Research Project Proposal: Transfer of generative models in reinforcement learning

Pierluca D'Oro, pierluca.doro@gmail.com

1. Introduction to the problem

Reinforcement learning (RL) is the area of machine learning that studies sequential decision-making problems, in which an agent acts in an environment with the aim of maximizing a cumulative reward. Training an agent to learn a *policy* to choose actions optimally in every state of the environment is often quite expensive, slow or dangerous. Furthermore, many RL approaches struggle to transfer knowledge across similar environments or tasks. The application to RL of generative modeling, the area of machine learning that addresses the problem of estimating the generating distribution for a given set of data, has shown to be able to improve both the efficiency of learning, drastically reducing the need for difficult interactions with the environment, and the generalization of policies among related tasks.

A learning agent can leverage generative models in various ways, but often they are used in estimating environmental dynamics, either by modeling the distribution of entire state-action trajectories or single-step state transitions. This class of approaches goes under the name of *model-based* reinforcement learning. Model-based RL is a powerful paradigm: if an agent is able to predict the effect that actions will have on the environment, it can use this knowledge to plan or learn faster when presented with new problems.

In several scenarios, it is possible to collect interactions either from more than one agent acting in the environment according to its own policy, or from multiple related Markov Decision Processes (MDPs), which formalize reinforcement learning problems [13]. In both cases, a model-based agent could take advantage of these different experiences and use the additional information they provide about the environment during its learning process, approximating its dynamics. In fact, with this amount of information available, training a generative model for the sole estimation of trajectories generated by a single target policy would be a waste of precious information; nonetheless, a generative model that is trained to approximate the environmental dynamics for all available policies at the same time struggles to learn the complex, inherently multimodal distribution that fits all the differences in policies. Similarly, a single, monolithic model would be not able to estimate plausible trajectories for a whole family of related MDPs, but still knowledge of the dynamics of different MDPs can be exploited for better modeling in a related setting, with the ultimate goal of improving the performance of an agent.

An example scenario is that of *autonomous driving*, where an agent is asked to drive a vehicle, receiving input signals from its sensors and controlling its actuators. It is easy to imagine that there could be multiple and diverse sources of driving experience, such as different drivers, different vehicles, different driving conditions. Moreover, you could use simulations to obtain transitions that are not very likely in the real world (e.g., car crashes). A learning agent that uses a generative model for estimating the environment dynamics would waste all this experience if it only learns from its own interactions. Being data collection for autonomous driving expensive and time-consuming, a dynamic model able to understand how to reuse the trajectories generated by other policies or in other conditions to estimate the ones of a target policy or condition would be extremely beneficial.

Other application settings that feature great variability are related to the medical domain, in which RL methods can be used to help in controlling medical equipment. There can be significant variation of conditions among different patients or even for the same individual, due either to patient's state or practical issues. For instance, in *functional electrical stimulation* (FES) [9], a medical technology used for rehabilitation of individuals and to address problematic neuromuscular conditions, there is uncertainty about patients and equipment at the start of each session. Nonetheless, an agent must be able to learn the optimal behavior quickly, given a very short duration for the session itself, and flexible generative models can offer an effective performance boost.

2. Main related works

Model-based RL has been studied for many years to make reinforcement learning more efficient and more adaptable [15, 11, 17]. Recent work employed generative models to estimate the distribution of entire state-space trajectories, in place of state-to-state transitions. In [10], a generative model [14] is conditioned on past states and actions as well as on planned future actions and used to sample likely future trajectories. The method is evaluated in two control settings: *trajectory optimization*, maximizing rewards obtained over the predicted trajectories, and *policy optimization*, in which a trajectory-based policy is learned. [4] exploits variational inference [8] to deal with sparse rewards from the environment. State trajectories are embedded and the learned model is employed for performing hierarchical reinforcement learning: a lower level policy is constrained to be consistent with a predictive model for trajectories, and it is steered by a *model predictive controller* acting at the trajectory latent-space level. Both [10] and [4] obtained promising results, suggesting that modeling of entire trajectories can be effective. However, none of them exploited it for any kind of transfer.

