PROBABILISTIC SUCCESSOR FEATURES ALLOW FOR FLEXIBLE BEHAVIOUR

Jesse P. Geerts & Neil Burgess
University College London
London, W1T 4JG, UK
{jesse.geerts.14,neil.burgess}@ucl.ac.uk

Kimberly L. Stachenfeld
DeepMind
London, UK
stachenfeld@google.com

ABSTRACT

The assignment of credit to the appropriate preceding stimuli is of crucial importance in Reinforcement Learning (RL). One aspect of understanding the neural underpinnings of this process involves understanding what sorts of stimulus representations support generalisation. Successor Features (SFs) achieve generalisation through a predictive representation: states that predict similar futures are similarly represented. Another dimension of credit assignment involves understanding how agents handle uncertainty about learned associations, using probabilistic methods such as Kalman Temporal Differences (KTD). Combining these approaches, we propose using KTD to estimate a distribution over SFs. Kalman SF captures uncertainty about the estimated SFs as well as covariances between different SFs. We show that, unlike vanilla SF methods, Kalman SF exhibits partial transition revaluation, as humans do in a decision making experiment and as rodents do in an associative learning study. We conclude by discussing future applications of Kalman SF as a model of the interaction between predictive and probabilistic reasoning.

1 INTRODUCTION

Predictive representations are useful for supporting planning, generalisation and transfer in artificial intelligence (Boots et al., 2011; Lehnert & Littman, 2018), and for understanding structure learning in humans and other animals (Stachenfeld et al., 2017; Whittington et al., 2019). To understand how the brain achieves flexible inference of such predictive models would be of great value for cognitive science as well as for brain-inspired machine learning research. Here we focus on a predictive representation known as the Successor Representation (Dayan, 1993) – and its generalisation to function approximation known as Successor Features (SFs; Barreto et al., 2016). SFs generalise over stimuli that predict similar futures and can provide a useful balance between efficiency and flexibility. As in model-based (MB) algorithms, and in contrast to model-free (MF) algorithms, the separate representation of transition dynamics and reward allows for flexible re-evaluation of value in the face of changes to the reward function. Furthermore, unlike MB algorithms, evaluation using SFs does not require expensive forward simulation. However, SFs are worse than MB at handling changes in the environment's transition structure because they are based on caching long-run future predictions. In cognitive science and neuroscience, SFs offer a compelling explanation for a range of behavioural and neural findings (Momennejad et al., 2017; Stachenfeld et al., 2017; Gardner et al., 2018; Garvert et al., 2017; Bellmund et al., 2019).

While SFs offer a solution to some of the shortcomings of MF learning, existing methods for estimating SFs do not take into account uncertainty. Representing uncertainty explicitly is useful for multiple reasons: optimally combining prior and novel information, updating jointly predictions that covary and guiding exploration. Here, we incorporate uncertainty. into the SF framework by drawing on the Kalman Temporal Difference (KTD) method for value learning (Geist & Pietquin, 2010; Gershman, 2015). The resulting algorithm, Kalman SF, gives the agent an estimate of its uncertainty as well as the covariance between different features. We show how this augments the SF's capacity to support revaluation following changes in transition structure, and how it explains aspects of human decision making, as well as rodent behaviour during an optogenetics experiment.

2 RESULTS

2.1 SUCCESSOR FEATURES AND UNCERTAINTY

An RL environment is a Markov Decision Process consisting of *states* s the agent can occupy, *transition probabilities* $T_\pi(s'|s)$ of moving from state s to states s' given the agent's policy $\pi(a|s)$ over actions a, and the reward available at each state, for which $R(s)$ denotes the expectation. An RL agent is tasked with finding a policy that maximises its expected discounted total future reward, or *value*:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t\right) | s_0 = s \right] \tag{1}$$

where t indexes time step and $\gamma \in [0, 1)$ is a discount factor that down-weights distal rewards.

