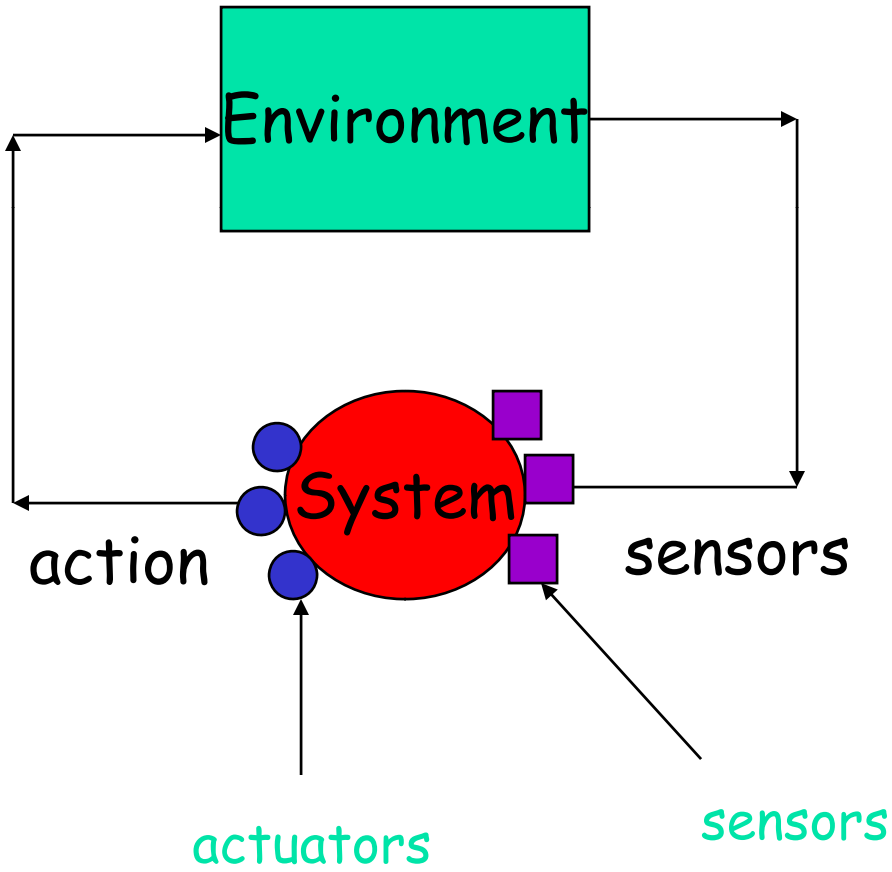
Statistical Machine Learning

Devika Subramanian
Computer Science and Electrical Engineering
Rice University

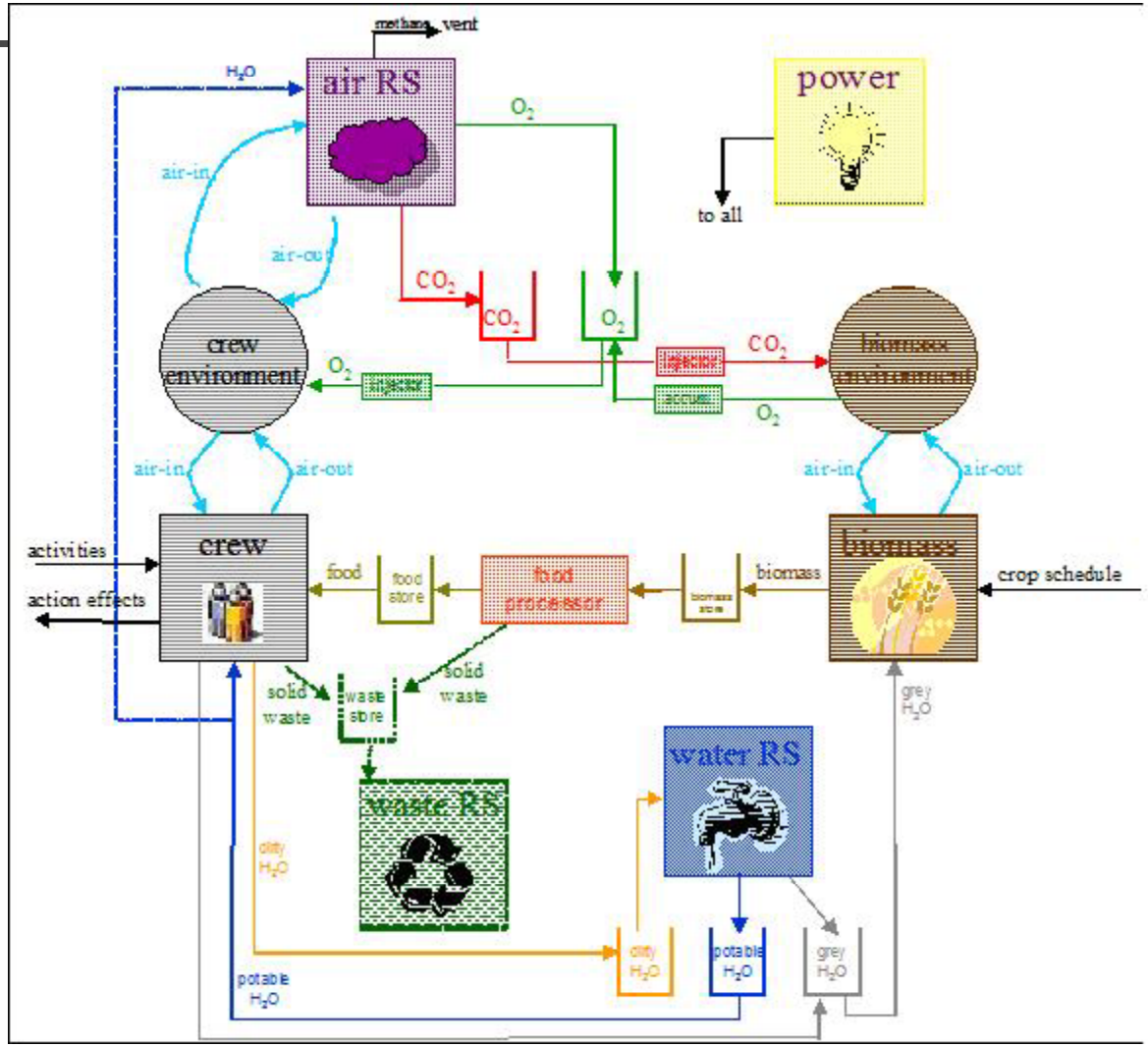
The context



Goal: build systems that "learn from experience"

The Mars Bioplex

Need for models learned by the system itself to determine control actions.



Questions

- How can we design and build systems that adapt and learn from their experience?
- Is there a general theory of embedded learning?
- Can we build machine learning tools that can be used "off the shelf"?

How we build systems now

- Analyze problem
  - Interview human experts, gather requirements, system dynamics, understand how decisions are made.
- Design a solution
  - Handcraft system models and devise algorithms for decision making
- Implement
- Test

When methodology breaks down (1)

- There is no human expertise for the task.
  - Example 1: Energy allocation policies in Bioplex.
  - Example 2: Optimization sequences for a compiler.

When methodology breaks down (2)

- There are human experts but they cannot articulate decision making criteria.
  - Example 1: any perceptual task (e.g., face recognition, speech and handwriting recognition).
  - Example 2: most perceptual-motor tasks (e.g., playing video games).

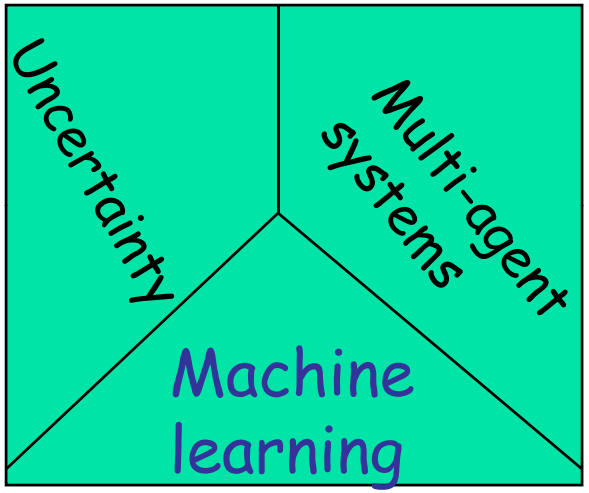
When methodology breaks down (3)

- Task solutions need to be customized for each individual.
  - Example 1: Spam filtering/Speech recognition.
  - Example 2: Tracking human learning.
  - Example 3: Program-specific compiler optimization sequences.

When methodology breaks down (4)

- The system dynamics or the environment changes rapidly.
  - Example 1: Outdoor robot navigation.
  - Example 2: Mars Bioplex energy allocation decisions.
  - Example 3: Predicting conflict levels in regions of the world.

Learning in context

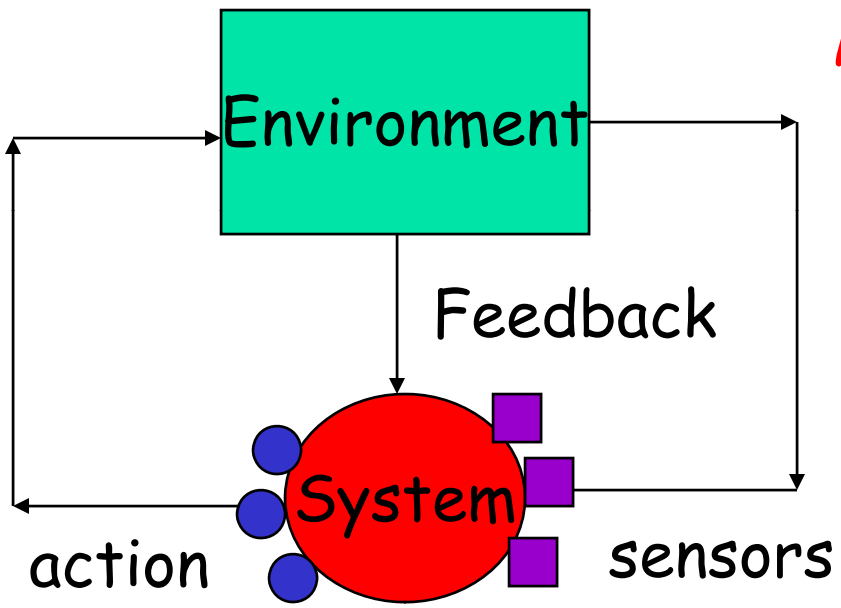


Artificial Intelligence

Algorithms	Systems/ Software E.	Data Mining

Control Th.	OR (MDPs)	Statistics	Appl. Math.

Statistical machine learning



Model of environment

$$M : S \times A \rightarrow \Pr(S)$$

policy → $\pi : S \rightarrow A$

Optimal policy → $\pi^* = \arg\max_{\pi} \left[E\left(\sum_t r(s_t, \pi(s_t)) \right) \right]$

Machine learning (1)

- There is no human expertise for the task.
  - Example 1: Energy allocation policies in Bioplex.
  - Example 2: Optimization sequences for a compiler.
- Machine learners learn
  - Value functions Q:SxA→Re by interaction
    - Actively choosing actions a_t using Q.
    - Observing system transitions (s_t, a_t, s_{t+1}, r_t)
    - Updating Q on the basis of observations

$$Q^{\pi}(s,a) = E\left\{r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \mid s_t = s, a_t = a, \pi\right\}$$

Machine learning (2)

- There are human experts but they cannot articulate decision making criteria.
  - Example 1: any perceptual task (e.g., face recognition).
  - Example 2: most perceptual-motor tasks (e.g., playing video games).
- Machine learners learn
  - Policies: pi:S*→A or pi:S*→Pr(A) by observing (s_t,a_t,s_{t+1}) system transitions over time, where trajectories are chosen by human experts.

Machine learning (3)

- Task solutions need to be customized for each individual.
  - Example 1: Spam filtering/Speech recognition.
  - Example 2: Tracking human learning.
  - Example 3: Program-specific compiler optimization sequences.
- Machine learners learn
  - Policies: pi:S*→A or pi:S*→Pr(A) by observing system trajectories (s_t,a_t,s_{t+1}) over time, where trajectory is chosen by individual.

Machine learning (4)

- The system dynamics or the environment changes rapidly.
  - Example 1: Outdoor robot navigation.
  - Example 2: Mars Bioplex energy allocation decisions.
  - Example 3: Predicting conflict levels in regions of the world.
- Machine learners learn
  - To detect changes (discontinuities and trends).
  - Reacquire policies and value functions as soon as change is detected.

Statistical Machine Learning

- Replaces guesswork/intuition with analysis of real data.
- Instead of guessing a control policy/prediction function/
  - Machine learning provides a basis for deriving them from observed data.
- Result: more robust, more adaptive systems!

Outline

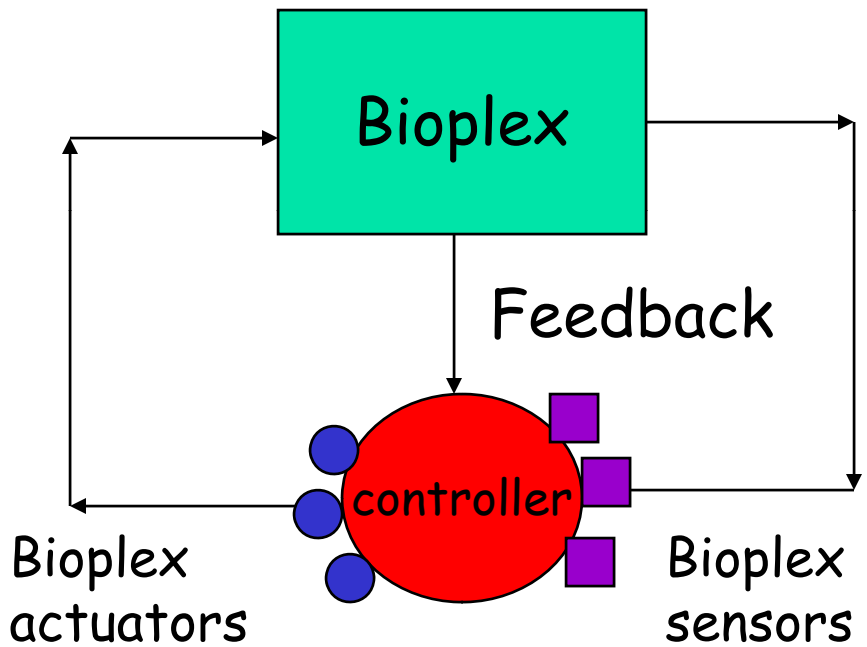
- Motivation
- Statistical machine learning
  - Scenario 1: Mars Bioplex
  - Scenario 2: Virgil
  - Scenario 3: Adaptive compilers
  - Scenario 4: Analyzing conflict
  - Scenario 5: Tracking human learning
- Conclusions

Scenario 1

- Controlling the Mars Bioplex
  - Approach 1: assemble a team of control engineers to write software to make resource allocation decisions for the system.
  - Approach 2: use available sensor/actuator/reward data to learn dynamic models and value functions for the Bioplex and calculate optimal resource allocation policies.

Thanks to NASA-NRA and D. Kortenkamp and P. Bonasso of NASA JSC.

Statistical machine learning



Model of environment

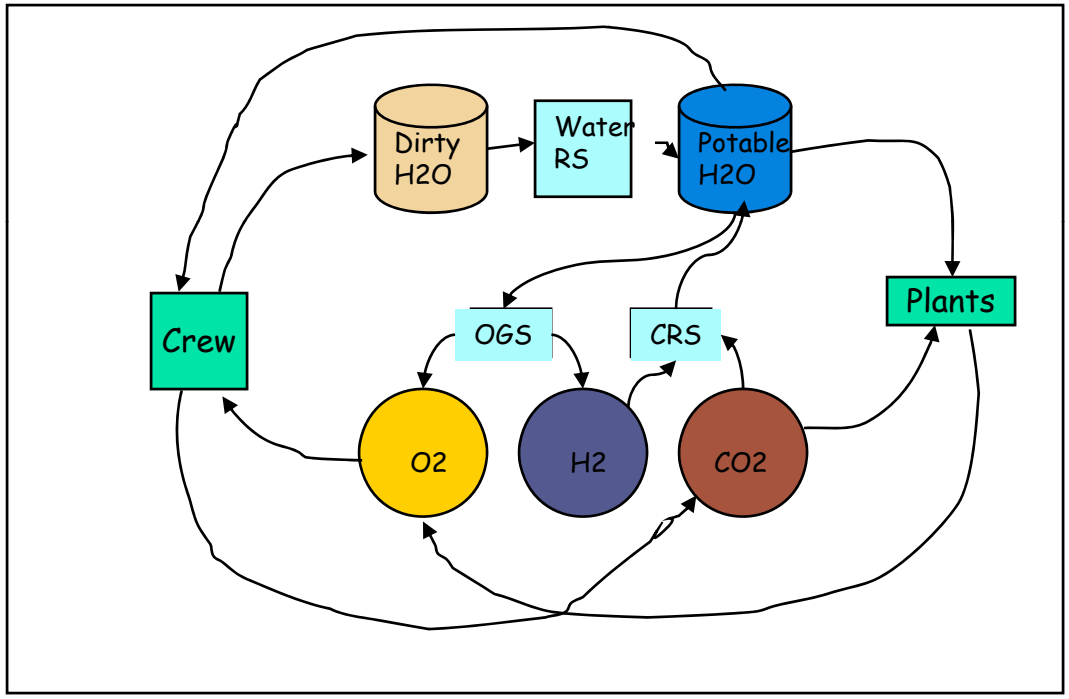
$$M : S \times A \rightarrow \Pr(S)$$

Bioplex treated as a Markov Decision Process

policy → $\pi : S \rightarrow A$

Optimal policy → $\pi^* = \arg\max_{\pi} \left[E\left(\sum_t r(s_t, \pi(s_t)) \right) \right]$

The Bioplex: take 1



Air and water recycling systems interact! One major source of complexity in the overall system.

Modeling Bioplex as an MDP

- Action space: turning on/off the CRS, OGS, WRS.
- Sensor space: O_2, H_2, CO_2, potable H_20 and dirty H_2O values.
- Local reward function
  - Rewards for keeping crew alive (in proportion to length of time they are kept alive)
  - Punishment for starving crew of water and/or air and/or food.
  - Punishment for killing plants.
  - Punishment for overflowing tanks.

Result from MDP Theory

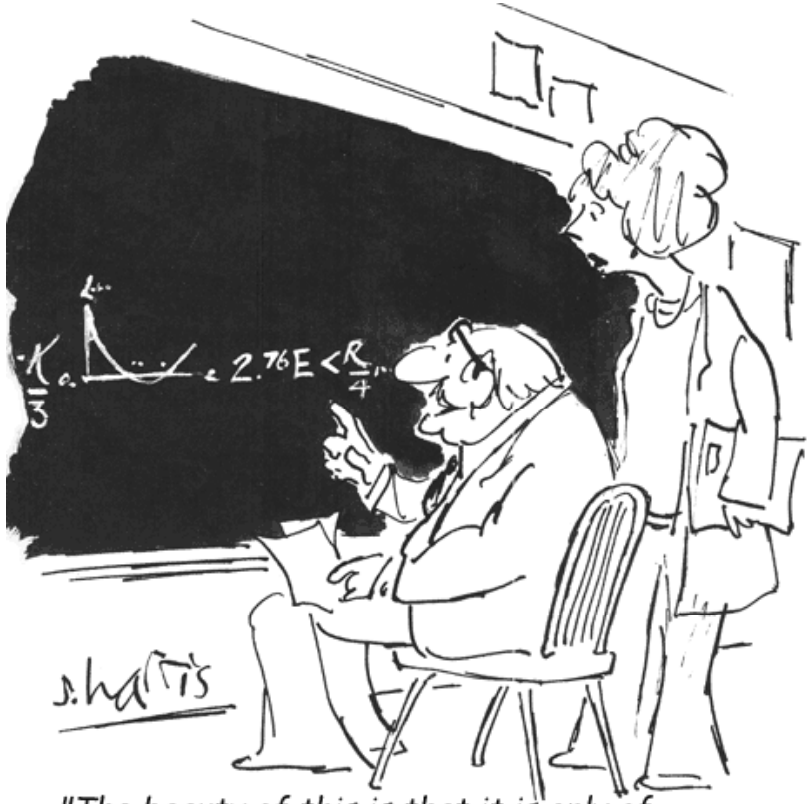
- There exist optimal value functions

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$$

- and corresponding optimal policies

$$\pi^*(s) = \arg\max_{a} Q^*(s,a)$$

So what?



