Data-efficient Reinforcement Learning Based on Koopman Operator Theory

Junheng Wang, advised by Dr. Lixing Song

Department of Computer Science & Software Engineering, Rose-Hulman Institute of Technology

Introduction and Motivation

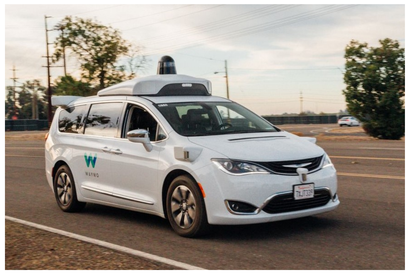

Fig. 1: A Self-driving Car on Road Test

Reinforcement Learning (RL) has been widely used to provide solutions to control problems and decision making, with various successful applications including but not limited to robotics, autonomous vehicles (Fig 1), and games. Given a certain task, *RL* leverages the intuition of "trial and error" (Fig 2) to search for optimal decisions. The accuracy of *RL* algorithms comes at the cost of training time and data efficiency, especially in *model-free RL* where the algorithm does not create a model of the environment. For example, over the span of weeks of training of OpenAI Five, an *RL* agent that plays Dota, more than 25 million observations were created every day, equivalent to 900 years of gaming experience of a human player. [2].

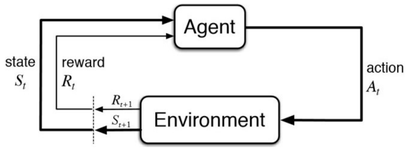

Fig. 2: Reinforcement Learning Explained

Part of this disadvantage can be overcome with *model-based RL*, where the algorithm learns a model of the environment and uses that model to improve the policy. This allows the agent to produce more educated guesses and thus increase data efficiency. But the performance of *model-based RL* is highly dependent on how well the model is constructed. Models that accurately represent the environment are typically complex due to nonlinearity, resulting in difficult control problems. In this work, we aim to apply Koopman Operator Theory to the optimal control problem of nonlinear model and use it as an data-efficient model-based reinforcement learning approach.

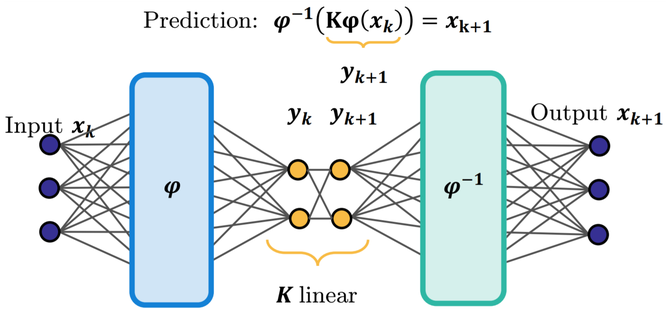

Fig. 3: High-Level Structure of Our Model

Methodology

Optimal linear control is a well studied problem with off-the-shelf, well developed solutions such as *linear–quadratic regulator (LQR)*. In a finite horizon and in discrete time, *LQR* can be expressed as

$$x_{k+1} = Ax_k + Bu_k$$

where x stands for state and u stands for action. Once the A and B matrices are determined, the state of the next timestep can be easily calculated by matrix arithmetic.

To bridge the gap between nonlinear models in *model-based RL* and linear control methods, we introduce Koopman Operator Theory [1], which maps the nonlinear component of the model to its linear equivalent in a higher dimensional space (Fig 4). This will then allow us to apply *LQR* to the linearized model for optimal control.

$$Kg(x^t) = g(x^{t+1})$$

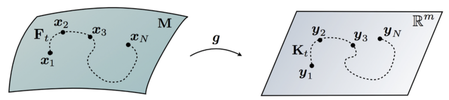

Fig. 4: Schematic Illustration of Koopman Operator

The problem is thus divided into two parts: finding the mapping between linear and nonlinear representations, and finding the *Koopman Operator K* that advance the measurement forward linearly. The first part of the problem is typically solved by using a dictionary that handles the mapping. In our approach, we use a pair of autoencoder and autodecoder to find the dictionary (Fig 3), which then maps the nonlinear representations to and from their linear counterparts. Once we obtain the linear representation, we can apply *LQR* to find the *Koopman Operator K*, solving the second part of the problem.

