



The RL Competition as Class Project



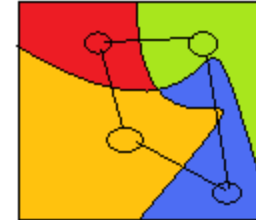
Presentation by Tom Walsh

Based on a class project from **Michael Littman's**
Sequential Decision Making course at
Rutgers University, Spring 2009.

John Asmuth, Monica Babes, Xinyi Cui, Sergiu Goschin, Baiyang Liu,
Chris Mansley, Paul Ringstad, Kevin Sanik, Brian Schubert, Daniel Shields,
Ravneet Singh, Tingting Sun, Fengming Wang, Ari Weinstein, John Wilder,
Michael Wunder, Yan Xiong, SaeHoon Yi

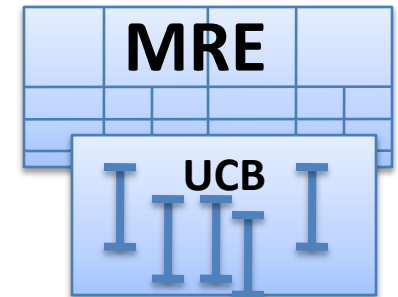
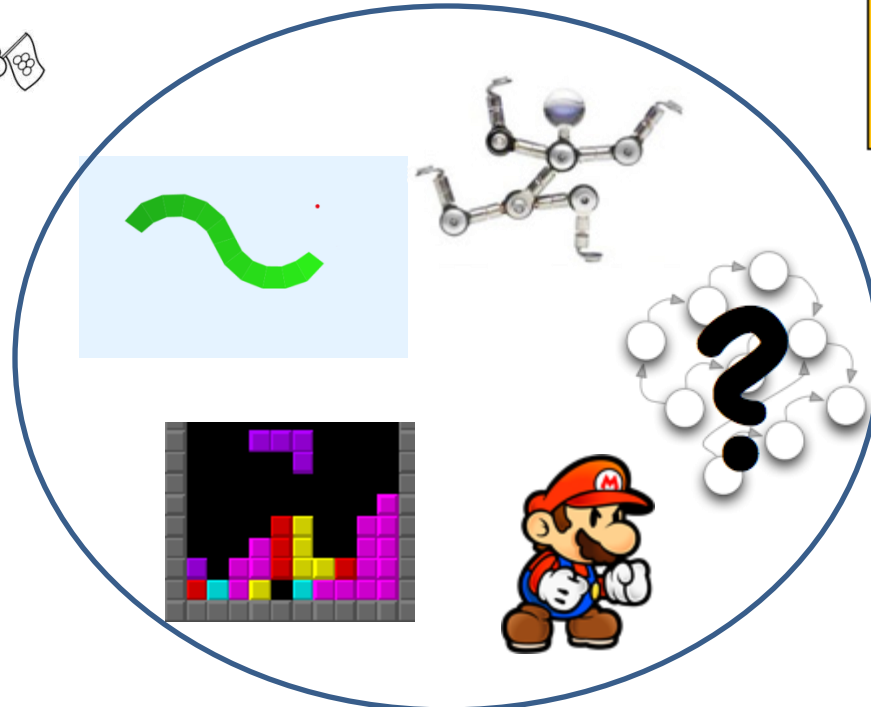
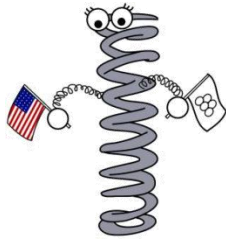
Students and the Competition

RL³ Namor
Ari Weinstein
Chris Mansley

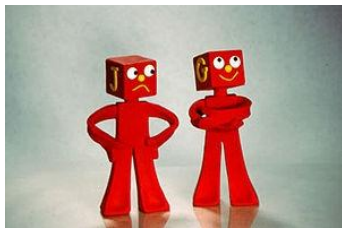


RL³ John
John Asmuth

WSCS
Kevin Sanik
Tingting Sun
John Wilder
Xinyi Cui



RU-Poly
Sergiu Goschin
Yan Xiong
Ravneet Singh
Brian Schubert



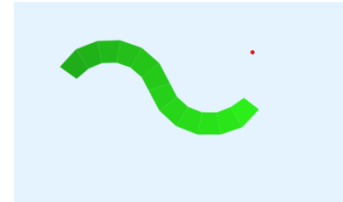
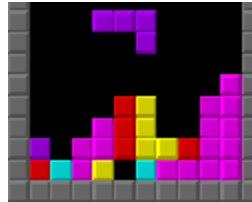
Block Heads
Monica Babes
Daniel Shields
Michael Wunder
Baiyang Liu

Pavlov's Pets
SaeHoon Yi
Paul Ringstad
Fengming Wang



Why use the Competition?

- Tested RL domains



- Environments and RL-Glue interface already set up



- Motivates students with cool prizes



Technical Comments

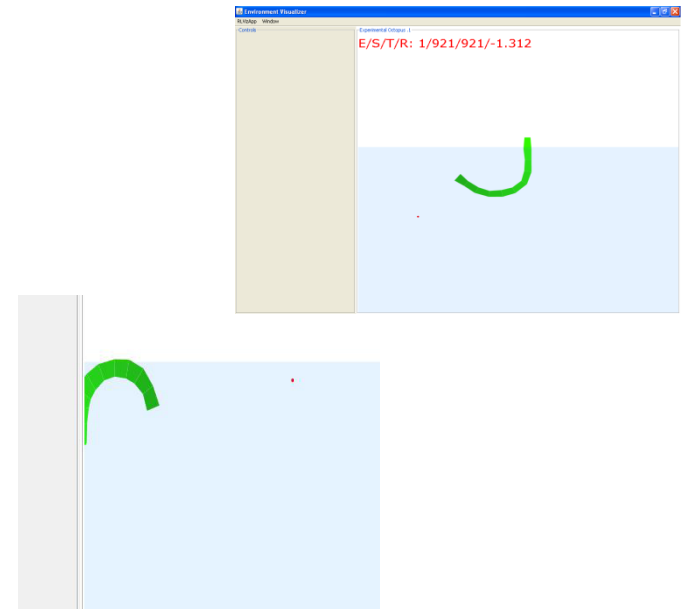
- The students found the code interface really easy to use.