"The beauty of this is that it is only of theoretical importance, and there is no way it can be of any practical use whatsoever."

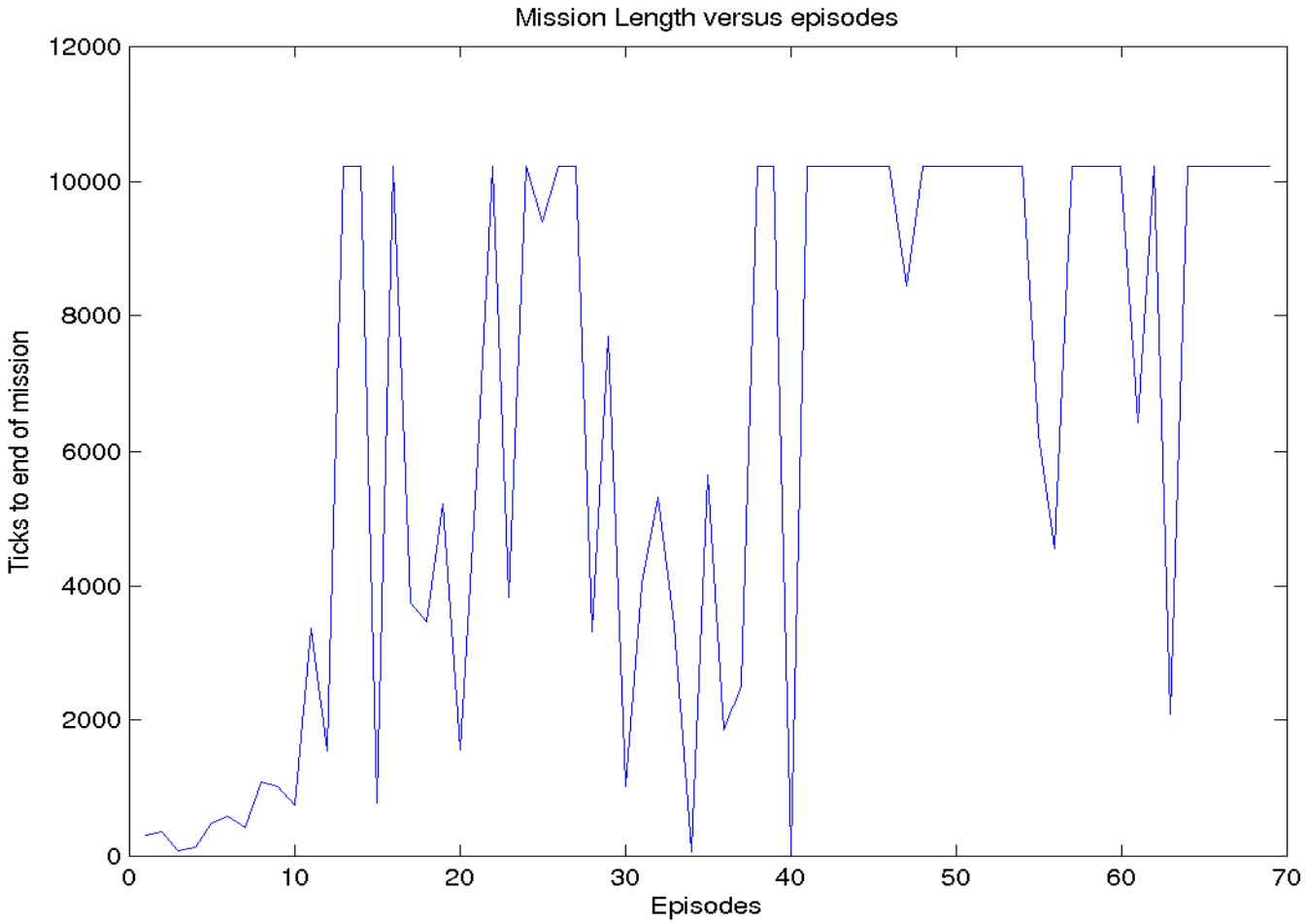
The Reinforcement Learning Algorithm

- $Q(s,a) = 0$ for all s in S and a in A.
- Repeat till end of trial
  - s = current state
  - Pick action a which maximizes $Q(s,a)$ [policy selection] (can be ε-greedy)
  - Do a and observe new state s' and local reward r.
  - Update
    - $Q(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \max\{a' \text{ in } A\}\ Q(s',a'))$. [Temporal difference update]
  - Set s to s'.
- Repeat step above till Q converges.

Speeding up convergence by nearest neighbor generalization

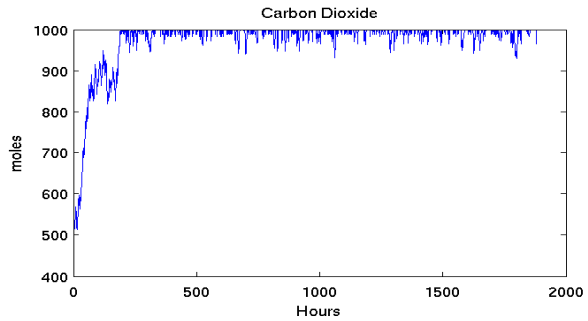
- When Q(s,a) is updated, updates are applied to N nearest states within a set radius, with the update decayed exponentially according to the distance from s.
- Allows for faster convergence, because we no longer have to visit every single state to update it. The trajectory of a simulation run cuts a wider swath through the state space.

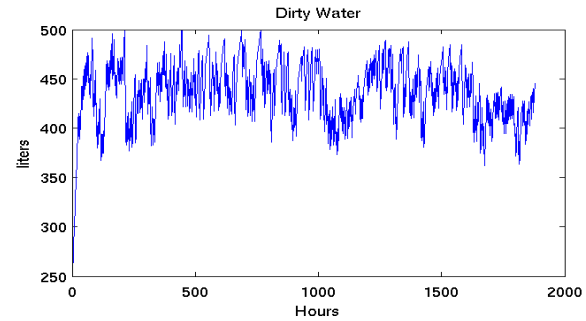
Generalized RL Results

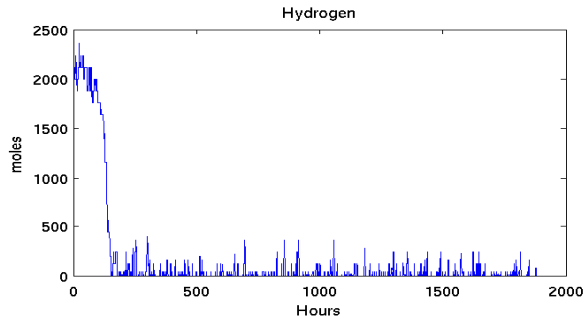


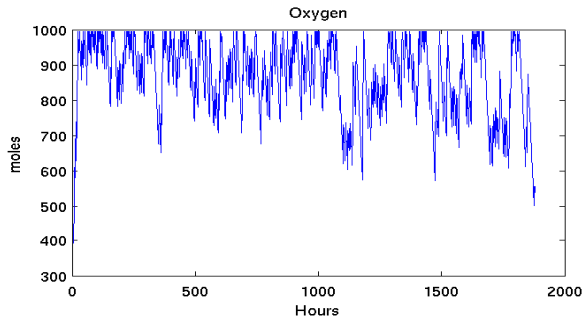
Klein, Subramanian, Kortenkamp, Bell ICES 2004

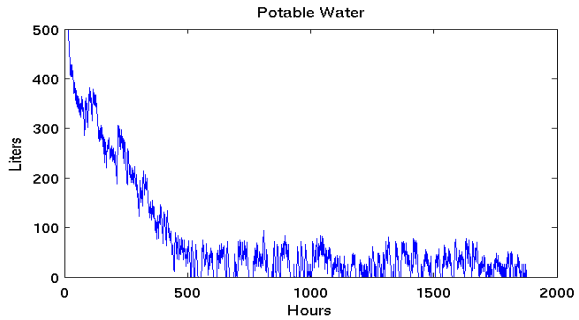
Simulation Behavior











Back of the envelope…

- Given the biomass store of 300 kg, and food stocks of 2000 kg, the maximum mission lifetime for this configuration is approximately 10220 hours: it is the time it takes for the crew to consume all available food. RL achieves 10000 hours.
- Since the simulator uses 10 square meters of wheat, it doesn't grow enough food for mission to be extended beyond 10220 hours.
- So given perfect water and oxygen recycling the limiting resource in this simulation configuration is food!

Scenario 1 summary

- Reinforcement learning is an effective tool for exploring the state space of a complex coupled dynamical system.
- It can find interesting policies rapidly. It can work fast enough to respond to abrupt changes to the system and redesign control policies (because convergence can occur as quickly as 10-20 trials).

Outline

- Motivation
- Statistical machine learning
  - Scenario 1: Mars Bioplex
  - Scenario 2: Virgil
  - Scenario 3: Adaptive compilers
  - Scenario 4: Predicting conflict
  - Scenario 5: Tracking human learning
- Conclusions

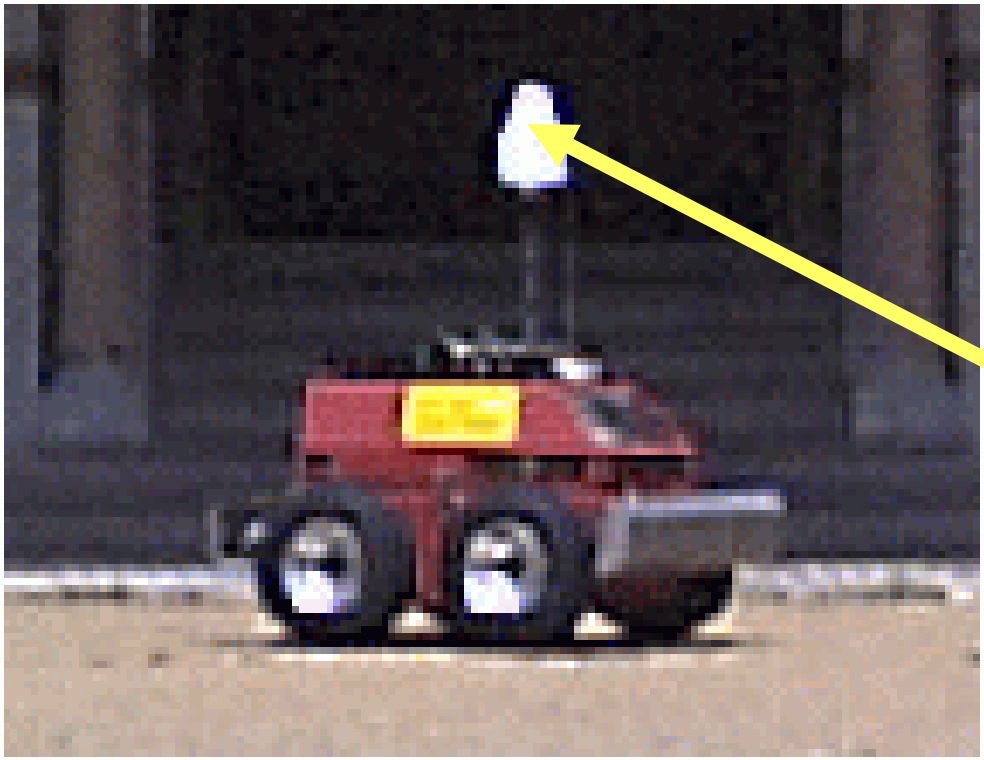
Scenario 2

- Outdoor robot navigation with GPS and odometry.
  - Approach 1: get a human expert to construct sensor error models for GPS.
  - Approach 2: get robot to gather correlated GPS and odometry data and learn error models.

Virgil: The Rice Tour Guide

On-board odometry

Sonars and bumpers for collision avoidance



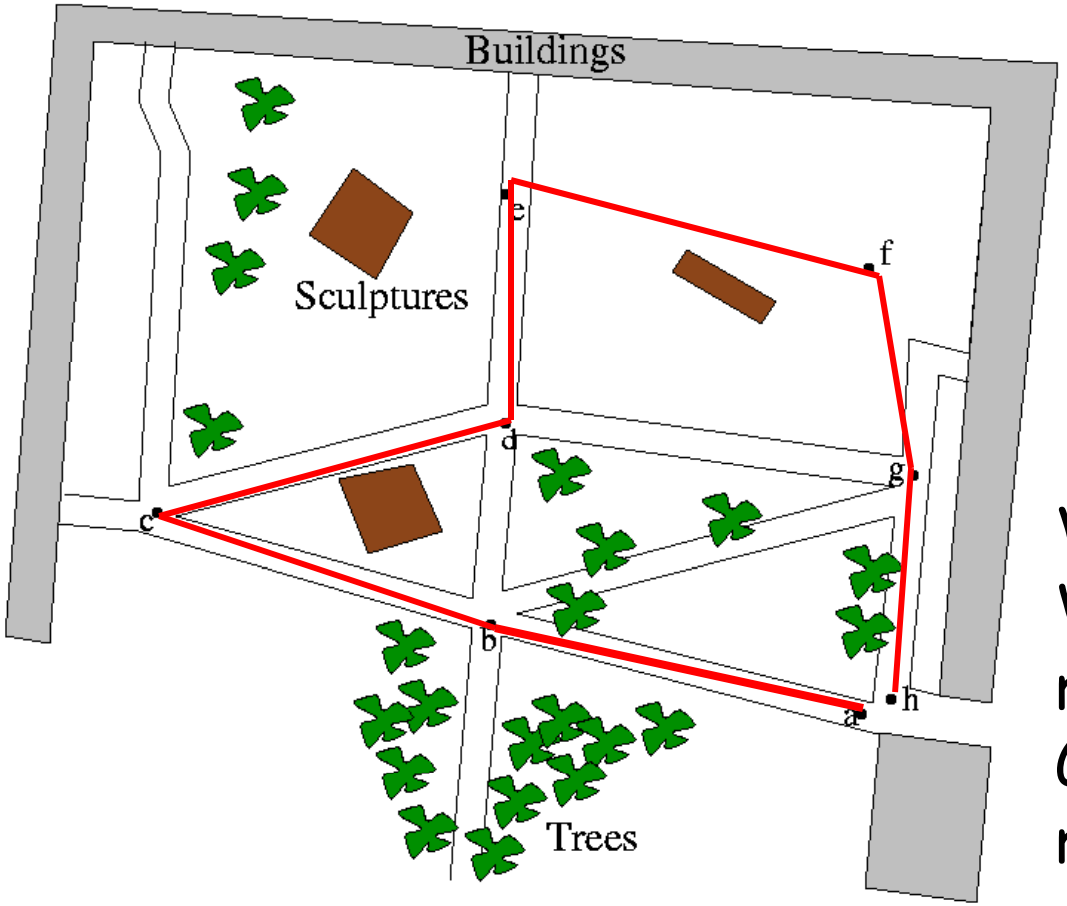
RWI ATRV Jr equipped with cheap differential GPS ($2K)

Supported by a grant from Rice's Engineering School

The task and its challenges

- Task: To give tours of the Rice University Campus.
- Challenges
  - Environment cannot be modified to assist robot.
  - Low budget project --- no expensive sensors!
  - Robot needs to interact with people.
  - Localization errors of more than 40 cm cannot be tolerated. Cost of failure high!

The tour

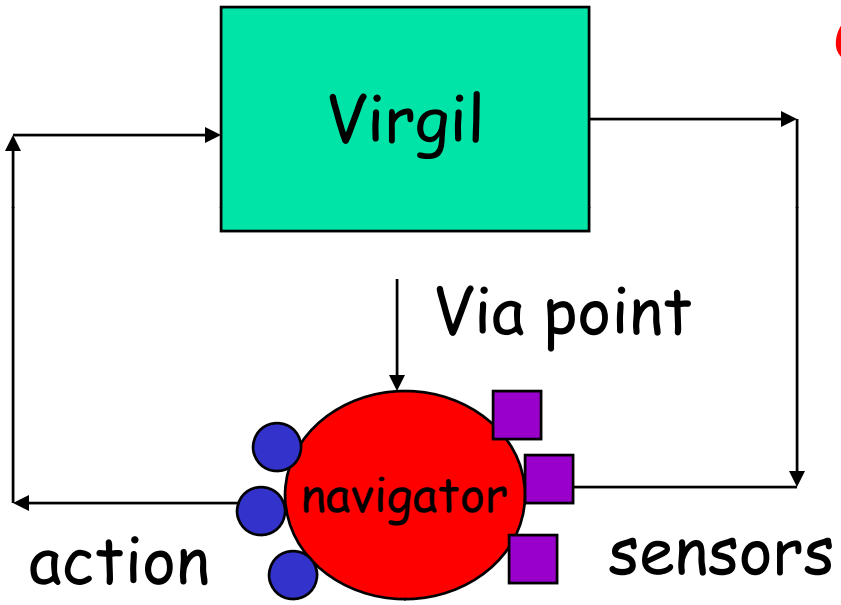


We can localize Virgil to 1m resolution w/ GPS of 10-40m resolution.

The Darpa Grand Challenge

CNN video of Darpa Grand Challenge

Statistical machine learning



Odometry model

$$M : S \times A \rightarrow \Pr(S)$$

Sensor model

P(GPS reading|s)

Proportional Controller $\longrightarrow$ $\pi : S \times G \rightarrow A$

Real problem is state estimation!

