Data Mining Project

Specification

Rasmus Holm

2017-09-09

Reinforcement Learning

Reinforcement learning problems involve learning what to do in a dynamic environment as to maximize the numerical rewards. This is done, similar to how humans learn many things, by trial and error, the learner is not guided by a supervisor like in supervised learning. The environment is formulated as a Markov decision process (MDP) and I assume a finite MDP which consists of

- a finite set of states S,
- a finite set of actions A,
- a reward function $R_a(s, s')$. The reward of going from state s to state s' by action a,
- a transition function $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$,
- a discount factor γ that determines the importance of future vs. present rewards.

The goal is to find a policy π, the agents action selection model, that maximizes the expected future reward.

Project Proposal

My proposal is to take some open source game (e.g. snake) and then implement/apply reinforcement learning algorithms to see if a learning agent can be trained sufficiently in a reasonable time. You can find the algorithms that are of interest below. I was thinking of implementing 1-3 of the proposed algorithms and compare their performance. Other factors to analyse are

- How do changes to the reward function influence the behavior of the agent?
- How do changes to the state representation influence the performance?

Algorithms

I am interested in three approaches to solving the reinforcement learning problem which are *Dynamic Programming*, *Monte Carlo Methods*, and *Temporal Difference Learning*. These families of algorithms are all model-free, i.e. the agent does not have an internal model of the environment. Possible algorithms that have caught my interest are:

- Policy Iteration
- Value Iteration
- Monte Carlo Prediction
- Monte Carlo Control
- TD(λ)
- Sarsa(λ)
- Q-learning