# Usable ABC Reinforcement Learning

**Christos Dimitrakakis**
Chalmers university of technology
chrdimi@chalmers.se

**Nikolaos Tziortziotis**
University of Ioannina
ntziorzi@cs.uoi.gr

## Abstract

The issues with the use of Approximate Bayesian Computation in Reinforcement Learning is the following. Firstly, that the model set may comprise simulators which are purely deterministic. Secondly, that there is a dependence between the policy used and the data collected, which necessitate maintaining a representation of the policy used as well as the data history. Thirdly, there is the question of the statistics used. Finally, there is the problem selecting a policy given the data observed so far. In this paper, we report some progress on using more sophisticated statistics and policy search algorithms and show that they have significant impact.

## 1 Introduction

Approximate Bayesian computation (ABC) methods for reinforcement learning (RL) were first introduced in (Dimitrakakis and Tziortziotis, 2013), with sufficient conditions for ABC to be a useful approach for reinforcement learning problems and provided some experimental evidence for its utility. There are, however a number of open problems relating to the use of ABC methods in RL. The first is that in many cases, the simulators used are deterministic. In that case, the assumptions given in Dimitrakakis and Tziortziotis (2013) do not hold any more. The second is the fact that in order to perform the posterior calculation in controlled systems using ABC, the policy used to generate the data must be remembered. The third concerns the choice of statistics, which can severely influence how easy it is to distinguish between different models. Finally, given an approximate posterior distribution computed over models, the question is how to compute an appropriate near-optimal policy.

**Setting.** To fix ideas, consider an agent acting in some discrete-time environment $\mu \in \mathcal{M}$, taking actions $a_t \in \mathcal{A}$ and obtaining observations $x_t \in \mathcal{X}$ and scalar rewards $r_t \in \mathbb{R}$. Let $\mathcal{H} = (\mathcal{X}, \mathcal{A}, \mathbb{R})^*$ be the space of all possible observation histories. The agent uses a policy $\pi : \mathcal{H} \to \mathcal{A}$ to take actions and is interested in maximising the utility

$$U \triangleq \sum_{t=0}^{\infty} \gamma^t r_{t+1}, \qquad \gamma \in (0,1) \tag{1.1}$$

where $\gamma$ is a discount factor. We assume that for each $\mu$, there exists an optimal policy $\pi^*(\mu)$ maximising expected utility $\mathbb{E}_\mu^\pi U$.

In the reinforcement learning problem, $\mu$ is unknown. Adopting a Bayesian perspective, we can assume a prior probability distribution $\xi$ on $\mathcal{M}$, encoding the agent's subjective belief about which is the most likely model. Then the optimisation problem is

$$\mathbb{E}_\xi^\pi U = \int_{\mathcal{M}} \left(\mathbb{E}_\mu^\pi U\right) \, \mathrm{d}\xi(\mu). \tag{1.2}$$

Solving this gives a policy that optimally trades off obtaining more information about $\mu$ (exploration) and maximising rewards in the short term (exploitation). However, there are three important problems. The first is how to select $\mathcal{M}$ and $\xi$, the second is how to perform inference and the third is how to find the optimal policy.

**Contribution.** In this work we shall focus on the case where $\mathcal{M}$ is a set of detailed simulators, which could model reality very well, but whose parameters are uncertain. This naturally lends to ABC inference. The optimisation problem is solved approximately through a combination of Thompson sampling (Thompson, 1933) and either approximate dynamic programming (ADP, c.f. Bertsekas, 2005) or Monte-Carlo planning (MCTS, c.f. Kocsis and Szepesvári, 2006). Our main contribution is the investigation of the properties of how the choice of statistic and policy search affect the final outcome. We conjecture that more informative statistics should allow us to more easily distinguish between models, and that more sophisticated policy search methods, such as Monte-Carlo planning would be less sensitive to the domain they are tested on.

## 2 ABC Reinforcement learning.

The basic ABC RL algorithm is composed of two parts. The first, given in Algorithm 1 involves sampling a model from the approximate posterior through rejection sampling.

**Algorithm 1** ABC-RL-Sample

```
input Prior $\xi$ on $\mathcal{M}$, history $h \in \mathcal{H}$, threshold $\varepsilon$, statistic $f : \mathcal{H} \to \mathcal{W}$, policy $\pi$, maximum number of samples $N_{\mathrm{sam}}$, stopping condition $\tau$.
$\widehat{M} = \emptyset$.
for $k = 1, \dots, N_{\mathrm{sam}}$ do
  $\mu^{(k)} \sim \xi$.
  $h^{(k)} \sim P^{\pi}_{\mu^{(k)}}$
  if $\left\| f(h) - f(h^{(k)}) \right\| < \varepsilon$ then
    $\widehat{M} := \widehat{M} \cup \{ \mu^{(k)} \}$.
  end if
  if $\tau$ then
    break
  end if
end for
return $\widehat{M}$
```

Given a posterior sampling mechanism, we can now perform Thompson sampling, given in Algorithm 2. This involves finding a policy that is optimal for the given sampled model. However, this is not trivial for general models, as approximations must be used.

