A SURVEY OF GENERAL VALUE FUNCTIONS AND ROBOTICS

Samuel Neumann
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
sfneuman@ualberta.ca

1 INTRODUCTION

Robotics is a vast field encompassing ideas from disciplines such as mechanical engineering to artificial intelligence. From the view of artificial intelligence, the goal is to create robots which are autonomous and can learn in real time. The emphasis is placed on *learning*. A robot should be able to learn about and adapt its behaviour to its environment.

The problem of creating autonomous robots using artificial intelligence has proven to be difficult. A central issue is that the world is partially observable, and many observations which are needed to make informed decisions are not available to the robot. Unlike robots, humans are able to make many predictions about the world in short periods of time, and such predictions inform the decisions we make. For example, if you need to make it to the airport before 9:30 a.m., you can make an informed guess on what time to leave your house. You can do this even though the amount of time it takes to drive to the airport is not directly known before-hand. The human brain is excellent at making multiple predictions about the future and taking all of these predictions into account to make an informed decision. This phenomenon though has proven difficult for robots.

This paper is a survey of methodologies used to account for such behaviour in robots. In particular, this paper considers the reinforcement learning formalism and discusses how general value functions (GVFs) can be used as both predictive state representations and predictive approaches to knowledge in order to enable robots to make informed decisions.

2 A SHORT SUMMARY OF REINFORCEMENT LEARNING AND GENERAL VALUE FUNCTIONS

Reinforcement learning is a way to formalize sequential decision making. In reinforcement learning, an intelligent agent (for example, a robot) finds itself in some environment and must take actions which alter the environmental state. Upon taking an action, the agent receives a reward. The goal of the agent is to maximize rewards, and it must therefore learn which actions lead to high rewards.

This process is modelled as a Markov Decision Process which is represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \gamma)$, where $\mathcal{S}$ is the set of possible states, $\mathcal{A}$ is the set of possible actions, $\mathcal{R}$ is the set of possible rewards, and $p(s', r \mid s, a)$ is the transition dynamics which measures the probability density of transitioning to state s' and receiving reward r after taking action a in state s[1]. The discount factor, $0 \leq \gamma \leq 1$ determines the relative importance of near and future rewards. The actions the agent takes in a state $s \in \mathcal{S}$ are drawn from its policy $a \sim \pi(\cdot \mid s)$, which is a function mapping states to probability distributions over actions. The agent must learn a policy which selects actions to maximize the reward received.

A popular method to determine which actions lead to high rewards is by using temporal-difference learning (Sutton, 1988) to learn value functions. A state-value function is a function which measures the expected, discounted sum of future rewards attainable after some state $s \in \mathcal{S}$ when following the

[1]Here, we consider continuous state spaces and continuous rewards, that is $\mathcal{S} \subseteq \mathbb{R}^n$ for some $n \in \mathbb{N}$ and $\mathcal{R} \subseteq \mathbb{R}$. It is also possible to consider discrete state spaces and discrete reward by considering p to be a probability mass function.

agent's policy:

$$v_\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s \right] = \mathbb{E}_\pi \left[\sum_{k=1}^{T} \gamma^{k-1} R_{t+k} \mid S_t = s \right] \tag{1}$$

where T denotes the final time step, which may be infinite[2], and G_t is defined implicitly. An action-value function measures the expected, discounted sum of future rewards attainable after taking some action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and then following the agent's policy thereafter:

$$q_\pi(s, a) = \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right] = \mathbb{E}_\pi \left[\sum_{k=1}^{T} \gamma^{k-1} R_{t+k} \mid S_t = s, A_t = a \right] \tag{2}$$

Value functions can be generalized to measure the expected, discounted sum of any future signal. The future signal is referred to as the *cumulant* while the discount factor is referred to as the *time scale*. These general value functions (GVFs) can be used to answer predictive questions about the agent's current state or future. For example, a robot could use GVFs to predict how long it might take to drive to the airport. Such predictive state representations are useful as they can be used to inform a robot's real-world decisions. Similar to value functions, GVFs are typically learned using the method of temporal-differences (Sutton, 1988). In the next section, we provide a survey on how GVFs have been combined with robots to both answer predictive questions and generate predictive representations about the robot's state.

3 A Survey of GVFs in Robots

One of the first major breakthroughs in utilizing GVFs in robotics was the Horde architecture (Sutton et al., 2011). This architecture was designed to allow a robot to answer many predictive or goal-oriented questions about it or its environment. Each question is answered by a single reinforcement learning agent which learns a GVF. Each of these GVF learners is referred to as a daemon, and has its own policy, timescale, and cumulant corresponding to the question the daemon answers. Sutton et al. (2011) demonstrated that Horde could be used to learn in real-time on a mobile robot to accurately predict the answers to questions such as:

1. *How much time will elapse before I hit an obstacle?*
2. *How much time do I need in order to stop before hitting the obstacle?*

The authors also demonstrated that Horde could be used to learn goal-oriented behaviours in real-time on a mobile robot. In particular, the authors showed that the Horde architecture could be used to train a mobile robot to stay near light even when the robot learned under a random behaviour policy.