In the context of imitation learning, [3] and [16] employed generative adversarial networks [6] for imitating diverse expert policies, learning meaningful latent representations for experts' trajectories. Nonetheless, the results obtained are limited to imitation and not easily transferable to the model-based reinforcement learning setting.

In [7], the problem of learning a dynamic model based on uncertainty estimates, conditioned on a latent representation of the MDP, is addressed. This model is then employed in the learning procedure of an agent tested for transfer across multiple tasks. The work does not consider trajectories, but only single-step environment transitions. Moreover, the different settings of the MDPs are only implicitly modeled, using latent variables for representing a given setting, without taking into account the case in which some of the parameters of the MDP are known in advance. For instance, in the FES setting, you can infer important information about the dynamics in which the policy will act from features such as patient age, sex or pathology.

3. Research plan

The goal of the research is to investigate the use for model-based reinforcement learning of generative models that can effectively leverage, to improve a target policy in a target MDP, trajectories generated by different policies or in different MDPs. The research will have a theoretical aspect, covering the mathematical formulation and the theoretical analysis, and an experimental aspect, including the development of algorithms and their empirical evaluation. The data on which this evaluation will be done may be real data about autonomous driving and FES, as well as experience collected in standard simulation environments for the approach to be compared with similar methods. Although the theoretical part will be the first one to be addressed, iterations between theory and experiments will be performed.

We divide the research plan into two distinct phases, to be carried out sequentially for an approximate period of, respectively, six and five months starting from November 2018.

In the *first phase*, we will consider the problem of understanding generative models that approximate trajectory distributions under different policies in a given MDP and how to properly leverage them for model-based RL. These generative models should be *flexible*, understanding during their training process how much they have to learn from each one of the similar settings. We will start with a theoretical analysis concerning the limits of generative modeling in using this variety of source policies, with particular attention to sample efficiency. Then, we will turn to the development of an algorithm for dynamically weighting different trajectories during the learning procedure of a generative model, integrating it with model-based reinforcement learning. Afterwards, an implementation will be used to perform experiments that aim to validate the theoretical insights collected in the first months and to possibly obtain new insights to guide a further theoretical analysis. The algorithm will be compared against similar approaches on real data or standard benchmarks, using implementations provided by authors or custom implementations when the former are not available. After an analysis of the results, a *first milestone* will be reached, with the writing of a document synthesizing the results that have been achieved and its possible submission to an appropriate conference, for instance *Neural Information Processing Systems* (NIPS), whose submission deadline is usually in May.

In the *second phase*, we will look into the use of generative models for approximating environment dynamics at the trajectory level across families of related MDPs, again in order to improve the efficiency and transfer capabilities of an agent. The research schedule for this phase is similar to that of the first one: after a theoretical analysis of the mathematical properties of the problem, an algorithmic approach will be devised, implemented and empirically verified. As in the first phase, the empirical evaluation will be performed in comparison to other methods on standard simulation benchmarks and on real data. For the implementations included in both the phases of the research, we plan to use the Python programming language and its vast ecosystem of machine learning software libraries, such as *Pytorch* [12] or *Tensorflow* [1] for implementing parameterized models and their training procedures, *Gym* [2] and *Baselines* [5] from OpenAI to simulate environments and evaluate baselines in a reliable way. Naturally, results and insights obtained in the first phase of the research plan will be used to ease the whole process during the second phase. In the end, the *second milestone* will be reached, consisting in the completion of the master thesis, together with a paper to be submitted to a relevant conference with a compatible submission deadline, such as the *AAAI Conference on Artificial Intelligence* (AAAI) or the *International Conference on Learning Representations* (ICLR). A tentative schedule for the various tasks that compose the research plan is shown in Figure 1.