- The MATLAB and JAVA codecs were especially appreciated.
- There was plenty of time to produce results on most of the environments.
 - The later release of the Polyathlon proving runs left the RUPoly team's class report somewhat incomplete.

Octopus and Domain Trouble

- Definition did not conform to the actual environment (ranges of variables, and the reward signal)
- The arm could reach out of the tank and could also go out of the sides
- Response time was very quick
- The forum worked really well.
- More unit testing?



Using Multiple Environments in Class

- Lets you use different algorithms that might work better in some environments than others.



[Nouri & Littman, 08]



3rd in Polyathlon
(1st on leaderboard)



6th in Acrobot*

- You can't always directly compare algorithms from different groups in the class.



*on the leaderboard

The Competition as Tutorial

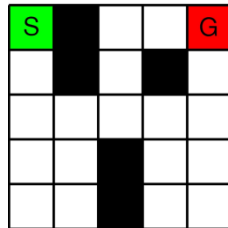
- A lot of people didn't quite understand RL when they got into the competition
 - The Machine Learning class at Rutgers really doesn't cover RL .
 - Many final reports included situations where “value iteration/Q-learning didn't converge”
- The competition has no “beginners” track or U-21 division



Simple Algorithms



- The RL-Library and RL-Glue provide implementations of simple algorithms like Sarsa and Q-learning.
- Students should be encouraged to test drive and extend these before trying harder algorithms.



- Versions of these algorithms and a very simple test domain linked into the competition framework would be helpful.

Meta Point

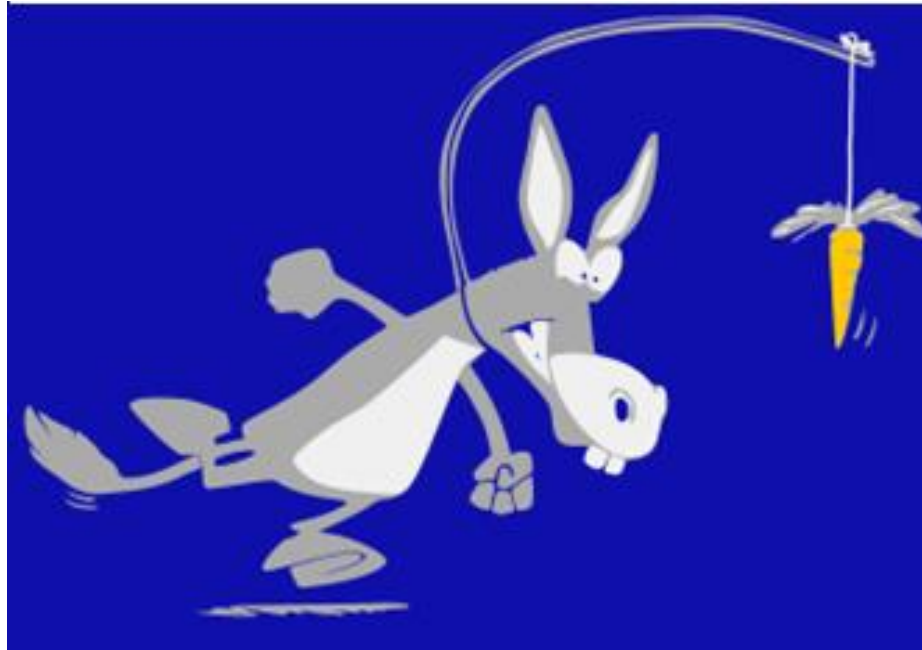
- There are different goals in a tutorial class versus a worldwide competition for fabulous prizes.



- The environments for a competition require very innovative solutions
- If you've never gotten Q-learning to converge before, perhaps octopus isn't for you.



Reinforcement Learning Lessons

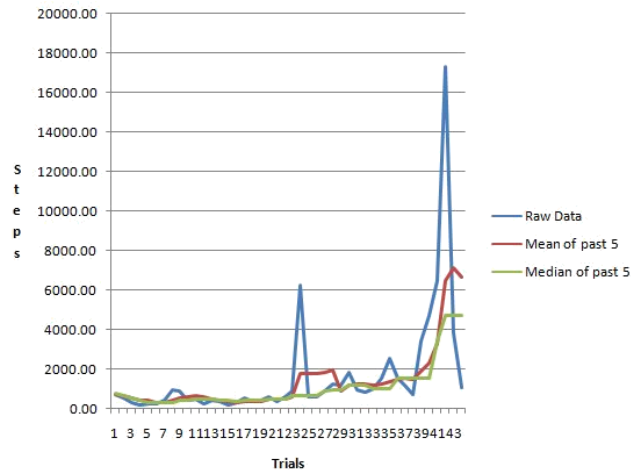
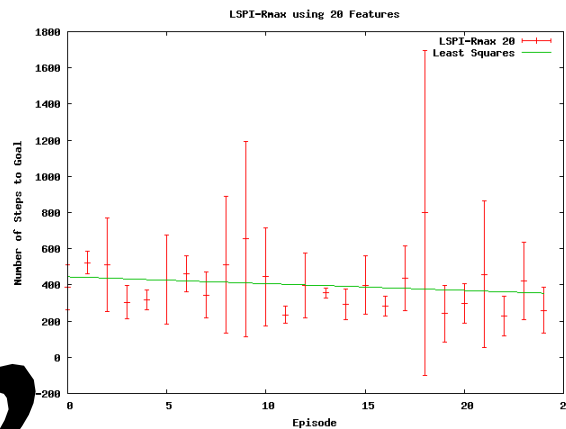
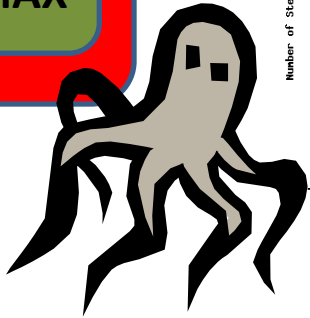


