A Baseline of Discovery for General Value Function Networks under Partial Observability.

Matthew Schlegel, Adam White, Martha White
University of Alberta
{mkschleg, amw8, whitem}@ualberta.ca

1 Introduction

A reinforcement learning agent's representation of its world is critical to maximizing its return, sum of discounted reward, in an environment. This is particularly important under partial observablity where the observations of the environment are typically insufficient for learning. There have been several mechanisms developed to build representation, or state, under the constraints of partial observability. One method is to keep a history, where the environment is assumed markov given sufficient length of history. The agent can also maintain a distribution over a proposed set of states as in a partially observable markov decision process (POMDP). Another promising approach is to build state through a predictive representation, as in Predictive State Representations (PSR) [1, 10], Temporal-Difference (TD) networks [11], and General Value Function (GVF) networks [8]. A representation built from predictions is appealing because each predictive unit can be learned through interactions with the world, and the units naturally provide temporally extended meaning to the representation. A predictive representation can also expand and modify the set of predictive units, allowing the agent to continually improve its understanding of the world. GVF networks are particularly appealing because they propose a familiar language for defining predictive questions, and they can be trained using standard reinforcement learning algorithms [12, 14].

A common problem in using predictive representations is in the specification of the predictive units. A discovery system addresses this problem directly, by enabling an agent to propose predictive units independently of an expert. There are several characteristics of an effective approach to discovery such as maintaining stability of the current representation, effectively removing dysfunctional or not learnable predictive units, and providing a representation which is generalizable to use with new tasks. There have been several approaches to discovering core tests in a PSR [4, 16], and the nodes of a TD network [3]. Extensions of these approaches to GVF networks should be considered in future work.

In this work we describe a general framework for discovery in GVF networks, and define a simple variant to act as a baseline for future work. We also provide a demonstration in compass world, a partially observable domain, to evaluate the components of the system.

2 Background

The dynamics of the environment are modeled as a markov decision process with state space $\mathcal{S} \in \mathbb{R}^d$, actions $\mathcal{A} \in \mathbb{R}^b$, and transition probabilities $P = \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, \infty)$. On each step the agent is presented with an observation $\mathbf{o}_t \in \mathcal{O} \subset \mathbb{R}^d$, which are determined by a lossy function of the underlying state $\mathbf{o}_t = \mathbf{o}(s_t), s_t \in \mathcal{S}$. The agent takes an action $a_t \in \mathcal{A}$ given an observation and receives a new observation corresponding to the new underlying state. The goal of an agent under partial observability is to find a state representation $h_t \in \mathbb{R}^n$ through interactions with the environment to enable the learning of other tasks.

We use the generalized form of value functions [13, 5] as the base predictive units of a predictive representation. A general value function (GVF) [13, 14] is defined by a cumulant, policy, and continuation function. The cumulant $c \in \mathbb{R}$ can be any signal (internal or external) available to the agent. We use the usual definition of a policy where $\pi(a_t|o_t) \in [0, 1]$ is the probability of

Preprint. Work in progress.

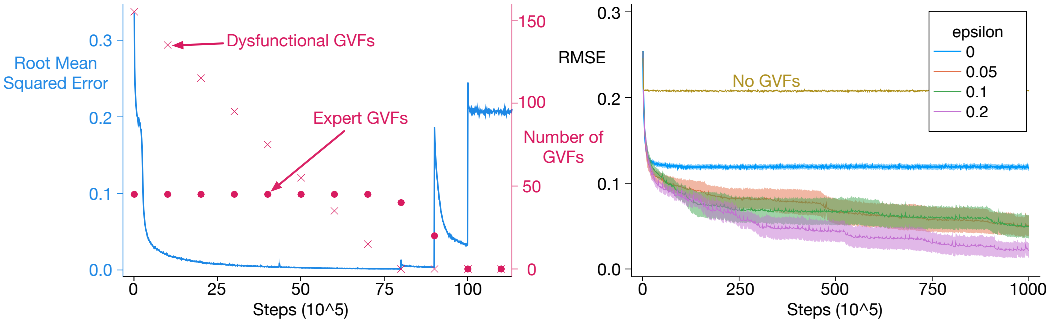

Figure 1: **(left)** Pruning predictive units occurs every million steps with no regeneration $\alpha = 0.001, \lambda = 0.9, \epsilon = 0.1, \sigma^2 = 1$ **(right)** Learning curves of the evaluative GVFs $N = 1000000, \epsilon = 0.1, \alpha = 0.001, \lambda = 0.9, n = 100$.

taking an action a_t given the current observations o_t. The continuation function is a function $\gamma_t = f_\gamma(o_t, a_t, o_{t+1}) \in [0, 1]$ [15] determining the horizon (or temporal distance) of the prediction. A GVF network [8] is a network of n interconnected GVFs that recurrently use previous predictions as a form of state. We can define the state update function $h_t = \sigma(\boldsymbol{\theta}_t[\mathbf{o}_t, h_{t-1}]^\top) \in \mathbb{R}^n$, where $\boldsymbol{\theta} \in \mathbb{R}^{nxm}$ is the weight matrix with each row representing a specific GVF. There are several algorithms which can be used to learn the weights of a GVF network ranging from simple TD(0) to a fully recurrent gradient algorithm using back propagation through time [9, 8]. Here we simply use off-policy semi-gradient TD(λ).