Bayesian localization

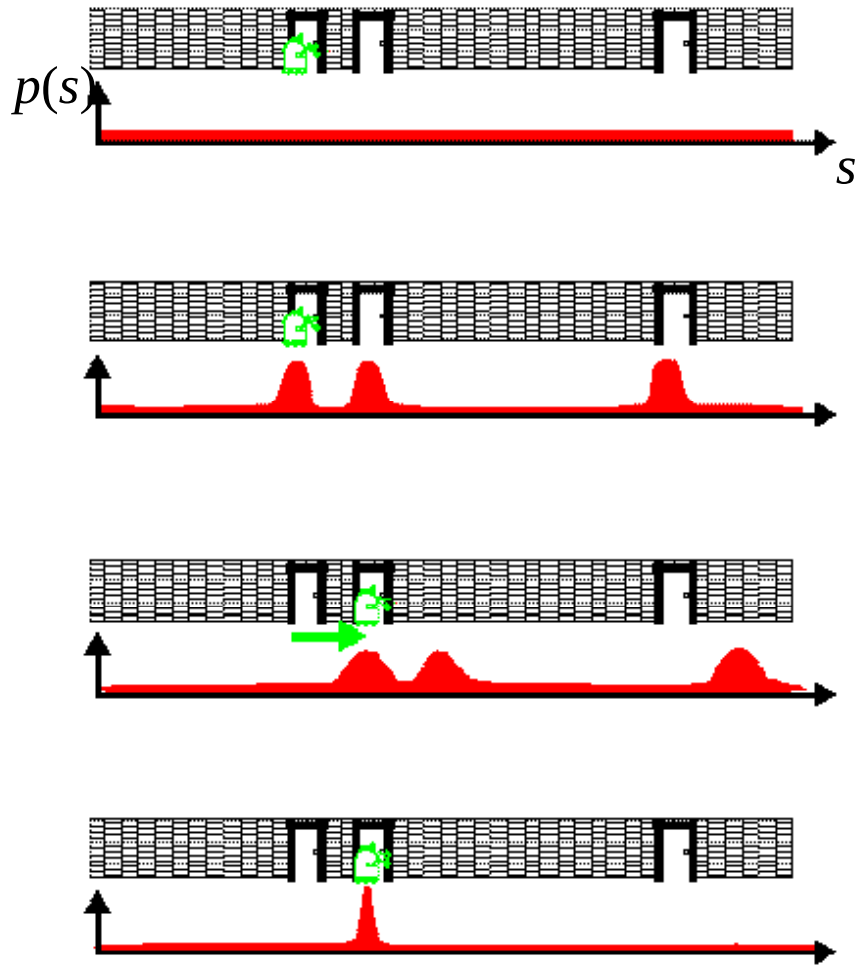
- Estimate robot's coordinates $s=(x,y,\theta)$ from sensor data (GPS) by using Bayesian filtering.

$$b(s_t) = \eta \, \overbrace{p(o_t \mid s_t)}^{\text{correct}} \overbrace{\int p(s_t \mid s_{t-1}, a_{t-1}) \, b(s_{t-1}) \, ds_{t-1}}^{\text{predict}}$$

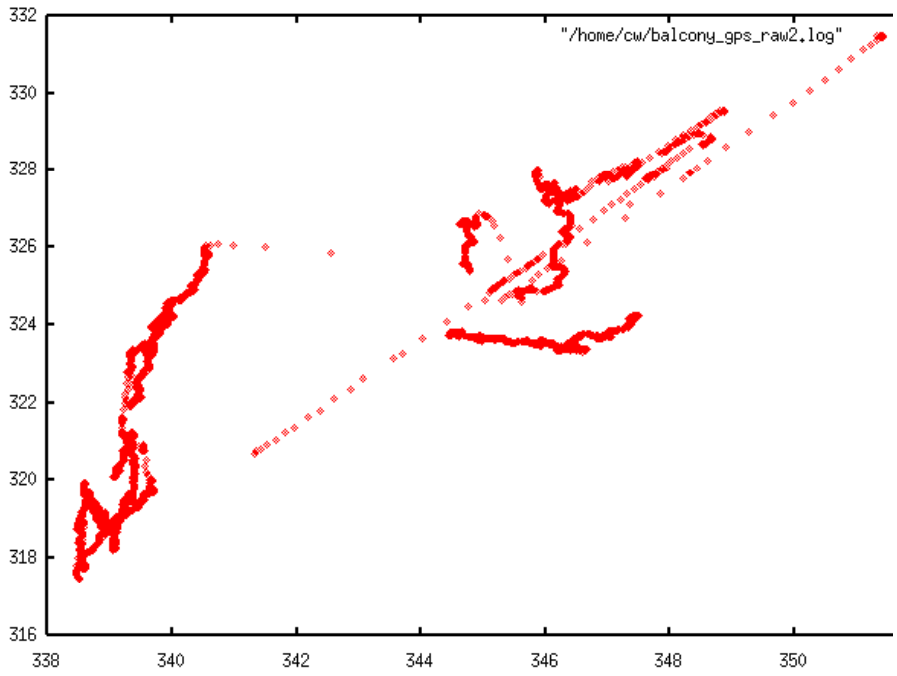
GPS model

odometry

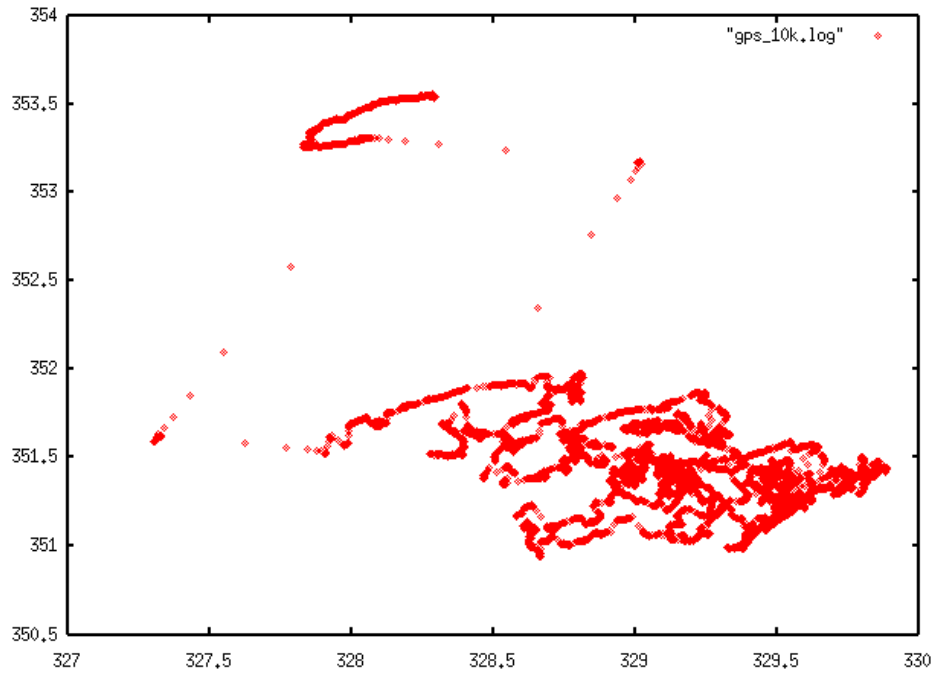
Bayesian filter at work



GPS sensor characteristics



Near buildings



In free space

Characteristics of GPS data

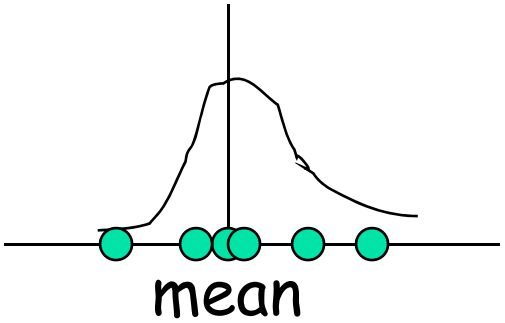
- Abrupt shifts in GPS data quality when satellites drop out of view (e.g., when robot is near concrete buildings or travels under trees).
- Gradual drifts in GPS data quality caused by atmospheric effects.

Sensor models

- A single Gaussian for the GPS sensor model P(o|s) requires a large variance to accommodate the variation in data quality with location.
  - This causes slow convergence in localization estimates even when data quality is high!
  - Shifts in GPS data quality are not handled well; e.g., when robot emerges into a clear area with many visible satellites, it takes a while before localization accuracy reflects the quality of the GPS data.

Unsupervised learning

- Model is a conditional probability density function p(o|s) for GPS reading o in location s.
  - Parametric (e.g., Gaussian)
  - Non-parametric (histogram)
  - Semi-parametric (mixture)



- Training data are samples (o_1,s_1), …, (o_n,s_n).

Learning GPS sensor models

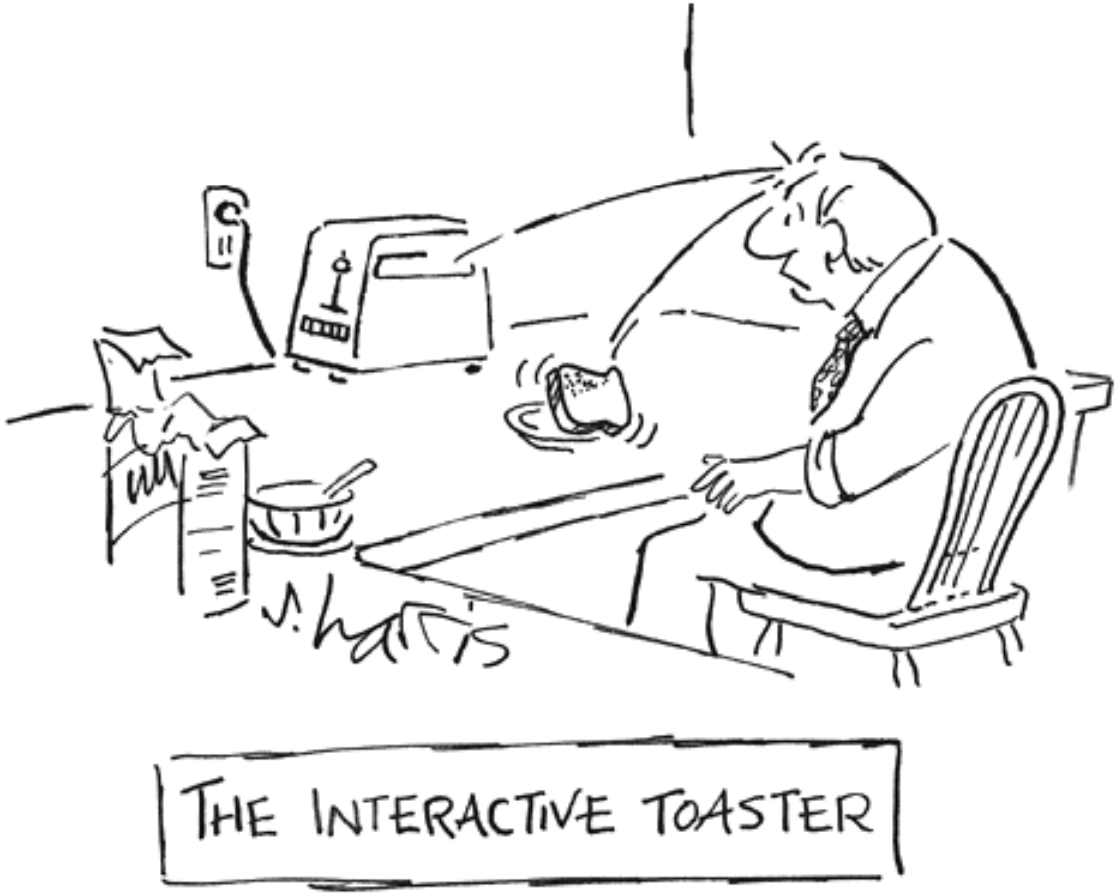
- Need location-specific P(o|s).
- Which locations? How to learn them?
  - Joystick Virgil through course where we allow it to gather correlated GPS and odometry data.
  - Learn mean and variances of GPS readings throughout the route; merge distributions that are "close enough".
  - We discovered a very strong correlation between the number of visible satellites and GPS mean and variance.

Adaptive localization

- For abrupt shifts in GPS data quality: Learn p(o|s) indexed by number of visible satellites. Use "gain scheduling" and swap in appropriate sensor model for current number of visible satellites at each time t.
- For gradual shifts in GPS data quality: use "exponential forgetting", i.e., use a window of N (=100) GPS observations and update state estimate based on it.

Virgil video

Toast localization



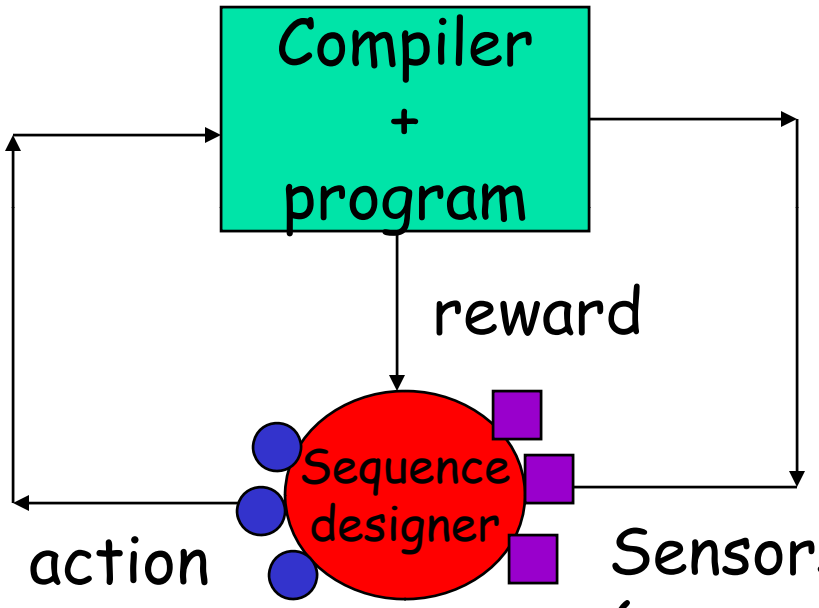
Outline

- Motivation
- Statistical machine learning
  - Scenario 1: Mars Bioplex
  - Scenario 2: Virgil
  - Scenario 3: Adaptive compilers
  - Scenario 4: Analyzing conflict
  - Scenario 5: Tracking human learning
- Conclusions

Scenario 3

- Find good optimization sequences for compiling programs with respect to given objectives.
  - Approach 1: get human experts to devise good compilation sequences for a whole range of programs.
  - Approach 2: learn a probabilistic model of what constitutes a good compilation sequence and use it to perform biased random sampling of the combinatorial compilation space.

Statistical machine learning



action
(sequence of
optimizations)

Sensors
(measurements on program
and compiler state)

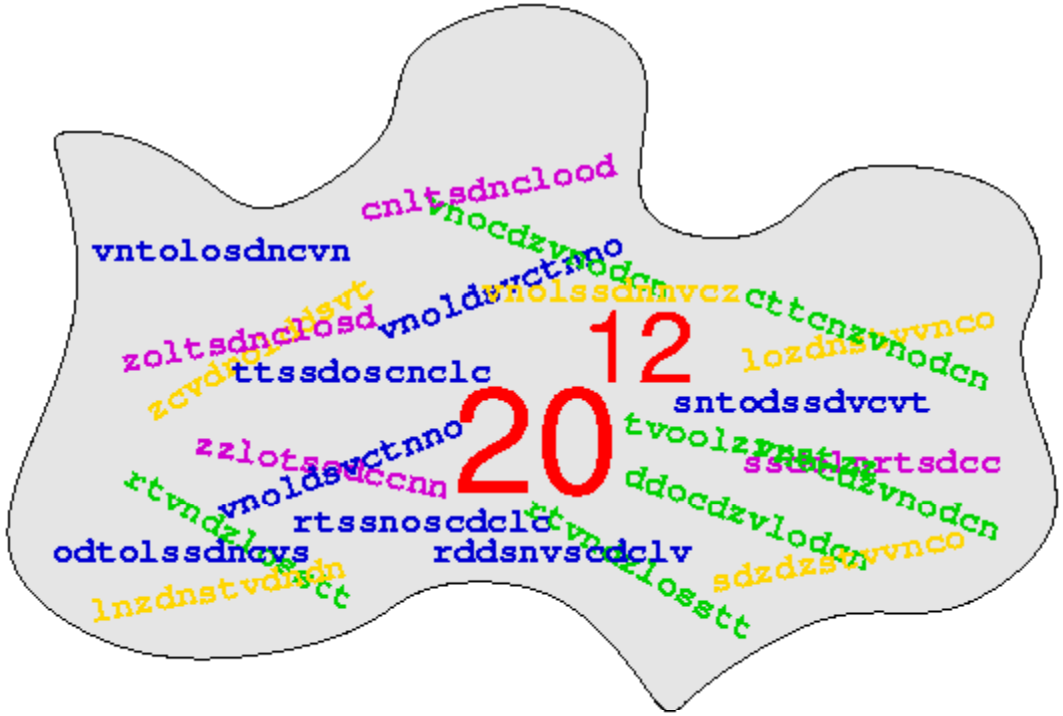
Opt. cost model

$$M : S \times A \rightarrow \Re$$

Not available,
will have to
be learned!

Find optimal sequence of optimizations for a specific program.

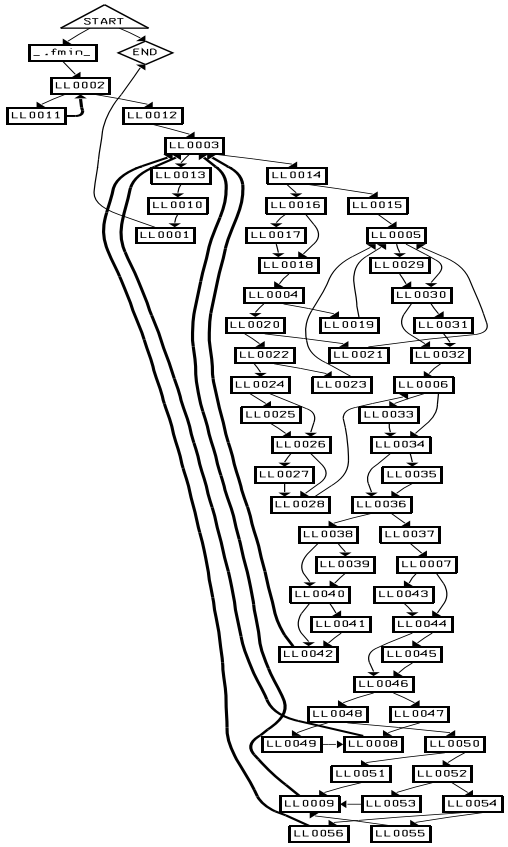
The problem



Combinatorial space of sequences (actions).
Discrete, unknown, non-linear model M:S x A → Re.

