\tilde{\mathbf{f}}(t, \overset{*}{\tau})$ to the current state of the stimulus $\mathbf{f}(t)$. In this way $\tilde{\mathbf{f}}(t, \overset{*}{\tau})$ functions like a temporal context for $\mathbf{f}(t)$ (Howard and Kahana, 2002). In this framework, we can estimate future states of $\mathbf{f}$, $\mathbf{f}(t+\delta)$ by probing the associative memory with $\tilde{\mathbf{f}}(t+\delta, \overset{*}{\tau})$. This is accomplished by implementing a translation operator in the Laplace domain operating on $\mathbf{F}(t, \overset{*}{\tau})$ to obtain an estimate of $\mathbf{F}(t+\delta, \overset{*}{\tau})$. 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 $\tilde{\mathbf{f}}$ is on a logarithmic scale (Shankar and Howard, 2013); $\tilde{\mathbf{f}}$ 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, Singh, and Howard, 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 model-free estimate of the future.

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 (state). 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, $\mathbf{F}(t, \overset{*}{\tau})$, act as leaky integrators (first order low-pass filters) with a spectrum of time constant $k/\overset{*}{\tau}$, where k is positive integer (Figure 1):

$$\frac{\mathbf{F}(t, \overset{*}{\tau})}{dt} = -\frac{k}{\overset{*}{\tau}}\mathbf{F}(t, \overset{*}{\tau}) + \mathbf{f}(t). \tag{7}$$

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 $k/\overset{*}{\tau}$, denoted as $\mathbf{F}^{(k)}(t, \overset{*}{\tau})$. The inverse is derived based on Post's inversion formula (Post, 1930):

$$\tilde{\mathbf{f}}(t, \overset{*}{\tau}) = C_k \left(\frac{k}{\overset{*}{\tau}}\right)^{k+1} \mathbf{F}^{(k)}(t, \overset{*}{\tau}), \tag{8}$$

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:

$$\tilde{f}_A(t, \overset{*}{\tau}) = C_k \frac{1}{\overset{*}{\tau}} \left(\frac{t}{\overset{*}{\tau}}\right)^k e^{-k\frac{t}{\overset{*}{\tau}}}, \tag{9}$$

where C_k here is a different constant that depends only on k. The activity of each node in $\tilde{f}_A(t, \overset{*}{\tau})$ is the product of an increasing power term $\left(\frac{t}{\overset{*}{\tau}}\right)^k$ and a decreasing exponential term $e^{-k\frac{t}{\overset{*}{\tau}}}$. Consequently, each node in $\tilde{f}_A(t, \overset{*}{\tau})$ has a peak that corresponds to the $\overset{*}{\tau}$ value of that node: $\frac{d\tilde{f}_A(t, \overset{*}{\tau})}{dt} = 0 \Rightarrow t = \overset{*}{\tau}$. 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 perfect accuracy in the limit when $k \to \infty$. In our implementation where k is finite and $\overset{*}{\tau}$ 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. Choosing $\overset{*}{\tau}$ to be logarithmically spaced makes the dynamical memory representation compressed. Figure 2a shows the sequential, spreading activation with logarithmically spaced $\overset{*}{\tau}$ for three different transient stimuli. This implementation is neurally plausible as described in Howard et al. (2014). See Tiganj, Hasselmo, and Howard (2015) and Tiganj, Shankar, and Howard (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, Soares, and Paton, 2015).