We assume that the reward can be expressed as a linear combination of the state features $\boldsymbol{\phi}(s_t)$ and the reward expectation per feature $\mathbf{u}$: $R(s) = \boldsymbol{\phi}(s)^T \mathbf{u}$. In these cases the value function can be decomposed into a product of the reward expectation $\mathbf{u}$ and the *Successor Features* $\boldsymbol{\psi}^\pi(s)$ (Barreto et al., 2016):

$$V(s) = \boldsymbol{\psi}^\pi(s)^T \mathbf{u}. \tag{2}$$

$\boldsymbol{\psi}_i$ is defined as the the expected discounted future sum of the occurrence of feature $\boldsymbol{\phi}_i$:

$$\boldsymbol{\psi}^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{\phi}(s_t) | s_0 = s \right] \tag{3}$$

We use linear function approximation $\boldsymbol{\psi}^\pi(s) = W^T \boldsymbol{\phi}(s)$, where W is a matrix parameterising the approximation. SFs are the extension of Dayan's (1993) Successor Representation (SR) to function approximation, and they reduce to the SR in the tabular case. The logic is the same: SFs represent each state in terms of the future (successor) states they predict under the current policy. Factorising value into an SF term and a reward term permits greater flexibility because if one term changes, it can be relearned while the other remains intact. Standard RL algorithms, such as TD learning, produce a point estimate of the SFs (see equation 9 in Supplement A for the standard TD SF update). While useful for approximating expected value, it is not capable of expressing uncertainty in these estimates. We therefore propose using Kalman Filtering (Kalman, 1960) to optimally handle uncertainty while learning the SFs. Applying the Kalman Temporal Differences (KTD) algorithm (Geist & Pietquin, 2010; Gershman, 2015) to SFs, we update the weight matrix W and covariance matrix Σ using the Kalman Filter equations:

$$W_{t+1} = W_t + \mathbf{k}_t (\boldsymbol{\delta}_t^{\boldsymbol{\psi}})^T \tag{4}$$

$$\Sigma_{t+1} = \Sigma_t + Z - \frac{\mathbf{k}_t \mathbf{k}_t^T}{\lambda_t} \tag{5}$$

where $\boldsymbol{\delta}_t^{\boldsymbol{\psi}} = \boldsymbol{\phi}(s_t) + \gamma \boldsymbol{\psi}(s_{t+1}) - \boldsymbol{\psi}(s_t)$ is the vector-valued *successor prediction error*, $\mathbf{k}_t = (\Sigma_t + Z)\mathbf{h}_t$ is an adaptive learning rate or Kalman gain, $\lambda_t = \mathbf{h}^T(\Sigma_t + Z)\mathbf{h}_t + \sigma_o^2$ is the residual variance, and $\mathbf{h}_t = \boldsymbol{\phi}(s_t) - \gamma \boldsymbol{\phi}(s_{t+1})$ is the discounted temporal derivative of the features – see Appendix A for details and parameter values. Crucially, the Kalman Filter takes into account the variance of and covariance between features when learning the SFs. This means that the agent can learn about features that are not currently present, as long as they show nonzero covariance with the current features. In the next sections, we show how this can be leveraged for flexible updates in the face of changes to the environment's transition structure.

2.2 PARTIAL TRANSITION REVALUATION SIMULATIONS

A key prediction of standard temporal difference SF learning is that "reward revaluation" (changes in the reward function) should be easy to transfer to while "transition revaluation" (changes in the transition dynamics) should not. Momennejad et al. (2017) tested whether or not this is the case in human learning. In the first phase of their experiment, participants learned two different sequences of states terminating in different reward amounts: 2→4→6→\$1 and 1→3→5→\$10 (see Figure 1A). In the next stage, half of the participants were exposed to the transition revaluation condition, observing novel transitions 4→5→\$10 and 3→6→\$1. The other half experienced "reward revaluation" in the

form of novel reward amounts. 6→\$10 and 5→\$1 (Figure 1A). Importantly, the novel experiences start from intermediate states such that transitions from 1 or 2 are not seen following phase 1. While participants were significantly better at reward revaluation than transition revaluation, they were capable of some transition revaluation as well (Figure 1C). Accordingly, the authors proposed a hybrid SF model: an SF-TD agent that is also endowed with capacity for replaying experienced transitions, permitting updating the SF vectors of states 1 and 2 through simulated experience (Figure 1F).

We simulated this experiment and found that the Kalman SF accounts for partial transition revaluation even without replay (Figure 1C). Kalman SF correctly learns the SF matrix after phase 1 (Figure 2 in the Supplementary Material) as well as an estimate of the covariance between features, Σ. Unlike TD-SF, Kalman SF uses the covariance matrix to estimate the Kalman gain and uses that to update the whole matrix. This means that after seeing $3 \rightarrow 6$, it updates not just $\psi(3)$ but also $\psi(1)$ because these entries have historically covaried (same for $\psi(4)$ and $\psi(2)$) (Figure 2). To estimate direct reward $\hat{r}$, the agent uses a delta rule (Rescorla & Wagner, 1972). Model parameters are listed in Supplement A, Table 1, and experimental parameters are kept the same as in Momennejad et al. (2017).