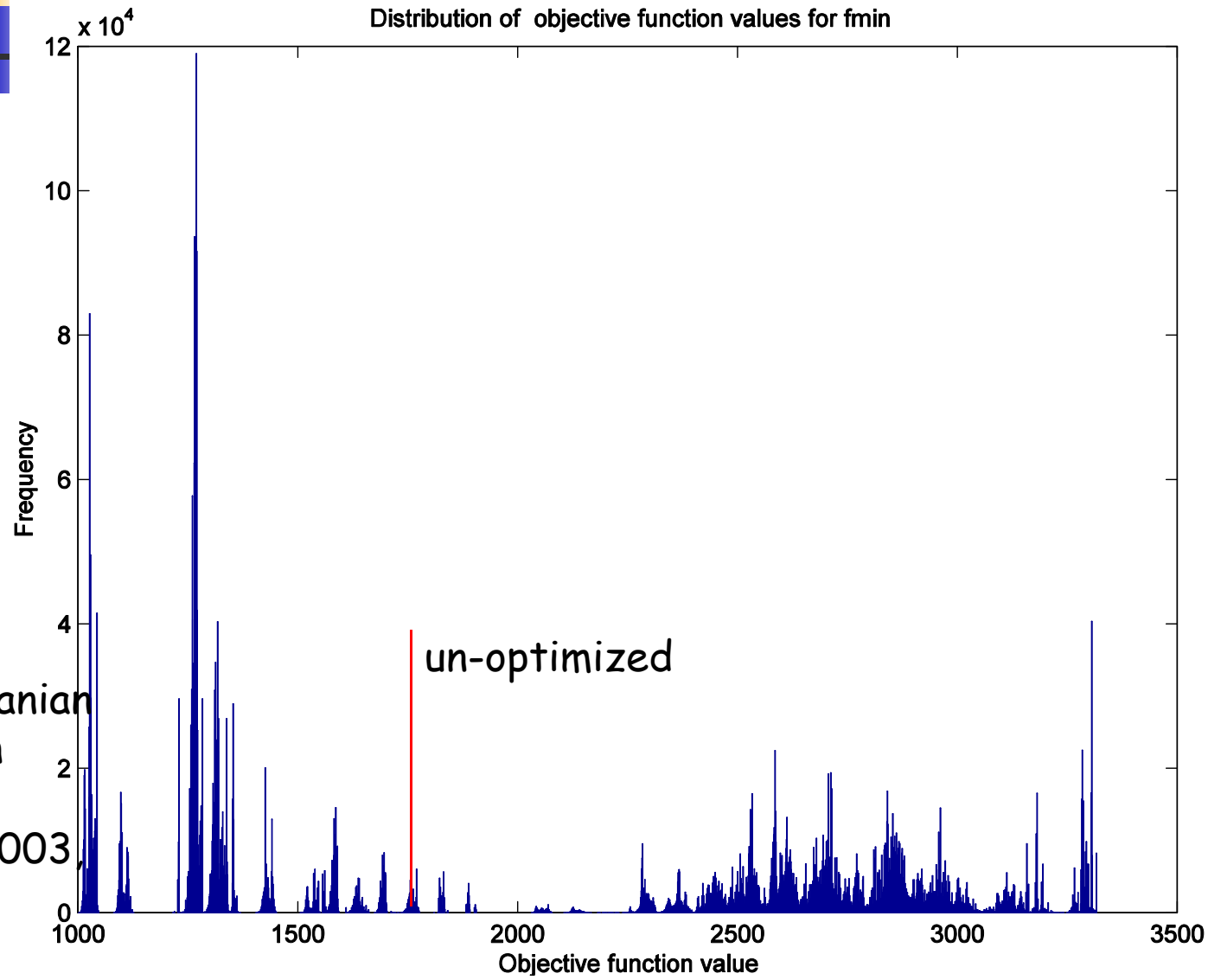
The fmin experiment

- Took one program, fmin
  - 150 lines of Fortran, 44 basic blocks
  - Exhibited complex behavior in other experiments
- Generated all sequences of length 10 from 5 opt
  - 9765625 combinations



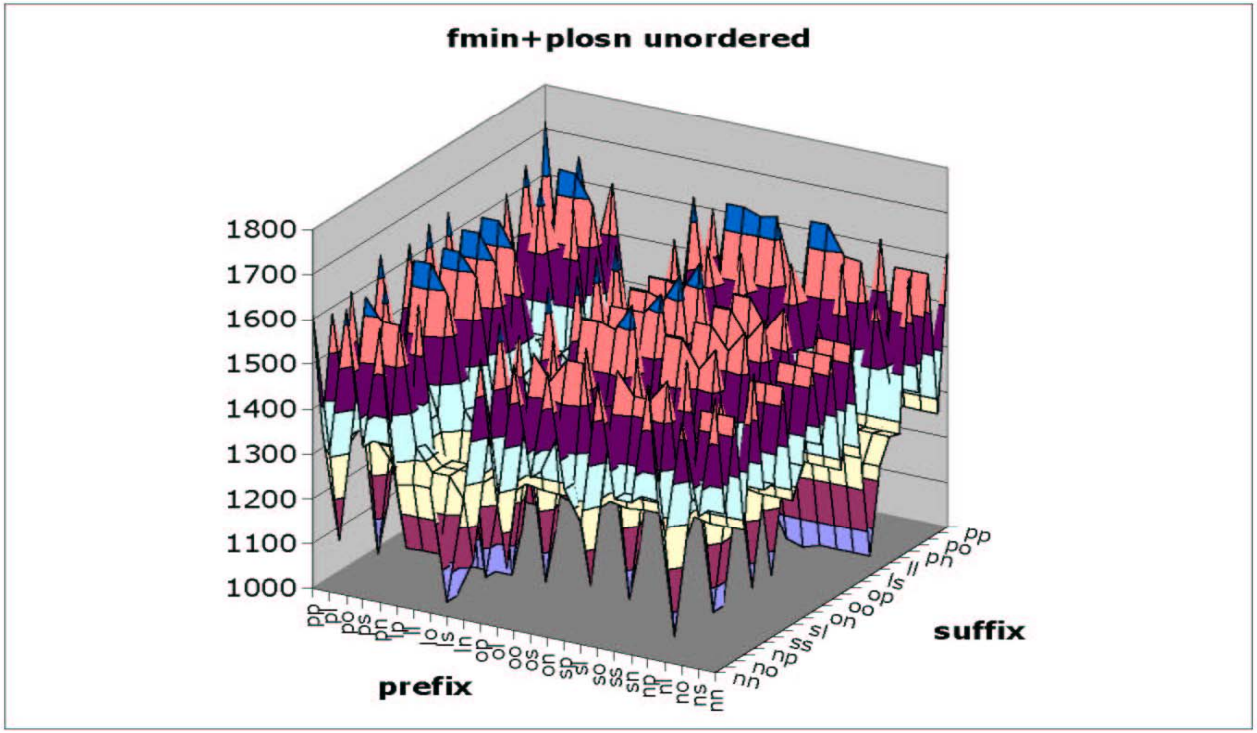
FMIN

Cost distribution

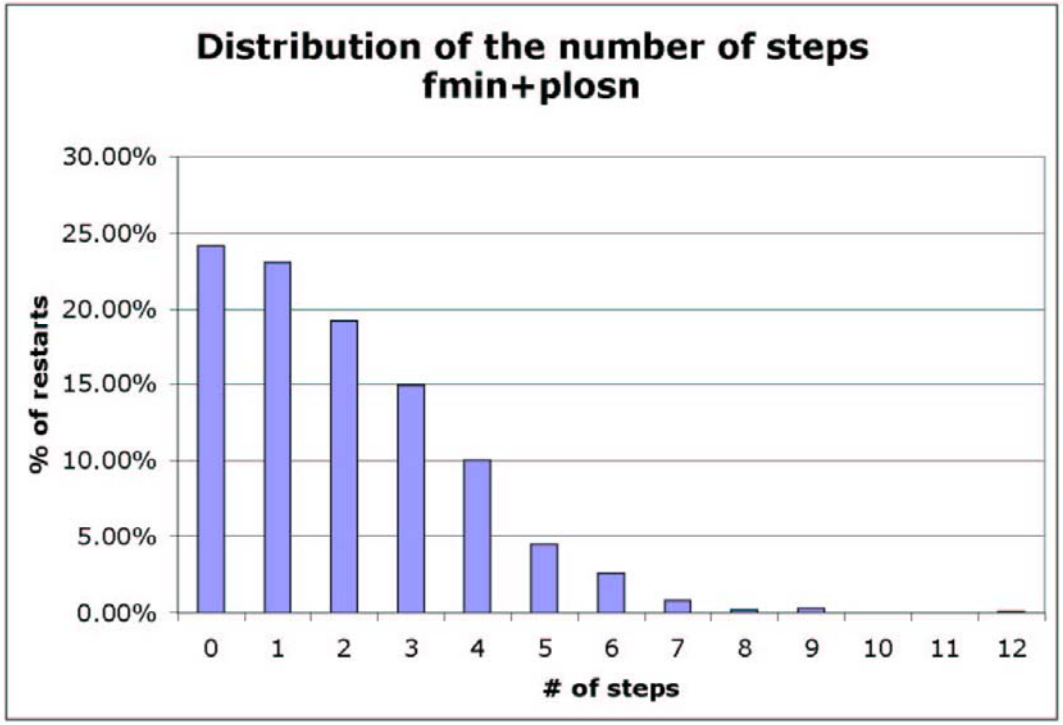


Cooper
Subramanian
Torczon
et. al.
2002, 2003,
2004

The landscape



Steps to local minima



Structure of sequence space

- Many local minima!
- 90% local minima are within 10% of best solution. 10% are 20-30% worse.
- Short hill-climbing runs.
- Suggests that hill climbing with randomized restarts is an appropriate algorithm.

Hill climbing with randomized restarts (HCRR)

- for m restarts do
  - 1. Start with s = a randomly generated sequence.
  - 2. (local improvement) for k random neighbors of s
    - Generate Hamming 1 neighbor s' of s
    - If s' better than s, make s = s' and go back to 2.
  - 3. If no local improvement, restart at 1.

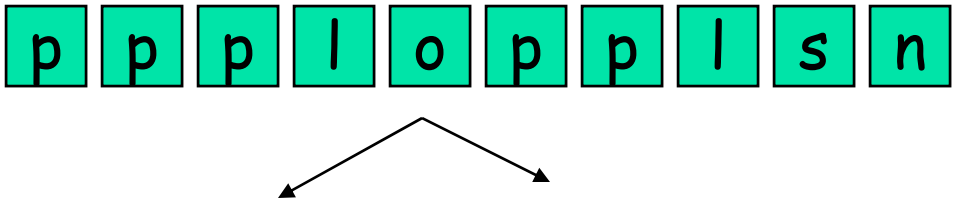
Role of learning

- for m restarts do
  - 1. Start with s = a randomly generated sequence.
  - 2. (local improvement) for k random neighbors of s
    - Generate neighbor s' of s
    - If s' better than s, make s = s' and go back to 2.
  - 3. If no local improvement, restart at 1.

Bias neighbor generation by N-gram models over subsequences.

Neighbor generation

p,l,o,s,n



40 neighbors (each with the same probability of being generated)

- HCRR runs yield sequences and their costs.
- Sort out good sequences.
- Learn n-gram models of optimizations on good seq

Summary of Scenario 3

- Approach:
  - Learn n-gram models $p_i(o_j \ldots o_{i+n})$ of the impact of optimizations on specific programs.
  - Use these models to perform a biased random HCRR search of the combinatorial space.
- Results
  - On Spec and Media benchmarks, approach has yielded improvements of 15-40% over human-designed universal optimization sequences.

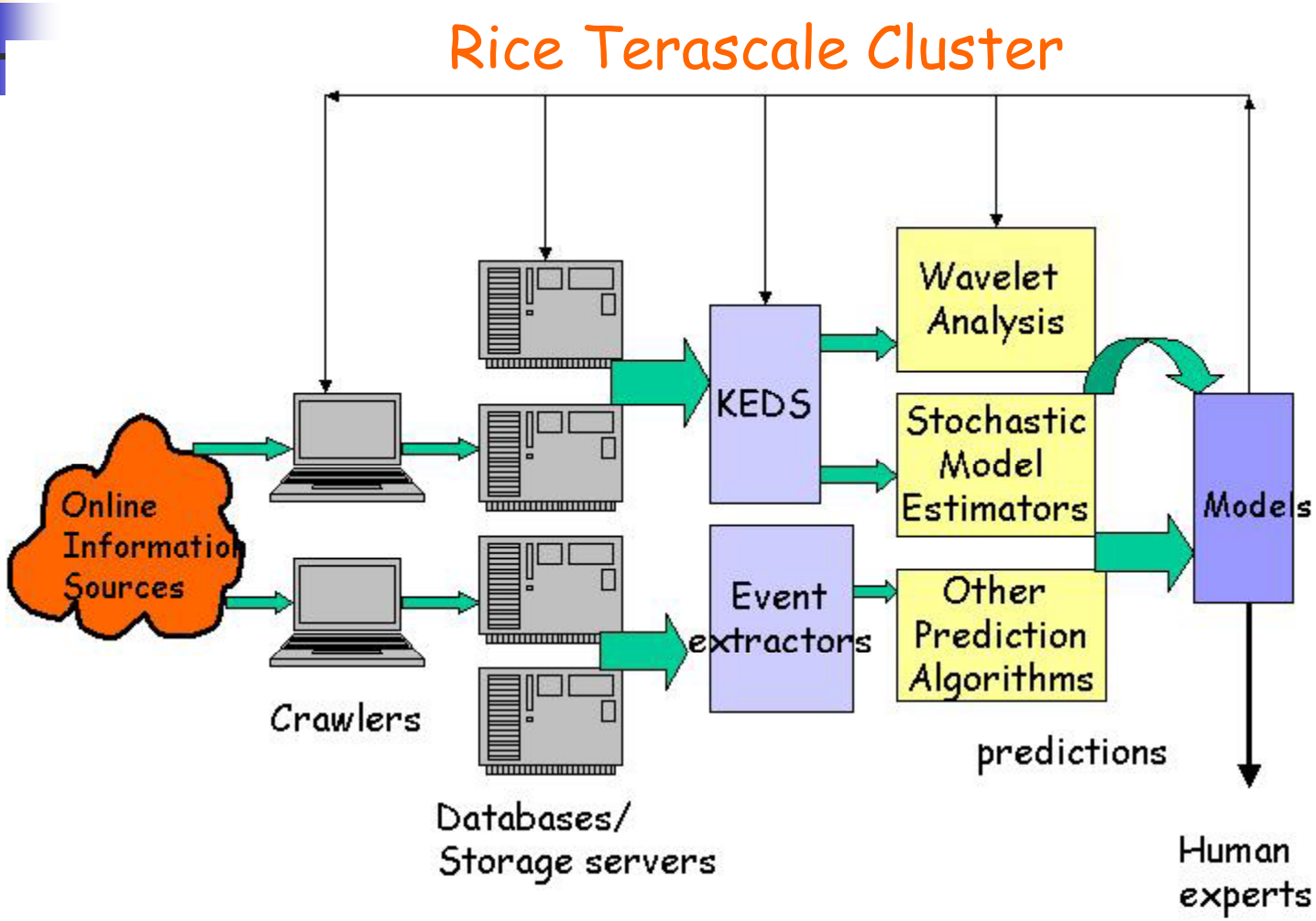
Outline

- Motivation
- Statistical machine learning
  - Scenario 1: Mars Bioplex
  - Scenario 2: Virgil
  - Scenario 3: Adaptive compilers
  - Scenario 4: Analyzing conflict
  - Scenario 5: Tracking human learning
- Conclusions

Scenario 4

- Is it possible to monitor news media from regions all over the world over extended periods of time, extracting low-level event information from them, and piece them together to automatically track and predict conflict in all the regions of the world?

Analyzing and predicting conflict



Thanks to NSF ITR- **0219673** (with Ric Stoll)

Example of event extraction

NY Times

Headline 03/31/2003: Iraq moving more troops to guard Baghdad from South.
Text: .....

Coding: 03/31/2003 IRQ US/Brit -9.5

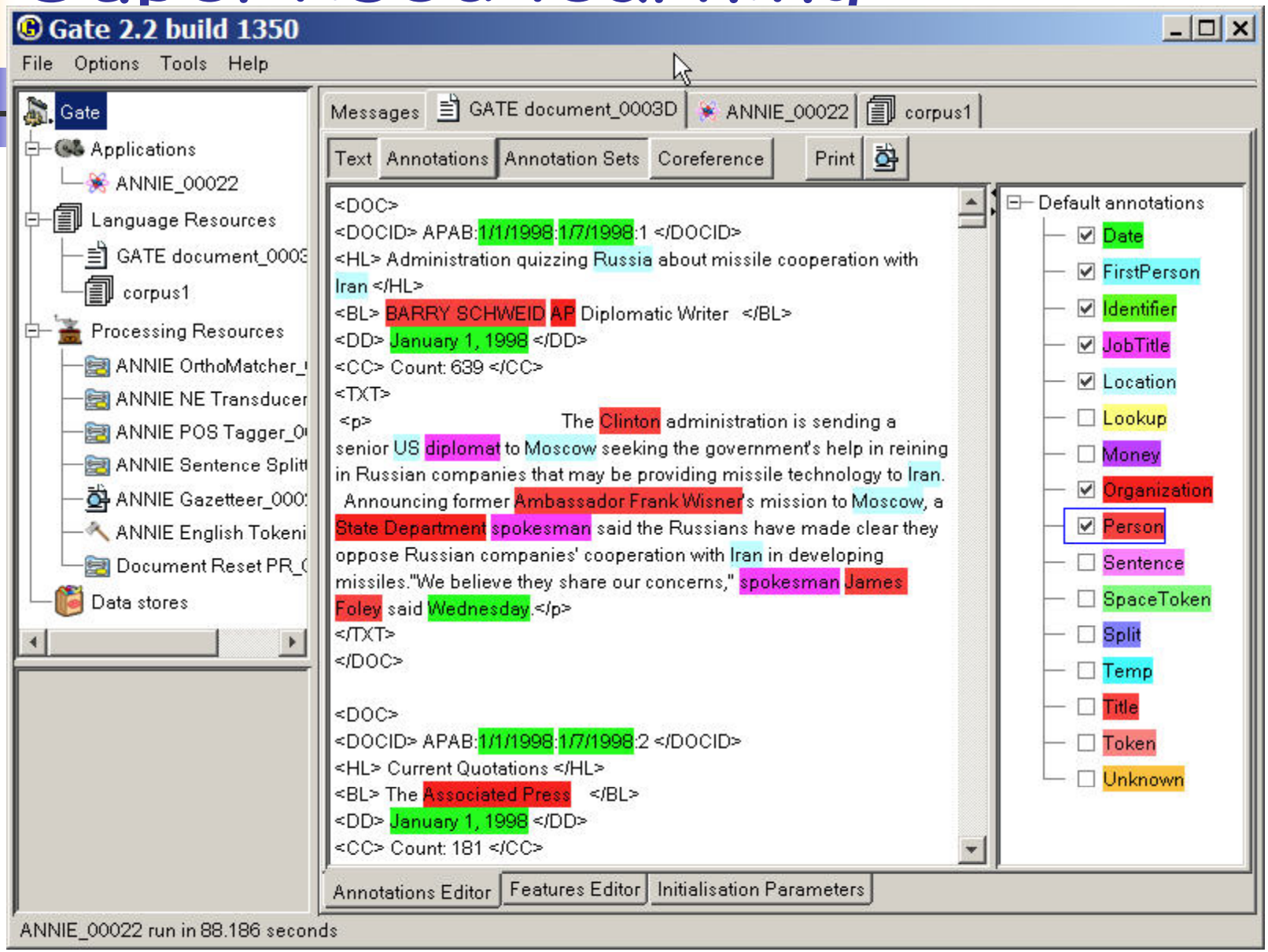
Who did what to whom and when?

Not in headline, needs to be inferred!

Extracting events

- Information extraction from news stories
  - Approach 1: assemble a team of human experts to write rules for extracting actors, targets and actions from sentences.
  - Approach 2: assemble a training corpus of sentences with extracted components, and learn rules for identifying different components in news stories.

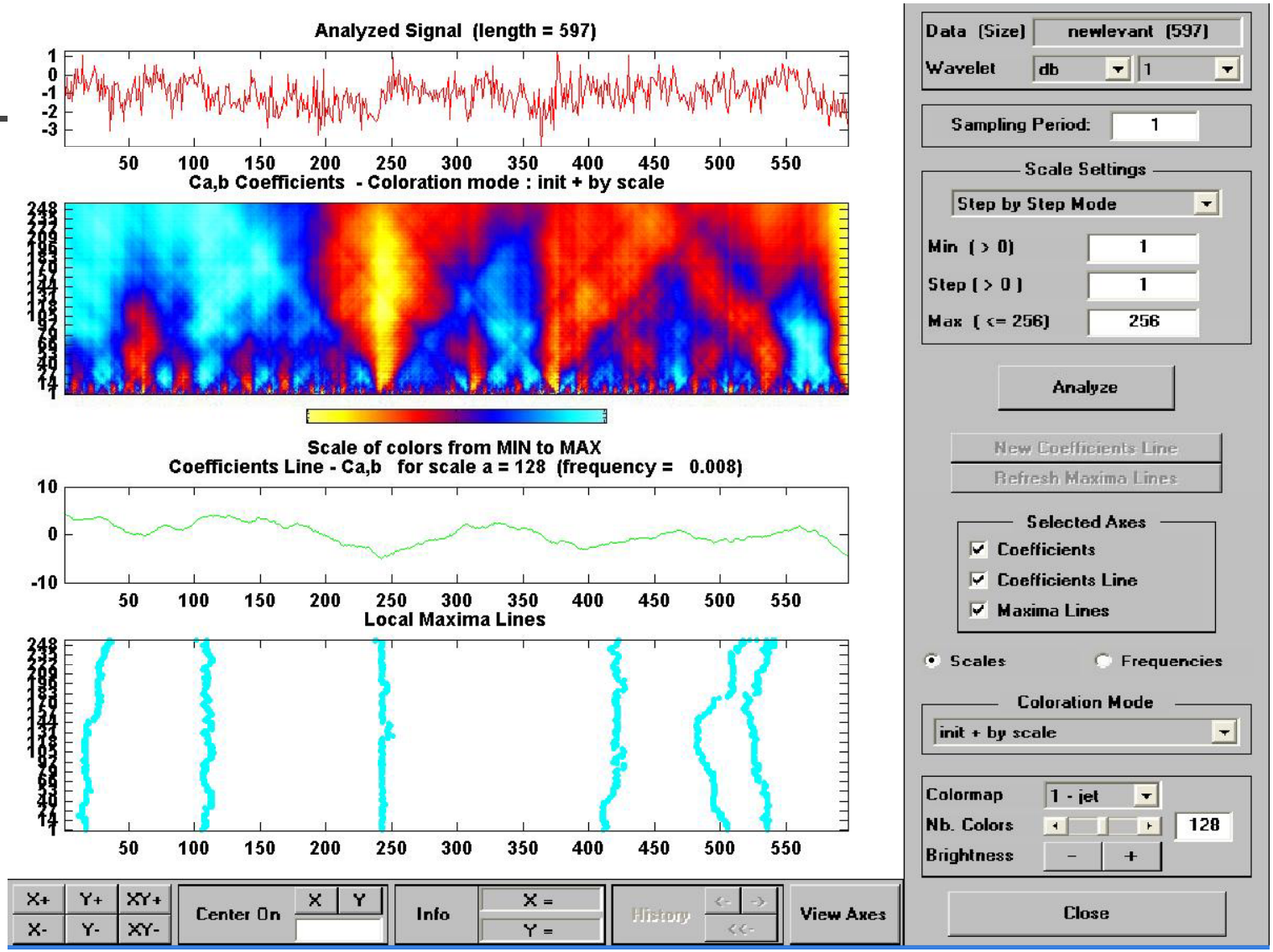
Supervised learning



Identifying relevant stories

- Filtering stories not pertaining to conflict.
  - Only about 10% of the stories contain events that are to be extracted.
  - The rest are interpretations of conflictual events (e.g., op-eds), or are events not about conflict (e.g., sports, human-interest stories, etc.).
- We have trained a Naïve Bayes classifier (with specificity and sensitivity of over 90%) for this phase. Further improvements using semantic information about words (WordNet) are being implemented.