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

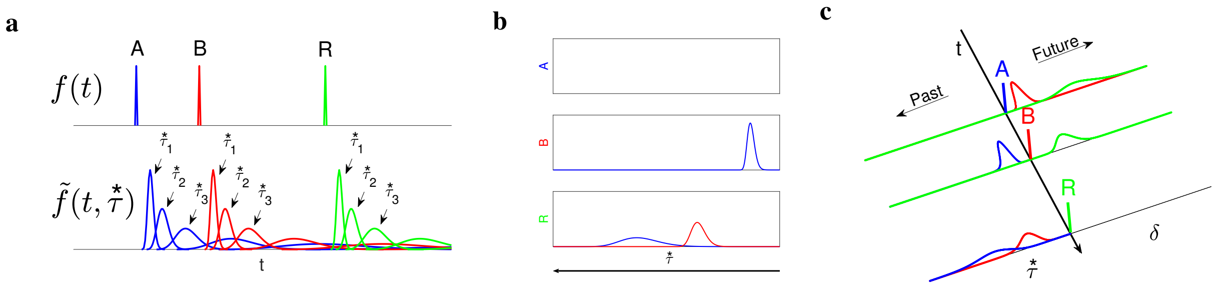

Figure 2: Constructing a scale-invariant future using an associative memory. **A.** Scale-invariant dynamical compressed memory representation. Stimuli A, B (non-rewarding) and R (rewarding) presented at different times (top row) induce a logarithmically spaced spreading sequential activation (bottom row) corresponding to $\tilde{\mathbf{f}}(t,\overset{*}{\tau})$. Activation corresponding to different stimuli are represented with different colors. For clarity, only a handful of $\overset{*}{\tau}$ nodes are displayed. **B.** For each value of $\overset{*}{\tau}$ 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 show 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 $\overset{*}{\tau}$ and weaker, less temporally precise association with R at a larger value of $\overset{*}{\tau}$ (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, $\tilde{\mathbf{f}}(t,\overset{*}{\tau})$, 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. When stimuli A and B are presented, the expected power-law discounted future reward can be computed for each of the two stimuli by integrating the prediction of the reward R along the *future* axis.

the dynamical compressed memory representation $\tilde{\mathbf{f}}$. Hence $\mathbf{M}(t)$ is a three-tensor that stores the temporal relationships along the $\overset{*}{\tau}$ axis across the space of all the stimuli[1] (Fig. 2b):

$$\Delta\mathbf{M}(t;\overset{*}{\tau}) = |\mathbf{f}(t)\rangle\,\langle\tilde{\mathbf{f}}(t,\overset{*}{\tau})|\,. \qquad (10)$$

The dimensions of the three-tensor $\mathbf{M}(t)$ are $N \times N \times L$, where L is the number of $\overset{*}{\tau}$ 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 $\mathbf{M}(t;\overset{*}{\tau})$ will be proportional to the number of times the stimulus i followed stimulus j after a time $\overset{*}{\tau}$. In the limiting case, $\tilde{\mathbf{f}}$ takes the form of a shift register (in such case, with linear $\overset{*}{\tau}$, each node in $\tilde{\mathbf{f}}$ would correspond to one slot in an evenly spaced shift register), and column normalized $\mathbf{M}(t;\overset{*}{\tau})$ (which we shall denote by $\bar{\mathbf{M}}(t;\overset{*}{\tau})$) will give an exact measure of the conditional probability of observing stimulus i given that the stimulus j was observed $\overset{*}{\tau}$ time ago.

Estimating future: mechanism for time-local temporal translation

Constructing dynamical memory representation of the recent past through described integral transform allows straightforward implementation of temporal translation. This means that future states of the memory representation (under the assumption of no new inputs) can be obtained by applying a linear operator $\mathbf{R}^\delta \equiv e^{-k\delta/\overset{*}{\tau}}$ on the leaky integrators, where δ is the time by which the memory representation is translated (Shankar, Singh, and Howard, 2016). The impulse response of the memory nodes translated in time by δ has the following form:

$$\tilde{\mathbf{f}}(t+\delta,\overset{*}{\tau}) = C_k \frac{1}{\overset{*}{\tau}}\left(\frac{t+\delta}{\overset{*}{\tau}}\right)^k e^{-k\frac{t+\delta}{\overset{*}{\tau}}}. \qquad (11)$$

Here C_k is the same as in Equation 9.

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

Time-local estimate of the future rewards is obtained by probing the temporal associations stored in $\mathbf{M}$ with a time-translated stimulus representation (Fig. 2c):

$$p_A(t,\delta) = \langle\mathbf{r}|\,\bar{\mathbf{M}}(t;\overset{*}{\tau})\,\mathbf{R}^\delta\,|F_A(t,\overset{*}{\tau})\rangle\,, \qquad (12)$$

where $p_A(t,\delta)$ is a prediction computed at time t of the reward amount at time $t+\delta$ based on the stimulus A. The reward $\mathbf{r}$ is a vector of length N and contains the amount of reward given for each stimulus. If a stimulus is non-rewarding its entry in $\mathbf{r}$ is equal to zero. Translation of the input stimulus is done using $\mathbf{R}^\delta$, such that $\mathbf{R}^\delta\,|F_A(t,\overset{*}{\tau})\rangle$ gives the state of the dynamical memory representation at time δ in the future.

