SuperPAC: Extended Reinforcement Learning for Pacman

Justin Bare
In assignment #3:

- python pacman.py -p PacmanQAgent -x 2000 -n 2010 -l smallGrid

- Same world, over and over
Now:

- Pacman is not a common slave…
- He is a gladiator!
  - Adaptable to very different situations
- To model this:
  - Multi-task RL with Hierarchical Bayesian Model
Multi-task RL:

• Pacman still experiences several worlds (MDP’s) in sequence
• But the worlds differ significantly in:
  – Transition dynamics: $T(s,a,s')$
  – Reward functions: $R(s,a,s')$
• Optimal policies differ significantly
Hierarchical Bayesian Model:

Prior distribution over world types

- World Category 1
  - World 1
  - World 30

- World Category 2
  - World 2

- World Category N
  - World 17

• Knowledge transfer within category => better policy is learned faster!
Implementation:

• Most of the learning algorithm is built

• Surprises
  – Coding based off of a research paper can be difficult
  – Changing the Pacman framework may be unfeasible given the deadline
    • Gridworld may be used for testing instead
Experiments:

• Planning comparison to:
  – Q-learning
I’ve learned:

• Approximate Bayesian inference
• Sampling techniques
• Dirichlet Process
Acknowledgements:

- Mike Chung
- “Multi-Task Reinforcement Learning: A Hierarchical Bayesian Approach” -