Kalman SF thus provides an alternative explanation of the participants' choices, without the need for the memory buffer and computation time that replay requires. However, the updates in Kalman SF are on-policy, meaning that the algorithm will do worse in off-policy planning problems, which could prove to be a way to distinguish between the Hybrid and Kalman SF models.

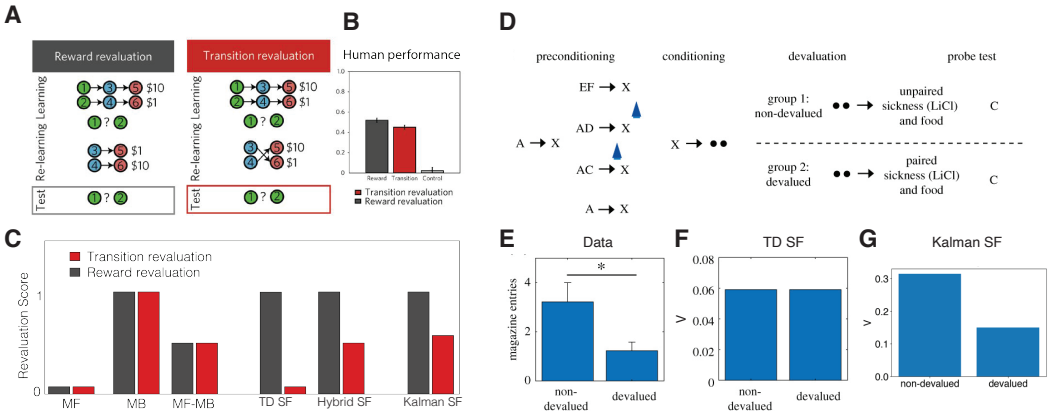

Figure 1: Kalman SF performance on transition and reward revaluation experiments. (A) Task structure for reward revaluation and transition revaluation experiments. (B) Human performance on transition and reward revaluation tasks. (C) Model predictions for classic model-free, model-based or a hybrid of model free and model-based algorithms, TD-SF, hybrid SF and Kalman SF. (D) Preconditioning and optogenetic unblocking paradigm designed by Sharpe et al. (2017). Blue light cones indicate optogenetic activation of dopamine. (E) behaviour of rodents to preconditioned cue that is unblocked by activation of dopamine neurons is sensitive to devaluation of the predicted reward. (F) vanilla SF model is insensitive to reward devaluation in this paradigm. (G) Kalman SF is sensitive to reward devaluation. Panels A– B reprinted with permission from Momennejad et al. (2017). Panels D-F from Gardner et al. (2018).

2.3 Simulating dopamine-dependent reward devaluation sensitivity

The fact that the vanilla TD SF cannot acquire state transitions that are not directly experienced can also impair behaviour in the context of associative learning. To illustrate this, consider the experiment shown in Figure 1D, designed by Sharpe et al. (2017) to show that the sensitivity to reward devaluation, a hallmark of model-based learning, is dependent on dopamine transients. In this experiment, animals started with two preconditioning phases, where they learned associations between nonrewarding stimuli ($A \rightarrow X$ and $AC \rightarrow X$). Normally, the $A \rightarrow X$ association

impairs learning of the $C \rightarrow X$ association – a phenomenon known as *blocking* (Kamin, 1967) – but the authors *unblocked* this learning by activating dopamine neurons using optogenetics during preconditioning. X was then paired with a reward, after which the reward was devalued by pairing it with sickness for half of the animals.

The key feature of this experiment is that the food reward was paired with illness in the absence of any of the lettered stimuli introduced in the preconditioning stage. Thus, unlike the animals, a vanilla SF agent is not sensitive to the reward devaluation (c.f. Figures 1E and 1F) (Gardner et al., 2018). This is because in the vanilla SF, only stimuli that directly predict reward will change value after devaluation. In this paradigm, however, C was never directly associated with food. Hence, any algorithm that only updates associations with the currently active features will be insensitive to devaluation.

