Reasoning and Learning in Real-World Systems for Long-Term Autonomy (LTA)
Papers from the AAAI 2018 Fall Symposium

Evaluating Predictive Knowledge

Alex Kearney, Anna Koop, Craig Sherstan, Johannes Günther
Richard S. Sutton, Patrick M. Pilarski, Matthew E. Taylor
Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada;
kearney@ualberta.ca

Abstract

Predictive Knowledge (PK) is a group of approaches to machine perception and knowledgability using large collections of predictions made online in real-time through interaction with the environment. Determining how well a collection of predictions captures the relevant dynamics of the environment remains an open challenge. In this paper, we introduce specifications for sensorimotor baselines and robustness-to-transfer metrics for evaluation of PK. We illustrate the use of these metrics by comparing variant architectures of General Value Function (GVF) networks.

Predictive Knowledge

A key challenge for machine intelligence is that of representation: a system's performance is tied to its ability to perceive and represent its environment. Predictive knowledge representations use large collections of predictions to model the environment. An agent continually anticipates its sensation from its environment by making many predictions about the dynamics of its environment with respect to its behaviour (Modayil, White, and Sutton 2014). These predictions about expected sensation can then be used to inform an agent's internal representation of its environment (Littman and Sutton 2002). Other proposals describe inter-relations of predictions, similar to TD Networks (Tanner and Sutton 2005; Makino and Takagi 2008) to enable abstract, conceptual representations by making predictions of predictions (Schapire and Rivest 1988).

In this paper we discuss the subtleties of evaluation predictive representation and propose two complimentary techniques. We specifically consider PK methods that 1) are able to expand their representations by proposing new predictions, 2) are able to self-verify their predictions through interaction with their environment, and 3) are able continually learn their predictions on-line.

To examine these evaluation metrics we use the General Value Function framework for predictive representations (White 2015). GVFs estimate the expected discounted return of a signal C defined as $G_t = \sum_{k=0}^{\infty}(\prod_{j=1}^{k}(\gamma_{t+j}))C_{t+k+1}$. Value is estimated with respect to a specific policy π, discount function γ, and cumulant c: $v(s;\pi,\gamma,c) = \mathbb{E}_\pi[G_t|S_t = s]$.

The parameters c, π, and γ are the *question parameters* which specify what a GVF is about; the *answer parameters*—such as the step size α and eligibility decay λ—describe how a learning method learns to answer the GVF question. GVFs can be learnt online, incrementally through methods such as Temporal-difference (TD) learning (Sutton 1988). The representational power of a given GVF network depends not just on the quality of the answers, but also in the architecture of the network, as illustrated in Figure 1.

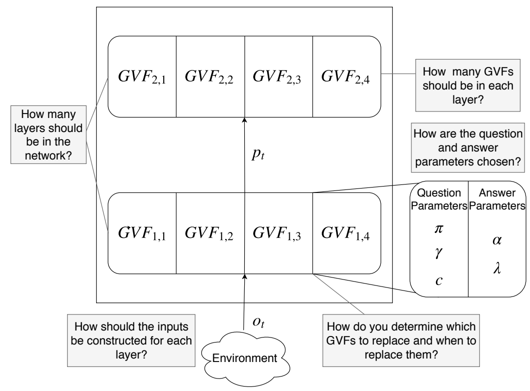

Figure 1: Many of the decisions which specify a PK architecture.

PK systems have been shown to be a scalable way to update and verify an agent's representation of the world, with examples of real-world robotic prediction tasks making thousands or tens of thousands of predictions in real-time on consumer-grade devices (Sutton et al. 2011; White 2015; Pilarski and Sherstan 2016).

Evaluating PK Architectures

Existing evaluation metrics for PK fall into two categories: 1) reporting the **average** error over all predictions within the PK system and 2) reporting errors on a known, challenging **subset** of the predictions within the system. Reporting the average error penalizes the accuracy of every prediction equally, when some predictions may have high error (such as for inherently random signals) but still provide representational power. Conversely, a representation that makes irrelevant but constant predictions will perform well according to average error, while providing no useful signals. Reporting errors on a subset of predictions requires identification of said subset across all architectures and lends itself to overfitting for those particular questions. It is difficult to identify predictions of interest without biasing towards particular architectures or network structures. Identifying predictions that require more complex representations in real-world settings requires extensive domain knowledge. In addition, it forces the inclusion of those pre-defined predictions when a goal of PK is to independently construct a useful representation. Neither of these are entirely satisfactory proxies for the real question: What is the representational power of a given PK system?

