

RL and Planning Under Uncertainty (ANU, Sem2, 2008)
Lab3: Monte Carlo and TD(λ) for Tic-Tac-Toe & Othello
Lab Instructor: Scott Sanner

READ THE SUTTON AND BARTO BOOK THROUGH CHAPTER 8!!!

Agenda:

1. Familiarize yourself with games (i.e., Othello if you haven't seen it). Implement TD(λ) within current code framework, experiment with different parameter settings and its effect on learning rate.
2. Discuss properties of different game domains and how learning algorithms must be adapted to each case. Two important lessons:
 - a. **Real-world RL is as much an art as it is a science! May take some effort to find tradeoffs / tune parameters for optimal performance.**
 - b. **Function approximation is required in large-scale applications: need (to learn) good set of features and/or flexible (e.g., non-linear) function approximation to approximate optimal value function.**

No amount of algorithm tuning can correct for an impoverished set of features or overly-limited function approximation method! These issues trump the MC vs. TD(λ) vs. other RL method debate.

Student lab work:

1. Run Tic-Tac-Toe and Othello (*Main* class in respective directories, gives usage when run with no parameters), watch Monte Carlo train.
2. See method *endgame(.)* in class *rl.svrrl.TDPlayer*. Implements Monte Carlo learning... extend to handle offline TD(λ) (i.e., update end of the game only).
3. Experiment with performance on Tic-Tac-Toe and Othello with varying λ , learning rate α , and Boltzmann softmax noise parameter. Note the early learning performance and how λ helps or hurts this. $\lambda=0$ for TTT vs. Othello?
4. What modifications would you make to improve performance?

For further reading:

===

Reinforcement Learning (Sutton and Barto)

<http://www.cs.ualberta.ca/~sutton/book/the-book.html>

Chapter 11 has good coverage of case studies including games (Backgammon and Checkers).

===

Temporal Difference Learning and TD-Gammon (Tesauro)

<http://www.research.ibm.com/massive/tdl.html>

Description of world's best RL algorithm for Backgammon: TD-Gammon (also discussed in Sutton & Barto, Ch. 11)

===

Overview of Logistello (Buro)

<http://www.cs.ualberta.ca/~mburo/log.html>

Description of world-class Othello player and various RL / search methods used.

===

Overview of Computer Go

http://en.wikipedia.org/wiki/Computer_Go

Wikipedia seems to have the most current entry on general Computer Go methods. Lot's of good historical references.

Original UCT (Upper Confidence Tree generalization of UCB – for one-shot bandits – to handle sequential decisions)

<http://zaphod.aml.sztaki.hu/papers/ecml06.pdf>

MoGo: UCT and other ideas for Go, very strong 9x9 player

<http://www.lri.fr/~gelly/MoGo.htm>