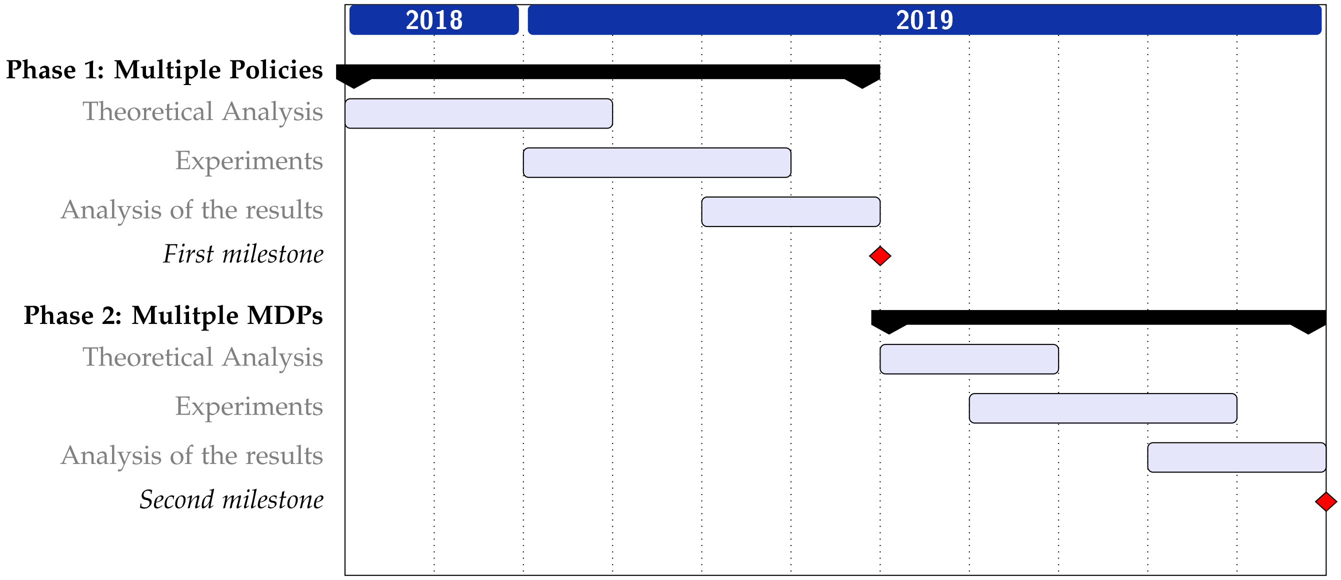

Figure 1: *Gantt chart* picturing the research plan, starting from November 2018 to September 2019. The plan is divided into two phases, corresponding to the two related research problems on the application of generative models for model-based reinforcement learning. The first one considers the presence of multiple policies, while the second one considers the existence of multiple MDPs. Each phase will consist of both a theoretical part, including mathematical formalization and derivation of relevant properties or bounds, and an experimental part. The milestones, namely the completion of scientific papers presenting the results of the research, are represented in the diagram by means of red marks.

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

[2] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *arXiv:1606.01540 [cs]*.

[3] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180.

[4] Co-Reyes, J. D., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., and Levine, S. (2018). Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *arXiv preprint arXiv:1806.02813*.

[5] Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. (2017). Openai baselines. `https://github.com/openai/baselines`.

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[7] Killian, T. W., Daulton, S., Konidaris, G., and Doshi-Velez, F. (2017). Robust and Efficient Transfer Learning with Hidden Parameter Markov Decision Processes. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6250–6261. Curran Associates, Inc.

[8] Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*.

[9] Lynch, C. L. and Popovic, M. R. (2008). Functional electrical stimulation. *IEEE Control Systems Magazine*, 28(2):40–50.

[10] Mishra, N., Abbeel, P., and Mordatch, I. (2017). Prediction and Control with Temporal Segment Models. In *International Conference on Machine Learning*, pages 2459–2468.

[11] Moore, A. W. and Atkeson, C. G. (1993). Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130.

[12] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch.

[13] Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[14] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491.

[15] Sutton, R. S. (1991). Dyna, an Integrated Architecture for Learning, Planning, and Reacting. *SIGART Bull.*, 2(4):160–163.

[16] Wang, Z., Merel, J. S., Reed, S. E., de Freitas, N., Wayne, G., and Heess, N. (2017). Robust Imitation of Diverse Behaviors. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5320–5329. Curran Associates, Inc.

[17] Wiering, M. and Schmidhuber, J. (1998). Efficient Model-Based Exploration. In *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior: From Animals to Animats 6*, pages 223–228. MIT Press/Bradford Books.