[1] We use here bra-ket notation, where $|\cdot\rangle$ represents a tensor (or column vector for a first order tensor), $\langle\cdot|$ represents its transpose (or a row vector for a first order tensor), $\langle\cdot|\cdot\rangle$ represents the inner-product of two tensors of the same rank, and $|\cdot\rangle\,\langle\cdot|$ denotes the outer product of two tensors.

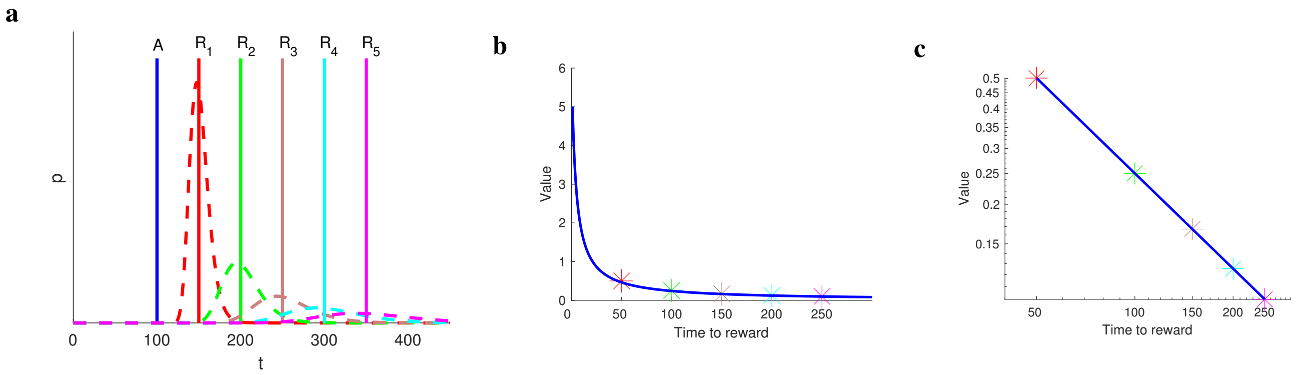

Figure 3: Scale-invariant value computation. **a.** Constructing prediction of the future reward. An agent observed a temporal sequence consisting of a non-rewarding stimulus A, followed by five rewarding stimuli, R_1 to R_5 (solid lines). After observing the sequence, the next time A is presented, the agent computes a prediction for each of the five rewarding stimuli (dashed lines with colors corresponding to the rewarding stimuli). The peak time of the prediction for each of five rewarding stimuli is at $d\frac{k}{k+2}$, where d is a temporal distance between A and the respective rewarding stimulus. The prediction was computed with $k = 40$ making the peak of the prediction slightly earlier than the time of the rewarding stimulus. 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.

The value of a stimulus (or state) is computed as an integral of the prediction over the future time:

$$V_A(t) = \int p_A(t,\delta) g(\delta) d\delta, \tag{13}$$

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 3a). Prediction form a biologically realistic neural model for constructing time-local estimate of the future is that $\alpha = 1$ (Shankar, Singh, and Howard, 2016). $V_A(t)$ provides a power-law discounted estimate of the future reward (Figure 3b and Figure 3c).

In a probabilistic world where we transition through different states obeying 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 (13) we are summing the conditional probability of observing all rewarding states over all time lags.

Discussion

The proposed approach provides a novel method for computing value in a scale-invariant way, using power-law discounting of the expected future reward. This approach reflects real-life setting in which stimuli occur at different moments in time, which is represented as a continuous dimension, rather than a discretized set of states as it is commonly done in reinforcement learning. In fact, states in our approach correspond only to moments when a stimulus is presented, not to the time between the stimuli. Therefore there is no error propagation through a discrete set of states separating different stimuli. The associations are made directly between the stimuli using a dynamical compressed memory representation. Superficially, the memory representation is similar to the one in Ludvig, Sutton, and Kehoe (2008, 2012), but it is constructed mechanistically through a neural network with analytically derived wights and its use to compute the value is rather different. Instead of considering each node corresponding to a different value of $\overset{*}{\tau}$ to be a separate state (or a microstate as in Ludvig, Sutton, and Kehoe (2008, 2012)), we use these nodes to compute the extent to which different stimuli predict each other at lag $\overset{*}{\tau}$. This dramatically reduces the number of states and therefore does not restrict us on using the Bellman equation and exponential discounting which is not scale-invariant. Instead, we use future time translation of the compressed memory representation with a property of power law discounting, giving us scale-invariant estimate of the value for each stimulus.