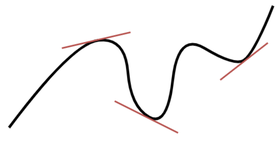

Fig. 5: Local Linear Models

One key challenge of this approach is that for complex nonlinear models, it is difficult to find a single yet global mapping to produce a linear representation. To maintain high data-efficiency and produce reliable models, we can learn multiple local models using *Operator Sampling* (Fig 5) instead of a global one. For each sequence of observations, we fit a local linear model that can be used to estimate the global model, around where the data is collected. Multiple linear models are then used together when making predictions, and the best performer is sampled out and used in the control process. This *Operator Sampling* process increases the possibility of finding better models.

References

[1] B. O. Koopman. “Hamiltonian Systems and Transformation in Hilbert Space”. In: *Proceedings of the National Academy of Sciences* (1931).

[2] OpenAI. “Dota 2 with Large Scale Deep Reinforcement Learning”. In: *arXiv:1912.06680* (2019).

Experiment

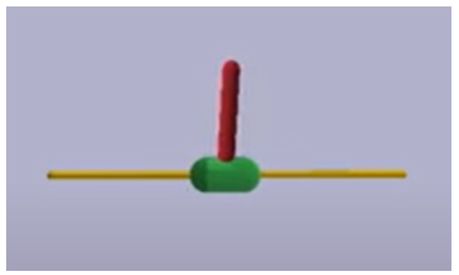

Fig. 6: Inverted Pendulum Environment

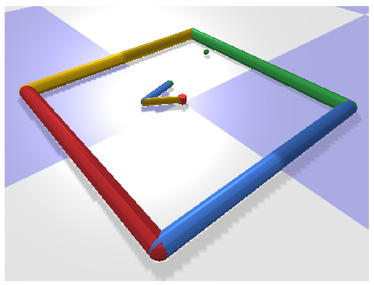

Fig. 7: Reacher Environment

Our approach is evaluated in simulated environments from OpenAI Gym and PyBullet: the Inverted Pendulum environment (Fig 6) and the Reacher environment (Fig 7). Both our models, with and without *Operator Sampling*, are tested in these environments, and several other reinforcement learning approaches are also included for comparison. Particularly for the out approach with *Operator Sampling*, multiple sequence lengths and sampling size are used to evaluate the performance of local models. Results from the Inverted Pendulum experiment are provided below.

Results

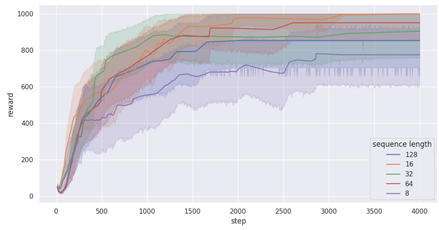

Fig. 8: Reward Curves w.r.t. Sequence Length

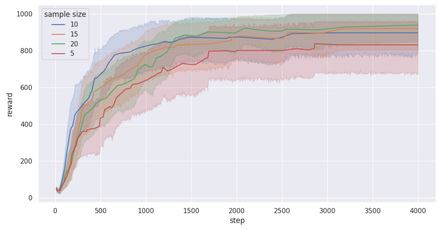

Fig. 9: Reward Curves w.r.t. Sample Size

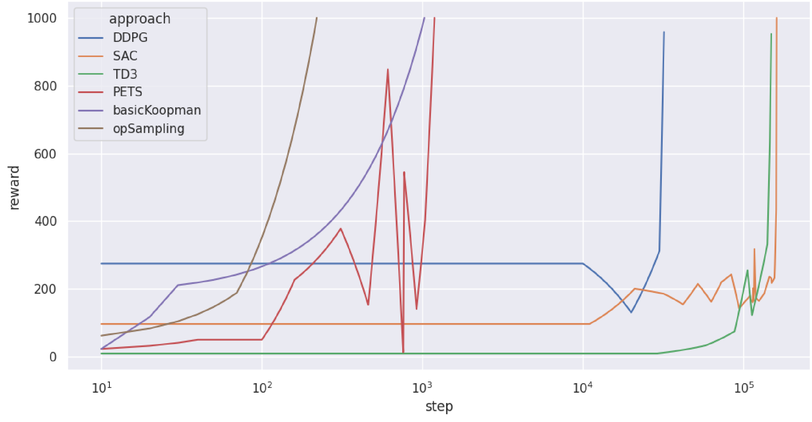

Fig. 10: Reward Curves Comparison Between RL Approaches