3 A framework for discovery in GVF Networks

We base the structure of our framework for discovery on prior methods of representation search [2] focusing on two main components: an evaluator, and an generator. The evaluator determines which predictions should be removed to make room for new proposals, and determines when predictions should be tested. The generator is responsible for proposing new questions for the predictive representation. These components act in a cycle, continuously proposing new questions and pruning infrequently used or not learnable predictions. One variant of the two aforementioned components is a random generator, and an evaluator based on magnitude of weights. The generator proposes new GVFs from a set of policies, cumulants, and continuation primitives. These primitives are described fully in appendix A.2. The evaluator measures the usefulness of a GVF based on the magnitude of its corresponding weight in the external tasks. We evaluate all the predictive units every $N \in \mathbb{N}$ steps and prune the $0 \leq n\epsilon \leq n$ least useful GVFs where $\epsilon \in [0, 1]$. Pseudo code for this evaluator can be found in appendix A.1.

4 Experiments

We evaluate the performance of our system on two experiments in compass world [7]. Both experiments use five evaluative GVFs that are not learnable through the observations. These questions correspond to a question of "which wall will I hit if I move forward forever?" The first experiment, figure 1 (left), provides a check to ensure the evaluation strategy targets dysfunctional representational units for removal. We initialize the GVF network with 200 GVFs: 45 used to form the expert crafted TD network [7], and 155 defective GVFs predicting noise $\sim \mathcal{N}(0, \sigma^2)$. We report the learning curve and pruned GVFs over 12 million steps. The second experiment, figure 1 (right), uses the full discovery approach to find a representation useful for predicting the evaluation GVFs. We report the learning curves of the evaluative GVFs over 100 million steps.

While this baseline performs surprisingly well in the demonstrations, there is ample room for improvement. Primarily, the random generation strategy does not take into account the current set of proposed predictions, potentially resulting in redundancy. A more principled method would look to generate a wide variety of predictions dependent on the current set of predictions, proposing a diverse set of predictive units. Unfortunately, measuring how related questions are from their specification is not particularly straightforward. Another issue is the proxy used to determine a predictions usefulness. Currently, the system will potentially cut predictions which are useful for the internal representation. This could harm of the predictive state or cause other instabilities within the GVF network.

References

[1] Michael L Littman, Richard S Sutton, and S Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, 2001.

[2] Ashique Rupam Mahmood and Richard S Sutton. Representation Search through Generate and Test. In *AAAI Workshop: Learning Rich Representations from Low-Level Sensors*, 2013.

[3] Takaki Makino and Toshihisa Takagi. On-line Discovery of Temporal-difference Networks. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

[4] Peter Mccracken and Michael Bowling. Online Discovery and Learning of Predictive State Representations. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, 2006.

[5] Joseph Modayil, Adam White, and Richard S Sutton. Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior*, 22(2):146–160, 2014.

[6] Eddie Rafols, Anna Koop, and Richard S Sutton. Temporal Abstraction in Temporal-difference Networks. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, 2006.

[7] Eddie J. Rafols, Mark B. Ring, Richard S. Sutton, and Brian Tanner. Using Predictive Representations to Improve Generalization in Reinforcement Learning. In *IJCAI*, 2005.

[8] Matthew Schlegel, Adam White, Andrew Patterson, and Martha White. General Value Function Networks. *arXiv:1807.06763 [cs, stat]*, 2018.

[9] David Silver. Gradient Temporal Difference Networks. In *European Workshop on Reinforcement Learning*, 2013.

[10] S Singh, M.L. Littman, N K Jong, D Pardoe, and P Stone. Learning predictive state representations. In *International Conference on Machine Learning*, 2003.

[11] Richard S Sutton and Brian Tanner. Temporal-Difference Networks. In *Advances in Neural Information Processing Systems*, 2004.

[12] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009.

[13] Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. Horde: A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[14] Adam White. Developing a predictive approach to knowledge. *University of Alberta*, 2015.

[15] Martha White. Unifying Task Specification in Reinforcement Learning. In *International Conference on Machine Learning*, 2017.

[16] David Wingate and Satinder Singh. On Discovery and Learning of Models with Predictive Representations of State for Agents with Continuous Actions and Observations. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, New York, NY, USA, 2007. ACM. ISBN 978-81-904262-7-5.

A Approach to Discovery

A.1 Evaluation

To evaluate the usefulness of a predictive unit we look at the magnitude of the associated weight in the tasks using the predictive representation. See algorithm 1 for the pseudo code.