Acknowledgments

We gratefully acknowledge helpful discussions with Samuel Gershman and Per Sederberg. This work was supported by NIBIB R01EB022864, NIMH R01MH112169 and MURI N00014-16-1-2832.

References

Alexander, W. H., and Brown, J. W. 2010. Hyperbolically discounted temporal difference learning. *Neural Computation* 22(6):1511–1527.

Bellman, R. 1957. A Markovian decision process. Technical report, DTIC Document.

Dayan, P. 1992. The convergence of TD (λ) for general λ. *Machine learning* 8(3-4):341–362.

Howard, M. W., and Kahana, M. J. 2002. A distributed representation of temporal context. *Journal of Mathematical Psychology* 46(3):269–299.

Howard, M. W.; MacDonald, C. J.; Tiganj, Z.; Shankar, K. H.; Du, Q.; Hasselmo, M. E.; and Eichenbaum, H. 2014. A unified mathematical framework for coding time, space, and sequences in the hippocampal region. *Journal of Neuroscience* 34(13):4692–707.

Kurth-Nelson, Z.; Bickel, W.; and Redish, A. D. 2012. A theoretical account of cognitive effects in delay discounting. *European Journal of Neuroscience* 35(7):1052–1064.

Ludvig, E. A.; Sutton, R. S.; and Kehoe, E. J. 2008. Stimulus representation and the timing of reward-prediction errors in models of the dopamine system. *Neural computation* 20(12):3034–3054.

Ludvig, E. A.; Sutton, R. S.; and Kehoe, E. J. 2012. Evaluating the TD model of classical conditioning. *Learning & Behavior* 40(3):305–319.

MacDonald, C. J.; Lepage, K. Q.; Eden, U. T.; and Eichenbaum, H. 2011. Hippocampal “time cells” bridge the gap in memory for discontiguous events. *Neuron* 71:737–749.

Mello, G. B. M.; Soares, S.; and Paton, J. J. 2015. A Scalable Population Code for Time in the Striatum. *Current Biology* 25(9):1113–1122.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Post, E. 1930. Generalized differentiation. *Transactions of the American Mathematical Society* 32:723–781.

Salz, D. M.; Tiganj, Z.; Khasnabish, S.; Kohley, A.; Sheehan, D.; Howard, M. W.; and Eichenbaum, H. 2016. Time cells in hippocampal area CA3. *The Journal of Neuroscience* 36(28):7476–7484.

Shankar, K. H., and Howard, M. W. 2012. A scale-invariant representation of time. *Neural Computation* 24:134–193.

Shankar, K. H., and Howard, M. W. 2013. Optimally fuzzy scale-free memory. *Journal of Machine Learning Research* 14:3753–3780.

Shankar, K. H.; Singh, I.; and Howard, M. W. 2016. Neural mechanism to simulate a scale-invariant future. *Neural Computation* 28(12).

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* 3(1):9–44.

Sutton, R. S. 1995. Td models: Modeling the world at a mixture of time scales. In *ICML*, volume 12, 531–539. Citeseer.

Tiganj, Z.; Kim, J.; Jung, M. W.; and Howard, M. W. 2016. Sequential firing codes for time in rodent mPFC. *Cerebral Cortex* (1-9).

Tiganj, Z.; Hasselmo, M. E.; and Howard, M. W. 2015. A simple biophysically plausible model for long time constants in single neurons. *Hippocampus* 25(1):27–37.

Tiganj, Z.; Shankar, K. H.; and Howard, M. W. 2013. Encoding the laplace transform of stimulus history using mechanisms for persistent firing. *BMC Neuroscience* 14(Suppl 1):P356.