As a result of this evaluation bottleneck, examples of PK on real-world problems are largely proof-of-concept applications which serve to highlight the type, quantity, and diversity of predictions which can be made (Pilarski and Sherstan 2016; Modayil, White, and Sutton 2014; Sutton et al. 2011). Where evaluation exists, it focuses on prediction error as a means of evaluating the quality of a collection of predictions. This is insufficient, as *the reliability of predictions does not necessarily equate to the quality of a learned representation*. While low prediction error describes the quality of a single predictor, low average prediction error is not necessarily indicative of the best collection of predictions for constructing representations of the environment.

For example, one could maintain a diverse collection of GVFs for different time-scales γ and policies π that exclusively anticipate the voltage of servos on a robotic limb—a signal that is often constant. These trite predictions would likely have a lower error than a collection of predictions which represent the environment more completely. Moreover, comparing the average error between two collections of predictions with different question parameters is inappropriate, as the errors are with respect to different signals. When we compare the average error of different sets of predictions in PK architectures, we are unable to meaningfully quantify how changes in the architectural proposal impact the knowledgability of a system.

We propose sensorimotor predictions as a baseline which balances our ability to meaningfully assess the representational capacity of a collection of predictions in a meaningful way, while being general enough to be extensible to real-world prediction problems.

Evaluation by Sensorimotor Baselines

A scalable alternative to comparison by hand-crafted predictions is to maintain a collection of baseline sensorimotor predictions common between each architecture being evaluated. Instead of hand-crafting predictions based on the idiosyncrasies of a particular domain, a sensorimotor baseline uses the observations from the environment as prediction targets. The identification of good features is integral to being able to make reliable predictions; in evaluating the ability of a system to predict its raw stimuli, we are in fact evaluating the ability of the system to perform representation learning for the simplest predictions we could want to make.

By comparing architectures based on how well they can represent their stimuli, we are prioritizing architectures that are able to find better representations for learning low-level sensory input, rather than better representations of the environment in general. While a limitation, it is a natural approach to evaluation: approaches to PK have been motivated by being able to anticipate their environment (Modayil, White, and Sutton 2014), and low-level anticipatory predictions are useful as inputs in applications of PK (Sherstan, Modayil, and Pilarski 2015).

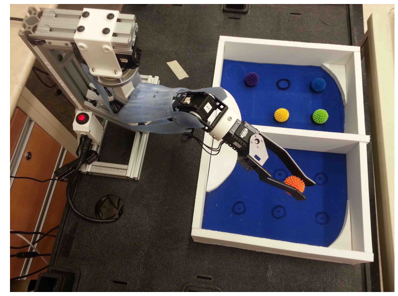

Figure 2: The data source for the experiments in this work: The Bento Arm, controlled by a human participant, generating a stream of multimodal sensory data from participants' interactions with a modified Box and Blocks task.

Sensorimotor baselines are a balance between the two aforementioned methods of evlauation: Baseline predictions enable us to assess the representation generated by our PK system with no designer intervention, making them a general scalable alternative for evaluation of real-world systems. By assessing representation quality, we can begin to precisely quantify the impact of different construction methods in real-world domains. Using sensorimotor baselines is a fair first step in bridging the evaluation gap between toy domains and real-world problems.

Evaluation by Transfer

Perhaps one of the most natural qualities of an effective PK system is generality. PK systems are intended for use in life-long, continual learning methods—methods that are expected to learn for the duration of their deployment. In such a setting it is imperative that the predictions being made are resilient to changes in their environment. A method of evaluating the ability of a continual learning system to produce general representations is through transfer-learning (Taylor and Stone 2009). We can evaluate the generality of PK by constructing GVFs in one setting and testing their generality on experience in a transfer environment that shares some traits with the source setting. An architecture that is able to propose and interrelate GVFs such that they are more robust to such transfers is an architecture that produces more general representations.