We simulated this task using Kalman SF and found that, like the animals in Sharpe et al. (2017), Kalman SF was sensitive to the reward devaluation paradigm. Like in the transition revaluation task, this is because Kalman SF estimates a covariance between features, and uses this for a non-local upate of SFs corresponding to features that are not currently active. Specifically, during the pre-conditioning phase, a positive covariance between C and X is learned, which means that during conditioning, C becomes directly associated to the food. Subsequent devaluation thus directly affects C as well as X.

3 Discussion

Successor Features constitute a middle ground between MB and MF RL algorithms by separating reward representations from long-run state predictions. Here we learn a probabilistic SF model that supports principled handling of uncertainty about state predictions and inter-dependencies between these predictions. We exploit this feature to show that, unlike standard TD-SF, Kalman SF can perform partial transition revaluation. In later work, we plan to test our model on other tasks that could benefit from Kalman SF in a similar way, such as policy revaluation (a well-known weak spot of TD-SR; Lehnert et al., 2017).

For both experiments modelled here, an alternative explanation would be that subjects used offline replay to update the SFs. We therefore note the relative strengths and weaknesses of Kalman SF when compared to this hybrid-SF approach. Replay requires a buffer to store experienced episodes and a sufficient number of replays that information is propagated throughout the SF model. While Kalman SF can incorporate information about long-range in a single update, it must store a covariance matrix (although dimensionality reduction can reduce this burden; Fisher, 1998), and updates are on-policy. There is compelling evidence in favor of both replay (Carr et al., 2011; Ólafsdóttir et al., 2018) and probabilistic representations (Ma et al., 2006). Future work will consider how the relative tradeoffs of these approaches constrain hypotheses.

We made several assumptions. The Gaussian assumption is clearly violated in the case of one-hot state vectors. However, the model is sufficiently expressive that a good approximation can still be found, and the Kalman SF model could be applied over arbitrary features for which the assumption might hold. The random walk process noise might be ill-suited for step changes or sub-optimal when the dynamics are predictable (Didier & Kayo, 2001). While we assume deterministic transitions and linear function approximation here, it is straightforward to extend KTD to stochastic transitions and nonlinear function approximation (Geist & Pietquin, 2010).

Probabilistic models provide a number of advantages for RL in terms of optimal credit assignment (Kruschke, 2008) and uncertainty-minimising exploration (Dearden et al., 1998). Distributional RL-trained neural network agents achieve state of the art performance (Bellemare & Dabney, 2017). Furthermore, a range of animal learning findings suggest that animals are capable of probabilistic reasoning (Gershman, 2015; Kruschke, 2008; Courville et al., 2006). Future work will involve exploring these advantages in the context of SR learning.

Acknowledgments

This work is funded by the Gatsby Foundation and the Wellcome Trust. We thank Samuel Gershman, Eszter Vértes, Talfan Evans, Steven Hansen and Matthew Botvinick for helpful comments.

REFERENCES

Andre Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Zidek, and Remi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. *35th International Conference on Machine Learning, ICML 2018*, 2:844–853, 2018.

André Barreto, Rémi Munos, Tom Schaul, and David Silver. Successor Features for Transfer in Reinforcement Learning. *arXiv*, pp. 1–13, 2016. URL http://arxiv.org/abs/1606.05312.

Marc G Bellemare and Will Dabney. A Distributional Perspective on Reinforcement Learning. 2017.

Jacob L S Bellmund, William de Cothi, Tom A Ruiter, Matthias Nau, Caswell Barry, and Christian F Doeller. Deforming the metric of cognitive maps distorts memory. *Nature Human Behaviour*, 2019. ISSN 2397-3374. doi: 10.1038/s41562-019-0767-3. URL https://doi.org/10.1038/s41562-019-0767-3.

Byron Boots, Sajid MSiddiqi, and Geoffrey J. Gordon. Closing the learning-planning loop with predictive state representations. *International Journal of Robotics Research*, 30(7):954–966, 2011. ISSN 02783649. doi: 10.1177/0278364911404092.

Margaret F. Carr, Shantanu P. Jadhav, and Loren M. Frank. Hippocampal replay in the awake state: A potential substrate for memory consolidation and retrieval. *Nature Neuroscience*, 14(2):147–153, 2011. ISSN 10976256.

Aaron C. Courville, Nathaniel D. Daw, and David S. Touretzky. Bayesian Theories of Conditioning in a Changing World. *Trends in Cognitive Sciences*, 10(7):294–300, 2006. ISSN 13646613. doi: 10.1016/j.tics.2006.05.004.