Established Algorithms

- Many established algorithms didn't fare so well in competition domains

[Li, Littman, Mansley, 09]

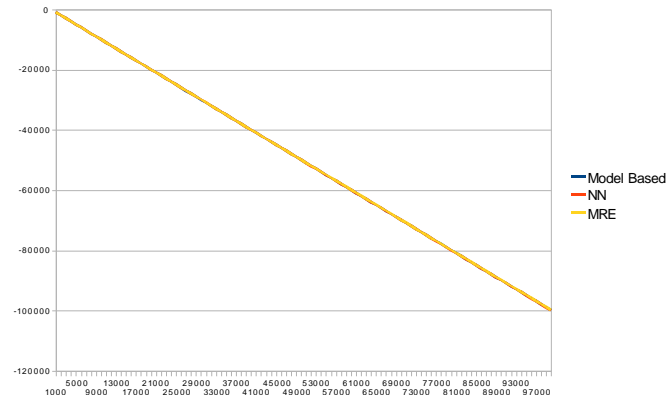
Chris Mansley
Ari Weinstein

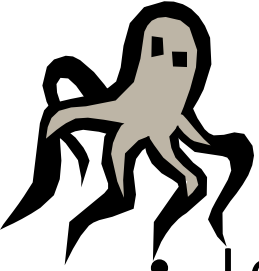


[Nouri & Littman,08]



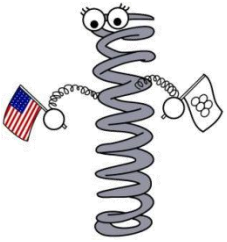
Sergiu Goschin
Yan Xiong
Ravneet Singh
Brian Schubert



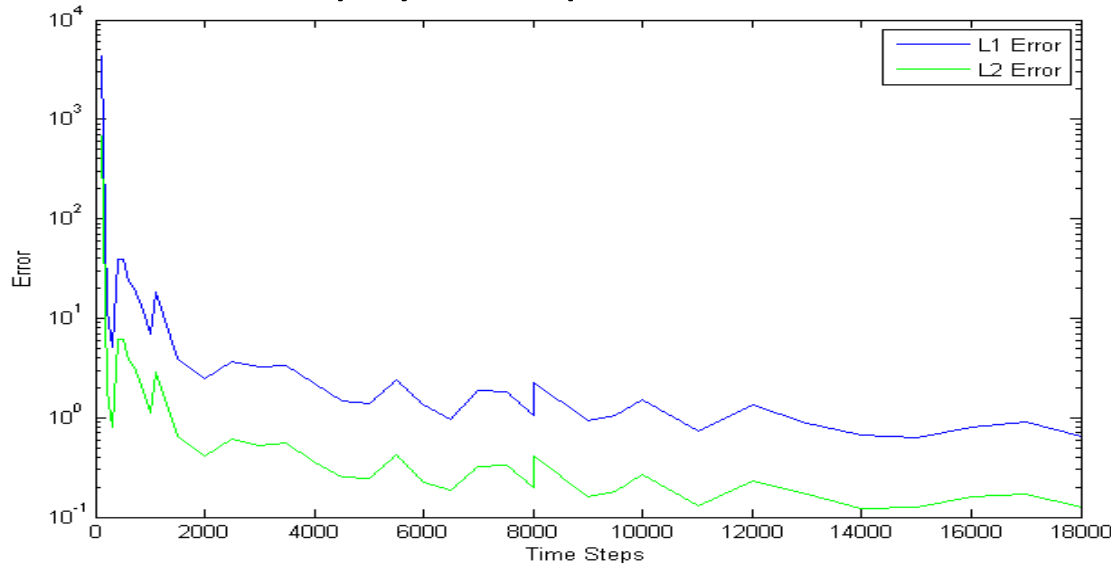


Partial Success

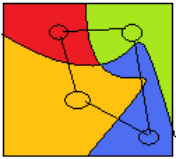
- Learned an extremely accurate model, but couldn't get the planning together for an 82 dim. space.
 - Model based on physics equations for the motion of springs.



