Representation-Induced Algorithmic Bias: An empirical assessment of behavioural equivalence over 14 reinforcement learning algorithms across 4 isomorphic gameform representations

Supplementary Material A
Revised SMV function

Simon C Stanton[1][0000-0003-0312-4407]

[1] University of Tasmania
simon.stanton@utas.edu.au

Abstract. This document corrects for an error found in the implementation of one algorithm (of 14) analysed in the paper *Representation-Induced Algorithmic Bias*, accepted to AJCAI21. The error is in the inappropriate use of *self.prev_state* in the action-selection mechanism of the algorithm Actor/Critic only. Two variants of Actor/Critic in the original study use the correct variable (*self.state*) in the appropriate way. Whether error or typo, or an incorrect interpretation regarding the sequence order of *now* (t+1) and the *immediate past* (t), is discussed. Importantly, the code has been corrected and experiments re-run. Results from re-processing the analysis indicate that although the algorithm's behaviour does change, the change is not counter to the hypothesis presented in the original paper. Before and after revisions to tables are provided. The conclusions in the original paper remain.

1 Introduction

The issue manifests in a single algorithm in the test set of fourteen algorithms and appears as the use of a variable in the action-selection mechanism such that the algorithm was drawing its next action from its internal representation of the *prior state of the world*, one step behind its *current understanding of the world*. This resulted in action-selection such that at time $\boldsymbol{t+1}$ (*now*) the processing of the incoming reward signal had been completed and internal state had been updated, but the action for the next timestep was being drawn using the state at timestep $\boldsymbol{t}$ (*the immediate past timestep*).

Reprocessing the experiments for the affected algorithm indicate four entries in *Tables 3-6* in the original paper be updated. *Table 7* in the submitted paper also requires minor changes. **The conclusions of the submitted paper remain.**

2 The SMV Function

Whether this occurred as simple error, a typo, or as a result of a lack of clarity in interpreting the literature is not clear: *Mea Culpa.* Sutton and Barto introduce a SoftMax action-selection technique in the first edition, and continue its use in the second edition, of *Reinforcement Learning: An Introduction* (1998, 2018). Their presentation is not explicit with regard to the final step in action-selection when using the SoftMax technique in Actor/Critic. However, their treatment of various Binary Bandit algorithms (that utilise SoftMax) follow the procedure of a) receive incoming reward, b) process incoming reward and update internal state, and c) draw action-selection using the current state. Szepesvari is explicit regarding the sequence of updates to state and subsequent action-selection.

This study includes two Actor/Critic variants (Actor/Critic with Eligibility Traces, and Actor/Critic with Replacing Traces). Both of these implementations conform to the expected sequence of operations.

The specific code can be found at line #165 in the agent_model_hpc *equivalence study* repository release v0.1-alpha in agent_model::strategies::actor_critic_1ed.py. See **Code Availability** for URL.

3 Size of Effect

This issue affects the data generated from four (4) sub-experiments. The experiment type is selfplay_parameter_study over four game form representations assessing the Actor/Critic algorithm. The data from these four experiments is fed into a sequence of analysis steps that results in Wilcoxon paired-treatment tests independently comparing four pairings of the data generated from the gameform representations.

Reprocessing the experiments with the corrected code it is evident that there is some change in the comparisons of variance. The pairing of scalar and scalar_norm.3 (Experiment Group One) changes status from not exhibiting significant variance (V=38811, p-value=.5776) to *weakly* not exhibiting significant variance (V=35742, p-value=.07125). In contrast, the pairing in Experiment Group Two changes from exhibiting significant variance (V=31507, p-value=.0002), to no longer exhibiting significant variance (V=41239, p-value=.6227). The effect in Experiment Group Three (V=36299, p-value=.1005) and Experiment Group Four (V=46908, p-value=.0032) is to strengthen the observed significant variance: (V=47086, p-value=.0025) and (V=48983, p-value=.0001), respectively.

4 Revised and Original Data

The values shown in **Table 1** are the original results for the Wilcoxon tests and additionally shows peak cooperative outcomes. **Table 2** shows the revised values. *Table 7* in the original paper is replaced by **Table 3**, where Actor/Critic is removed from Exp Grp Three and added to Exp Grp Two.

Table 1. Original Paper Aggregated Distributions Wilcoxon Tests.

Algorithm	Exp Grp	Peak % CC		Wilcoxon			
				V	p-value	CI L	CI U
Actor/Critic	One	32.6	27.8	38811	.5776	-0.0127	0.0052
	Two	32.6	12.4	31507	.000204	-0.0182	-0.0081
	Three	32.6	14.9	36299	.1005	-0.010	0.0014
	Four	27.8	14.9	46908	.003261	0.0080	0.0577

Table 2. Revised Paper Aggregated Distributions Wilcoxon Tests.

Algorithm	Exp Grp	Peak % CC		Wilcoxon			
				V	p-value	CI L	CI U
Actor/Critic	One	32.1	30.2	35742	.07125	-0.0176	0.0006
	Two	32.1	14.0	41239	.6227	-0.0049	0.0074
	Three	32.1	19.7	47086	.002535	0.0048	0.0195
	Four	30.2	19.7	48983	.0001235	0.0285	0.0712

Table 3. Algorithms that do not exhibit significant variance between behavioural profiles.

Experiment		**Experiment**	
Group	**p-value > .05**	**Group**	**p-value > .05**
One	Actor/Critic	Three	Double Q-Learning
	Actor/Critic with Replacing Traces		**Watkins Q, Linear Function Approximation**
	Watkins Q, Linear Function Approximation		
Two	Actor/Critic	Four	Actor/Critic with Replacing Traces
	Actor/Critic with Replacing Traces		SARSA Lambda
	Q-Learning		SARSA Lambda, with Replacing Traces
	Double Q-Learning		Watkins (naïve) Q, Lambda
	Expected SARSA		Watkins (naïve) Q, Lambda, Replacing Traces
	R Learning		Watkins Q, Lambda
	SARSA		**Watkins Q, Linear Function Approximation**
	SARSA Lambda, with Replacing Traces		
	Watkins Q, Lambda		
	Watkins Q, Linear Function Approximation		

References

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction (1st ed.).* The MIT Press.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction (2nd ed.).* The MIT Press.

Szepesvári, C. (2010). *Algorithms for Reinforcement Learning.* Morgan and Claypool Publishers.

Code Availability A repository of code releases and supplementary material is available at https://github.com/simoncstanton/equivalence_study.