Knowledge-Based Support Vector Regression for Reinforcement Learning

Richard Maclin
rmaclin@d.umn.edu
Computer Science Department
University of Minnesota – Duluth

Jude Shavlik, Trevor Walker, Lisa Torrey
{shavlik.twalker,torrey@cs.wisc.edu}
Computer Sciences Department
University of Wisconsin – Madison

This research was supported by DARPA IPTO grant HR0011-04-1-0007 and US Naval Research grant N00173-04-1-G026.

Two Approaches to Creating Intelligent Agents

**Learning from Experience**
- IF At First Junction THEN Move South
- Bottom Line: Hand coding solutions to real world problems requires LOTS of instructions AND those instructions have to be right (and hopefully general)

**Learning from Instruction**
- IF At Second Junction THEN Move South
- Combined

**Advice-Taking Learning**
- Idea: combine teacher instructions (advice) with learning from experience
- Advantages:
  - Fewer experiences needed
  - Learner can use experience to refine/correct advice

Desiderata for Advice-Taking Systems:
- Human observer expresses advice “naturally” and w/o knowledge of ML agent’s internals
- Agent incorporates advice directly into function it is learning
- Additional feedback (rewards, more advice) used to refine learner continually

**Support Vector (Kernel) Regression**
- Find a function f(x) to fit set of example data points
- Problem phrased as constrained optimization task
- Solved using LP problem solver

**Knowledge-Based Kernel Regression**
- In addition to sample points, give advice:
  - If (x ≥ 3) and (x ≤ 5) Then y ≥ 5
  - Rules add constraints about regions

Constraints added to LP and a new solution (with advice constraints) is constructed
Note: advice need not be followed completely

**RoboCup Soccer Simulator Task: KeepAway**
- Object: yellow team, keep the ball away from the blue team
- Learn: player with ball learns whether to hold ball or pass to a teammate
- State: inter-player distances & angle
- Action: hold or pass
- Reinforcement: +1 for each time step

Sutton & Stone (2001) demonstrated RL can be effectively used on this task

- **Reinforcement Learning**
  - Given a task environment
  - States of the world
  - Actions that can be performed
  - Reinforcements (feedback)
    - +100 – get money
    - -100 – eaten by alligator
    - -1 – run into wall
    - 0 – otherwise
  - Do
    - Learn policy to maximize total future reward by exploring environment
    - Learn Q(s,a) function – the expected future reward for performing action a in state s

- **Background**

References (more in paper)