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,ltorrey@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.

Abstract

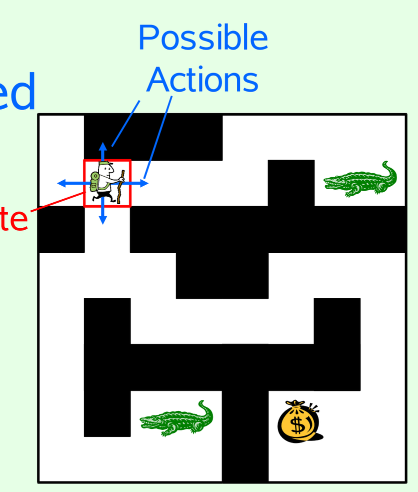
Reinforcement learning (RL) methods have difficulty scaling to large, complex problems. One approach that has proven effective for scaling RL is to make use of advice provided by a human. We extend a recent advice-giving technique, called Knowledge-Based Kernel Regression (KBKR), to RL and evaluate our approach on the *KeepAway* subtask of the RoboCup soccer simulator. We present empirical results that show our approach can make effective use of advice. Our work not only demonstrates the potential of advice-giving techniques such as KBKR for RL, but also offers insight into some of the design decisions involved in employing support-vector regression in RL.

Reinforcement Learning

Background

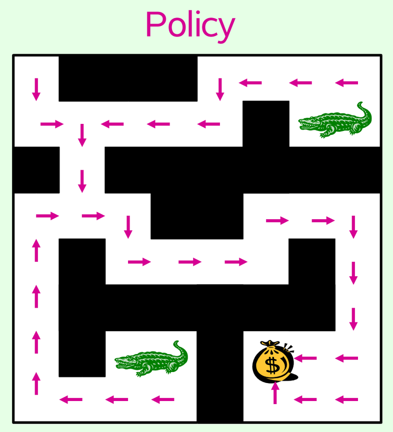
Given a task environment

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

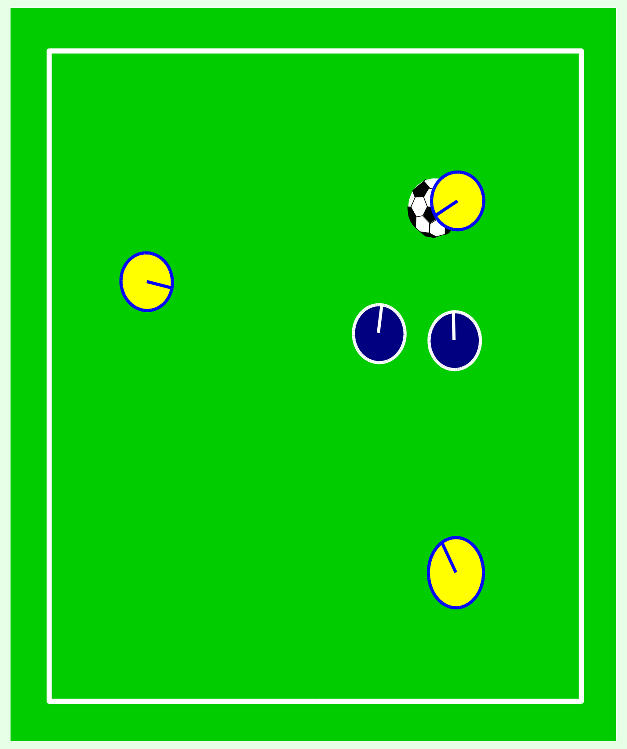
Learn *policy* to maximize total future reward by exploring environment



Learn Q(s,a) function – the expected future reward for performing action *a* in state *s*

RoboCup Soccer Simulator Task: *KeepAway*

Testbed



Object: yellow team, keep the ball away from the blue team

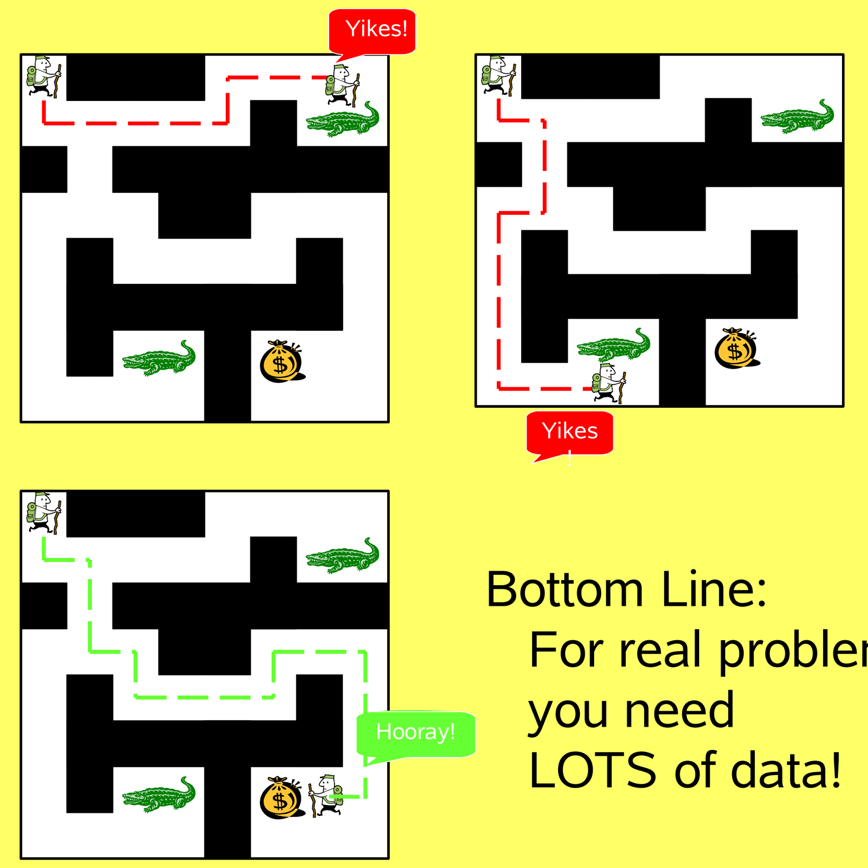
Learn: player with ball learns whether to hold ball or pass to a teammate

