Robot Module 2: General Value Functions

Amir Samani*

***Abstract*— In Robot Module 1, we built the basic tools required for real life system to test various applications of reinforcement learning (RL). In Robot Module 2, we build General Value Functions (GVFs) on top of the experimental setup from Robot Module 1. We design two on-policy and one off-policy questions in GVF framework and use raw sensorimotor data stream from the robot to answer those questions.**

I. Introduction

General Value Functions (GVFs) can be used to as an approach to representing predictive knowledge [1]. GVFs take advantage of conventional value functions to represent useful predictive knowledge about the environment, while using the strong conventional value function learning methods. In this module, we design three GVF questions and use reinforcment learning methods to answer these questions. The outline for the rest of write-up is in the following. In Section II, we discuss experimental setup. Section III is dedicated to how to design a GVF question. In Section IV, we design two on-policy GVF questions and one off-policy GVF question and illustrate the result of answering the questions based on raw sensorimotor data stream from the robot. Last but not least, Section V summarizes the write-up and point out the most noticeable conclusions.

II. Robot Setup

In this module we use the same robot setup as Robot Module 1 [1]. The details of the setup is available in the write-up for module 1 as well. For the purpose of clarity the final robot setup is illustrated in Figure 1. However, for the purpose of simplicity in this work, we only use one of the servos for the experiments.

Fig. 1. Robot setup for the experiments.

III. GVFs

For applications in artificial intelligence such as robotics, we need to keep an accurate knowledge of the world. This is specially true in a complex and changing environment. General Value Functions (GVFs) is an approach to represent predictive knowledge. Using GVFs we can use conventional value function learning methods to represent useful predictive knowledge about the world.

Intuitively, GVFs enable us to ask questions about the raw or processed sensorimotor data stream of the robot and answer them using conventional value function learning methods. In order to design a question, we need to think about two elements, cumulant and termination. Cumulant is a signal that plays a similar role to the reward target construction, but the agent goal is not maximizing the cumulant all the time [1]. The termination can be considered as indicator for "how long we are asking this question?". To shed light on the matter, we use an example that we will expand in Section IV. Let's assume we want to answer the question of "while I am following the behavior policy (rotating between angle 1.5 and -1.5), what is the next angular position?". First of all, this is an on-policy question, because the behavior policy (the policy that the agent follows) and the target policy (the policy that we ask the question about) are the same. To formalize the question in terms of a GVF, we need to design cumulant and termination signals. The cumulant can be the next angle. Since the question is a one-step prediction, the termination is always 0. In the next section we discuss this GVF question and two more in details.

IV. Experiments

In this section we discuss three different experiments. First one is an on-policy question with fixed termination. Second question is also an on-policy question but with a state-dependent termination. While, the third question is an off-policy question.

A. On-policy question with fixed termination

This question is the same as the question that we used as an example in Section III. We want to ask "while I am following the behavior policy (rotating between angle 1.5 and -1.5), what is the next angular position?". The behavior policy (since it is an on-policy question, the behavior policy and the target policy are the same), is rotating servo angle

*Amir Samani is with Faculty of Computing Science, University of Alberta, Edmonton, Canada samani@ualberta.ca

[1] available at: https://github.com/Amir-19/M1-Signs_Of_Life

between 1.5 and -1.5. This question is an one-step prediction, since we want to know the next angular position of the servo. Hence, the termination, γ, is always zero. Intuitively, termination is indicating that whether we are asking the question or not in this state (that is why termination can be state-dependant). Since termination is always zero, the value function estimates only the next cumulant. Moreover, we want to know the next angular position, so we use the angular position as the cumulant. To summarize, we used angular position as the cumulant and the termination is always 0. Now we need to think about how to answer this question. As mentioned earlier, one of the advantages of using GVFs is the ability to use the conventional reinforcement learning methods to estimate the value function. Since the question is on-policy, we can TD(λ) [2]. We also need to think about the state representation to be used in the answer part. Since the servo rotating between -1.5 and 1.5 we decided to use the angle and direction of rotation (towards 1.5 or towards -1.5) as the state. Then we digitized this into some bins (thus, the number of bins is a parameter of the answer). The other parameters set as: step size (α) is 0.1 (divided by the number of active features in state representation which is 1), eligibility trace parameter (λ) is 0.99 and the number of bins is 15, so feature vector is a vector of size 16. Figure 2 illustrates the beginning of learning for this question, while, Figure 3 shows the verifier with 100 time step delay. As the learning continues, Figure 4 illustrates the learning process in time step around 300 and Figure 5 compares the ideal return and the prediction (with 100 time step delay). In order to make sure that the learner is not tracking the target and actually learning, we freeze the learning (by setting the step size to zero) around time step 600 and we can observe the prediction in Figure 6.