Modayil et al. (2012) showed that *nexting*, the ability of humans to predict what might happen next, is possible on robots. Whereas Sutton et al. (2011) focused on answering off-policy questions using the Horde architecture, Modayil et al. (2012) focused on using GVFs to answer thousands of on-policy questions in parallel and at different time scales. In their experiments, Modayil et al. (2012) showed that a mobile robot could successfully predict both future state representations and changes in its sensor readings at multiple time scales. This was one of the first times GVFs had been used in real-time on a robot to answer thousands of questions about both the state of the robot and the state of the environment.

Until this point, learning thousands of GVFs in parallel had only been demonstrated when learning on-policy. This was a significant limitation to the utility of GVFs, since an important part of life-long learning, where the robot continually learns over the course of its lifetime, is off-policy learning. White et al. (2012) demonstrated the ability of a mobile robot to learn thousands of GVFs off-policy and in real-time using the Horde architecture (Sutton et al., 2011) and a random behaviour policy. To do so, learning algorithms which are stable under off-policy updating such as GTD(λ) (Maei, 2011) were utilized. This work demonstrated that Horde could be utilized to learn hundreds of GVFs from 6 different policies. The authors also demonstrated that their methodology scaled to many policies. Using the same, random behaviour policy, the authors learned GVFs for 1,000 randomly

[2]For value functions to be well-defined, we require $0 \leq \gamma < 1$ when T is infinite.

generated policies over 4 different time scales. This demonstrated that learning about many different behaviours (through GVFs) is possible in real-time on a robot. Computation was performed on a laptop via a wireless link to the robot, but given sufficient computational power these computations could have been performed directly on the robot. This was the first demonstration of large-scale off-policy learning of GVFs in real-time on a robot.

A large area of research in robotics is modular prosthetic limbs, and how these limbs can be properly controlled by humans. Such a task has proven difficult due to a disparity between the number of electrical signals the human user can send to the prosthetic limb through muscle tissue and the degrees of freedom of the prosthetic limb's many actuators. GVFs may be able to rectify this issue and have been used in the past to increase the utility of prosthetic limbs (Sherstan, 2020; Parker et al., 2019; Pilarski et al., 2013; Vasan & Pilarski, 2018).

Pilarski & Sherstan (2016) used approximately 18,000 GVFs to predict information about the velocity, position, impedance, and temperature of the many actuators in a robotic prosthetic arm. After only six minutes of training, the prosthetic arm not only could detect errors due to human perturbation but also could anticipate when future errors would occur.

Günther et al. (2018) used GVFs to learn to predict and anticipate signals on a robotic prosthetic arm. Sensor data from the robotic arm was packed into UDP packets of 3,520 bits. These bits were then used as both state inputs and cumulants for a first Horde of GVFs (one GVF for each bit). Using the prediction of this first Horde as input, a second Horde of GVFs predicted surprise as unexpected daemon error (UDE) (White, 2015), a measure of unexpected change in predicted signal due to changes in the environment. UDE compares the prediction error of the current signal to the average past prediction error and will remain low both during regular learning and when observing noise in the learned signal. UDE will only significantly increase if changes in the environment alter the TD error of the learned signal. In this way, UDE can be viewed as a sort of surprise due to changes in the environment. In their experiments, Günther et al. (2018) showed that a robotic prosthetic arm was able to learn to anticipate surprise, measured as UDE, when it was perturbed in a recurring fashion. Günther et al. (2018) suggest that abstract predictive models such as predictions of surprise can serve to increase a robots understanding of itself and its environment under continual learning.

Humans are excellent at utilizing past experiences to generalize to new situations; one major issue with GVFs is that when a GVF is newly added in the middle of training, a robot's past experience cannot be utilized to learn the newly added GVF. In a continual learning setting on a prosthetic arm, Sherstan et al. (2018) demonstrated that successor representations (Dayan, 1993) could be used mitigate this issue. They showed that successor representation can improve both sample efficiency and learning speed when incrementally adding new GVFs during training. In their experiments, Sherstan et al. (2018) had a human user control a robot arm by guiding its end effector through a maze 12 times over 50 minutes. GVFs for six different predictive targets were learned: the current, position, and speed of both the elbow and shoulder joints. During the experiment, new GVFs were added at set time periods; GVFs were learned from two different sets of features – successor representation features and direct state features. They found that successor representations improved both the sample efficiency and speed of learning of the newly added GVFs.