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

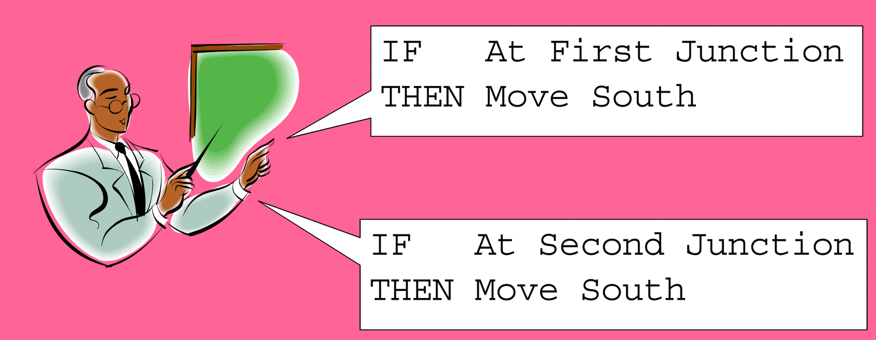
Two Approaches to Creating Intelligent Agents

Learning from Experience



Bottom Line:
For real problems you need LOTS of data!

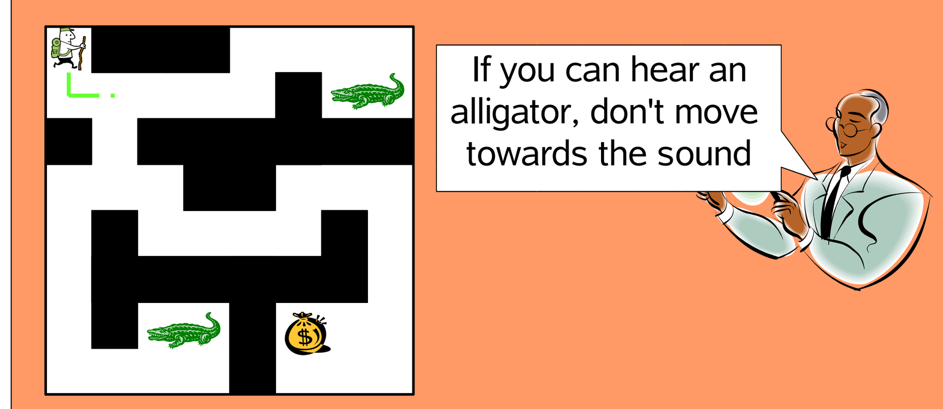
Learning from Instruction



Bottom Line:
Hand coding solutions to real world problems requires LOTS of instructions *AND* those instructions have to be right (and hopefully general)

Combined

Advice-Taking Learning



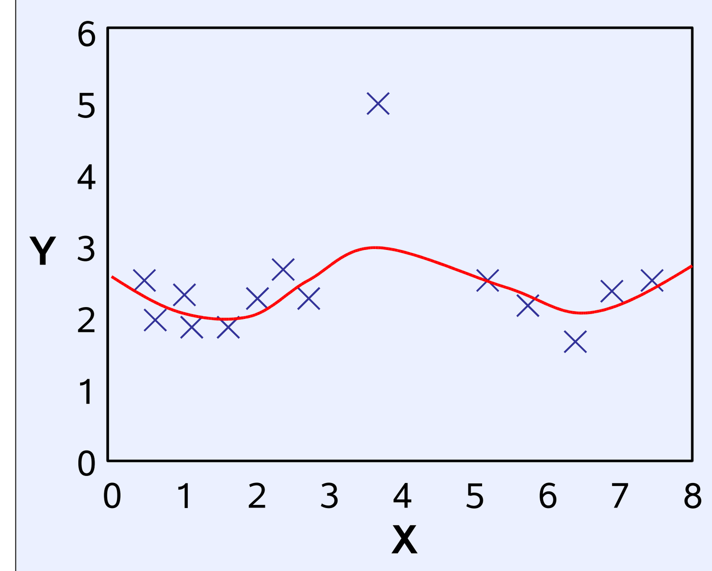
Idea: combine teacher instructions (advice) with learning from experience