**Algorithm 2** ABC-RL Thompson sampling

```
parameters $\mathcal{M}$, $\xi$, $h$, $\pi$, $f$
$\tau = \{|\widehat{M}| = 1\}$
$\hat{\mu} = \texttt{ABC-RL-Sample}(\mathcal{M}, \xi, h, \pi, f, \tau)$
return $\hat{\pi} \approx \arg\max_{\pi} \mathbb{E}^{\pi}_{\hat{\mu}} U$
```

**Sufficient conditions for deterministic models.** When the model is deterministic, we still need a way to guarantee that ABC behaves reasonably well. First, we recall the assumption and theorem proved in (Dimitrakakis and Tziortziotis, 2013).

**Assumption 1.** *For a given policy $\pi$, for any $\mu$, and histories $x, h \in \mathcal{H}$, there exists $L > 0$ such that $\left|\ln \left[\mathbb{P}^{\pi}_{\mu}(h) / \mathbb{P}^{\pi}_{\mu}(x)\right]\right| \leq L \|f(h) - f(x)\|$.*

**Theorem 1.** *Under a policy $\pi$ and statistic $f$ satisfying Assumption 1, the approximate posterior distribution $\xi_{\epsilon}(\cdot \mid h)$ satisfies:*

$$D\left(\xi(\cdot \mid h) \parallel \xi_{\epsilon}(\cdot \mid h)\right) \leq \ln |A^{h}_{\epsilon}| + 2L\epsilon, \tag{2.1}$$

*where $A_\epsilon^h \triangleq \{ z \in \mathcal{H} \mid \|f(z) - f(h)\| \leq \epsilon \}$ is the $\epsilon$-ball around the observed history $h$ with respect to the statistical distance and $|A_\epsilon^h|$ denotes its size.*

If some models are deterministic, we need some different assumptions, as $\mathbb{P}_\mu^\pi$ becomes a delta function and the log likelihood ratio becomes unbounded. Then in fact the statistic's value uniquely determined by the policy and the model. Let that be $f_\mu^\pi$.

**Assumption 2.** *Let $\mu^*$ be the model from which the data is generated. For any $f : \mathcal{H} \to \mathcal{W}$, $\exists L, U > 0$ such that $L\epsilon \leq \xi\left(\left\{ \mu \in \mathcal{M} \;\middle|\; \|f_\mu^\pi - f_{\mu^*}^\pi\| \leq \epsilon \right\}\right) \leq U\epsilon$.*

Under these conditions, it is trivial to show that the KL divergence is bounded.

**Remark 1.** *Under Assumption 2, the approximate posterior satisfies:*

$$D\left(\xi(\cdot \mid h) \parallel \xi_\epsilon(\cdot \mid h)\right) \leq \ln \frac{U\epsilon}{\xi(\mu^*)} \tag{2.2}$$

*Proof.* For the discrete $\mathcal{M}$ case, we can write

$$D\left(\xi(\cdot \mid h) \parallel \xi_\epsilon(\cdot \mid h)\right) = \sum_{\mathcal{M}} \ln \frac{\xi(\mu \mid h)}{\xi_\epsilon(\mu \mid h)} \xi(\mu \mid h) = -\ln \xi_\epsilon(\mu^* \mid h) \leq \ln \frac{U\epsilon}{\xi(\mu^*)}.$$

□

While the above assumptions gives us some formal guarantees about the quality of the posterior approximation, it is hard to interpret them. In practice, one must strive to simply select sufficiently informative statistics.

**Choice of statistics.** One simple choice, used in (Dimitrakakis and Tziortziotis, 2013), is the utility of a history $h = (x_{h,t}, a_{h,t}, r_{h,t})_{t=1}^{T_\tau}$, $f(h) = U(h) = \sum_t r_{h,t}$. We call this the $U$-statistic.

The $U$-statistic could be insufficient to distinguish between different models. A natural candidate is the discounted set of features $f(h) = \phi(h) = \sum_t \gamma^t \phi(x_t)$, where $\phi$ maps from $\mathcal{X}$ to some vector space. For the case where the model is a finite Markov decision process and $\phi$ is an indicator vector, we can construct the discounted state occupancy matrix $(\mathbb{E}_\mu^\pi \phi \mid x_0 = i)_i = (I - \gamma \mathbb{P}_\mu^\pi)^{-1}$, which can be used to calculate the value function of any policy $\pi$. In the general case, this link does not exist, but it is nevertheless suggestive for constructing statistics.

