Reinforcement Learning David Silver - Lecture 8 Notes: Integrating Learning and Planning

Name: Eli Andrew

- **Advantages of Model-based RL**
  - Can efficiently learn model by supervised learning methods
  - Model is like the teacher that provides the supervised learning
  - Example:
    - Domain where learning policy or value function is hard (i.e. chess)
    - Many different states
    - Has sharp value function (single move of a piece can change from won position to lost position)
    - Hard to learn this type of value function directly
    - Model is straight forward - essentially just rules of the game
    - If you can use model to "look ahead" you can estimate the value function by planning (by tree search)
    - This is easy compared to learning the value function because you are just learning that you have 0 reward for all positions except check mates and draws
    - As compared to learning the value function where you are evaluating how likely you are to win from all the many configurations of the pieces
  - Model can be a more useful (and compact) representation of the information than a value function
  - Can reason about model uncertainty
    - Helps you see what you know and don't know about the world
    - This way you can strengthen your true understanding of the world and not just your current understanding
  - Disadvantage: learn model and then construct value function (2 sources of error)
- **What is a model**
  - Model M is a representation of an MDP $< S, A, P, R >$ parameterized by η
  - Assume state space and action space are known
  - Model $M =< P_\eta, R_\eta >$ represents state transitions $P_\eta \approx P$ and rewards $R_\eta \approx R$

$$S_{t+1} \sim P_\eta(S_{t+1}|S_t, A_t)$$
$$R_{t+1} = R_\eta(R_{t+1}|S_t, A_t)$$

- **Model learning**

- Goal: estimate model M_η from experience $\{S_1, A_1, R_2, \ldots, S_T\}$
- Supervised learning problem

$$S_1, A_1 \to R_2, S_2$$
$$S_2, A_2 \to R_3, S_3$$
$$\ldots$$
$$S_{T-1}, A_{T-1} \to R_T, S_T$$

- Learning $s, a \to r$ is a regression problem
- Learning $s, a \to s'$ is a density estimation problem (since it is likely stochastic we are learning the distribution)
- Pick loss function (MSE, KL divergence, ...)
- Find parameters η that minimize empirical loss

- **Examples of Models**
  - Table lookup model
  - Linear expectation model
  - Linear Gaussian model
  - Gaussian process model
  - Deep belief network model
  - ...
- **Sample-based Planning**
  - Use the model *only* to generate samples
  - Unlike DP where you look at probabilities of transitions and integrate over the probabilities
  - You sample experience from the model (rather than knowing all the transition probabilities)

$$S_{t+1} \sim P_\eta(S_{t+1}|S_t, A_t)$$
$$R_{t+1} = R_\eta(R_{t+1}|S_t, A_t)$$

  - Apply *model-free* RL to samples: Monte-Carlo control, Sarsa, Q-learning, etc.
  - Sample based planning methods are often more efficient
  - Planning is essentially done by solving for the simulated experience drawn from the agents imagined world (its model)
  - Sampling is more efficient, even in the case when you know the entire model, because you are essentially focusing on the things that are most likely to happen