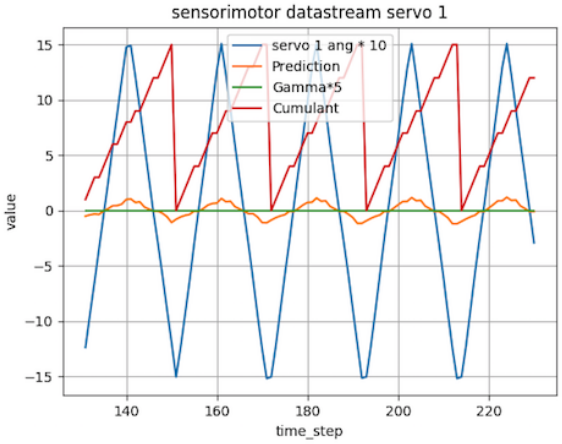

Fig. 2. Beginning of learning for first on-policy question.

B. On-policy question with state-dependent termination

Unlike the first question, the second question is no longer an one-step prediction. The second question asks "while I am following the behavior policy (rotating between angle 1.5 and -1.5), how many steps I am away from the angle

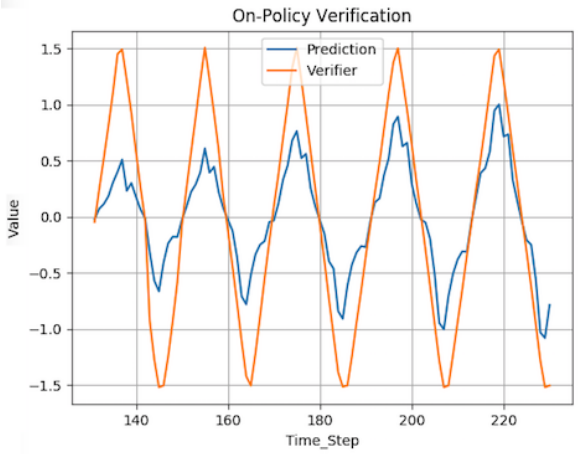

Fig. 3. Begining of learning for first on-policy question (comparing verifier and prediction).

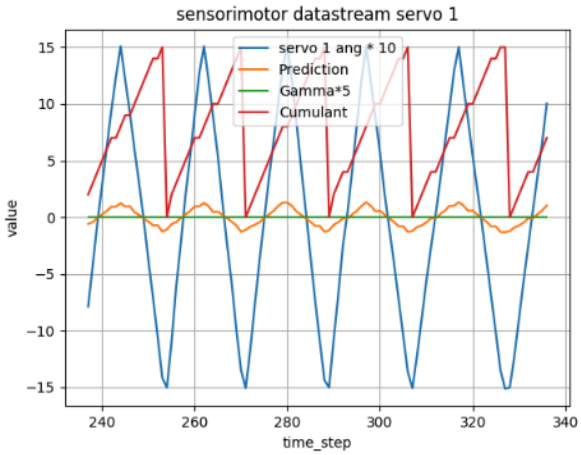

Fig. 4. Time step 300 of learning the first on-policy question.

-1.5?". Similar to the previous question, we need to design cumulant and termination to specify the question in terms of a GVF. Since we want to ask a question about "how many time steps till I get to angle -1.5", we can have the cumulant 1 for each state expect for the state that servo is at angle -1.5. This is the same for the termination also. To clarify, we want to keep asking this question until the servo is at angle -1.5. Thus, for this question, termination is dependent on the state. For the answer, again we use TD(λ) with same function approximation and state representation. The other parameters set as: step size(α) is 0.1 (divided by the number of active features in state representation which is 1), eligibility trace parameter (λ) is 0.99 and the number of bins is 15, so feature vector is a vector of size 16. Figure 7 illustrates the beginning of learning for this question, while, Figure 8 shows the verifier with 100 time step delay. As the learning continues, Figure 9 illustrates the learning process in time step around 300 and Figure 10 compares the ideal return and the prediction (with 100 time step delay). In order to make sure that the learner is not tracking the target and actually learning, we freeze the learning (by setting the step

