Reinforcement Learning

Other topics and successful applications

Mario Martin

CS-UPC

May 14, 2020

Other topics

Topics not discussed

- There are still some hot topics we haven't covered
  - Model based Reinforcement Learning
  - Inverse Reinforcement Learning (IRL)
  - Partial Observability: Memory approaches
  - Exploration vs. Exploitation
  - Hierarchical reinforcement learning
  - Curriculum learning
  - Transfer Learning
  - Meta-Learning
  - Multi Agent Reinforcement Learning (MARL)
  - Robotics applications
- Today, we'll mention some of them
- We will also see some successful applications of RL

Model based Reinforcement Learning

- Model-free methods do not need transition probabilities ($P^{a'}_{s,a}$)
- In case where we have these transitions we can do more things:
  - Planning: You can learn with complete backups using probabilities instead of samples. Or with different degrees of depth (n-steps)
  - Rollouts: You can learn generating predicted trials
  - Monte-Carlo Tree search: Combination of two previous cases.
- In case you don't have the model, learn it explicitly from samples and use it as it were given (samples taken at the beginning or while learning the policy).
- Nice classification and comparison of latest MBRL algorithms.
- Nice interactive paper on a MBRL proposal
- Goal is *sample efficiency*

Inverse Reinforcement Learning (1)

- Consists in, given a policy (or examples of the target policy), obtain the reward function.
- In some cases we cannot apply RL because the reinforcement function is unknown or too complex (f.i. driving)[1]
- But we have examples of the policy we want to learn
- In these cases, IRL allows (1) to catch the reward function and (2) from reward function that learn the policy
- More robust (less drifting and more general) than learning from examples (that is known as *Behaviour cloning*)

[1] We assume that reward function is easy to design but see how agents cheat in AI

Inverse Reinforcement Learning (2)

- State of the art methods (f.i. AIRL and GAIL) use Adversarial Networks in to generate the reward function (or examples)
- See also impressive DeepMimic video presentation [warning, RL but from examples, not using IRL]

Partial Observability

- In a lot of cases the agent has not complete information of the *true* state and uses its perception as state.
- The problem is not anymore an MDP.
- How to solve these case?
  1. Formalize as a POMDP: MDP extended with set of observations O and probability of each observation given the true state. Agent work with a *belief vector* of probabilities of being in each state. Solve with dedicated algorithms
  2. Works with memory as a way to disambiguate the true state. Simple approaches like window of last n perceptions (DQN), or more interesting ones using LSTM

Exploration

- Very sparse reward is one of the main problems in RL
- In this cases we want to explore *efficiently* the state space
- Complex mechanisms for exploration based on different criteria
  - Less explored state, action pairs (Counting)
  - Higher changes in value of state action pair
  - Bases on recency of last exploration
  - Uncertainty on estimation of values
  - Error in an agent's ability to predict the consequence of action (curiosity) ... but avoiding procrastination.
  - Reachability of states criteria.
  - ...
- Some of this approaches are under the umbrella of *Intrinsic Motivation*. See review here.
- See more references about the topic in course web page

Hierarchic RL

- Also helps in sparse rewards, but also useful for transfer learning.
- Natural way of learning.
- In some cases a complex task can be decomposed in simpler tasks.
- Learning is simplified when first these tasks are learnt.
- Several ways to find that:
  1. Using subrewards for subactions (reward shaping)
  2. **Discover them automatically**
- Actions can be reused to learn other tasks
- See references about the topic in course web page

Curricular Learning

- Some tasks are too difficult to be learnt from a sparse reward.
- In animals, learning is done step by step. You can only learn something that you are ready to learn
- Curriculum learning propose subgoals to be learnt in sequence (curriculum) to solve a complex tasks.
- See for a nice review of latest techniques
- See more references about the topic in course web page

Transfer RL

- Can we extend knowledge generated in one task to a different task?
- Changes in the task: different dynamics, different reward and/or different actions.
- Different kinds of information to transfer to transfer (Q-values, policy, reward, samples, model, features, etc.)
- Example: IRL for task disentangled from actions (AIRL)

Meta-Learning in RL

- An agent not only has to learn a single task. It has to efficiently learn a set of different tasks.
- Learning of each task has to be consistent and (hopefully) helpful for learning other tasks.
- Very popular in two last years, in ML and RL in particular (See course CS330 from Stanford here)
- See also a specific introduction and review of latest approaches for RL

Multi-agent RL (I)

- All cases we have seen assume the agent is the only one that executes actions in the environment
- In cases where there are also other agents, can we learn?
- Use of game theory and assumptions about the other agents (see f.i. old introduction.
- Depending on the goals of the agent, we have cooperative or competitive learning
- Two-players games are an special case.
  - Backgammon: Neurogammon (Tesauro 1994)
  - Go: Alpha-go (Silver et al. 2016) and Alpha-go Zero (Silver et al. 2017)
  - Chess: AlphaZero (Silver et al. 2017)

Multi-agent RL (II)

- One of the hottest topics at the moment.
- Very Fun example: hide and seek
- Selected bibliography on the topic of MARL
- Some Environments to play with
- Also game application in Star-Craft

Links with neuro-physiology

- In Artificial Intelligence it is important to compare techniques with actual techniques used by humans and animals
- Reinforcement learning has its roots in intuitive idea of learning in animals.
- Lately, a lot of papers support that RL is implemented in the brain
  - Link of RL with actual learning in brain
  - Dopamine as reward
  - Dopamine implements TD error.
  - Support for dopamine acting as distributed value estimation

Some succesful aplications of RL

Applications

- Games: Backgammon, Go, Chess (competition), Star-Craft, Dota 2, Poker (Pluribus)
- Robotics: Walking, Manipulation (also here), etc.
- Medicine: Review. Example: Sepsis treatment, or ventilation
- Drug dessign: For instance here.
- Physics
- Recommender systems
- Finances
- Optimization in general, f.i control power, or for IoT,
- Mathematics: For instance, Logic profs
- Natural Language processing: Summarizing texts.