Reinforcement Learning Tutorial 1, Week 2

—

Introduction (incl. Solutions)

Sanjay Rakshit, Pavlos Andreadis, Michael Herrmann

January 2025

Overview: The following tutorial questions relate to material taught at the start of week 1 of the 2024-25 Reinforcement Learning course. They aim at encouraging engagement with the course material and facilitating a deeper understanding.

This first tutorial starts with a couple of discussions on the concept of *Reinforcement Learning*. Use this tutorial as an opportunity to engage with your tutor and classmates, and to get a broader sense of the course and the subject area. This is also a good time to start thinking of how to use the resources provided by the course in setting up your personal learning plan.

As weeks progress, tutorials will also involve problem modeling exercises and algorithm applications, among other things. Answers provided to your tutorial questions and exercises will often pose further, "1-step ahead" questions. Consider these when studying and ask your tutor for advice, if you are stuck.

Reinforcement Learning is built around a few central concepts, and many algorithms are merely variations of the same basic approach. Trying to understand how one approach is similar to or different from another can be a good way to build an understanding of the fundamentals and how individual algorithms operate. You can also refer to the course algorithms chart (see the course resources page) to see how different approaches fit into the bigger picture of reinforcement learning. But all that in good time.

Problem 1 — Discussion

"How is Reinforcement learning different from Supervised learning?" Open ended question for discussion. Attempt to write a short answer before the tutorial. It could be interesting to start from the comparison at this link Shaikh, 2017, at the end of section 2, which is questionable. Can you give a better characterisation of the difference between these two types of learning? Consider also Section

1.1 in Sutton and Barto, 2018. Also, discuss the reward hypothesis Silver e.a., 2021 that was mentioned in the first lecture. Is there an analogous hypothesis underlying to supervised learning?

Solution

There are really two key differences: Supervised learning predicts the value of a variable (e.g. a class or continuous variable) and is trained using samples that include values for that variable. Reinforcement learning is trained to maximise an expected return based on a scalar reward signal, and predicts an action and/or a value function.

Furthermore, Reinforcement learning algorithms consider the existence of a dynamical system in how they operate; even when the model itself is not explicitly used (model-free RL). For example, even if you are running a Monte Carlo algorithm that simply accumulates sample returns for each state visit which it then averages to predict the value of each state (a supervised learning task), the algorithm itself takes into account that you have to take actions in the environment until termination of the episode.

One thing to point out about the linked-to article, is the ambiguous statement: "But in reinforcement learning, there is a reward function which acts as a feedback to the agent as opposed to supervised learning."

This in-and-of-itself, does not provide a good idea of how the two areas differ. Consider, that after fitting a model with Supervised Learning, it is not atypical to define a criterion to maximise over the input space. For example, you might have learned a model that predicts the age of a client, and then make a decision based on that and some *reward* function. If the element of control is missing, you are unlikely to want to model this as a Reinforcement Learning problem.

If you have time, it may also be interesting to consider Yann LeCun's famous (controversial) cake analogy LeCun, 2016.

The reward hypothesis suggests that a task that is well specified by a reward function can be solved, if enough training instances (state-action-reward samples) can be generated, and if a suitable reinforcement learning algorithm is in place. In supervised learning, we need an assumption of the form that the data are representative for the function that is to be represented (e.g. by a neural network). We can run into a similar issue in either case because of the amount of data avaiable or because of the suitability of the learning algorithm.

The discussion can also include that the problem with the reward hypothesis is due to goals not being "well defined" in the sense of Markovian RL, i.e. there may be information missing (as rock-paper-scissors it's unknown what action the opponent takes and, thus, what reward can be expected for each of one's own actions), the task may have value structure that cannot be expressed by exponential discounting, or there may be incomparable rewards at different states etc. Likewise in (un)supervised machine learning, there may be missing or conflicting data or nonstationarities. The difference is, however, that RL (to work safely) has more specific assumption need to be true including Markovianity

or exponential discounting, while the less specific points that can be made for (un)supervised learning mostly apply for RL as well.

Problem 2 — Discussion

Provide two different examples of applications of Reinforcement Learning in industry. What aspects of these problems make them solvable by Reinforcement Learning?

It may help to consider the possible state, action and reward spaces for your examples. You may find this link Chan, 2018 useful, as it has a few examples and relevant discussion in section II, although it is now a little outdated. Can you find other examples? Consider also the first paragraph of Section 1.1 in Sutton and Barto, 2018.

Solution

Problems solvable by reinforcement learning need:

- Some description of a (typically, but not always) changing state
- Some description of actions that can be taken (there is a decision to be made that should depend only on the current state; though the selected state description might differ from a natural interpretation of state. E.g. a state when playing ATARI might be the last 4 frames rather than the most recent frame)
- Some description of good/bad outcomes that can occur as a result of your actions (note that inaction can be modelled as an action) and that can be reasonably represented by scalar values
- That the best decisions/actions taken **can not** be computed separately for each state. Typically because they transition you to a different state from where more rewards are expected to be accumulated. However, there are some interesting exceptions, such as in [Boutilier, 2002] (optional!), where the environment is in a specific but unknown state.
- There are many (and in future most likely many more) applications of RL in industry including
  - Self-driving cars
  - Automation and robotics
  - Finance (e.g. trading strategies)
  - Recommendation systems
  - Gaming industry (e.g. NPCs or resource optimisation)
  - Marketing and advertising

- It may be necessary to apply many actions (some possibly in simulation or in parallel such pooling over many self-driving cars) before a good policy can be found. Exploration becomes non-trivial, because certain (combinations of) actions might be risky. Although RL seem quite general as an approach, it is still a good idea to have a some domain knowledge (or a business partner who has).

References

Gary Chan. Applications of reinforcement learning in real world. https://towardsdatascience.com/ applications-of-reinforcement-learning-in-real-world-1a94955bcd12/, 2018 (updated 2023).

JalFaizy Shaikh. Simple beginner's guide to reinforcement learning & its implementation. https://www.analyticsvidhya.com/blog/2017/01/ introduction-to-reinforcement-learning-implementation/, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, http://incompleteideas.net/book/the-book.html, 2018.

Yann LeCun: Cake Analogy, https://medium.com/syncedreview/yann-lecun-cake-analogy-2-0-a361da560dae, 2016.

Boutilier, Craig. A POMDP formulation of preference elicitation problems, 2002.

Silver, D., Singh, S., Precup, D. and Sutton, R.S., 2021. Reward is enough. *Artificial Intelligence* **299**, p. 103535.