Peter Dayan. Improving Generalisation for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 5(4):613–624, 1993. ISSN 0899-7667. doi: 10.1162/neco.1993.5.4.613.

Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian Q-Learning. *AAAI/IAAI*, 1998.

Sornette Didier and Ide Kayo. The Kalman-Levy Filter. *Physica D*, 151:142–174, 2001.

Michael Fisher. *Development of a simplified Kalman filter*. European Centre for Medium-Range Weather Forecasts, 1998.

Matthew P.H. Gardner, Geoffrey Schoenbaum, and Samuel J. Gershman. Rethinking Dopamine as Generalized Prediction Error. *Proceedings of the Royal Society B: Biological Sciences*, 285 (1891), 2018. ISSN 14712954. doi: 10.1098/rspb.2018.1645.

Mona M. Garvert, Raymond J. Dolan, and Timothy E.J. Behrens. A Map of Abstract Relational Knowledge in the Human Hippocampal-Entorhinal Cortex. *eLife*, 6:1–20, 2017. ISSN 2050084X. doi: 10.7554/eLife.17086.

Matthieu Geist and Olivier Pietquin. Kalman temporal differences. *Journal of Artificial Intelligence Research*, 39:483–532, 2010. ISSN 10769757. doi: 10.1613/jair.3077.

Samuel J. Gershman. A Unifying Probabilistic View of Associative Learning. *PLOS Computational Biology*, 11(11):e1004567, 11 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004567. URL https://dx.plos.org/10.1371/journal.pcbi.1004567.

Samuel J. Gershman. Dopamine, Inference and Uncertainty. *Neural Computation*, 29:3311–3326, 2017. ISSN 1221146X. doi: 10.1162/neco.

Samuel J Gershman. The Successor Representation: Its Computational Logic and Neural Substrates. *The Journal of Neuroscience*, 38(33):7193–7200, 8 2018. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.0151-18.2018. URL http://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0151-18.2018.

R E Kalman. A New Approach to Linear Filtering and Prediction Problems 1. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Leon J Kamin. Predictability, surprise, attention, and conditioning. 1967.

John K. Kruschke. Bayesian Approaches to Associative Learning: From Passive to Active Learning. *Learning and Behavior*, 36(3):210–226, 2008. ISSN 15434494. doi: 10.3758/LB.36.3.210.

Lucas Lehnert and Michael L. Littman. Transfer with Model Features in Reinforcement Learning. 2018. URL http://arxiv.org/abs/1807.01736.

Lucas Lehnert, Stefanie Tellex, and Michael L. Littman. Advantages and Limitations of using Successor Features for Transfer in Reinforcement Learning. 2017. URL http://arxiv.org/abs/1708.00102.

Wei Ji Ma, Jeffrey M. Beck, Peter E. Latham, and Alexandre Pouget. Bayesian Inference With Probabilistic Population Codes. *Nature Neuroscience*, 9(11):1432–1438, 2006. ISSN 10976256. doi: 10.1038/nn1790.

I. Momennejad, E. M. Russek, J. H. Cheong, M. M. Botvinick, N. D. Daw, and S. J. Gershman. The Successor Representation in Human Reinforcement Learning. *Nature Human Behaviour*, 1(9): 680–692, 2017. ISSN 23973374.

H. Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The Role of Hippocampal Replay in Memory and Planning. *Current Biology*, 28(1):R37–R50, 2018. ISSN 09609822.

Robert A Rescorla and Allan R Wagner. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, 2:64–99, 1972.

Evan M Russek, Ida Momennejad, Matthew M Botvinick, and Samuel J Gershman. Predictive Representations Can Link Model-Based Reinforcement Learning to Model-Free Mechanisms. *PLoS Computational Biology*, pp. 1–42, 2017.

Melissa J. Sharpe, Chun Yun Chang, Melissa A. Liu, Hannah M. Batchelor, Lauren E. Mueller, Joshua L. Jones, Yael Niv, and Geoffrey Schoenbaum. Dopamine transients are sufficient and necessary for acquisition of model-based associations. *Nature Neuroscience*, 20(5):735–742, 2017. ISSN 15461726. doi: 10.1038/nn.4538.

Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The Hippocampus as a Predictive Map. *Nature Neuroscience*, 20(11):1643–1653, 2017. ISSN 15461726. doi: 10.1038/nn.4650.

James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The Tolman-Eichenbaum Machine: Unifying space and relational memory through generalisation in the hippocampal formation. *bioRxiv*, pp. 770495, 2019. doi: 10.1101/770495. URL https://www.biorxiv.org/content/10.1101/770495v1.

A Supplementary Materials

A.1 Learning Successor Features

The SFs ψ^π can be seen as a summary of the dynamics of the environment under the current policy π. This allows for a *factorised* representation of V^π in which the environment's dynamics are decoupled from the reward expectation $\mathbf{u}$. One advantage of such a factorised representation is that, when either the rewards or the dynamics change, only one of the modules needs to be relearned (Dayan, 1993). Another consequence is that the agent now has two terms to learn: ψ^π and $\mathbf{u}$. In some cases, ϕ is also learned from data (e.g. Barreto et al., 2018), but here we assume that ϕ is given. In this section, we describe the standard temporal difference method for learning a point estimate, and how this can be adapted to incorporate uncertainty.

Name	Symbol	Value
Discount factor	γ	0.9
Process covariance	Z	$(1 \times 10^{-3})I$
Observation variance	σ_ϕ^2	1
Prior covariance	Σ_0	$0.1I$
Prior SF matrix	W_0	I
Reward learning rate	α_r	0.1

Table 1: Parameter values

Firstly, note that $R(s) = \boldsymbol{\phi}(s)^T \mathbf{u}$ is a supervised learning problem, and can be solved with a simple delta rule:

$$\hat{\mathbf{u}}_{t+1} = \hat{\mathbf{u}}_t + \alpha_r(R(s_t) - \boldsymbol{\phi}(s_t)^T \hat{\mathbf{u}}_t)\boldsymbol{\phi}(s_t), \tag{6}$$

where α_r is a learning rate and $\delta^R \equiv R(s_t) - \boldsymbol{\phi}(s_t)^T \hat{\mathbf{u}}_t$ is the reward prediction error.

For learning the SFs, we note that, like value, they satisfy a Bellman equation, recursively relating the SFs of subsequent states:

$$\boldsymbol{\psi}^\pi(s) = \boldsymbol{\phi}(s) + \gamma \mathbb{E}_{s'} \left[\boldsymbol{\psi}^\pi(s') | s_o = s \right], \tag{7}$$

which means that in principle *any* RL method can be used to learn $\boldsymbol{\psi}^\pi(s)$. Here we assume linear function approximation:

$$\boldsymbol{\psi}^\pi(s) = W^T \boldsymbol{\phi}(s), \tag{8}$$

where W is a matrix parameterising the approximation. Intuitively, W encodes how much each feature predicts every other feature. Combining this with temporal difference learning leads to the following SF-TD weight update:

$$W_{t+1} = W_t + \alpha^\psi \boldsymbol{\phi}(s_t)(\boldsymbol{\delta}_t^\psi)^T \tag{9}$$

where α^ψ is a scalar learning rate and $\boldsymbol{\delta}_t^\psi$ is the vector-valued *successor prediction error*:

$$\boldsymbol{\delta}_t^\psi = \boldsymbol{\phi}(s_t) + \gamma \boldsymbol{\psi}(s_{t+1}) - \boldsymbol{\psi}(s_t) \tag{10}$$

encoding surprise about the occurence of each feature.

It can be easily seen that, in the *tabular* case (where discrete states are encoded with one-hot feature vectors $\boldsymbol{\phi}$) this reduces to the Successor Representation originally described by (Dayan, 1993). Furthermore, the learning rule in equation 9 will only update the features currently active in state s_t, which, in the tabular case, will mean that only the row in W that corresponds to the current state will be updated. We refer to such updates as *local*. Exclusively local updates lead to problems when something in the transition structure changes because SFs rely on caching long-run state estimates (Russek et al., 2017). Hence, local changes in the transition structure require non-local updates to the SF matrix.