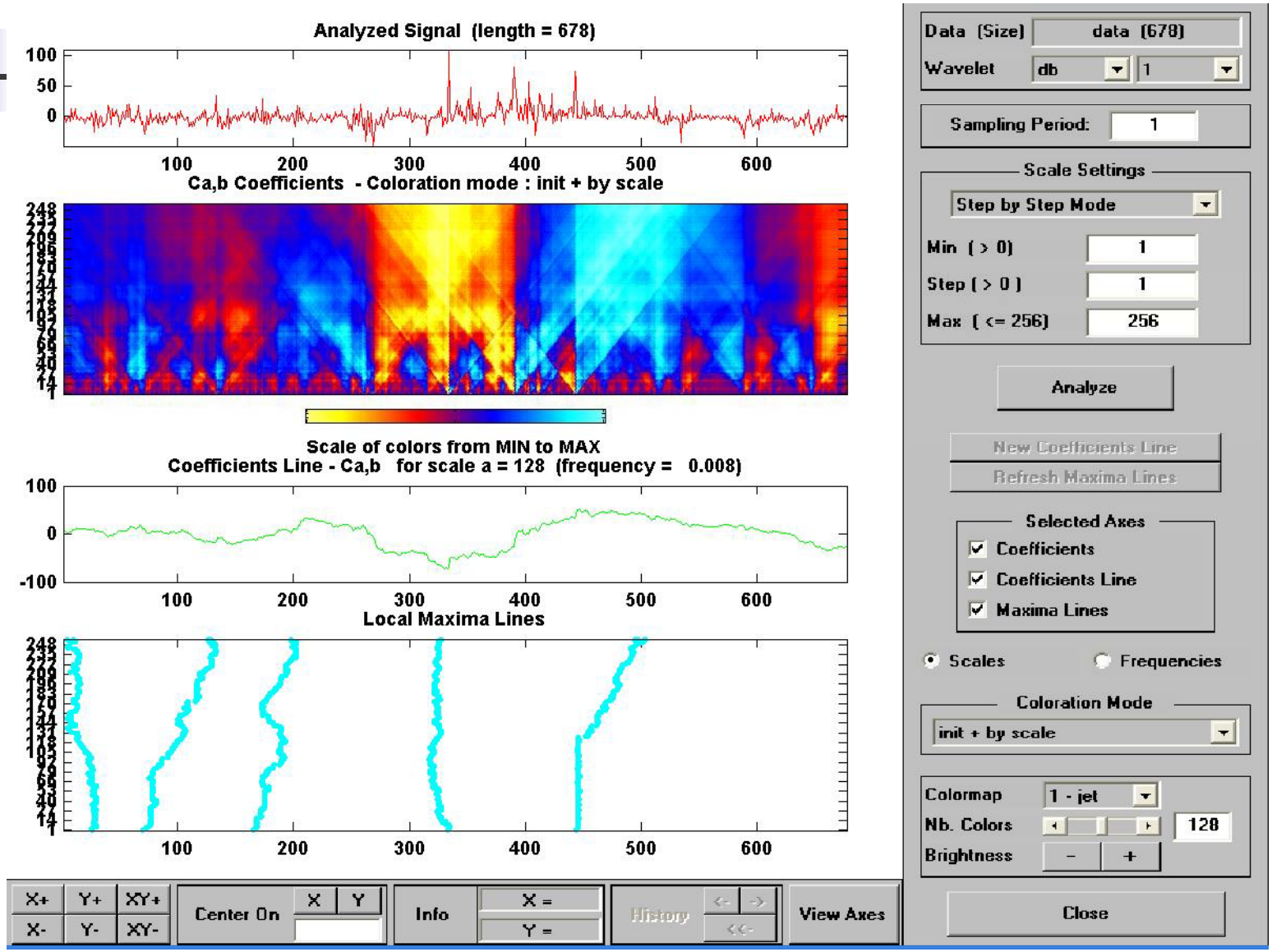
Analyzing conflict (Mideast)



Change point detection

biweek	Date range	event
17-35	11/79 to 8/80	Start of Iran/Iraq war
105-111	4/83 to 7/83	Beirut suicide attack
244	1/91 to 2/91	Desert Storm
413-425	1/95 to 7/95	Rabin assassination/start of Intifada
483-518	10/97 to 2/99	US/Iraq confrontation via Richard Butler/arms inspectors
522-539	4/99 to 11/99	Camp David summit hosted by Clinton

Analyzing conflict (Cold war)



Change point detection

Week(s)	Date(s)	Comments
56 – 83	67/1 – 67/8	[Red] Six Day War
132	68/7	[Red] Czech Crisis (138; 68/8), but NPT signed
187-265	69/8 – 71/1	[Red] Cienfuegos (Cuban sub base)
271-289	71/3 – 71/7	[Blue] Soviet ships leave Cienfuegos.
295-312	71/8 – 71/12	[Red] Soviets criticize Nixon trip to China, Moscow summit
316-334	72/1 – 72/5	[Blue] Moscow summit, SALT, ABM Treaties
341-363	72/7 – 72/12	[Red] US "Christmas bombing" of North Vietnam
364-390	72/12 – 73/6	[Blue] US-Soviet Summit (Brezhnev to US)
391-585	73/6 – 77/3	[Red] October War Crisis; Angola; Carter criticizes Soviets on human rights
620-649	77/11 – 78/6	[Red] Soviet UN employees arrested. Carter criticizes dissident trials. Soviet stage large maneuvers in Central Europe. Ginzburg & Shchransky sentenced.

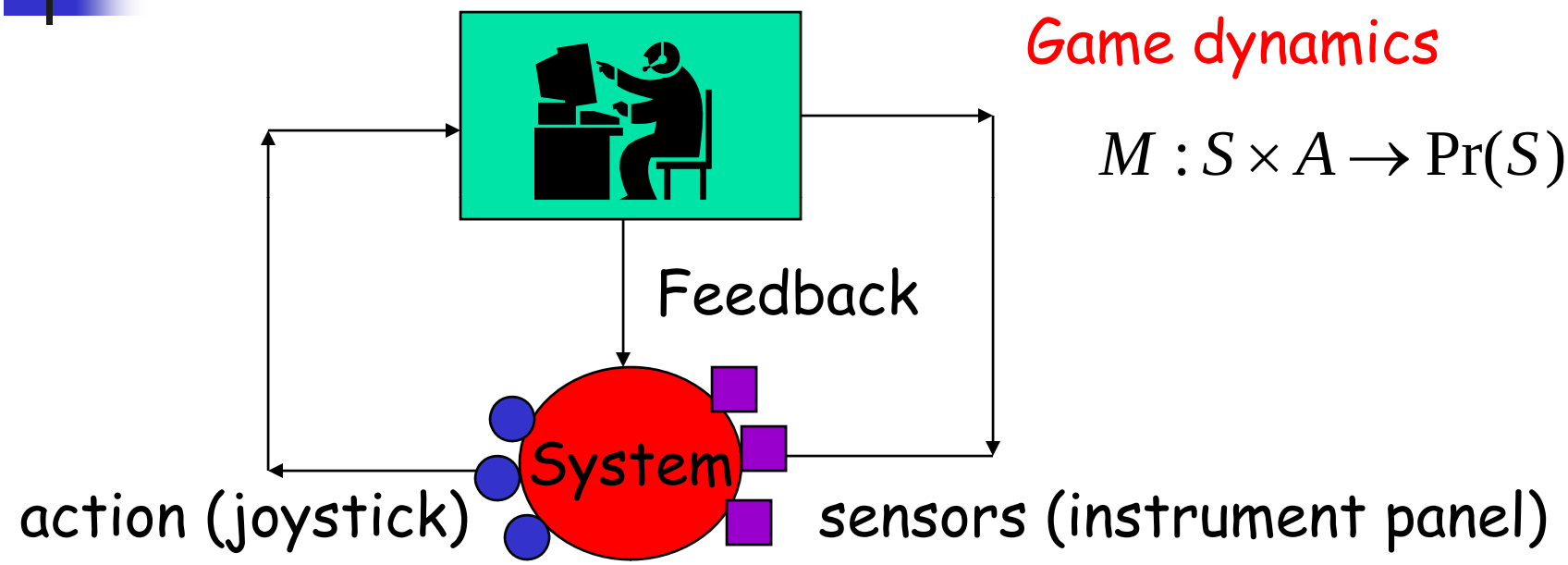
Outline

- Motivation
- Statistical machine learning
  - Scenario 1: Mars Bioplex
  - Scenario 2: Virgil
  - Scenario 3: Adaptive compilers
  - Scenario 4: Analyzing conflict
  - Scenario 5: Tracking human learning
- Conclusions

Scenario 5

- Tracking human learning on the NRL Navigation task
  - Approach 1: get cognitive scientists to write models of how humans learn.
  - Approach 2: gather visual and motor data from subjects and learn models directly from them.

Statistical machine learning



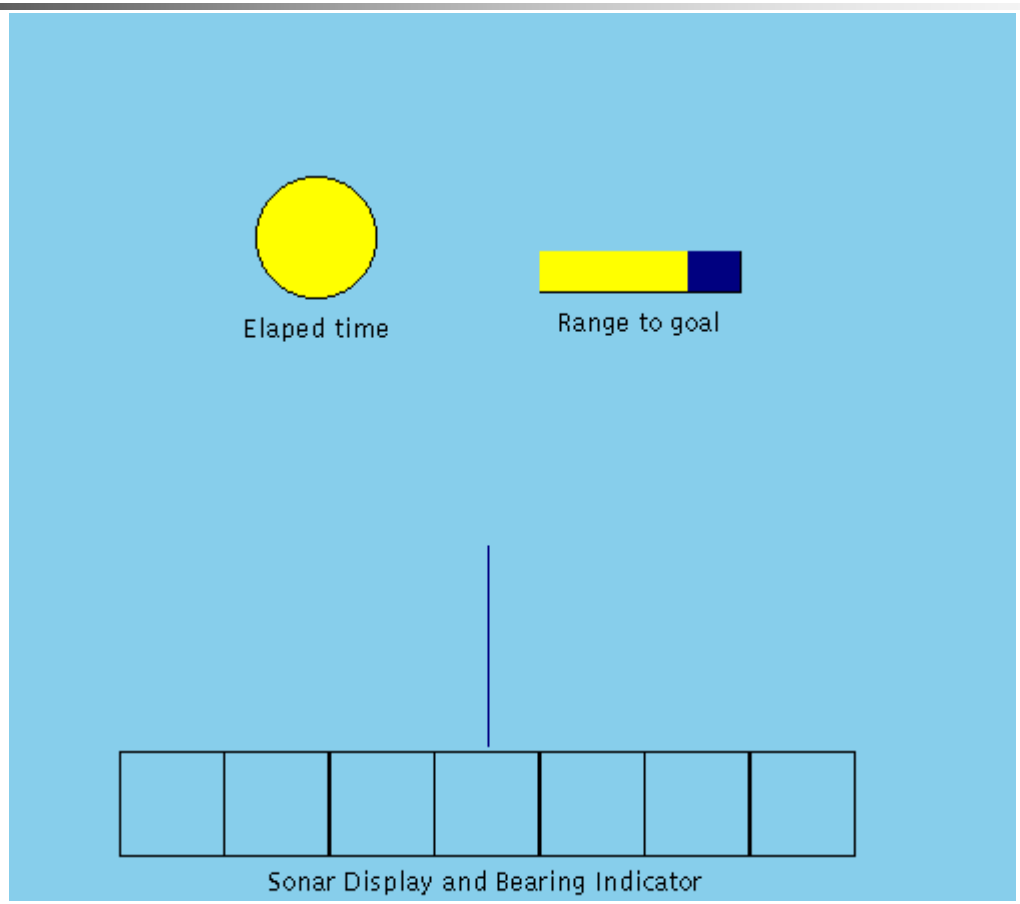
Game dynamics

$$M : S \times A \rightarrow \Pr(S)$$

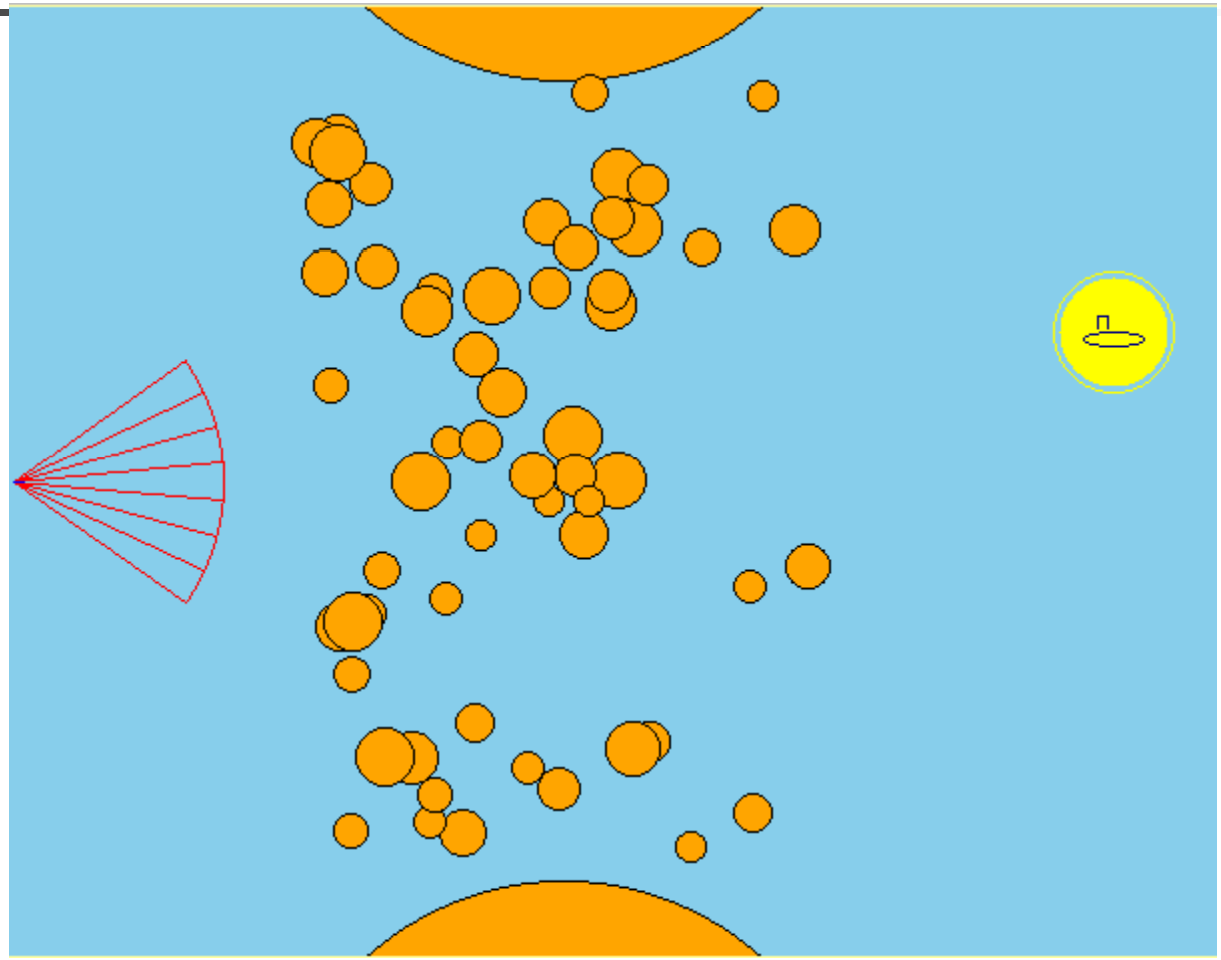
policy $\longrightarrow \pi : S \rightarrow \Pr(A)$

Track the evolution of the policy used by human, and alter training protocol to improve the speed and efficacy of learning.