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

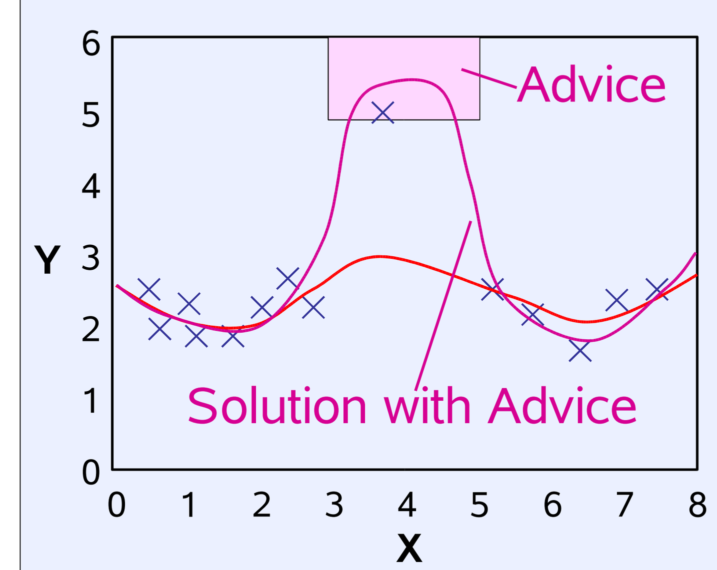
Background



Find a function *f(x)=y* 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 $(x \geq 3)$ and $(x \leq 5)$
Then $y \geq 5$

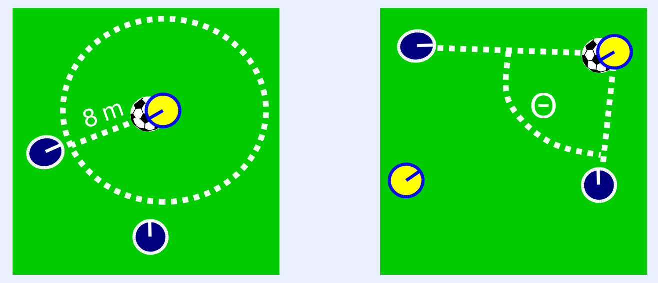
Rules add constraints about regions

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

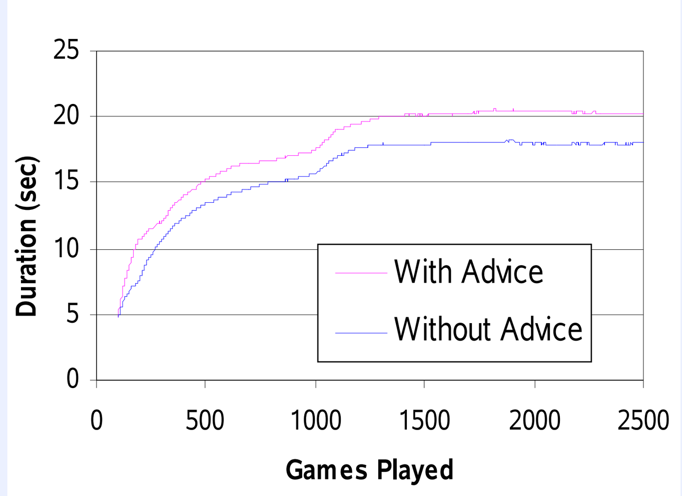
KBKR for Reinforcement Learning

Key refinements for effective use of KBKR:

- Define legal range for input features in advice – otherwise advice has to work even for input values that are not possible
- Strongly penalize threshold of learned function – otherwise often simply learn to predict average Q value
- Tile-coding features very useful – similar to those used in Sutton and Stone, 2001
- Useful to allow dynamic properties in advice (e.g., average Q value) – allows advice to change as learned Qs change



Hold Advice *Pass Advice*



References (more in paper)

1. **R. Maclin, J. Shavlik, L. Torrey, T. Walker & E. Wild (2005).** Giving Advice about Preferred Actions to Reinforcement Learners Via Knowledge-Based Kernel Regression. *AAAI '05.*
2. **L. Torrey, J. Shavlik, T. Walker, & R. Maclin (2005).** Using Advice to Transfer Knowledge Acquired in One Reinforcement Learning Task to Another. *ECML'05.*
3. **P. Stone and R. Sutton (2001).** Scaling reinforcement learning toward RoboCup Soccer, *ICML '01*.
4. **R. Maclin & J. Shavlik (1996).** Creating Advice-Taking Reinforcement Learners. *MLJ* 22: 251-281.
5. **R. Maclin & J. Shavlik (1994).** Incorporating Advice into Agents that Learn from Reinforcements. *AAAI '94.*