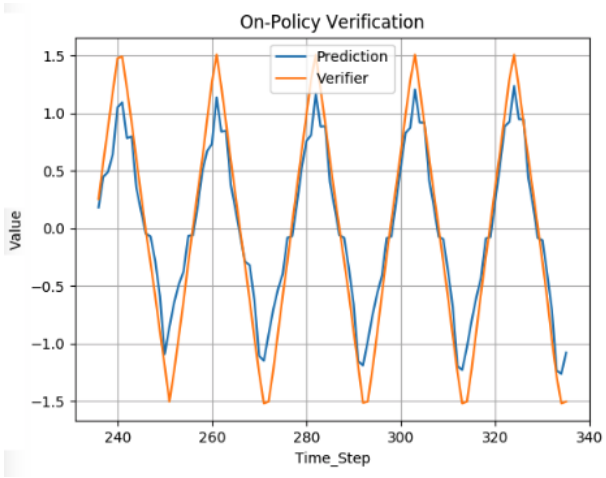

Fig. 5. Time step 300 of learning the first on-policy question (comparing verifier and prediction).

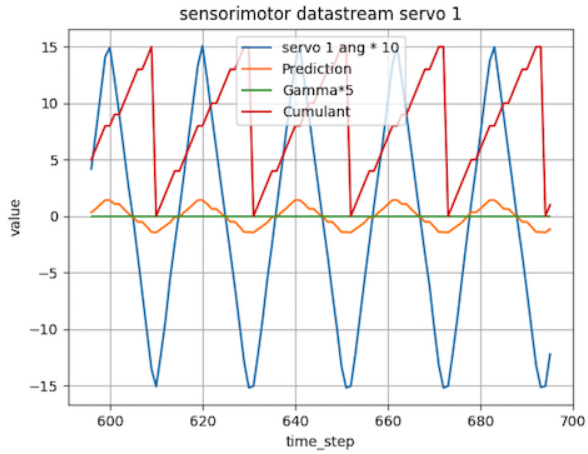

Fig. 6. Freezeing the learning for first on-policy question to make sure the learning is not tracking the target.

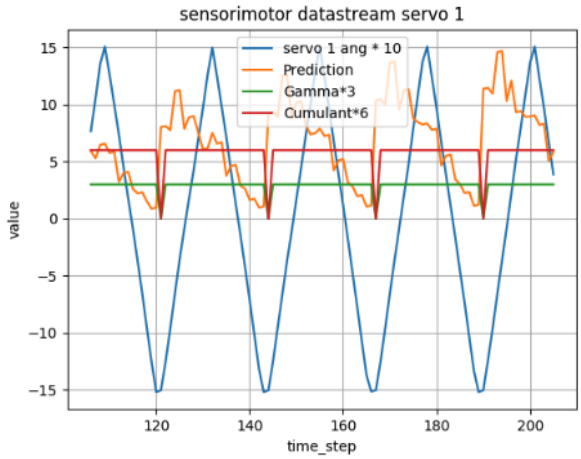

Fig. 7. Beginning of learning for the second on-policy question.

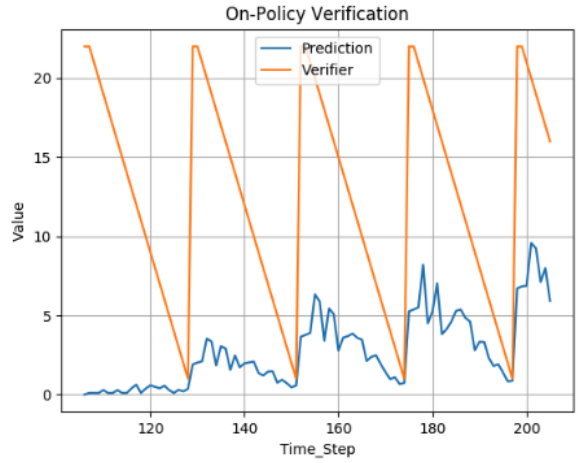

Fig. 8. Begining of learning for the second on-policy question (comparing verifier and prediction).

size to zero) around time step 600 and we can observe the prediction in Figure 11.

