Reproducibility study of the Value Equivalence principle for Model Based Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Reproducibility Summary

Scope of Reproducibility

Grimm et al. [2020] introduces and studies the concept of equivalence for Reinforcement Learning models with respect to a set of polices and value functions. It further shows that this principle can be leveraged to find models, constrained by representational capacity, which are better than their maximum likelihood counterparts.

Methodology

The code for this project was closed sourced, therefore we re-implemented the three sets of experiments (including the baseline) and the custom environments used by the authors. All experiments were performed using Google Colab and required a total of 160 hours of Google Colab GPU.

Results

Since all the results in the original paper are presented in graphical form, we cannot provide precise numbers. For experiments with $\mathrm{span}(\mathcal{V}) \approx \ddot{\mathcal{V}}$, our results match the reported results. For experiments with $\mathrm{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and linear function approximation, our results for both the baseline and the author's method, diverge from the reported graphs. For experiments with $\mathrm{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and neural networks our results follow the reported trend, but not always with the same values.

What was easy

Even though we had to re-implement everything from scratch, the general pipeline for all experiments were straightforward and well described in the original paper. The environments used for all three experiments were fairly simple.

What was difficult

Grimm et al. [2020] is a combination of theory and experiments. It is crucial to understand the theorems presented in the paper which require a strong knowledge of linear algebra. For experiments with linear function approximation, feature selection was done using k-means which depends heavily on initialisations of centroids. It takes a large amount of time to repeatedly apply k-means on a large set of data to find the best fit (more than 10 hours for 10,000 initialisations on a dataset of 1,000,000).

Communication with original authors

We contacted the author for multiple queries related to custom environments, hyper-parameters, feature selections and other minute experimental details via email. The author replied to all of them thoroughly and in reasonable time.

Submitted to ML Reproducibility Challenge 2020. Do not distribute.

1 Introduction

Grimm et al. [2020] introduces the principle of Value Equivalent models with respect to a set of policies and a set of value functions. Consider, Π to be a set of policies and $\mathcal{V}$ to be a set of functions from $S \to R$. Any two models m and $\tilde{m}$ are value equivalent with respect to Π and $\mathcal{V}$ if and only if, for all $\pi \in \Pi$ and all $\nu \in \mathcal{V}$, the following is true.

$$\mathcal{T}_\pi \nu = \tilde{\mathcal{T}}_\pi \nu$$

where T_π is the bellman operator of the MDP with respect to policy π.

The space of models which are value equivalent to the true model of the MDP, with respect to Π and $\mathcal{V}$, is called $\mathcal{M}(\Pi, \mathcal{V})$. The main argument of the paper is that we can aptly choose $\mathcal{V}$ and Π and exploit the Value Equivalence Pinciple to find models in $\mathcal{M}(\Pi, \mathcal{V})$ which are simpler and save computation and memory.

2 Scope of reproducibility

Grimm et al. [2020] shows that it is possible to exploit some structure and define the sets $\mathcal{V}$ and Π for training models using the Value Equivalence principle. And that these models perform better than Maximum Likelihood models, under rank constraints.

Exploiting structure in the problem: The set of true value functions of an MDP forms a polytype $\ddot{\mathcal{V}}$ (Dadashi et al. [2019]). To span as much of the polytype as possible, we define $\mathcal{V}$ to contain true value functions ν_π associated with random deterministic policies π.
Exploiting structure in the solution: In most applications of model based RL, value functions or policies are represented using function approximation. Suppose the agent can only represent policies $\pi \in \tilde{\Pi}$ and value functions $\nu \in \tilde{\mathcal{V}}$, then any model belonging to the set $\mathcal{M}(\tilde{\Pi}, \tilde{\mathcal{V}})$ is as good as the true model of the MDP.

We aim to verify these claims by using rank constrained models to compare the performance of Value Equivalence and Maximum Likelihood Estimation.

3 Methodology

We re-implement the training, evaluation and the custom environments for all the experiments performed by the authors using the experimental pipeline and environment description provided in the appendix of Grimm et al. [2020].

3.1 Experimental setup and code

The total reproducibility can be divided into three type of experiments:

- $\mathrm{span}(\mathcal{V}) \approx \ddot{\mathcal{V}}$ and finite state space.
- $\mathrm{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and finite state space, using linear function approximation.
- $\mathrm{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and infinite state space, using neural network.

We used the Catch environment for the first two experiments and the CartPole environment for the last one. The authors have used a custom Four-Rooms environment as well which has been uploaded to our Github repository and can be easily used by future practitioners. Our entire code can be accessed at https://github.com/RajGhugare19/VE-principle-for-model-based-RL

3.2 Hyperparameters

All the hyper-parameters are provided by the authors in the appendix. We found that for experiments with $\mathrm{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and linear approximation, the specified learning rate ($5 \cdot 10^{-5}$) did not work. After doing a learning rate search we found that, $5 \cdot 10^{-3}$ works best for these experiments.

3.3 Computational requirements

All experiments were carried out on the free GPUs from Google Colab. The GPU memory requirements and the total GPU hours required by every experiment are provided below.

Table 1: Compute Requirements per experiment

Experiment	CUDA memory (MB)	GPU training time (hrs)
$\text{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and infinite state space	1600	8
$\text{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and finite state space	600	0.25
$\text{span}(\mathcal{V}) \approx \ddot{\mathcal{V}}$ and finite state space	560	0.25

4 Results

In experiments with $\text{span}(\mathcal{V}) \approx \ddot{\mathcal{V}}$ and $\text{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ with neural networks, we found that the Value Equivalence models consistently performed better than the baseline, under rank constraints. For experiments with $\text{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ with linear function approximation we couldn't reproduce the results for both Value Equivalence training as well as the baseline.

