6.882 Proposal: Bayesian Reinforcement Learning

Vickie Ye and Alexandr Wang

1 Background and Motivation

Because of the recent excitement regarding DeepMind's AlphaGo AI, we wanted to explore reinforcement learning, in a Bayesian framework. We will use the Strens's ICML-2000 paper, *A Bayesian Framework for Reinforcement Learning* as a guideline as to how to reason about the toy reinforcement learning problem of navigating an unknown maze with a Bayesian formulation.

In the reinforcement learning problem, the agent must explore the surroundings and determine the behavior that maximizes the expected return. In this case, our agent must discover the structure of the maze while minimizing the time required to find the end. A Markov decision process (MDP) model is traditionally used to model the interactive system over S the set of states, A the set of actions, $R : S \times A \to \mathbb{R}$ the reward function, and T the transition probabilities defined as

$$T(s, a, s') = P(X_{t+1} = s' | X_t = s, Y_t = a). \tag{1}$$

We also define a quality function Q as

$$Q(s, a) = \mathbb{E}[R(s, a)] + \gamma \sum_{s'} \max_{a'} Q(s', a') \tag{2}$$

where γ is the time-discount factor (Strens, 2000, p.3).

Our optimal behavior is then the policy that maximizes the quality function. Many approaches for finding the best policy uses dynamic programming in order to estimate the transition probabilities. The maximum likelihood estimate for $T(s, a, s')$ is the proportion of times the action a in state s leads to s', and the estimate of $\mathbb{E}[R(s, a)]$ is the average reward received whenever (s, a) is taken. This requires the keeping track of all pairs of states and actions - storage of large sparse data. In Strens (2000), the author places a prior over and estimates the posterior transition probabilities. At each time step, the author uses the current model over the system to estimate the quality function Q^* and choose the best action.

2 Plans and Consideration

For the two of us, our plan to implement this model will be the following:

- Implement Bayesian Markov decision processes (MDPs) as described in the Stren paper. This includes implementing the representation of the posterior, generating the prior, and posterior updates.

- Implement hypothesis generation using the posteriors over the MDP parameters as described in the Stren paper.
- Replicate paper results in three applications discussed in the paper ("Chain", "Loop", and "Maze" toy problems).
- Apply implementation to our maze problem and compare to non-Bayesian methods (NEED SOMETHING TO COMPARE TO)

The major risks of this project come from

References

Strens, M. (2000). A bayesian framework for reinforcement learning. In *International Conference on Machine Learning*.