C. Off-policy question

In the off-policy case we need to have a target policy different than the behavior policy. In our case, we decided to use the same behavior policy as the on-policy question. While, the target policy is always choose to rotate towards angle -1.5 (so it never moves towards angle 1.5). The off-policy question we try to answer is "while I am following the behavior policy (rotating between angle 1.5 and -1.5), how many steps I am away from the angle -1.5 If I were to rotate directly towards angular position -1.5?". The behavior policy sometimes moves towards angle -1.5 and sometimes moves away from -1.5 but the question we are asking is specifying a target policy that moves only moves towards angle -1.5. As we can see, the target policy and behavior policy overlap when the servo moving towards angle -1.5. The cumulant and termination is the same as the on-policy version of this question. However, to answer this question we need to use GTD(λ) instead of TD(λ) (since we are dealing with an off-policy question) [3]. One important parameter added to off-policy learning is importance sampling ration. Importance sampling ratio helps us to learn more when the target policy and behavior policy overlaps and not to learn when they are not overlapping. The importance sampling ratio in this experiment is either 0 or 2 (since the direction is either towards -1.5 or towards 1.5, and the importance sampling is 2 when servo moves towards -1.5, and 0 when servo moves towards 1.5). However, these value for importance sampling makes the algorithm diverge and after spending sometime tuning, we had to set importance sampling to either 0 or 1 by dividing the importance sampling ratio by two to stop algorithm from diverging. The other parameters set as: step size(α) is 0.1 (divided by the number of active features in state representation which is 1), eligibility trace parameter (λ) is 0.99 and the number of bins is 15, so feature vector is a vector of size 16. Since we are using GTD we need to specify the seconds step size for the second set of weights. We used 0.001 (divided by the number of active features

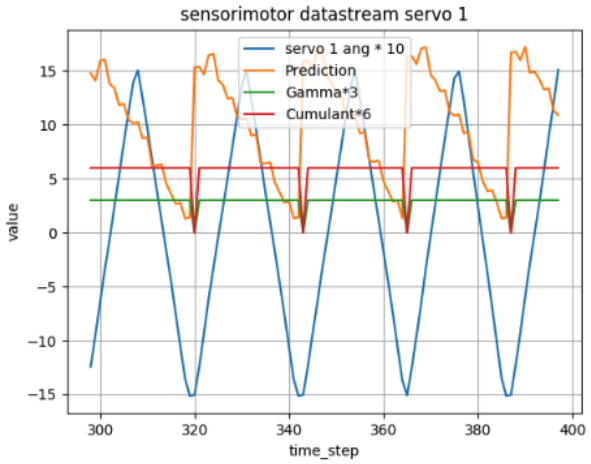

Fig. 9. Time step 300 of learning the second on-policy question.

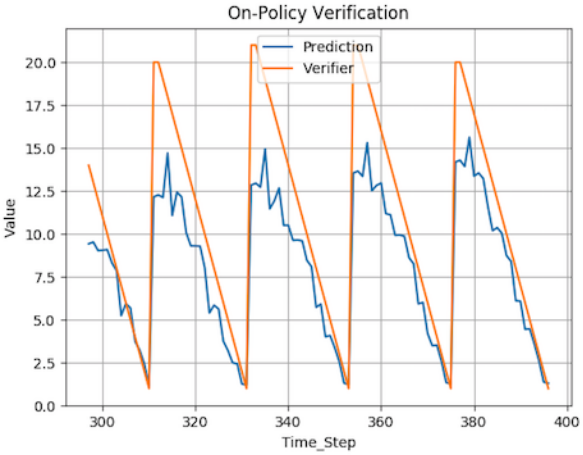

Fig. 10. Time step 300 of learning the second on-policy question (comparing verifier and prediction).

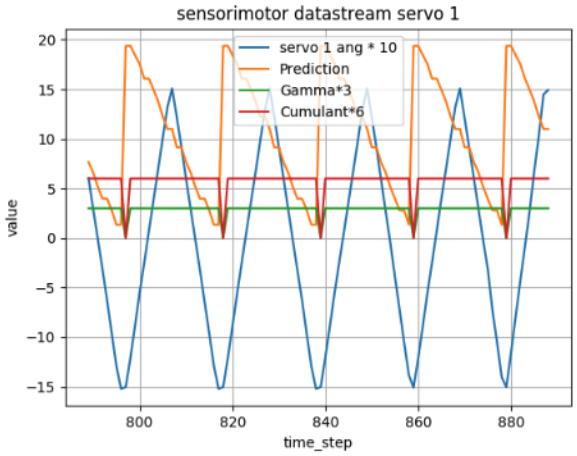

Fig. 11. Freezing the learning for the second on-policy question to make sure the learning is not tracking the target.

