There are still some hot topics we haven’t covered

- Model based Reinforcement Learning
- Inverse Reinforcement Learning (IRL)
- Partial Observability: Memory approaches
- Multi Agent Reinforcement Learning (MARL)
- Hierarchical reinforcement learning
- Exploration vs. Exploitation
- Transfer Learning
- Robotics

Today, we’ll mention some of them
Model-based Reinforcement Learning

- Model-free methods do not need transition probabilities ($P_{s,a}^{a'}$)
- In case where we have these transitions we can do more things:
  - Planning: You can learn with complete backups using probabilities instead of samples. Of with different degrees of depth (n-steps)
  - Rollouts: You can learn generating predicted trials
  - Monte-Carlo Tree search: Combination of two previous cases.
  - Smarter exploration
- In case you don’t have the model, learn it explicitly from samples and use it as it were given.
Inverse Reinforcement Learning

- Consists in, given an optimal policy (or examples of the policy), obtain the reward function.
- In some cases we cannot apply RL because we reinforcement function is very complex.
- But we have examples of the policy we want to learn.
- In these cases, IRL allows to get the reward function and from that learn the policy.
- More robust than learning from examples.
In a lot of cases the agent has not complete information of the 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 state. Simple approaches like window of last $n$ perceptions, or more interesting ones using LSTM

3. Find a stochastic policy using policy search methods or evolutionary methods
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

Useful for transfer learning
Can we extend knowledge generated in one task to a different task?
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.

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)