In particular we define the $\Phi$-statistic as follows. First, if $\mathcal{X} \subset \mathbb{R}^n$, we select some $k \in \mathbb{N}$ and generate a Gaussian $k \times n$ matrix $\Phi$, as commonly used in compressed sensing (Donoho, 2006). Then our statistic becomes $f(h) = \sum_t \Phi x_{h,t}$. We expect such statistics to be much more informative and to be generally robust.

**Choice of policy optimisation.** A final question is what policy optimisation method to use after a model has been sampled from the approximate posterior. While (Dimitrakakis and Tziortziotis, 2013) employed ADP[1], selection of appropriate parameters and issues with convergence made it problematic for general deployment. In this paper, we investigate the use of Monte Carlo tree search, and in particular the variant employed in (Hester and Stone, 2013), which combines upper confidence bounds with value function approximation in a discretised space.

**Experimental results** We performed some experiments to investigate whether there is a particular advantage from using a more sophisticated policy search method or more informative statistics. As this is only a preliminary investigation, we used fixed settings for

[1] While the paper reports experiments with LSPI, we had also experimented with fitted value iteration, which turn out not to be very robust either.

all algorithmic hyper-parameters.[2] In all cases, the experiments were performed by first generating 10 trajectories from a uniform random policy. These were then used to either directly find a policy via LSPI, or to first sample a model from the posterior, and then perform LSPI or UCT on the sampled model. From the results shown in Table 1, it is clear that in general Φ-statistics are an improvement over the simple $U$-statistic. In addition, UCT appears to be generally more robust than LSPI, especially when Φ-statistics are used.

| Domain | LSPI | ABC/LSPI-$U$ | ABC/UCT-$U$ | ABC/LSPI-Φ | ABC/UCT-Φ |
|---|---|---|---|---|---|
| Acrobot | -99 | -100 | **-97** | -100 | **-97** |
| Cart-Pole | 25 | 41 | **87** | 51 | 78 |
| Mountain Car | -71 | -56 | **-47** | **-47** | -50 |
| Pendulum | -0.9 | -0.3 | **0** | -0.06 | **0** |
| Puddle | -360 | -142 | -133 | -237 | **-108** |

Table 1: Average utility achieved from 10 randomly generated trajectories.

## 3 Conclusion.

The use of ABC methods in reinforcement learning is promising, as it allows us to use arbitrary simulator models for inference. These tie in rather well with simulation-based methods for policy optimisation. However, there are a number of challenges, such as the theoretical guarantees of ABC-RL, the choice of appropriate statistics, and the policy optimisation method.

In this paper, we provided some possible sufficient conditions for ensuring that ABC is a reasonable approach also for deterministic models. This is an assumption on the relation between the prior and the statistic use, and so it is hard to verify. In practice, we use the cumulative discounted feature Φ-statistic, which allows us to better identify models. Combined with UCT, this frequently results in very good policies from only small amounts of data. Major open questions are how to make such approaches more efficient, and how to automatically select the statistic, especially given the online nature of RL.

**Acknowledgements.** Thanks to Joel Veness for suggesting replacing ADP with MCTS and to Mahdi Milani Fard for suggesting the link to compressed sensing.

## References

Dimitri Bertsekas. Dynamic programming and suboptimal control: From ADP to MPC. *Fundamental Issues in Control, European Journal of Control*, 11(4-5), 2005. From 2005 CDC, Seville, Spain.

Christos Dimitrakakis and Nikolaos Tziortziotis. ABC reinforcement learning. In *ICML 2013*, volume 28(3) of *JMLR W & CP*, pages 684–692, 2013. See also arXiv:1303.6977.

David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4): 1289–1306, 2006.

Todd Hester and Peter Stone. Texplore: real-time sample-efficient reinforcement learning for robots. *Machine Learning*, 90(3):385–429, 2013.

Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of ECML-2006*, 2006.

W.R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of two Samples. *Biometrika*, 25(3-4):285–294, 1933.

[2] For LSPI, $2 \cdot 10^3$ trajectories were generated from the sampled model, and we used a squared-exponential basis functions on a 5-interval grid. For UCT we set the number of rollouts to $10^2$, the maximum tree depth to $10^3$, set the mixing constant to 0 and the learning rate to $1/k$, the grid interval to 20, and the exploration constant to $10^3$.