Kevin Sanik
Tingting Sun
John Wilder
Xinyi Cui

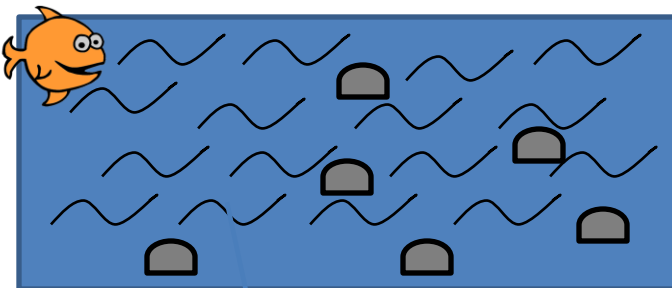


- The competition still might benefit from collecting code like this, even if it wasn't run competitively.
 - Similar to code collection in TAC or RoboCup

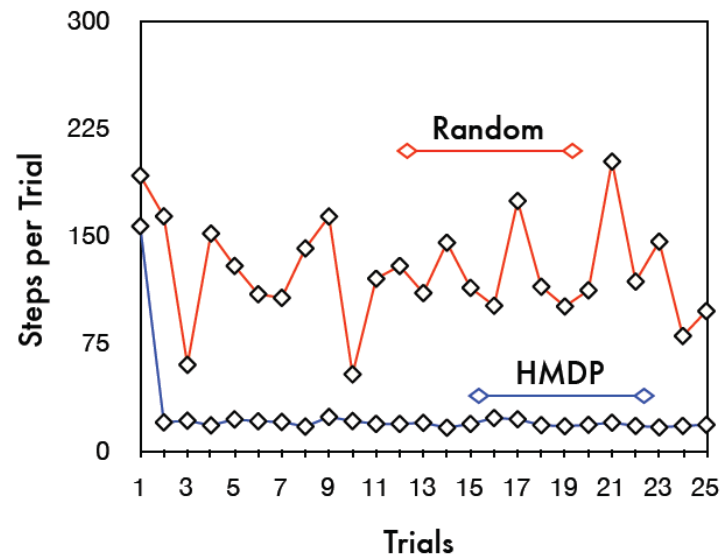


Too Innovative?

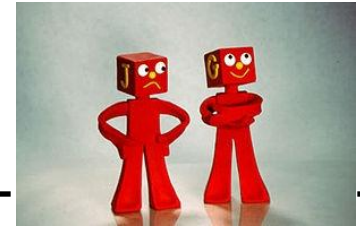
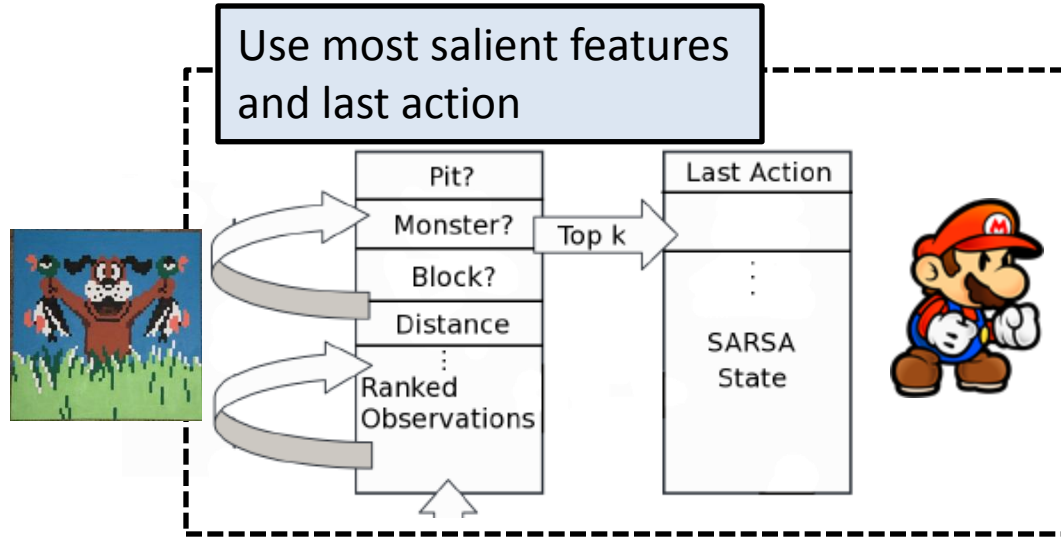
- Some groups overestimated the amount they could do in a short time
- Hidden MDPs [John Asmuth] – a model for capturing Markovian observations (*emissions*) produced by non-Markovian (and unobserved) states, which fit in *partitions*.
 - Learning: uses Bayesian clustering (CRP) to find partitions.



State partitions: proximity to a rock
Emissions: based on dynamics of the partition.

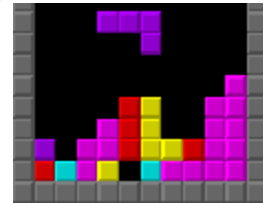
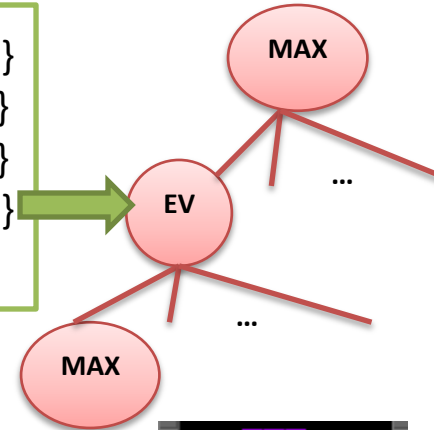