4.1 Results reproducing original paper

4.1.1 Result 1

The results from table 2 support the claim that Value Equivalence training converges towards a value equivalent model in a rank constrained space, i.e. a space with lesser span, when enforcing value equivalence with the value function polytype ($\text{span}(\mathcal{V}) \approx \ddot{\mathcal{V}}$).

Table 2: $\boldsymbol{span(\mathcal{V}) \approx \ddot{\mathcal{V}}}$ and Catch environment

Table 3: rank 40

	Our Results		Reported Results	
size $\mathcal{V}$	VE	MLE	VE	MLE
40	9.898	4.929	9.8	4.8
35	9.879	4.929	9.8	4.8
30	9.883	4.929	9.8	4.8
25	9.894	4.929	9.8	4.8
20	9.817	4.929	9.8	4.8
15	8.876	4.929	9.8	4.8
10	8.527	4.929	8.9	4.8
5	7.720	4.929	7.9	4.8
2	6.926	4.929	6.3	4.8

Table 4: size $\mathcal{V}$ (# of policies) = 10

	Our Results		Reported Results	
rank	VE	MLE	VE	MLE
20	9.479	4.303	8.5	5.1
30	9.883	5.975	9.4	4.3
40	8.527	4.929	9.6	6.2
50	8.914	8.002	9.8	9.1
100	9.049	9.887	9.8	9.6
150	9.089	9.910	9.9	9.9
200	9.895	9.910	9.9	9.9
250	8.053	9.910	9.9	9.9

The results are average state values corresponding to the policies formed using respective models.

4.1.2 Result 2

We followed the given pipeline and performed 10 experiments, the code for which can be found in our repository. But the values corresponding to the resulting policies for both VE and MLE models were highly stochastic and did not reflect the results reported by the authors.

To further evaluate the experiments with $\text{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and linear function approximation we used Double DQN to form a policy where the action value functions were linear function approximators (see table 5).

The number of experiments performed were relatively small, but their results clearly show that value equivalence models perform better than the baseline, under rank constrained.

4.1.3 Result 3

The results from table 6 indicate that even when the set $\tilde{\mathcal{V}}$ consists of neural networks, the Value Equivalent models with rank constraints tend to perform better than the baselines. But as the constraints are reduced the performance of the VE models seems to deteriorate.

These results are average returns for 100 consecutive episodes corresponding to the best policies formed using Double DQN.

Table 5: size $\tilde{\mathcal{V}} = 50$

	Our Results	
rank	VE	MLE
20	6.34	3.96
50	8.2	7.85
100	9.93	5.93

Table 6: $\boldsymbol{span(\mathcal{V}) \approx \tilde{\mathcal{V}}}$ and CartPole environment

Table 7: rank 10

	Our Results		Reported Results	
size $\tilde{\mathcal{V}}$	VE	MLE	VE	MLE
32	987.4	446.5	780	600
64	664.7	560.9	800	620
128	994.8	959.8	710	800
256	418.8	994.1	700	890
500	980.87	628.66	600	830
1000	945.3	562.61	650	810

Table 8: size $\mathcal{V}$ (width) = 128

	Our Results		Reported Results	
rank	VE	MLE	VE	MLE
2	152.9	112.3	180	60
4	224.7	524.2	780	100
6	726.6	981.0	800	400
8	927.1	994.3	790	690
10	959.8	994.8	800	740
12	981.9	880.7	690	700
14	972.3	684.5	700	760
16	974.2	842.7	690	780
20	993.8	671.1	800	800

4.2 Results beyond original paper

We used a different evaluation technique for the results of table 5. Instead of using approximate policy iteration with LSTD, we used Double DQN with linear action-value function approximators. The hyper-parameters used for this can be found in table 9.

Table 9: size $\tilde{\mathcal{V}} = 50$

Hyper-parameters	Values
DQN learning rate	1e-3
size of experience replay	10000
total iterations	1000000
DQN learning frequency	2

5 Discussion

Our results support the existence of Value Equivalent models in lower dimensional spaces and the ability of Value Equivalence training to converge to these models.

Even though we weren't able to reproduce the results for experiments with $\text{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ with linear function approximation, we have a few likely reasons behind this.

- The performance of both the VE and baselines models depend highly on the feature formation, which was done using k-means. The results of k-means are heavily dependent on the initialisation of centroids.
- LSTD with policy evaluation was used to evaluate the model. We are not sure if this method is guaranteed to converge to the best policy. We found that for most models this was not the case.
- The evaluation method not only diverged for the VE models but also for MLE models.

The results of table 5 also solidify the above reasons.

5.1 What was easy

The experiments in the paper were easy to follow and code, as the authors have divided the entire pipeline into different stages with clear descriptions for every stage.

5.2 What was difficult

Some points were left out or not clear in the original paper and we had to ask the authors to get them cleared.

- Feature formation for experiments with $\mathrm{span}(\mathcal{V}) \approx \tilde{\mathcal{V}}$ and linear function approximation was done by applying k-means on a set of 1000000 states collected using a random policy.
- The size of experience replay used in the paper was 1000000.
- Rank constrained matrices where optimised by applying gradient descent to the parameters F_D and F_K.

References

Robert Dadashi, Adrien Ali Taïga, Nicolas Le Roux, Dale Schuurmans, and Marc G. Bellemare. The value function polytope in reinforcement learning. *CoRR*, abs/1901.11524, 2019. URL http://arxiv.org/abs/1901.11524.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning, 2020.