A.2 Kalman Successor Features

Algorithm 1: Kalman Successor Features

Initialization: priors W_0 and Σ_0 ;
for $t \leftarrow 1, 2, ...$ **do**
  Observe transition (s_t, s_{t+1}) ;
  Compute statistics of interest ;
  $\lambda_t = \mathbf{h}^T(\Sigma_t + Z)\mathbf{h}_t + \sigma_\phi^2$;
  $\mathbf{k}_t = (\Sigma_t + Z)\mathbf{h}_t$;
  $\boldsymbol{\delta}_t^\psi = \boldsymbol{\phi}(s_t) + \gamma\boldsymbol{\psi}(s_{t+1}) - \boldsymbol{\psi}(s_t)$;
  Correction step ;
  $W_{t+1} = W_t + \mathbf{k}_t(\boldsymbol{\delta}_t^\psi)^T$;
  $\Sigma_{t+1} = \Sigma_t + Z - \frac{\mathbf{k}_t \mathbf{k}_t^T}{\lambda_t}$
end

To alleviate this problem, we draw on the Kalman Temporal Differences (KTD) method developed by Geist & Pietquin (2010), which combines Kalman Filtering Kalman (1960) and temporal difference learning to optimally handle uncertainty while learning the value function. Gershman has

shown that KTD captures a range of phenomena in animal behaviour (2015) and dopamine responses (2017) during associative learning.

Applying these ideas to SFs, we assume that there is an underlying, hidden Successor Representation weight matrix W, of which state observations $\boldsymbol{\phi}$ are noisy observations. The probabilistic model consists of an *evolution equation* describing the evolution of the hidden weights, and an *observation equation* describing how the hidden SF relates to the observations.

The evolution equation describes the evolution of the weights as a random walk:

$$W_t = W_{t-1} + \mathbf{n}_Z \tag{11}$$

where $\mathbf{n}_Z \sim \mathcal{N}(\mathbf{0}, Z)$ is the process or evolution noise, which is Gaussian with a diagonal covariance matrix Z.

The observation equation describes how the observations relate to the hidden SF:

$$\boldsymbol{\phi}(s_t) = W_t^T \mathbf{h}_t + n_\phi \tag{12}$$

where we have defined $\mathbf{h}_t = \boldsymbol{\phi}(s_t) - \gamma \boldsymbol{\phi}(s_{t+1})$ as the discounted temporal derivative of the features (see Geist & Pietquin, 2010), and $n_\phi \sim \mathcal{N}(0, \sigma_\phi)$ is the observation noise, which is drawn from a one-dimensional Gaussian with variance σ_ϕ.

We are thus faced with the problem of tracking the most likely W_t given the sequence of previous state observations $\boldsymbol{\phi}_{1\ldots t}$. Applying the Kalman Filter equations (Kalman, 1960) to this inference problem, we update the weight matrix W and covariance matrix Σ as follows:

$$W_{t+1} = W_t + \mathbf{k}_t (\boldsymbol{\delta}_t^{\boldsymbol{\psi}})^T \tag{13}$$

$$\Sigma_{t+1} = \Sigma_t + Z - \frac{\mathbf{k}_t \mathbf{k}_t^T}{\lambda_t} \tag{14}$$

where $\boldsymbol{\delta}_t^{\boldsymbol{\psi}}$ is the *successor prediction error* from equation 10,

$$\mathbf{k}_t = (\Sigma_t + Z)\mathbf{h}_t \tag{15}$$

is the Kalman gain, an adaptive, feature-specific learning rate. Crucially, as shown in equation 15, the Kalman gain is dependent on the covariance between features given by Σ. This means that agents can learn simultaneously about features that covary, even when those features are not currently present, which is a feature that vanilla SF learning methods lack (Russek et al., 2017; Gardner et al., 2018; Gershman, 2018).

Finally, the residual variance is given by $\lambda_t = \mathbf{h}^T(\Sigma_t + Z)\mathbf{h}_t + \sigma_\phi^2$. The Kalman SF algorithm is listed in Algorithm 1, and parameter values are given in Table 1.

A.3 Additional information

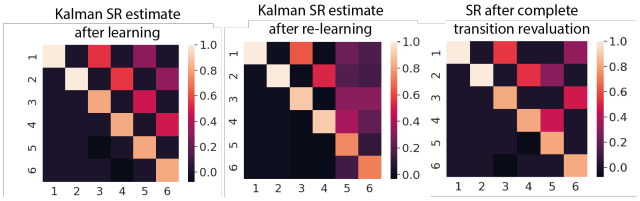

Figure 2: The SF matrix estimated by Kalman SF in the experiment of Momennejad et al. (2017) after (left) learning (phase 1), (middle) re-learning (phase 2) and (right) as it would be after complete transition revaluation.