Experiment: Prosthetic Prediction Task

We explore sensorimotor baselines and transfer using data from a human control task on the Bento Arm (Dawson et al. 2014), an open-source robot arm intended for use as a research prosthesis. Human control of a robotic prosthesis is an area with active interest in PK (Pilarski and Sherstan 2016), and GVFs have been previously used to improve the control in this domain (Pilarski et al. 2013).

Data for this experiment was sourced from the previous experiments of (Edwards et al. 2016). Four users performed a common manipulation challenge where they used the robot arm to move objects over a barrier (Figure 2). Each user

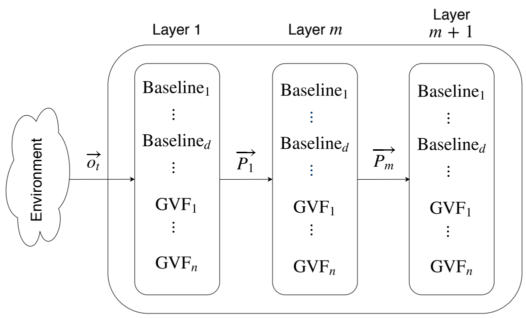

Figure 3: The architecture used for comparison. Each layer has d baseline predictions which predict each of the elements in the observations $\vec{o}$. Each layer has n additional predictions. The cumulants $c_{1...n}$ are functions of some output of the previous layer $\vec{p}_{m-1}$, or in the case of the first layer, the observations $\vec{o}_t$. For all predictions $\gamma = 0.95$, $\lambda = 0$, and step sizes are initialized to $\alpha_0 = \frac{1}{50}$, where 50 is the number of active features. Predictions are on-policy—π is always the robot arm's behaviour. Experiments vary the number of layers m and number of additional GVFs n.

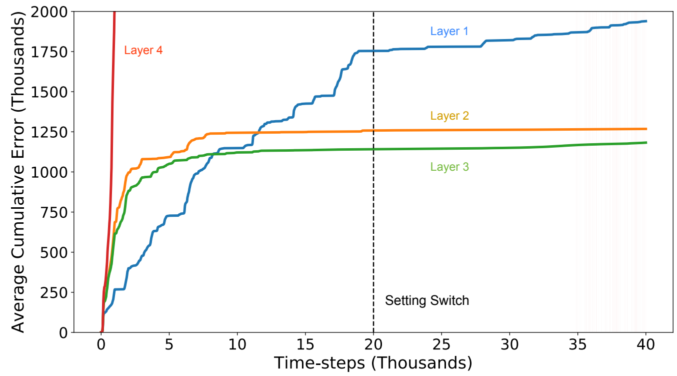

Figure 4: Accumulation of prediction error averaged over all sensorimotor baseline predictions in each layer. Our architecture has four layers and 100 constructed predictions in addition to the sensorimotor baseline. Error is averaged over 12 trials; variance is plotted but negligible.

performed the task three times using two different control schemes. We use one control scheme as the source environment and the other as a transfer environment, yielding 12 trials in total. The signals used to construct the observations are the position, load, velocity, and a binary movement signal for both the shoulder and hand joints.

The PK Architecture

To explore how choices in architecture impact the quality of learned predictions, we start with a straightforward representation using layers of GVFs (Figure 3). Inputs are produced in a feed-forward fashion: the base layer receives the observations from the environment o_t as state s_t, while each additional layer receives the output predictions from the previous layer. At each time-step, the position, velocity, and load of the shoulder and the gripper were used to construct the environment observations o_t. Step sizes are adapted using TIDBD (Kearney et al. 2017). We construct a binary representation of state by using a selective Kanerva coder (Travnik and Pilarski 2017) with 2000 prototypes and 50 active features.