Other Innovative Ideas



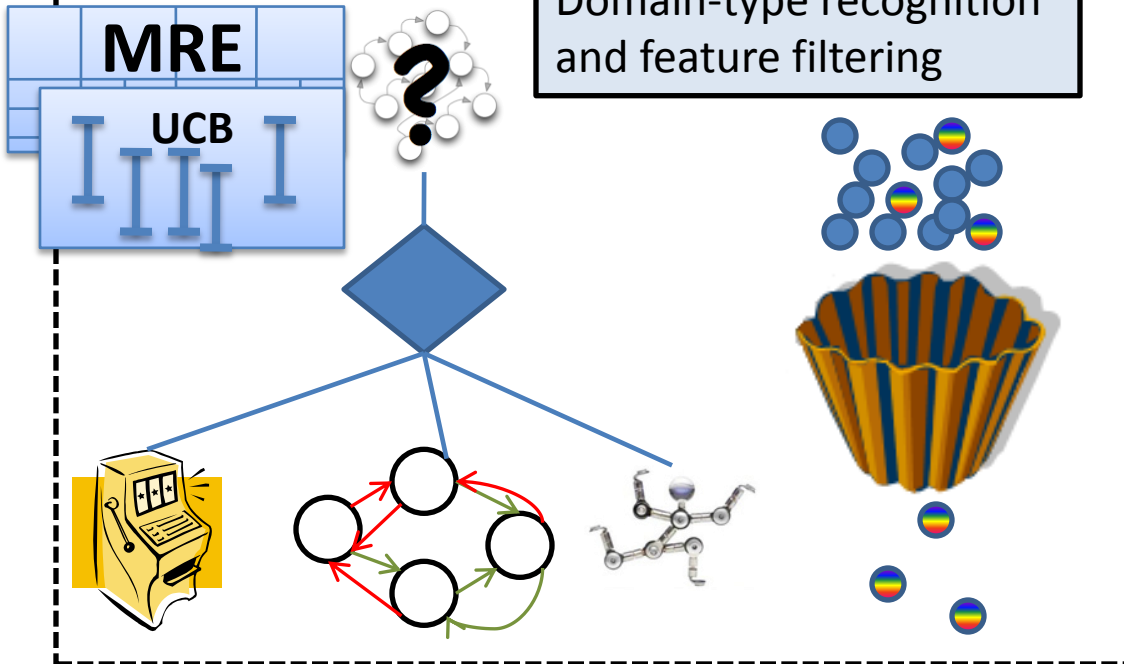
Cross-Entropy to Rank weights

$\{w_2, w_3, w_7\}$
 $\{w_1, w_5, w_7\}$
 $\{w_4, w_5, w_6\}$
 $\{w_3, w_4, w_5\}$



Look-ahead trees and Cross-Entropy

Domain-type recognition and feature filtering





Modified Sarsa for Mario

Paul Ringstad

Fengming Wang

SaeHoon Yi

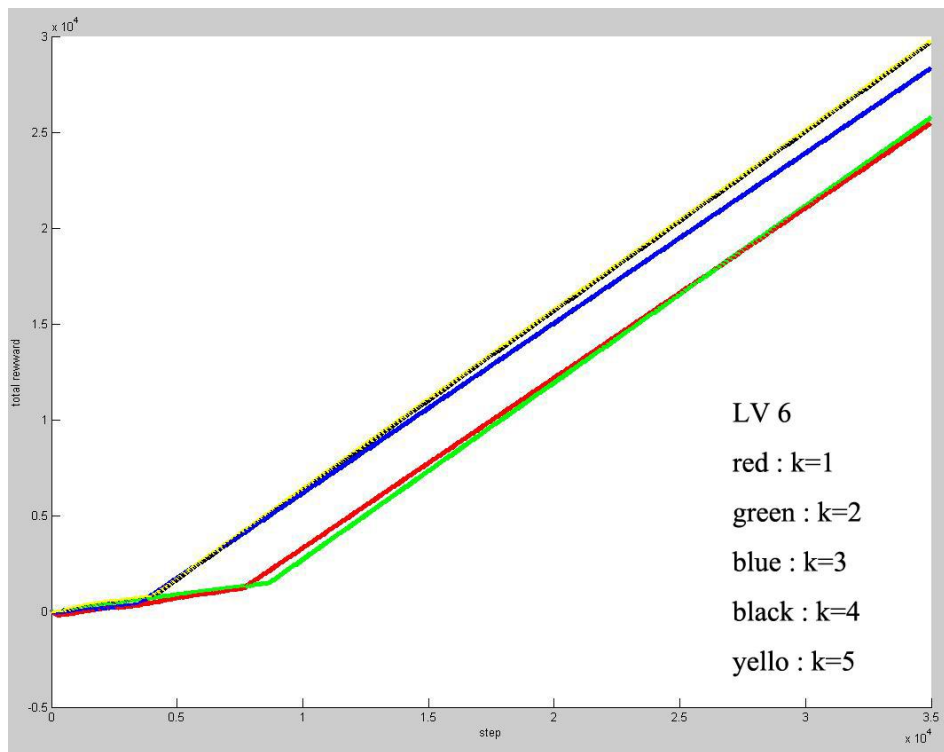


- Base Sarsa Algorithm with
 - Subgoals
 - Negative reward shaping for moving away from the goal
 - Using only the 3 most relevant objects
 - Ranked by danger to Mario and then by proximity
 - Also includes the last action taken
 - Multiple value updates (similar to eligibility traces)
- Ignores coins and just tries to finish the level