Until this point, GVFs were used to answer a question at a specific time scale. For example, *if I drive straight for **ten seconds**, how soon until I hit a wall?* Sherstan et al. (2019) introduced Γ-nets, which allow GVFs to generalize over time scale. Γ-nets work by training a standard GVF with two additional inputs – the timescale parameter γ and the expected number of steps until termination $\tau = \frac{1}{1-\gamma}$. In this way, a GVF can predict the answer to a question at a given time, the time indicated by γ. Sherstan et al. (2019) showed that Γ-nets could accurately predict the shoulder joint speeds of a robotic prosthetic arm at multiple time scales in the future. In their experiments, the end effector of a robotic arm was guided through a wire maze by a human controller. They trained three GVFs and one Γ-net to predict the shoulder joint speed at three different time scales. At each time scale, the predictions of the Γ-net were as accurate as the corresponding GVF, yet the Γ-net was significantly more flexible and possessed fewer parameters than the three GVFs combined.

Finally, Faridi et al. (2022) demonstrated that a robotic exoskeleton, the Indego exoskeleton, could utilize GVFs to learn the walking preferences of a human user. This was the first time GVFs had been used as prediction mechanisms for lower-limb controlled robotic prosthetics. The Indego exoskeleton is intended to assist humans in walking and has actuated hip and knee joints. In their experiments,

Faridi et al. (2022) had a human user control the exoskeleton by selecting three different walking speeds (slow, medium, fast) and two different walking directions (turn left, turn right). GVFs were used to anticipate the next most likely walking mode to be selected by the user at an accuracy of approximately 83%, almost double that of the non-adaptive baseline strategy.

4 Conclusion

A central barrier to applications of robotics to real-world problems is the inherent partial observability of our world. Such partial observability limits robotics by providing insufficient data to make informed decisions. One method to deal with such partial observability is the use of general value functions (GVFs) to form predictive state representations. GVFs have been used to improve both mobile and manipulator robots and hold much promise for increasing the applicability of robots to real-life situations.

References

Peter Dayan. Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, 1993.

Pouria Faridi, Javad Mehr, Don Wilson, Mojtaba Sharifi, Mahdi Tavakoli, Patrick Pilarski, and Vivian Mushahwar. Machine-learned Adaptive Switching in Voluntary Lower-limb Exoskeleton Control: Preliminary Results. *IEEE International Conference on Rehabilitation Robotics*, 2022.

Johannes Günther, Alex Kearney, Michael Dawson, Craig Sherstan, and Patrick Pilarski. Predictions, Surprise, Predictions of Surprise in General Value Function Architectures. *AAAI 2018 Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy*, 2018.

Hamid Maei. *Gradient Temporal-Different Learning Algorithms*. PhD thesis, University of Alberta, 2011.

Joseph Modayil, Adam White, and Richard S. Sutton. Multi-Timescale Nexting in a Reinforcement Learning Robot. *International Conference on Adaptive Behaviour*, 2012.

Adam Parker, Ann Edwards, and Patrick Pilarski. Exploring the Impact of Machine-Learned Predictions on Feedback from an Artificial Limb. *IEEE International Conference on Rehabilitation Robotics*, 2019.

Patrick Pilarski and Craig Sherstan. Steps Toward Knowledgeable Neuroprostheses. *IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2016.

Patrick Pilarski, Travis Dick, and Richard Sutton. Real-Time Prediction Learning for the Simultaneous Actuation of Multiple Prosthetic Joints. *IEEE International Conference on Rehabilitation Robotics*, 2013.

Craig Sherstan. *Representation and General Value Functions*. PhD thesis, University of Alberta, 2020.

Craig Sherstan, Marlos C. Machado, and Patrick M. Pilarski. Accelerating Learning in Constructive Predictive Frameworks with the Successor Representation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

Craig Sherstan, Shibhansh Dohare, James MacGlashan, Johannes Günther, and Patrick M. Pilarski. Gamma-Nets: Generalizing Value Estimation over Timescale. *AAAI Conference on Artificial Intelligence*, 2019.

Richard Sutton. Learning to Predict by the Method of Temporal Differences. *Machine Learning*, 1988.

Richard Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick Pilarski, Adam White, and Precup Doina. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. *The 10th International Conference on Autonomous Agents and Multiagent Systems*, 2011.

Gautham Vasan and Patrick Pilarski. Context-Aware Learning from Demonstration: Using Camera Data to Support the Synergistic Control of a Multi-Joint Prosthetic Arm. *IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2018.

Adam White. *Developing a Predictive Approach to Knowledge*. PhD thesis, University of Alberta, 2015.

Adam White, Joseph Modayil, and Richard S. Sutton. Scaling Life-long Off-Policy Learning. *IEEE International Conference on Development and Learning and Epigenetic Robotics*, Jun 2012.