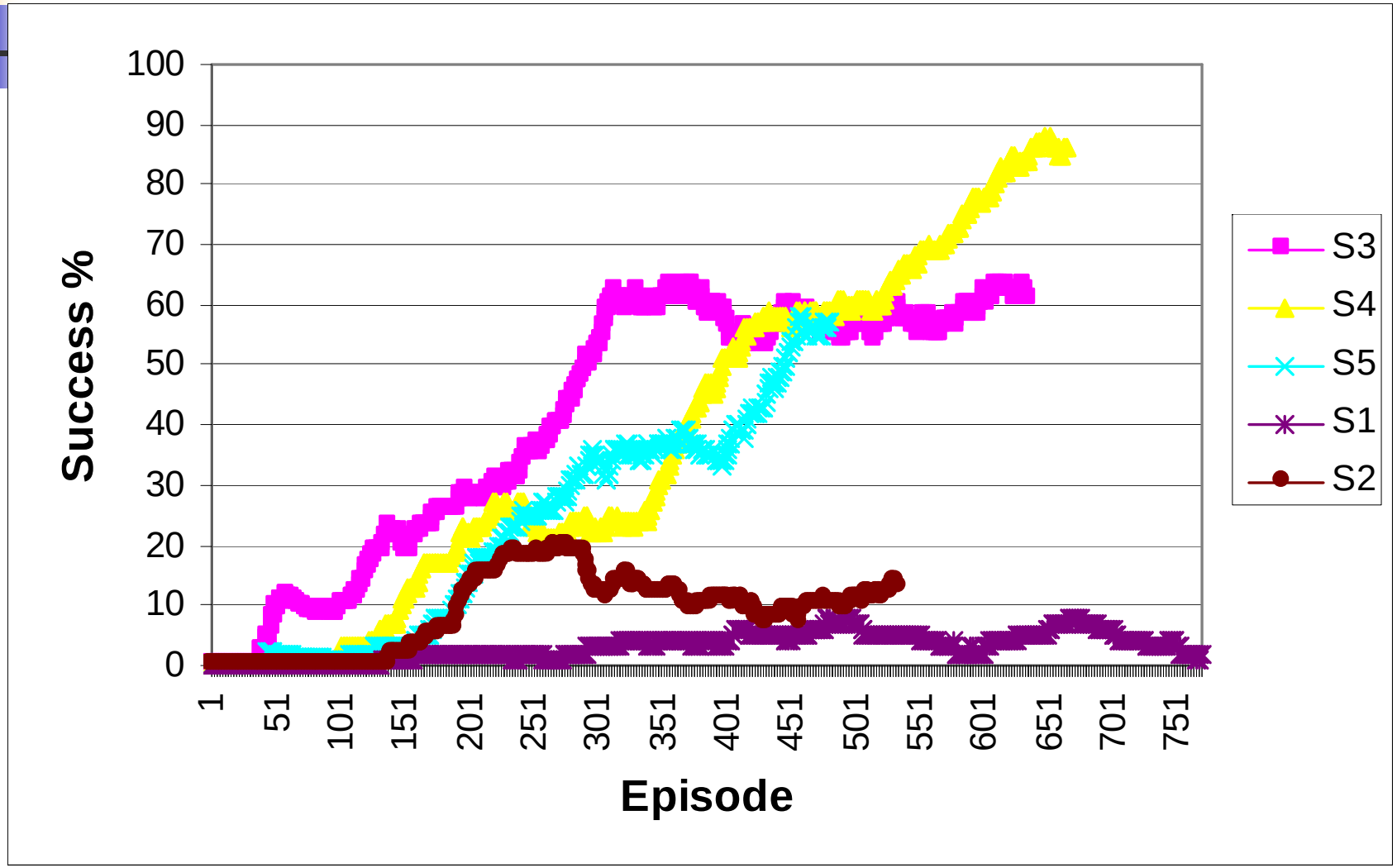
The NRL Navigation Task



The NRL Navigation Task



Learning curves (success)



Mathematical characteristics of the NRL task

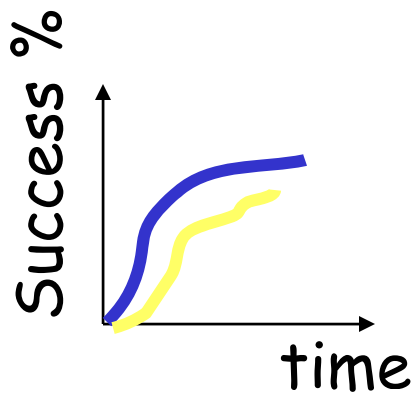
- A partially observable Markov decision process which can be made fully observable by augmentation of state with previous action.
- State space of size 10^{14}, at each step a choice of 153 actions (17 turns and 9 speeds).
- Challenging for both humans and machines.

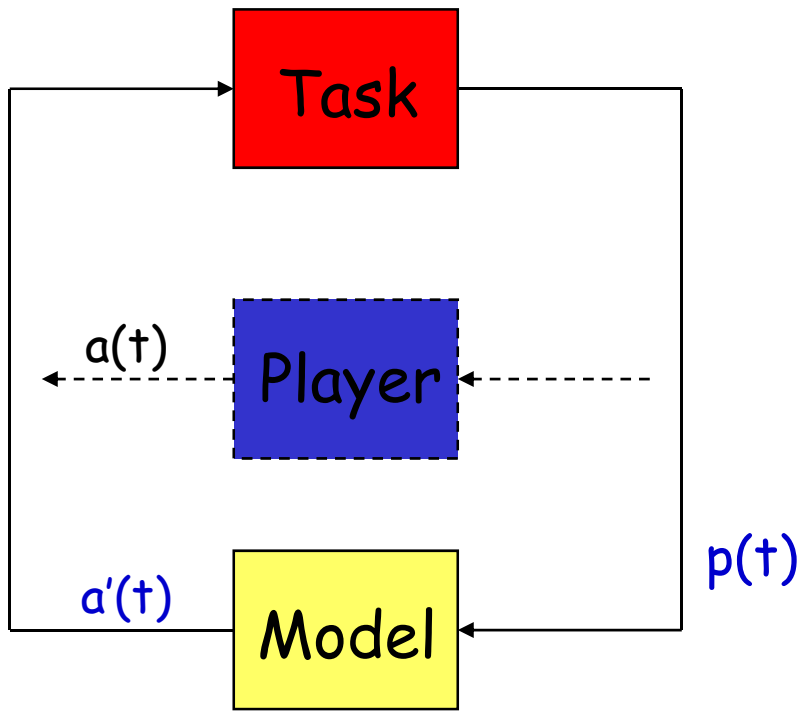
Challenges for a human learner

- A task with a significant strategic and a visual-motor component.
- Need for rapid decision making with incomplete information.
- The sheer number (10^{14}) of sensor panel configurations and action choices (153).
- Binary feedback at end of episode (200 steps).

Building Representative Models

- Behavioral equivalence (similarity in learning curves)





Models of policy

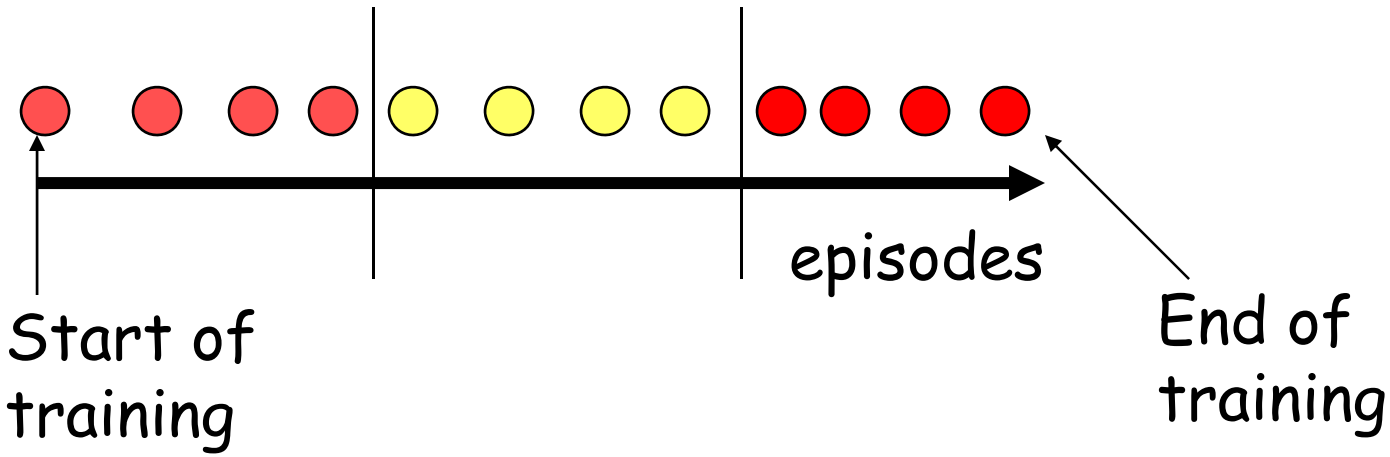
- Consider stateless stochastic models of the form pi:sensors $\rightarrow$ Pr(actions).
  - Associate with every observed sensor panel configuration, the distribution of actions taken by the player at that configuration.
- Advantage:
  - no need to abstract sensor space.
  - Model construction can be done in real time!

Surely, this can't work!

- There are 10^{14} sensor configurations possible in the NRL Navigation task.
- However, there are between 10^3 to 10^4 of those configurations actually observed by humans in a training run of 600 episodes.
- Exploit sparsity in sensor configuration space to build a direct model of subject's policy.

Model construction

- Segmentation of episodic data



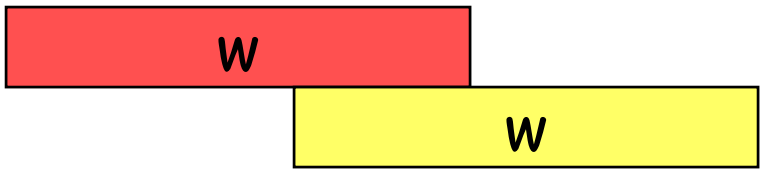
- Fit models of the form sensors → Pr(actions) on the stationary segments.

Model Derivative

- dm/dt =

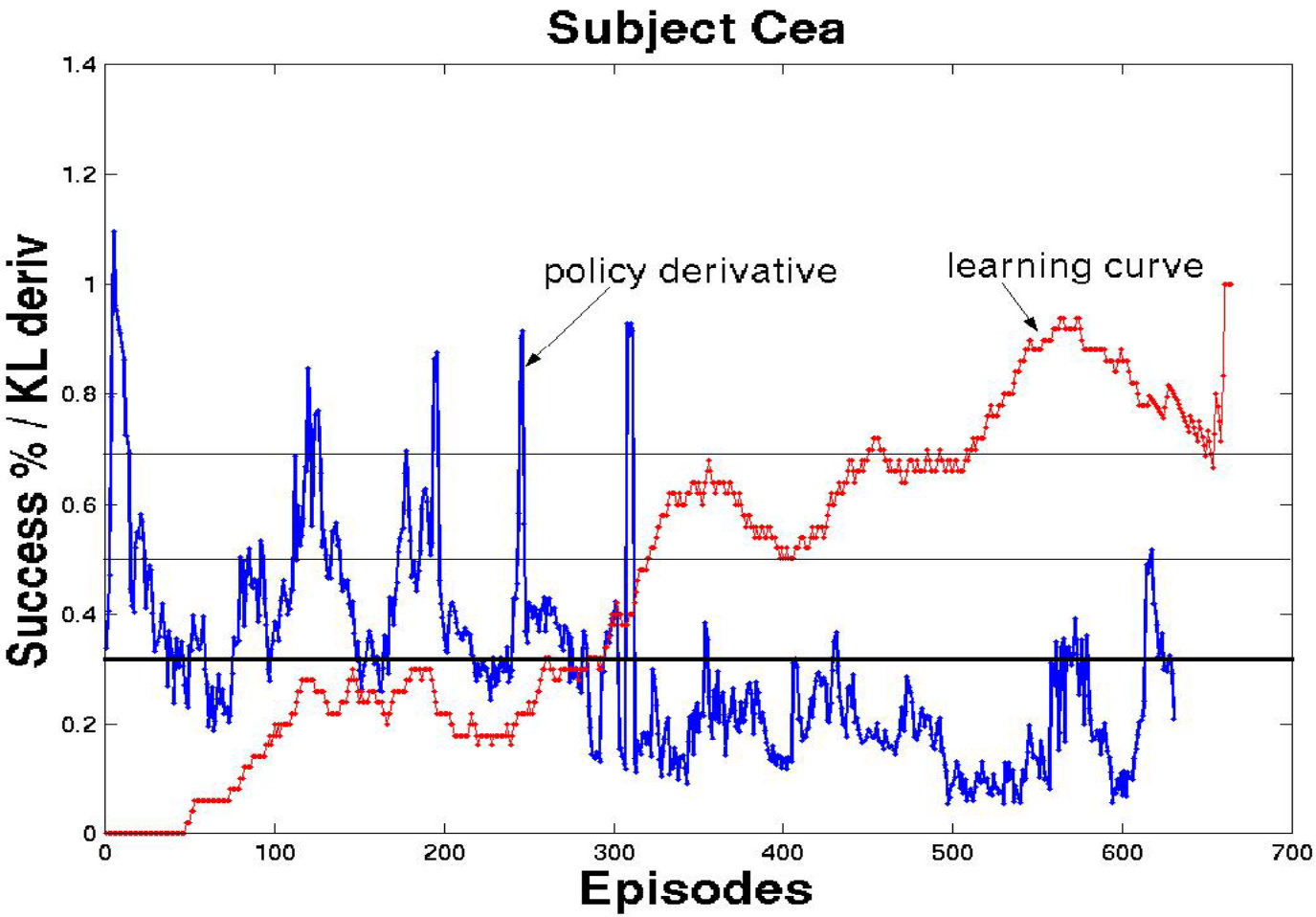
$$\frac{KLdiv(\Pi(i+w-s, i+2w-s), \Pi(i, i+w))}{w}$$

- empirical optimum: w = 20, s = 5
- Computed by Monte Carlo sampling (stabilizes after 5% of entries are sampled)