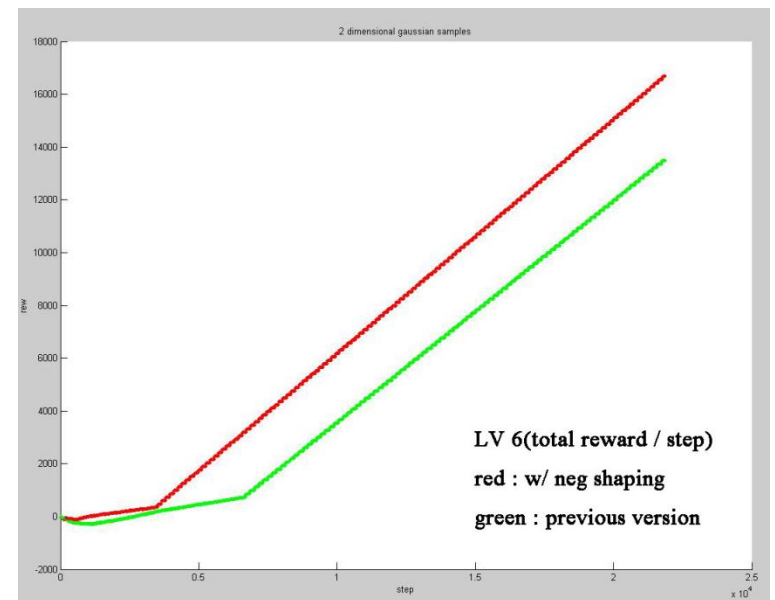


Mario Performance

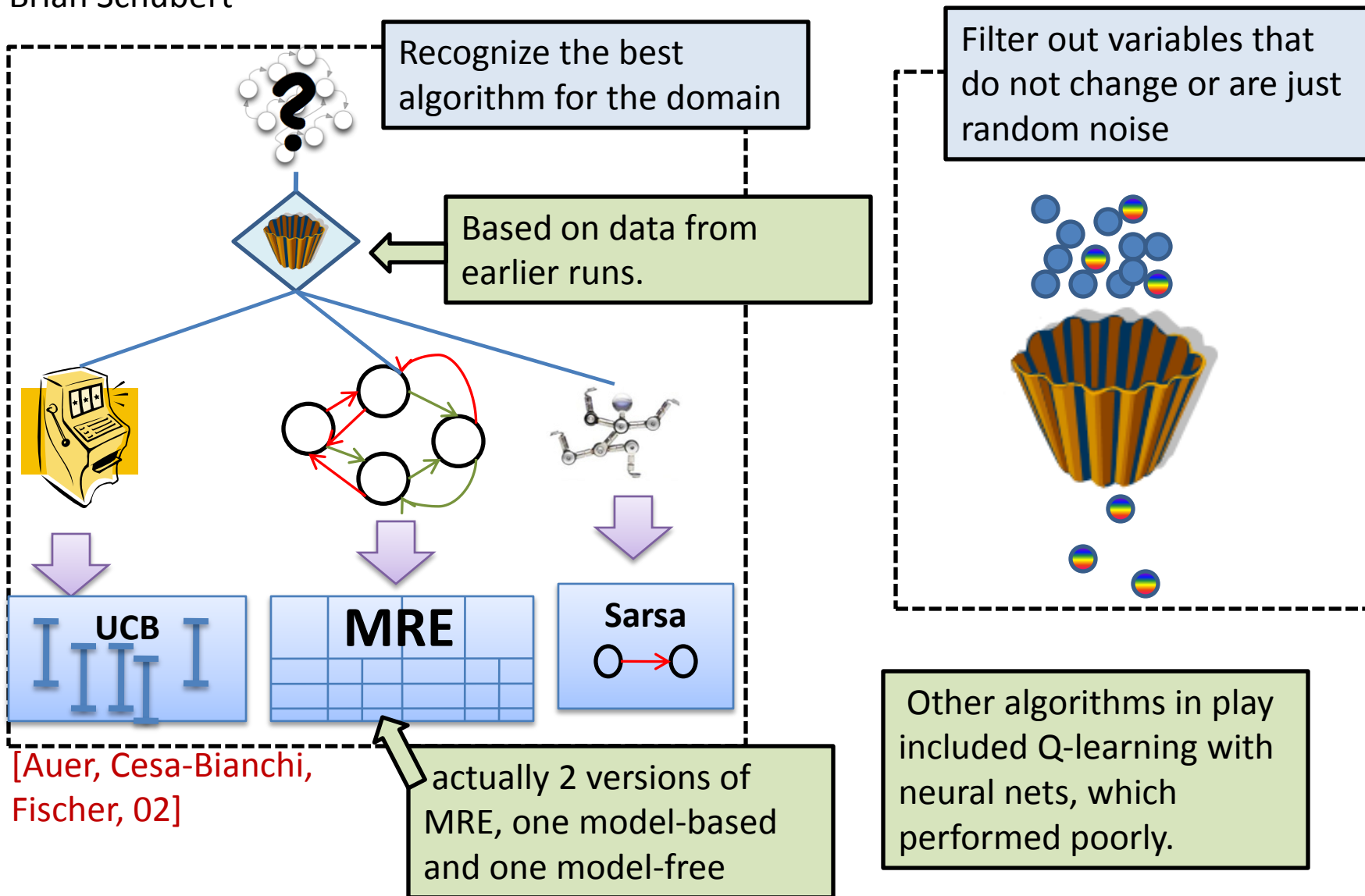
Number of Objects Considered



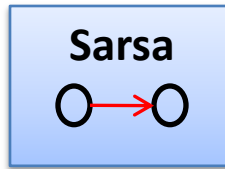
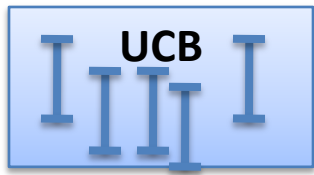
Effect of Negative Shaping