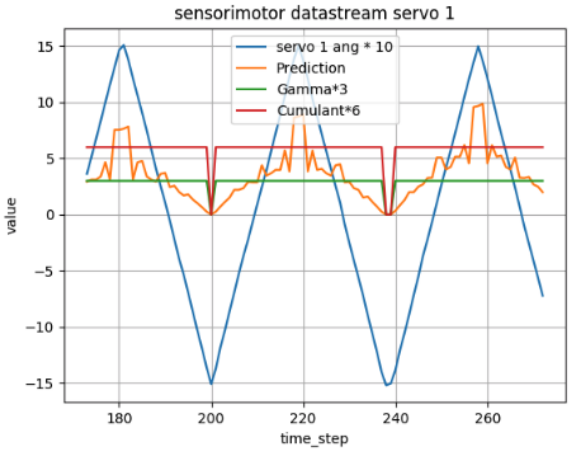

Fig. 12. Beginning of learning for the off-policy question.

in state representation which is 1) for β. Unlike TD(λ) which can be considered somehow robust to the parameter selection, our experiments shows GTD(λ) parameters plays an important role to its stability. Similar to the importance sampling ratio that we discussed, other parameters should be chosen carefully to make sure that the GTD(λ) is not diverging. In our experiments we always look at δ for the TD error and size of updates to make sure we are not diverging. Figure 12 illustrates the beginning of learning for the off-policy question. We can see learning is going on in Figure 13 and finally to make sure that the learning is not tracking the target we freeze the learning by setting α and β to zero and observe the prediction which is illustrated in Figure 14.

V. Summary

In this module we showed how to we can use raw sensorimotor stream of data to answer some question about robot sensorimotor data stream. We discussed how to design on-policy and off-policy questions and how to answer them using conventional reinforcement learning methods such as TD(λ) and GTD(λ). We illustrated the results of learning from for three different questions. This module can be expanded in many directions including, we can save the raw sensorimotor data stream and do parameter sweep for finding best step size, λ, and number of bins. In addition, the saved data can be used to possibly answer other GVFs. Moreover, we can used more complex function approximation scheme such as tilecoding or artificial neural networks.

References

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

[2] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[3] Hamid Reza Maei. *Gradient temporal-difference learning algorithms*. PhD thesis, University of Alberta, 2011.

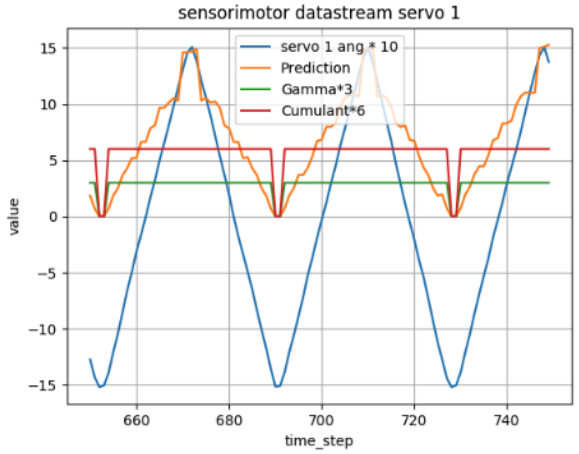

Fig. 13. Time step 700 of learning the first off-policy question.

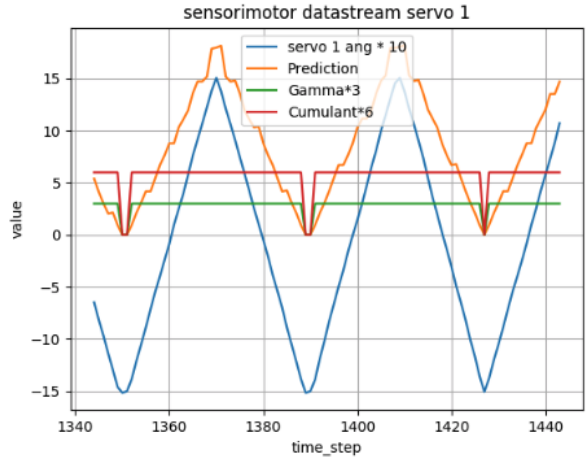

Fig. 14. Freezing the learning for the off-policy question to make sure the learning is not tracking the target.