In addition to the baseline sensorimotor predictions, there are n GVFs that are proposed and tested by the system. When proposing a new GVF, the architecture must specify both *what* the GVF is about by choosing c, γ, and π, and *how* the prediction is learnt by choosing appropriate learning parameters—in this instance, α and λ. Our architecture generates GVFs by randomly choosing cumulants, where c can either be an accumulation of a signal from the previous layer, or an operation on two signals—sums, differences, products, and ratios.

Each trial includes 20000 time-steps on a non-adaptive source setting where predictions are constructed, and 20000 time-steps on an adaptive switching transfer setting where the GVFs remain the same, but continue to be updated at each time-step. During the source setting, every 1000 time-steps the worst 10% of GVFs by average prediction error are culled and replaced with new GVFs, excluding the baseline predictions we use for evaluation. Prediction error is calculated online by estimating the discounted return from a sample of observed signals over approximately seven times $\frac{1}{1-\gamma}$, the expected time to termination.

Evaluation

To demonstrate evaluation using baseline predictions and setting transfer, we analyse the impact of two specification choices: 1) the number of layers in a network (Figure 4) and 2) the number of predictions in each layer (Figure 5). As indicated in Figure 1, these are decisions a designer must make when designing an architecture, and to date there is no clear intuition as to how these decisions impact the quality of the representation constructed. Our baseline predictions are of the load, position, velocity, and binary movement signal of the shoulder and gripper, or 8 predictions in total.

Since the first layer constructs its state exclusively from the observations from the setting o_t, it is not using any learned representations; by comparing each additional layer to the first layer, we assess how increasing representational abstraction impacts the baseline prediction error. Both the second and third-layer representations outperform predictions with no representation construction, while the fourth layer performs the worst. The sensorimotor baseline clearly illustrates the impact of abstraction on the ability to represent the environment.

There is a tension between what the GVFs in a layer can describe and the dimensionality of the representation for the following layer. Interestingly, the relationship between performance is not directly proportional to the number of predictions: while 10 additional predictions has the greatest performance, 100 additional predictions outperforms both 30 and 60 additional predictions. Of note is resilience to transfer to new prediction settings. Under none of the circumstances did the the methods accumulate substantially more baseline error after the switch to the transfer prediction setting. This demonstrates that the constructed representations generalized well between different control settings; however, in the future more complex transfer settings could be chosen.

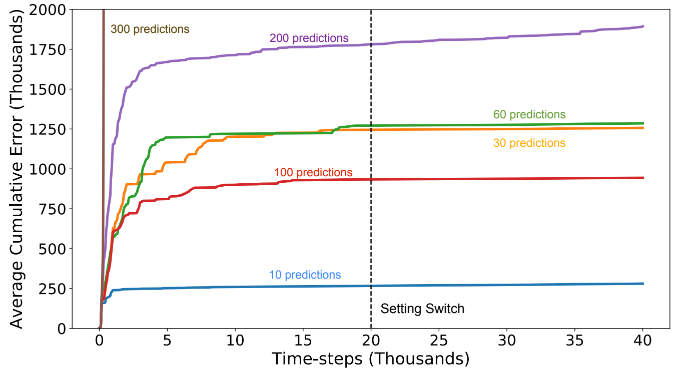

Figure 5: Accumulation of prediction error for averaged over all sensorimotor baselines predictions in the second layer. We vary the number of constructed predictions and the sensorimotor baseline. Error is averaged over 12 trials; variance is plotted but negligible.

By using a baseline of predictions, and performing transfer, we were able to elucidate how changes in the architecture impact predictive representations in a manner which requires little computational overhead. In doing so, we are providing a first step towards being able to study the impact of architectural choices on the learned representations of PK systems on real-world domains.

Limitations & Further Work

This paper's core contributions are a discussion of challenges in evaluation in PK and we do not perform an exhaustive evaluation of all the possible choices which could be made. For instance, we only consider the on-policy case, limiting the ability of our architecture to capture the impact of behaviour on the dynamics of the setting. In addition, future work could expand evaluation to include internal signals from predictions. For instance, internal signals of predictions corresponding to feature relevance could be used to identify the degree to which predictions used in the construction of state are impacting the model.

