Can the replayed sequences during sleep can be modelised by a reinforcement learning algorithm – Rapport de stage

Paul Ecoffet

15 avril 2016

Abstract

On the neurobiology side, replays of behavioral sequences has been observed in the hippocampus during sharp wave ripples complexes and is thought to be a mechanism involed in memory consolidation and learning. On the computer science side, some reinforcement learning algorithms also use replays of behavioral sequences to improve the learning speed. One can ask if this algorithms behave in the same way that hippocampus replays of sequences, and if these algorithms can model the replay mechanisms observed in the hippocampus. For instance, can those algorithms explain why certain sequences are replayed instead of others? We'll show that that ** we'll see **.

Introduction

Reinforcement learning

Reinforcement Learning is a kind of learning mechanism where an agent tries to maximise a reward by doing some specific action in an environment (R. S. Sutton & Barto, 1998). The environment is defined by *states* in which the agent can be. The agent can be in one state at a time. In each state, the agent can do *actions* from its repertoire. When doing a specific action in a specific state, the agent receives a *reward*, which is a real. The reward can be positive or negative. The goal of the agent is to maximise the amount of reward it receives.

To evaluate a state, the agent can estimate the *value* of it. The value of a state is the expected reward the agent will receive if he is in this state plus the future rewards it can get from future states (with a discount factor).

Notation

Learning by replay

In reinforcement learning

Reinforcement learning methods like Q-learning or Sarsa converge slowly and need a lot of samples to be efficient. These algorithms, based on the TD(λ), use the sample once to improve their solution, then discard it (Adam, Busoniu, & Babuska, 2012).

To improve the efficiency of reinforcement learning methods, experience replay can be use. Vanseijen and Sutton (2015) offers a good analysis about learning by replay. The goal of replay is to use the maximum of information an experience offers. Each sample is used several times, improving the solution several times. Compared to TD(0), replay techniques give a better convergence to the optimal solution with the same number of experiences. Reinforcement learning methods using replay are more expensive than TD(λ) both in memory and in computations, though the memory and computational power needed can be reduced a lot as Vanseijen and Sutton (2015) shows.

Algorithm 1: TD(0)

```
while test do
    works
end
```

Planning and replay?

RL algorithms with planning use a model of their environment to infer the value of a behaviour without actually doing it. The model is commonly a transition table between the possible states in the environment according to the action done by the agent. For exemple, if the agent has to solve a maze and has access to a transition table, then it can look up the table and know that if he is in the case $(3, 5)$ and it does the action "go to the east", it will be in $(3, 6)$.

The model can be either directly accessible (given by the developer) or inferred by the agent. When the model is inferred, the agent builds explicitly the transition table according to the experiences he has in its environment. In this document, we will mainly talk about the Dyna algorithms. Dyna is a class of RL algorithms where the model is inferred by the agent. There is two phase during an iteration of Dyna. The first phase is when the agent interacts with its environment: it observes its state and reward, do an action according to its policy and go to the next state. It also updates its model of the environment according to this new experience. The second part is the *planning* phase. During the planning phase, the agent improves its value function estimation with simulations according

to its model: The agent generates a sample state, an experience, according to some rules (a probability distribution) and uses its model to compute the value of this sample thanks its model. Then, it updates its solution according to this simulation (R. S. Sutton, Szepesvári, Geramifard, & Bowling, 2012). It is interesting to note that the distribution used has no influence on whether the solution will converge or not, but has an influence on the speed of convergence. Therefore, choosing a good distribution can greatly improve the performance of Dyna. R. S. Sutton et al. (2012) shows that choosing only already explored samples can have a good impact on performances. Therefore, Dyna can be efficient with a selection of already experienced samples, the planning phase is then a *replay* phase.

Vanseijen and Sutton (2015) highlights even more the strong similarity between RL methods using replay and RL methods using planning. First of all, both of them can be considered as model-based algorithms. Indeed, planning methods using an inferred model use an integrated version of samples experienced over time, thus they are by definition model-based algorithms. RL algorithms using experience replay do not store an explicit model, with transition tables and reward estimation. Yet, they store for each sample the state they were in, the reward they received and the next action they did. These data can be considered as a model. They implicitly contains an inferred model of the environnement. Transitions and rewards are available through all the samples the agent experienced and stored in memory (Vanseijen & Sutton, 2015).

In addition to that, Vanseijen and Sutton (2015) shows a complete equivalence in the value function approximation at each iteration between a TD(0) inspired algorithm with replay and Linear Dyna, a model-based RL algorithm. Though Linear Dyna is a model based algorithm and thought as a "looking ahead" algorithm and the replay algorithm as a "looking backward" algorithm, they do the exact same calculation. The Linear Dyna variation Vanseijen and Sutton (2015) presents is a batch Linear Dyna. Each iteration of planning compute using an integrated version of all the states and transition experienced. One can ask if different variations of Linear Dyna can also be considered as learning by replay methods. For instance, R. S. Sutton et al. (2012) reviewed different Linear Dyna algorithms which look at one possible state per planning iteration. They shows that using already experienced states and prioritizing about states where the difference between the estimated value and the real value was the greatest is an efficient strategy.

- Can one use Linear dyna with prioritised sweeping as a learning by replay method?