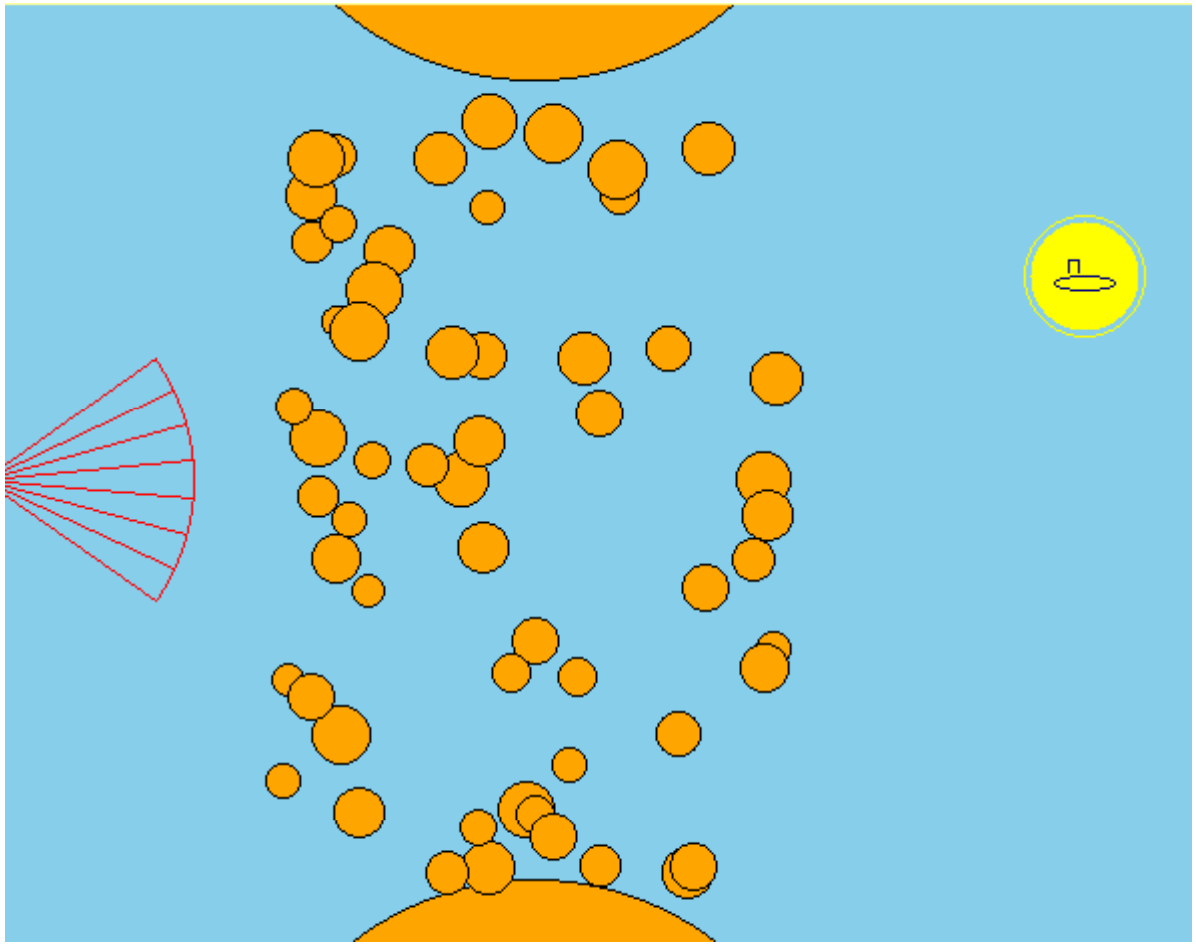


Overlap = s

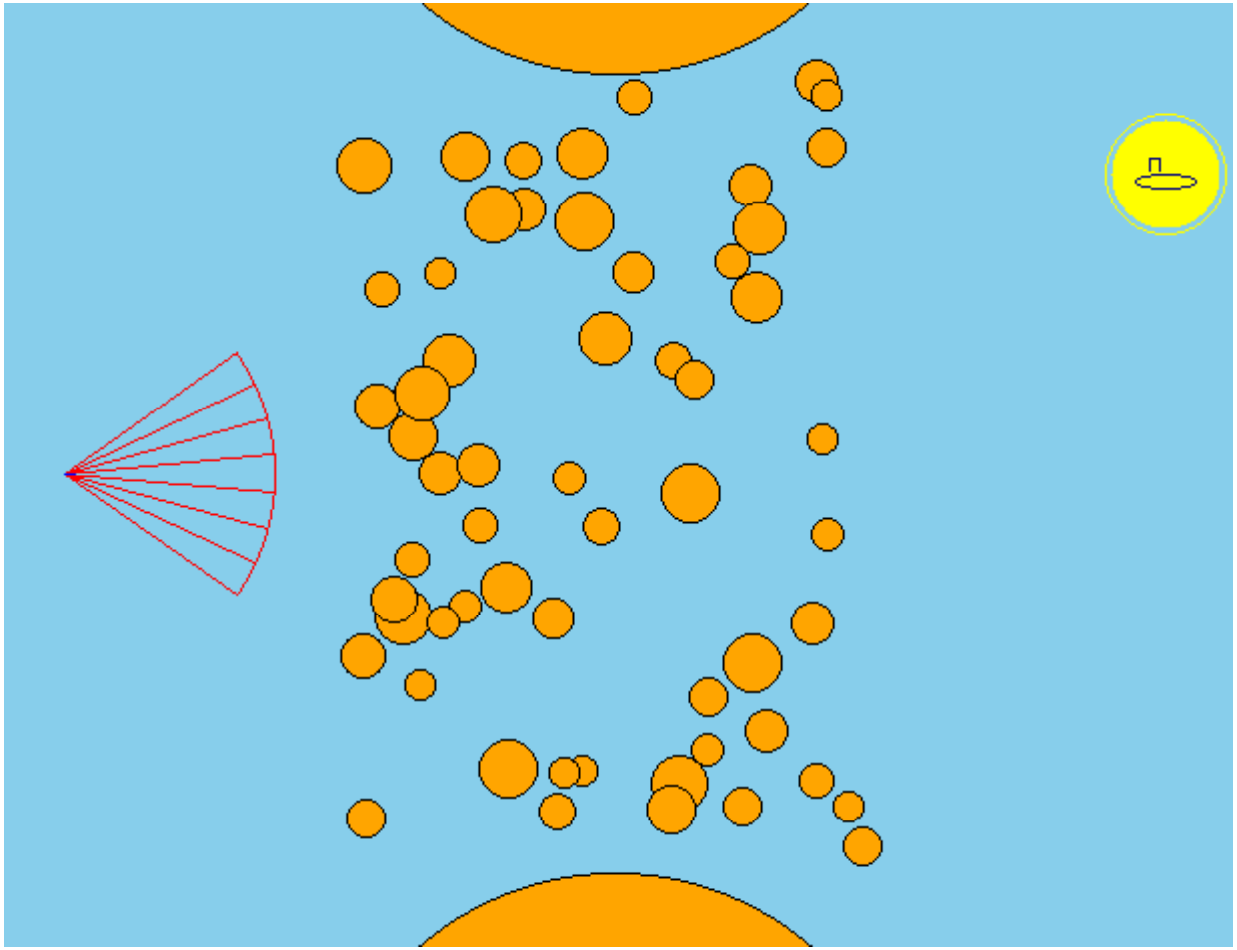
Model derivative for Cea



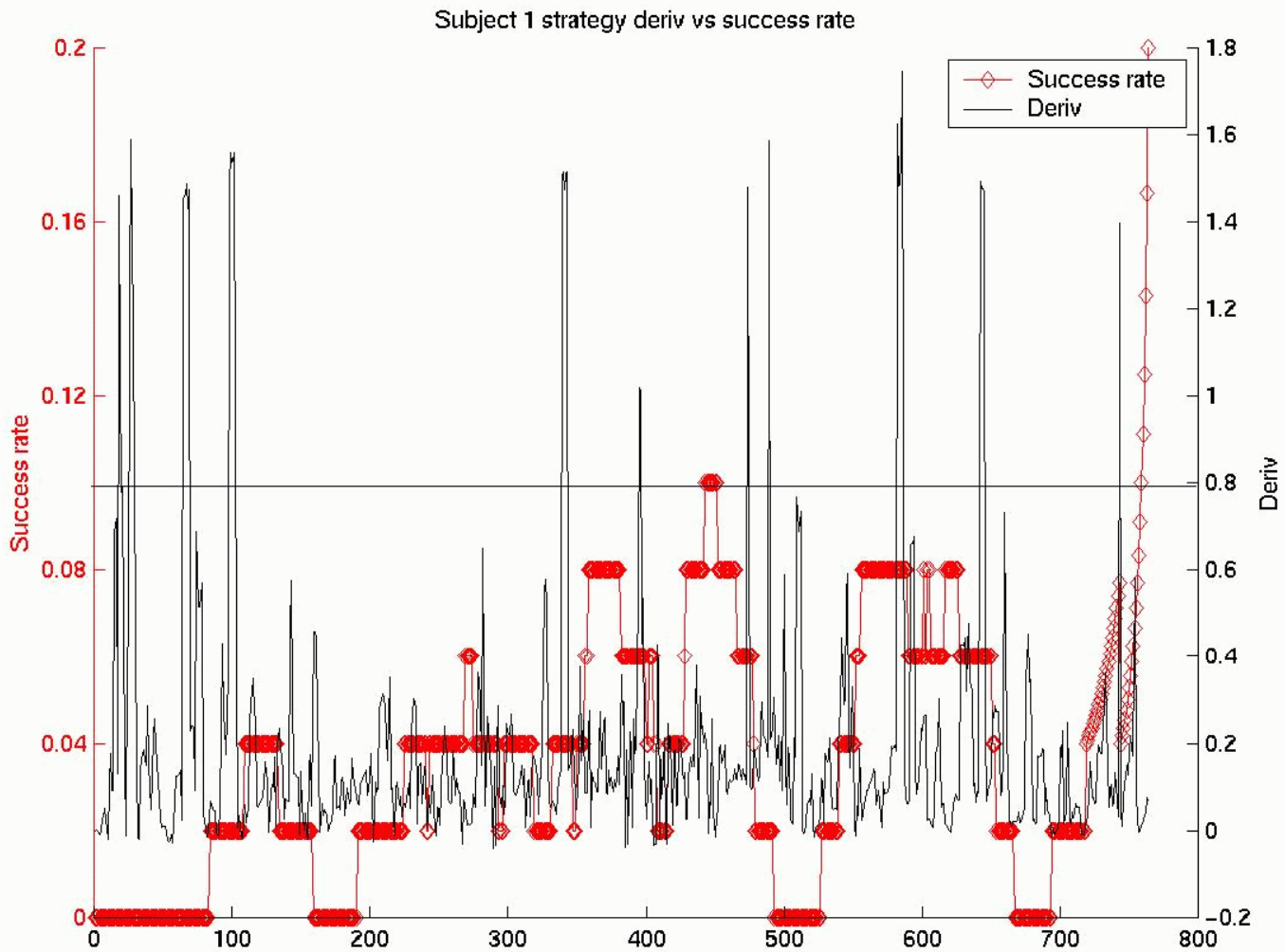
Before shift: Cea (episode 300)



After shift: Cea (episode 320)



Model derivative for Hei



Nearest-neighbor action computation

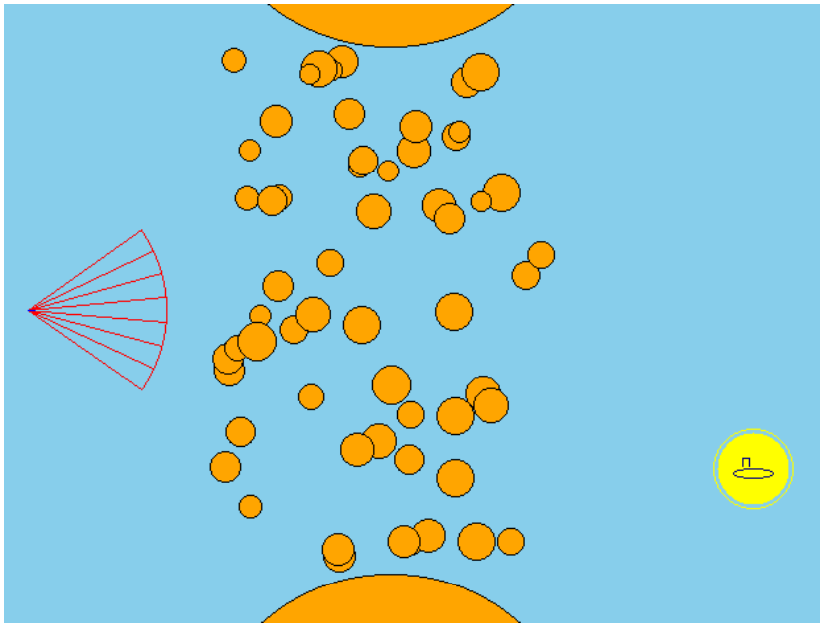
- To compute action a associated with current sensor configuration s
  - take 100 neighbors of s in policy table.
  - perform locally weighted regression (LWR) on these (s,a) pairs.

Evaluation protocol

- Same mine configurations as subject.
- Model switched on segment boundaries.
- Cross-validation method on each segment:
  - Train on 9/10^{ths} of data
  - Test on left-out chunk