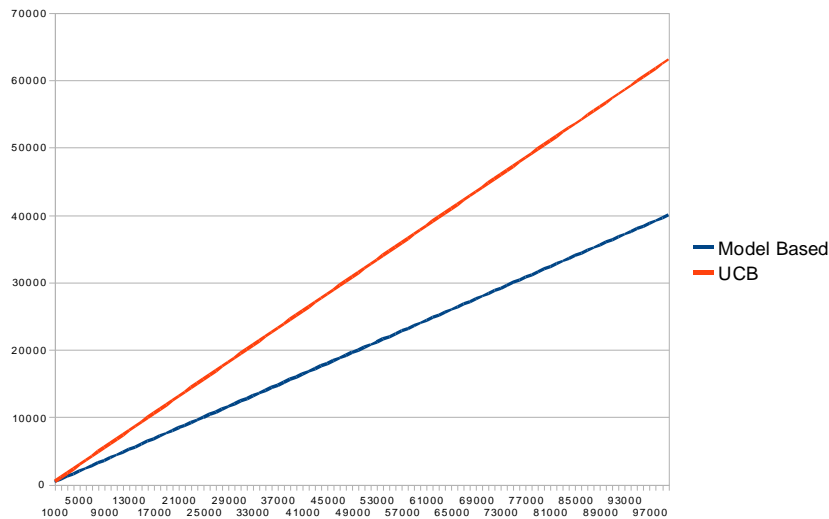
RUPoly & The Polyathlon



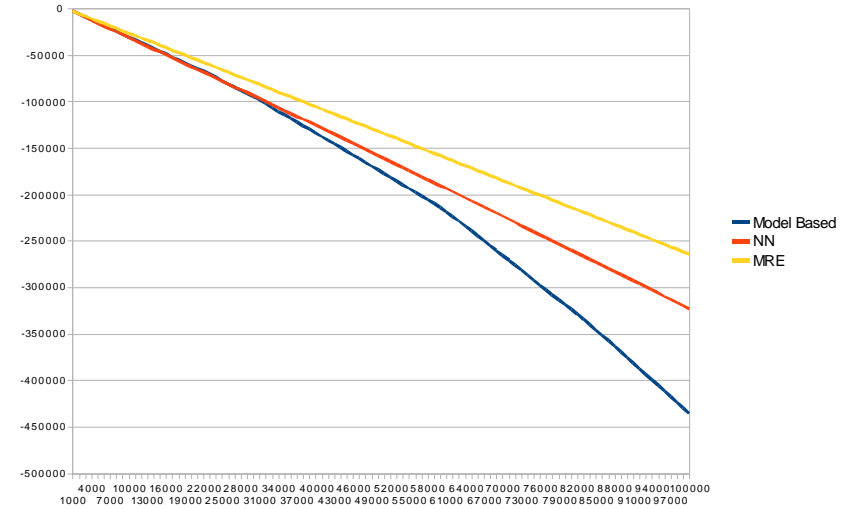
RUPoly Algorithms



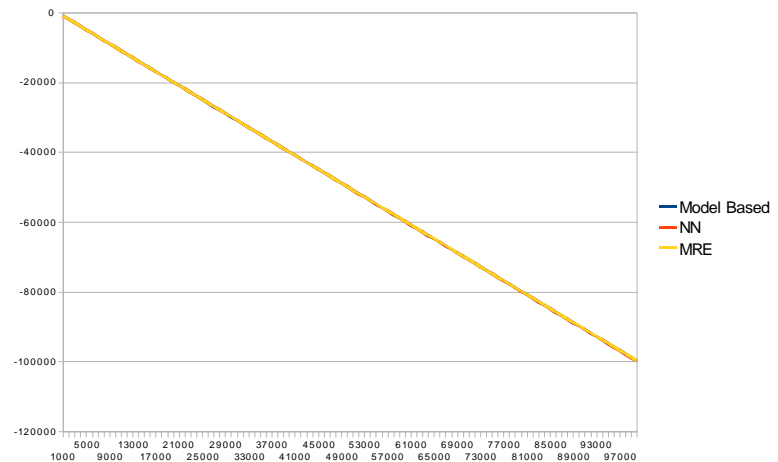
K-Arm Bandit



Cat & Mouse

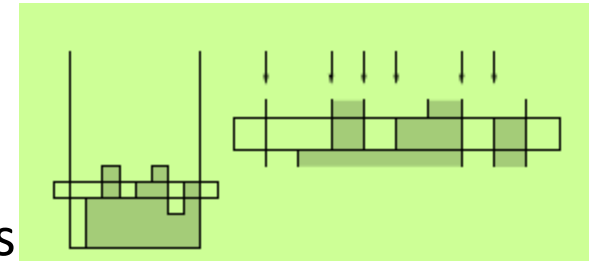
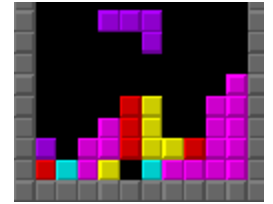
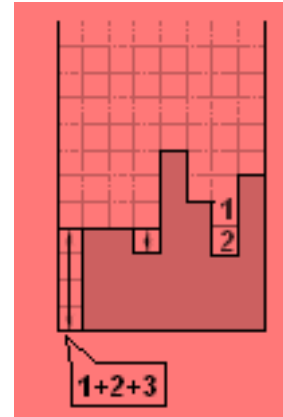


Acrobot Domain

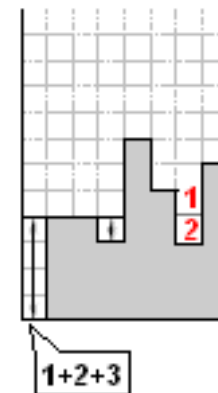


Tetris Features and Piece Distributions

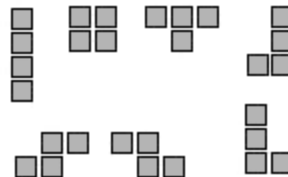
- Features [Bohm et al., 05]
 - Erased Lines
 - Number of Holes
 - **Well Count**
 - **Number of Row Transitions**
 - Number of Column Transitions
 - Individual Column Heights
 - Height Differences Between Adjacent Columns



- Since this Tetris is *adversarial*, probability distributions over next pieces were trained based on the “state” of the game
 - “State” is represented by a 7-tuple representing the maximum breakable lines for each piece.