In vivo

Place cells Place cells are high level integrative neurons in the hippocampus. They have a specific receptive field called a place field. When the animal is in

the receptive field of a place cell, the place cell spikes. It is on these cells that the replay of behavioral sequences is observed during sleep.

Replay Replays of sequences have been observed in the rat hippocampus after navigation tasks (Wilson & McNaughton, 1994; Skaggs & McNaughton, 1996; Davidson, Kloosterman, & Wilson, 2009; Gupta, van der Meer, Touretzky, & Redish, 2010). The place cells whose activations were highly correlated in time are also activated during sleep and respect the same correlation (Wilson & McNaughton, 1994). Replays of place cells activation sequences that occurs during the day can also be observed either forward (Skaggs & McNaughton, 1996) or even backward (Gupta et al., 2010). Coherent sequences that have never been done are also "replayed" by the rat (Gupta et al., 2010). Though, those articles do not propose explantion about why these specific sequences were chosen or about the length of these sequences. Gupta et al. (2010) shows that the recency of the experience is a wrong hypothesis but do not propose other explanation. We will try to provide a criterion to explain why specific sequences are replayed and not others. to check

- Replay during sleep and during hesitation
- Replay can be either forward or backward
- Replay can also be action that has never be done? –> Planning somewhat?
- Replay aren't necessary about the most recent event?
- Is "surprise" a good crterion?

Model navigation learning

We can model place cells in a reinforcement learning algorithm by using a tiling of the environment (see Figure 1) as **tamosiunaite_path-finding_2008** have done. Each feature of the feature vector (the state of the agent) represents the level of activation of a tile: a gaussian-like function of the distance of the agent from a kernel. For instance, the i^{th} feature of the feature vector will be the distance from the kernel k_i according to a gaussian function (see Figure **??**). The value of the i^{th} feature can be interpreted as the firing rate of a place-cell with a receptive field centered in k_i.

What are the sequences that are replayed? How are they selected?

Gupta et al. (2010) shows that the sequences that are replayed are not necessarly the most recent experiences. It shows also that the sequences that are replayed

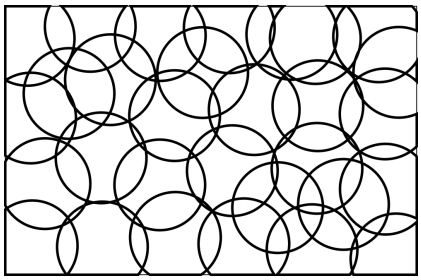

Figure 1: Tiling of an environment
Each disk can be thought as the receptive field of a place cell

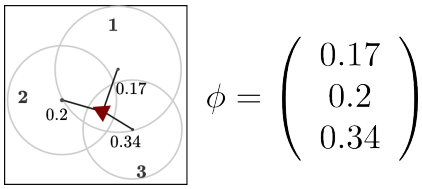

Figure 2: Features representation

can be either forward or backward. In forward replays, the sequence is replayed in the same order as the sequence the rat has done. In backward replays, the sequence is done in the reverse order. What is the explanation for the selection of the sequence which is replayed, and its order? No criteria are proposed in Gupta et al. (2010). Yet, we can make interesting hypothesis knowing that R. S. Sutton et al. (2012) suggests that there are very good samples to replay in a Dyna algorithm, and those sample are the sample where the difference between the expected value of a state and its real value is the greatest. It is consistant with Gupta et al. (2010) which states that the most common replays are the one that leads to the reward or comes from the reward.

Method

- Pourquoi x place-cells?
- Pourquoi ce radius?
- Quel densité?
- Quel type d'activation?

Results

Conclusion

Limitations

References

Adam, S., Busoniu, L., & Babuska, R. (2012, March). Experience replay for real-time reinforcement learning control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(2), 201–212. doi:10.1109/TSMCC.2011.2106494

Davidson, T. J., Kloosterman, F., & Wilson, M. A. (2009, August). Hippocampal replay of extended experience. *Neuron*, *63*(4), 497–507. doi:10.1016/j.neuron.2009.07.027

Gupta, A. S., van der Meer, M. A., Touretzky, D. S., & Redish, A. D. (2010, March). Hippocampal replay is not a simple function of experience. *Neuron*, *65*(5), 695–705. doi:10.1016/j.neuron.2010.01.034

Skaggs, W. E. & McNaughton, B. L. (1996, March 29). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science (New York, N.Y.) 271*(5257), 1870–1873.

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement learning: an introduction* (Nachdr.). Adaptive computation and machine learning. Cambridge, Mass.: MIT Press.

Sutton, R. S., Szepesvári, C., Geramifard, A., & Bowling, M. P. (2012). Dyna-style planning with linear function approximation and prioritized sweeping. *arXiv preprint arXiv:1206.3285*. Retrieved March 18, 2016, from http://arxiv.org/abs/1206.3285

Vanseijen, H. & Sutton, R. (2015). A deeper look at planning as learning from replay. In *Proceedings of the 32nd international conference on machine learning (ICML-15)* (pp. 2314–2322). Retrieved April 1, 2016, from http://machinelearning.wustl.edu/mlpapers/paper_files/icml2015_vanseijen15.pdf

Wilson, M. A. & McNaughton, B. L. (1994, July 29). Reactivation of hippocampal ensemble memories during sleep. *Science (New York, N.Y.) 265*(5172), 676–679.