Conclusion

Predictive approaches to knowledge are a rich and varied area of reinforcement learning research that focus on building internal representations of the environment through continual, life-long interaction. There has been recent success in refining fundamental aspects of PK architectures on toy domains; however, these evaluation methods do not transfer effectively to large, real-world problems, such as applications in robotics, a core domain for predictive approaches to knowledge. In this paper, we highlight challenges in developing PK architectures and, as a primary contribution, propose the use of sensorimotor baselines and setting transfer to assess the quality of representations learned using PK. We demonstrate the usefulness of sensorimotor baselines and setting transfer by elucidating the impact of increasing the numbers of layers and number of predictions in each layer on the ability of an architecture to predict its stimuli. In providing preliminary evaluation methods for knowledge construction, we are taking a necessary step in the development of predictive knowledge.

References

[Dawson et al. 2014] Dawson, M. R.; Sherstan, C.; Carey, J. P.; Hebert, J. S.; and Pilarski, P. M. 2014. Development of the Bento Arm: An improved robotic arm for myoelectric training and research. *Proceedings of MEC* 14:60–64.

[Edwards et al. 2016] Edwards, A. L.; Dawson, M. R.; Hebert, J. S.; Sherstan, C.; Sutton, R. S.; Chan, K. M.; and Pilarski, P. M. 2016. Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching. *Prosthetics and orthotics international* 40(5):573–581.

[Kearney et al. 2017] Kearney, A.; Veeriah, V.; Travnik, J.; Sutton, R. S.; and Pilarski, P. M. 2017. Every step you take: Vectorized Adaptive Step sizes for Temporal Difference Learning.

[Littman and Sutton 2002] Littman, M. L., and Sutton, R. S. 2002. Predictive representations of state. In *Advances in neural information processing systems*, 1555–1561.

[Makino and Takagi 2008] Makino, T., and Takagi, T. 2008. Online discovery of temporal-difference networks. In *Proceedings of the 25th international conference on Machine learning*, 632–639. ACM.

[Modayil, White, and Sutton 2014] Modayil, J.; White, A.; and Sutton, R. S. 2014. Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior* 22(2):146–160.

[Pilarski and Sherstan 2016] Pilarski, P. M., and Sherstan, C. 2016. Steps toward knowledgeable neuroprostheses. In *Biomedical Robotics and Biomechatronics (BioRob), 2016 6th IEEE International Conference on*, 220–220. IEEE.

[Pilarski et al. 2013] Pilarski, P. M.; Dawson, M. R.; Degris, T.; Carey, J. P.; Chan, K. M.; Hebert, J. S.; and Sutton, R. S. 2013. Adaptive artificial limbs: A real-time approach to prediction and anticipation. *IEEE Robotics & Automation Magazine* 20(1):53–64.

[Schapire and Rivest 1988] Schapire, R. E., and Rivest, R. L. 1988. Diversity-based inference of finite automata. Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science.

[Sherstan, Modayil, and Pilarski 2015] Sherstan, C.; Modayil, J.; and Pilarski, P. M. 2015. A collaborative approach to the simultaneous multi-joint control of a prosthetic arm. In *Rehabilitation Robotics (ICORR), 2015 IEEE International Conference on*, 13–18. IEEE.

[Sutton et al. 2011] Sutton, R. S.; Modayil, J.; Delp, M.; Degris, T.; Pilarski, P. M.; White, A.; and Precup, D. 2011. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *AAMAS 2011*, 761–768. International Foundation for Autonomous Agents and Multiagent Systems.

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

[Tanner and Sutton 2005] Tanner, B., and Sutton, R. S. 2005. Temporal-Difference Networks. In *International Conference on Machine Learning*.

[Taylor and Stone 2009] Taylor, M. E., and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10(Jul):1633–1685.

[Travnik and Pilarski 2017] Travnik, J. B., and Pilarski, P. M. 2017. Representing high-dimensional data to intelligent prostheses and other wearable assistive robots: A first comparison of tile coding and selective Kanerva coding. *IEEE International Conference on Rehabilitation Robotics: [proceedings]* 2017:1443–1450.

[White 2015] White, A. 2015. *Developing a predictive approach to knowledge*. PhD Thesis, PhD thesis, University of Alberta.