$\langle 3,0,0,1,0,0,0 \rangle$



Monica Babes
Daniel Shields
Michael Wunder
Baiyang Liu



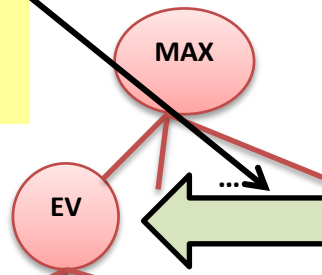
Look-ahead and Cross-Entropy for Tetris

Mean of best weights used as "master weights"

Cross-Entropy to Rank weights for a linear evaluation function

[Szita & Lincz, 06]

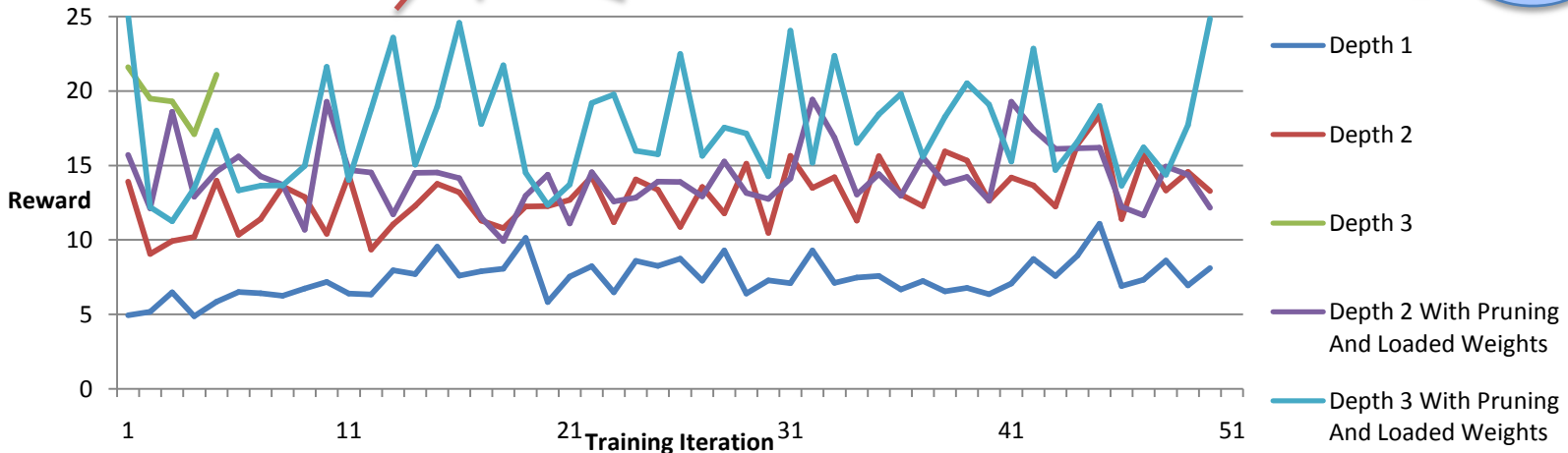
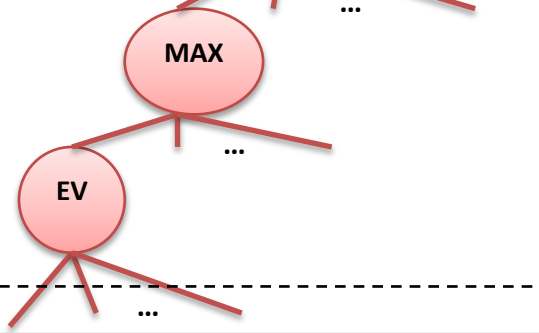
Depth 0



- {w₁₁, w₂₁, w₃₁}
- {w₁₂, w₂₂, w₃₂}
- {w₁₃, w₂₃, w₃₃}
- {w₁₄, w₂₄, w₃₄}

Weight sets evaluated on value prediction on one run each.

Depth 1



Ways of Improving Class Performance

- Competition coordinator TA
- Maybe on the competition side someone responsive specifically to newcomers and class participants.
- Check on student progress before the competition due dates.
 - Intermediate presentations / reports.
 - Status meetings.



Class Overview

- 3 very competitive teams at the top of their respective leaderboards
- Several other good ideas and partial successes
- Some over reaching and some algorithms that weren't able to participate.
- Not much overlap on the domains, so hard to make intra-class comparisons.
- Everybody learned some RL

Competition Recommendations

- A tutorial advisor / domain and rookies track to the competition
 - Special domain? “Rookie of the Year” award?
- Allow algorithms to be made public, even if they only had partial success.
- A little more unit testing of the domains and specs.
- Keep up the great work!

Thanks!

- The class:

John Asmuth, Monica Babes, Xinyi Cui, Sergiu Goschin, Baiyang Liu, Chris Mansley, Paul Ringstad, Kevin Sanik, Brian Schubert, Daniel Shields, Ravneet Singh, Tingting Sun, Fengming Wang, Ari Weinstein, John Wilder, Michael Wunder, Yan Xiong, SaeHoon Yi

- Michael Littman and Shu Chen
- The RL Competition!

Supplementary

Competition Limitations

- Run more than one trial at the same time
- Right now had to hack it to have multiple ports
- Need to make that more obvious
- Tutorial that says: hey here's how to do this hack