Algorithm 1 Evaluator

```
Input: $\theta \in \mathbb{R}, k =$ number of tasks, $\epsilon \in [0, 1], n =$ number of predictive units
$\bar{\theta} = \sum_{i=1}^{\mathrm{k}} |\theta_i|; \bar{\theta} \in \mathbb{R}^m$
for $i < n\epsilon$ do
    $j = \mathrm{argmin}\{\bar{\theta}\}$
    if $j$ is associated with a GVF $\in$ GVF Network then
        prune GVF
```

A.2 GVF Primitives

To enable generation of GVFs for this discovery approach, we introduce *GVF primitives*. The goal is to provide modular components that can be combined to produce different structures. For example, within neural networks, it is common to modularly swap different activation functions, such as sigmoidal or tanh activations. For networks of GVFs, we similarly need these basic units to enable definition of the structure.

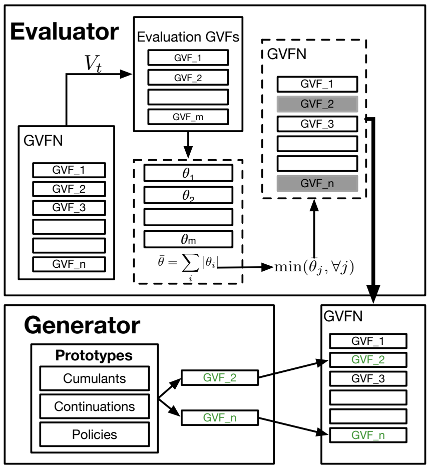

Figure 2: Visualization of discovery approach. (*Evaluator*) takes the weights of the evaluation GVFs and measures the usefulness on weight magnitude. (*Generator*) Randomly generates new GVFs from a set of prototypical parameter functions.

We propose basic types for each component of the GVF: discount, cumulant and policy. For discounts, we consider *myopic discounts* ($\gamma = 0$), *horizon discounts* ($\gamma \in (0, 1)$) and *termination discounts* (the discount is set to $\gamma \in (0, 1]$ everywhere, except for at an event, which consists of a transition (o, a, o')). For cumulants, we consider *stimuli cumulants* (the cumulant is one of the observations, or inverted, where the cumulant is zero until an observation goes above a threshold) and *compositional cumulants* (the cumulant is the prediction of another GVF). We also investigate *random cumulants* (the cumulant is a random number generated from a zero-mean Gaussian with a random variance sampled from a uniform distribution); we do not expect these to be useful, but they provide a baseline. For the policies, we propose *random policies* (an action is chosen at random) and *persistent policies* (always follows one action). For example, a GVF could consist of a myopic discount, with stimuli cumulant on observation bit one and a random policy. This would correspond to predicting the first component of the observation vector on the next step, assuming a random action is taken. As another example, a GVF could consist of a termination discount, an inverted stimuli cumulant for observation one and a persistent policy with action forward. If observation can only be '0' or '1', this GVF corresponds to predicting the probability of seeing observation one changing to '0' (inactive) from '1' (active), given the agent persistently drives forward.

B Experiments on discovering GVF networks in Compass World

We conduct experiments on our discovery approach for GVF networks in Compass World [6], a partially observable grid-world where the agent can only see the colour immediately in front of it.

There are four walls, with different colours; the agent observes this colour if it takes the action forward in front of the wall. Otherwise, the agent just sees white. There are five colours in total, with one wall having two colours and so more difficult to predict. The observation vector is five-dimensional, consisting of an indicator bit if the colour is observed or not.

The GVFs for the network are generated uniformly randomly from the set of GVF primitives. Because the observations are all one bit (0 or 1), the stimuli cumulants are generated by selecting a bit index i (1 to 5) and then either setting the cumulant to that observation value, o_i, or to the inverse of that value, $1 - o_i$. The events for termination are similarly randomly generated, with the event corresponding to a bit o_i flipping. The nonlinear transformation used for this GVF network is a clipping function. Every two million steps, the bottom 10% of the current GVFs are pruned and replaced with newly generated GVFs. Results are averaged over 5 runs.

Figure 1 (right) demonstrates that TD(λ) with randomly generated GVF primitives learns a GVF network—and corresponding predictive representation—that can accurately predict the five evaluative GVFs. The results show that the discovery approach is continually making the representation better for the evaluative GVFs. The discovered predictions outperform an initial random set, and significantly outperform a representation with no predictions.

B.1 Pruning random GVFs in Compass World

As we can see in figure 1 (left), TD(λ) does remove the dysfunctional GVFs first, and when the expert GVFs are pruned the representation isn't significantly damaged until the penultimate prune. These results also show how pruning dysfunctional or unused GVFs from a representation is not harmful to the learning task. The instability seen in the ends of learning can be overcome by allowing the system to generate new GVFs to replace those that were pruned and by pruning a small amount based on the size of network used as a representation.