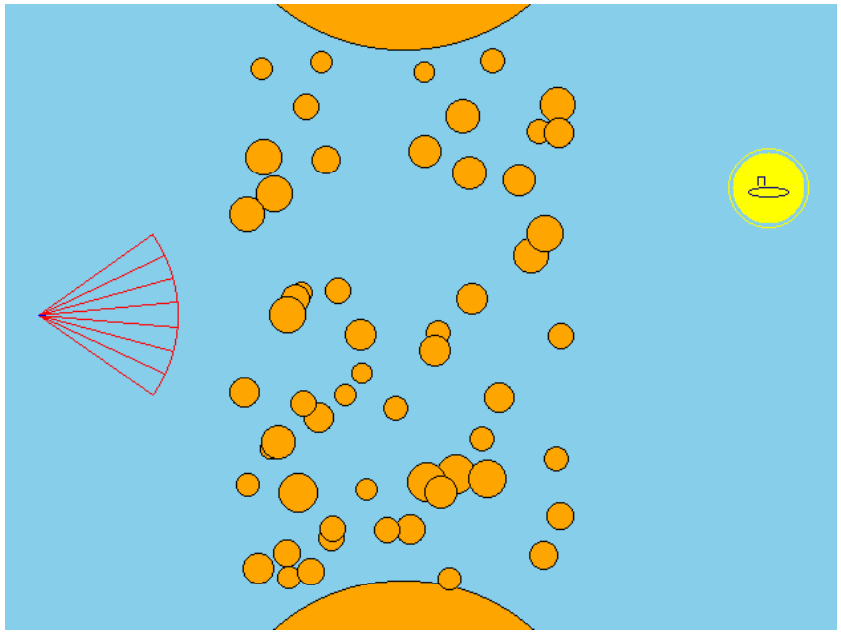
Subject Cea: Day 5: 1

Subject

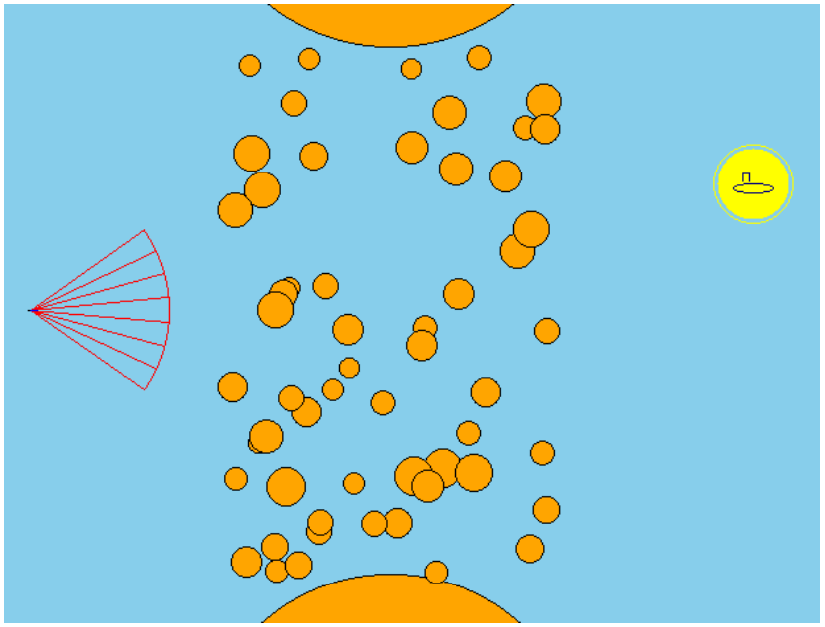


Model

Subject Cea: Day 5: 2

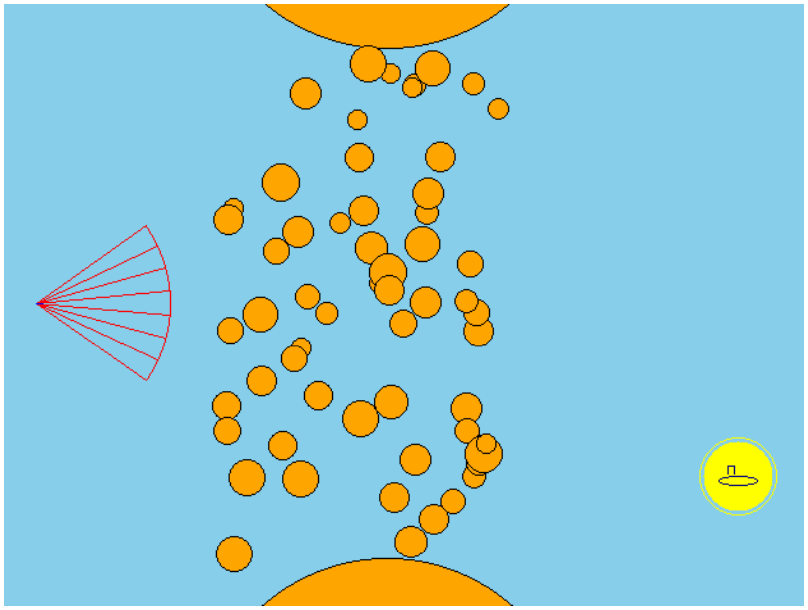


Subject

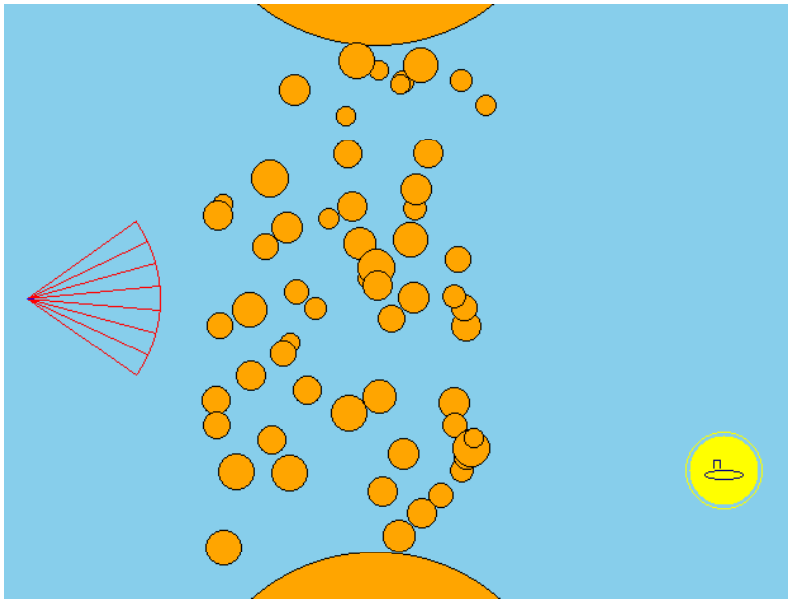


Model

Subject Cea: day 5: 3

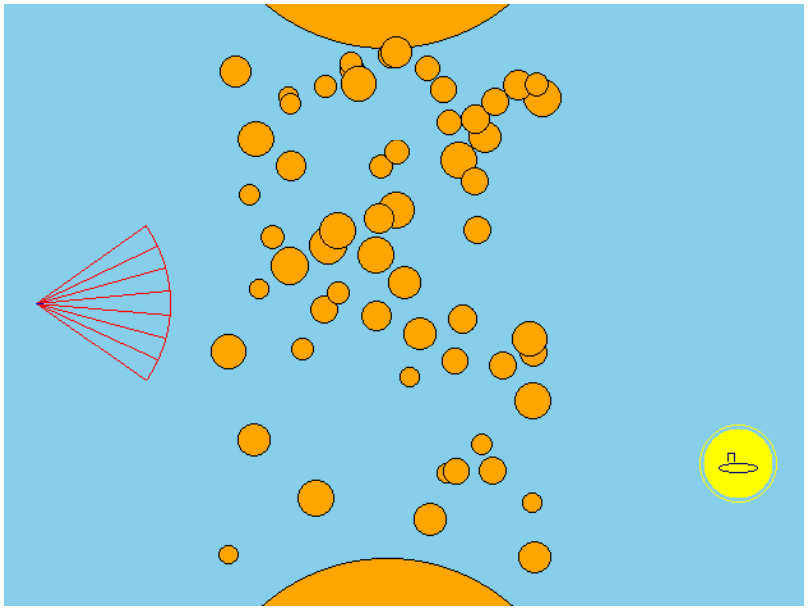


Subject

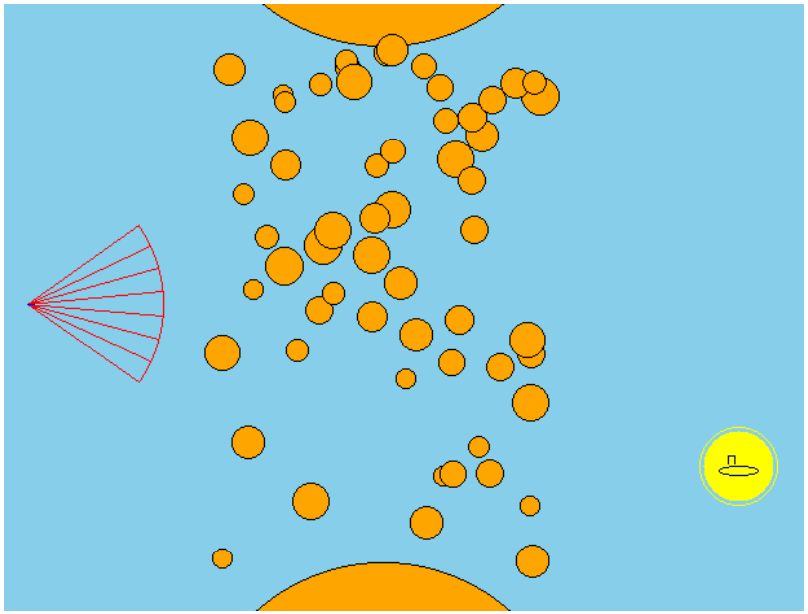


Model

Subject Cea: Day 5: 4

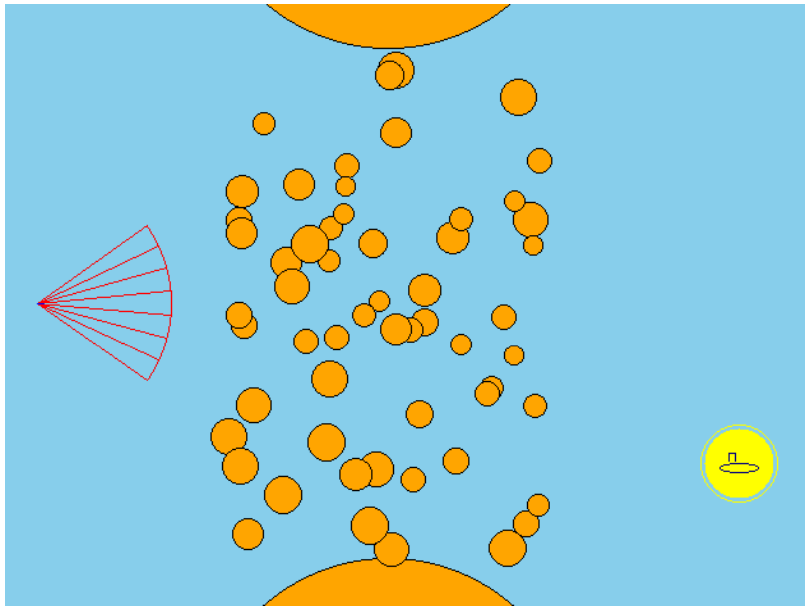


Subject

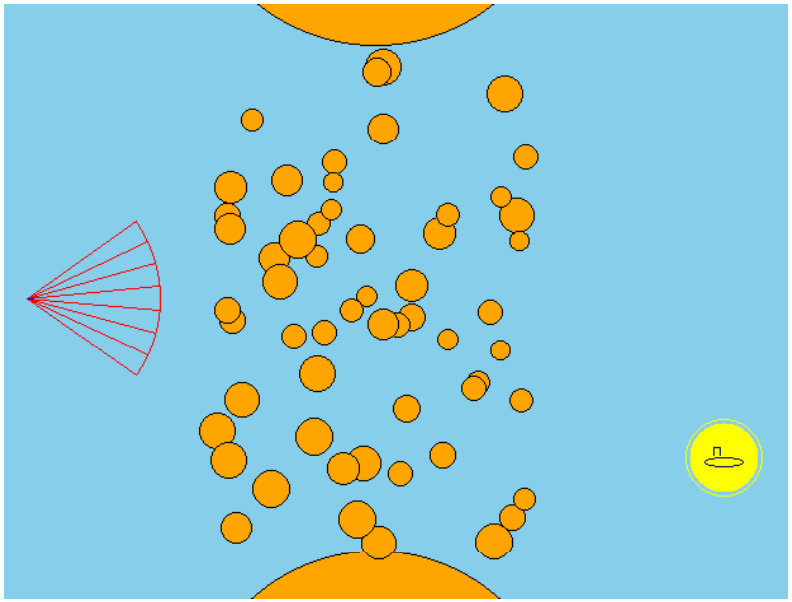


Model

Subject Cea: Day 5: 5

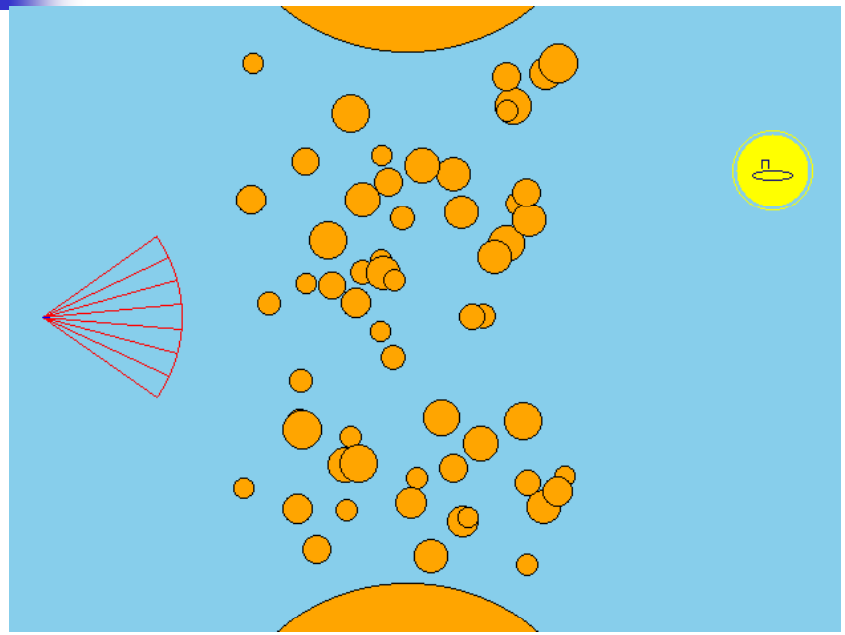


Subject

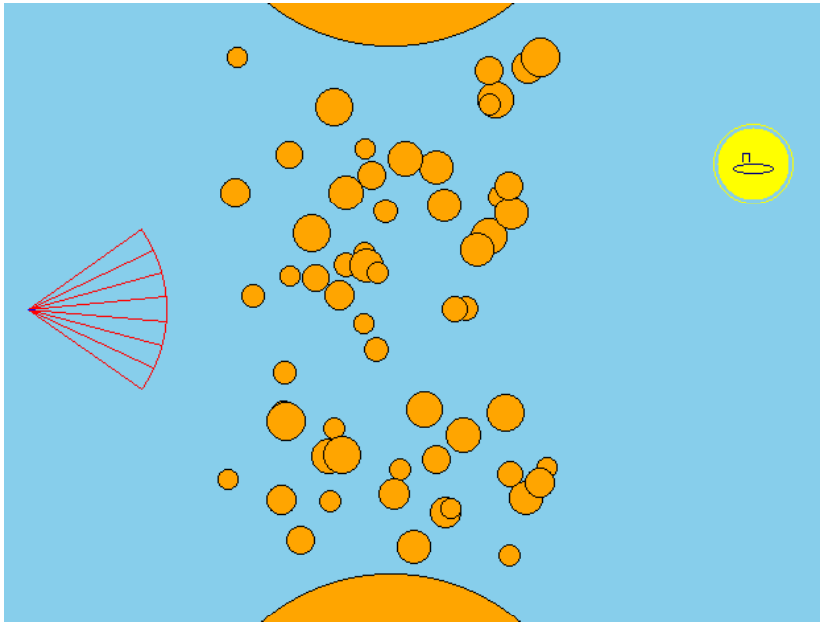


Model

Subject Cea: Day 5: 6

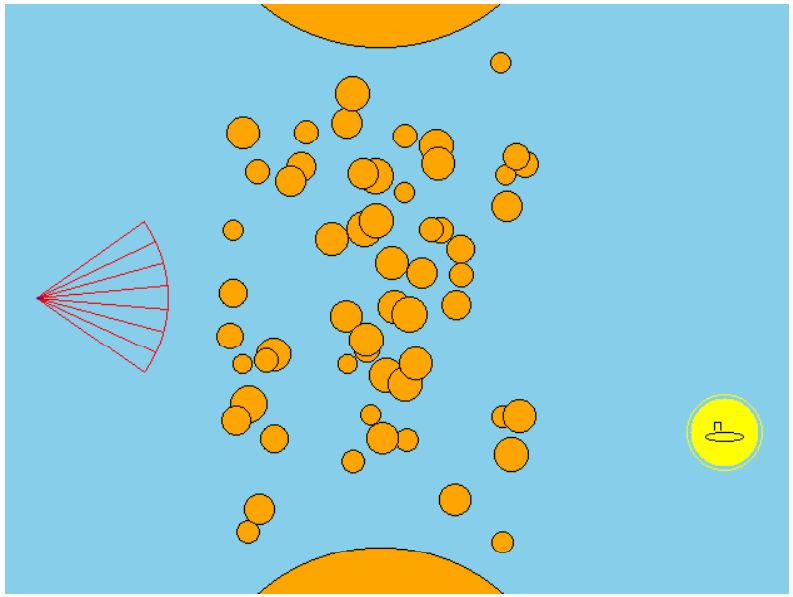


Subject

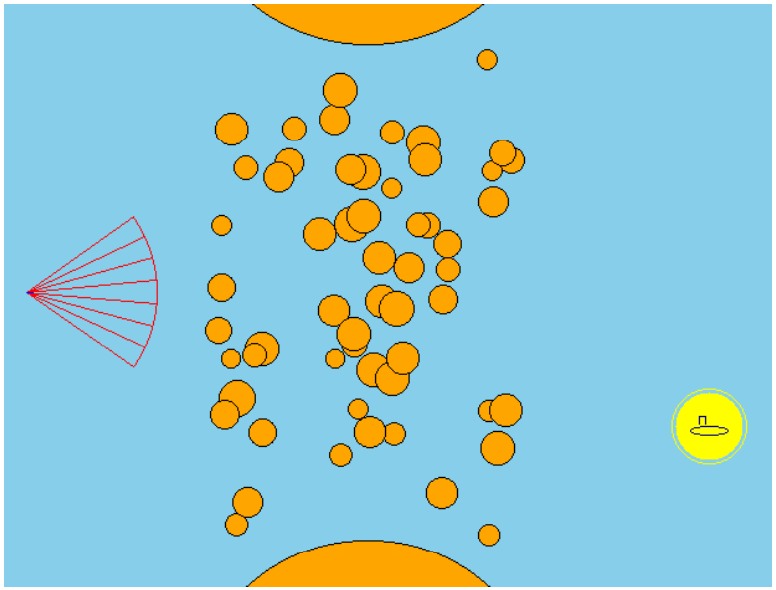


Model

Subject Cea: Day 5: 7

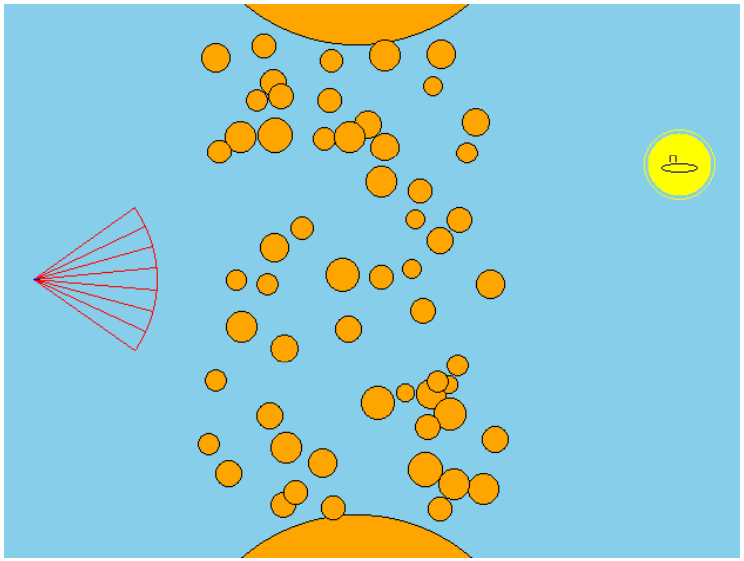


Subject

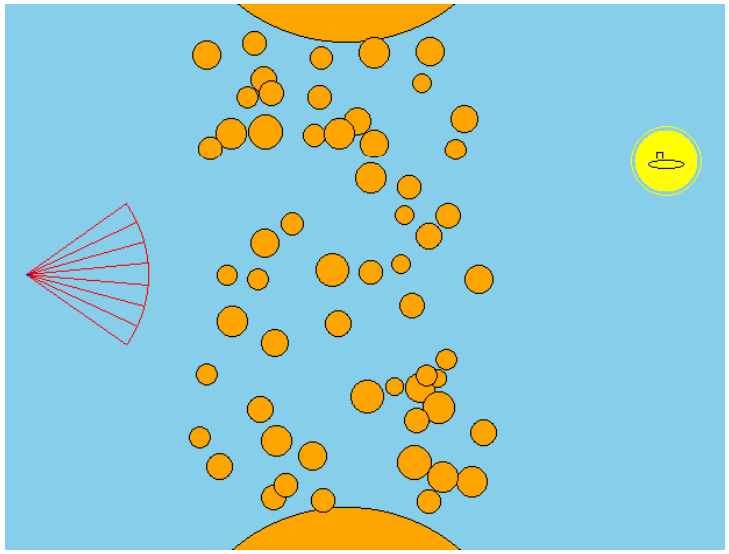


Model

Subject Cea: Day 5: 8

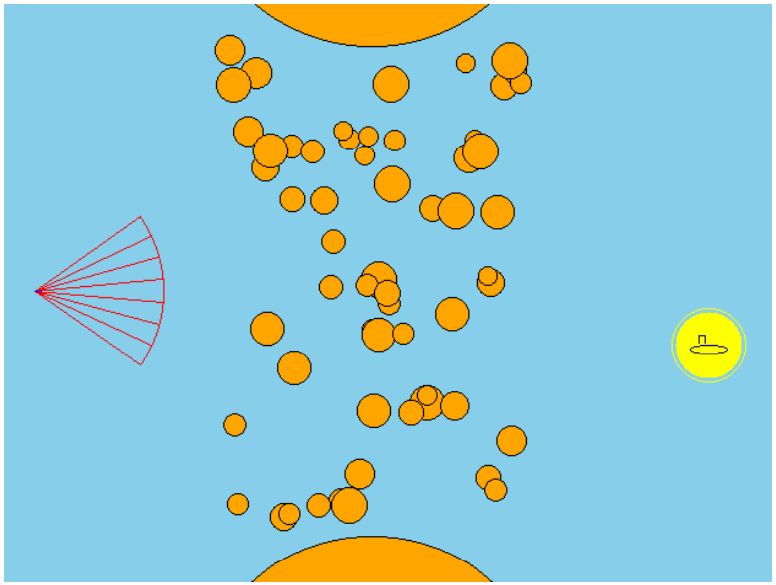


Subject

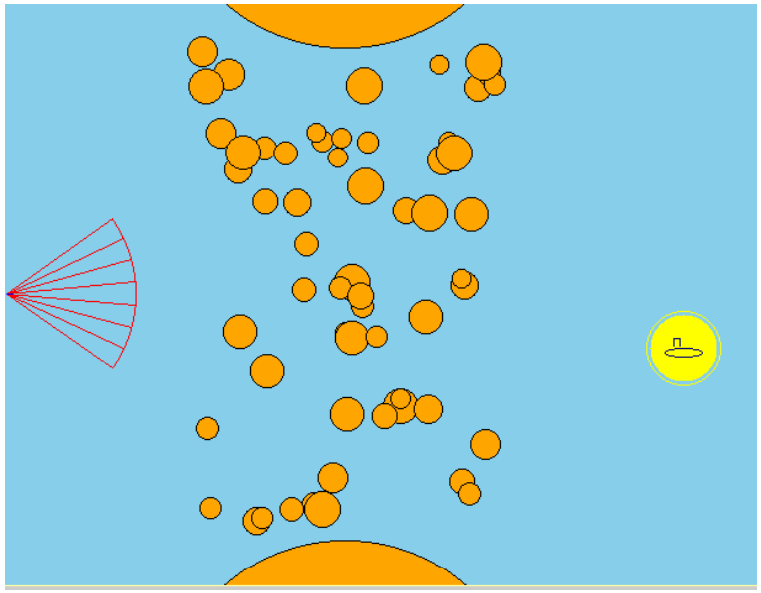


Model

Subject Cea: Day 5: 9

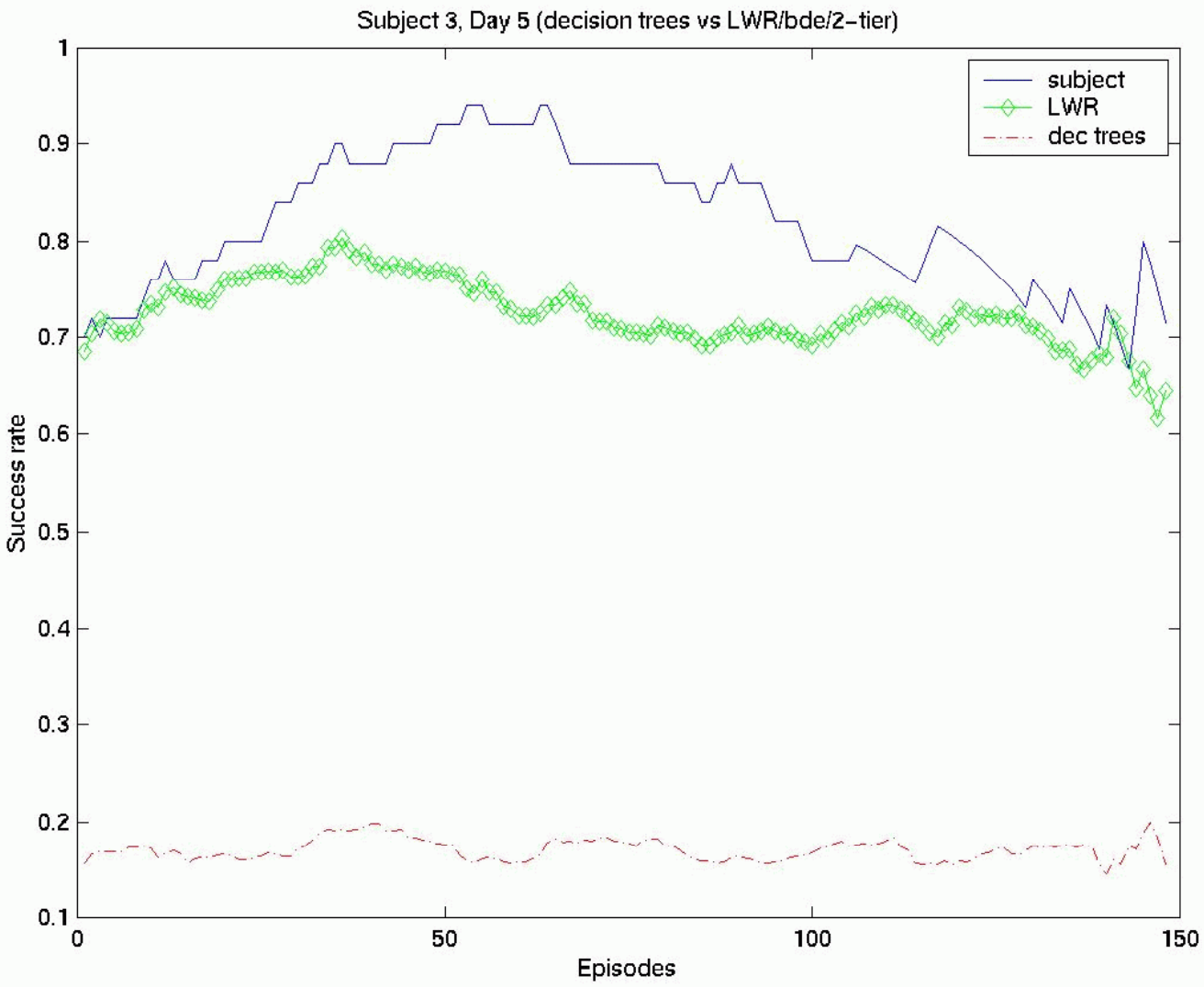


Subject

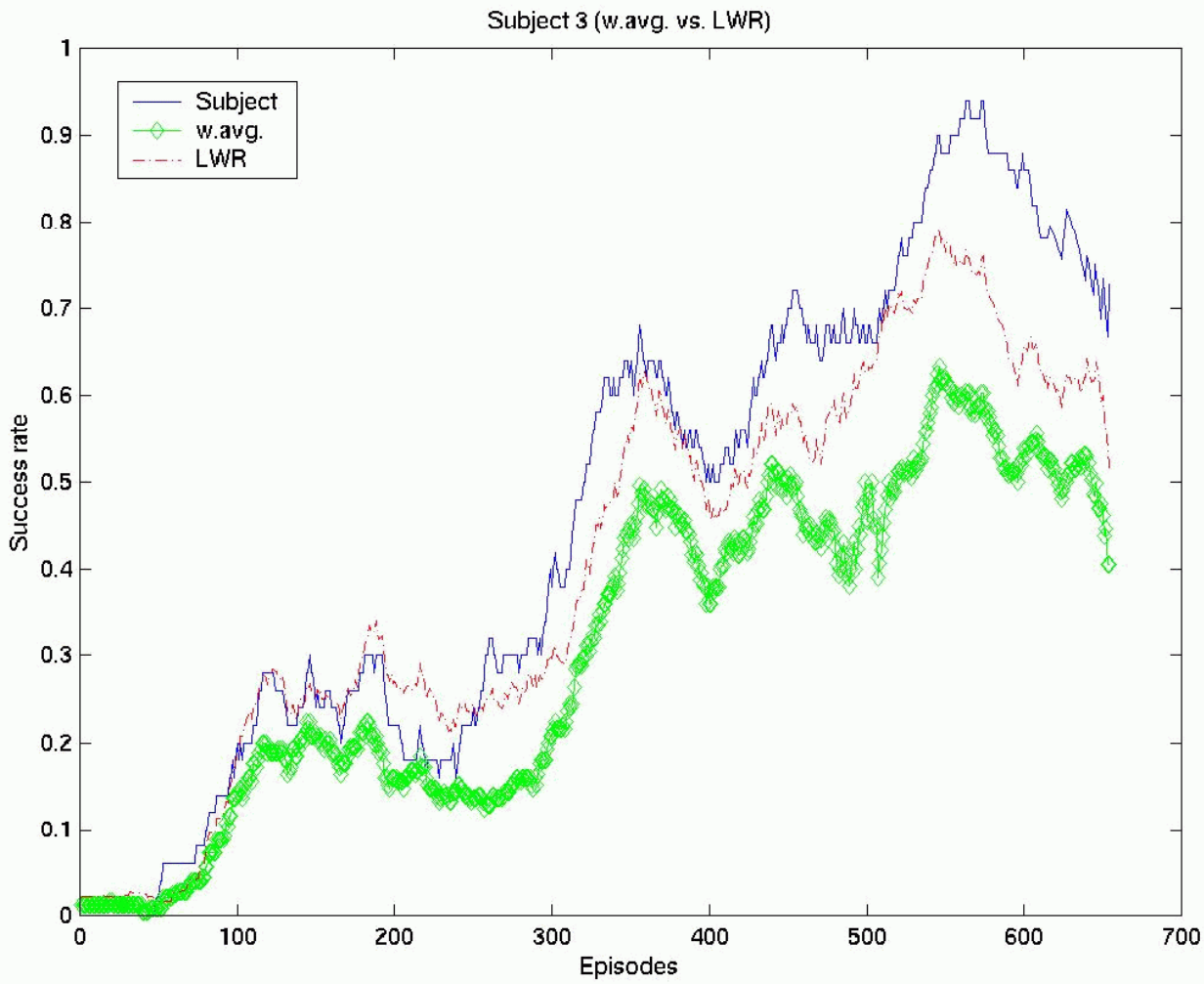


Model

Comparison with global methods



Tracking human learning



Summary of Scenario 5

- We can model subjects on the NRL task in real-time, achieving excellent fits to their learning curves, using the available visual-motor data stream.
- One of the first in cognitive science to directly use objective visual-motor performance data to derive high-level strategy models on a complex task.
- New algorithms for detecting change-points and building predictive stochastic models for massive, noisy, non-stationary, vector time series data.

Overall Summary

- Standard software system design methods fail in many applications.
- Statistical machine learning methods actively gather performance data to make robust, adaptive systems.

Statistical machine learning

- Machine learning is already at the heart of speech recognition and handwriting recognition.
- Statistical machine learning is transforming natural language, bioinformatics, data mining and mobile robotics.
- Statistical machine learning is creating opportunities in compilers, networking, and computer security.
- Statistical machine learning is opening up interactions with political science, biology and cognitive science.

Role of SML in S+E

- Data is a new source of power for scientists and engineers.
- Statistical machine learning should be taught to our students.
- By combining engineered frameworks with models learned from data, we can develop high-performance, robust, adaptive, systems of the future.

Acknowledgements

- NASA NRA (with D. Kortenkamp and P. Bonasso, NASA JSC)
- Conflict analysis: NSF ITR **0219673** (with Ric Stoll)
- Adaptive compilers: NSF ITR **0205303** (with Keith Cooper and Linda Torczon)
- Human learning: ONR N00014-96-1-0538
- Virgil: School of Engineering, Rice University

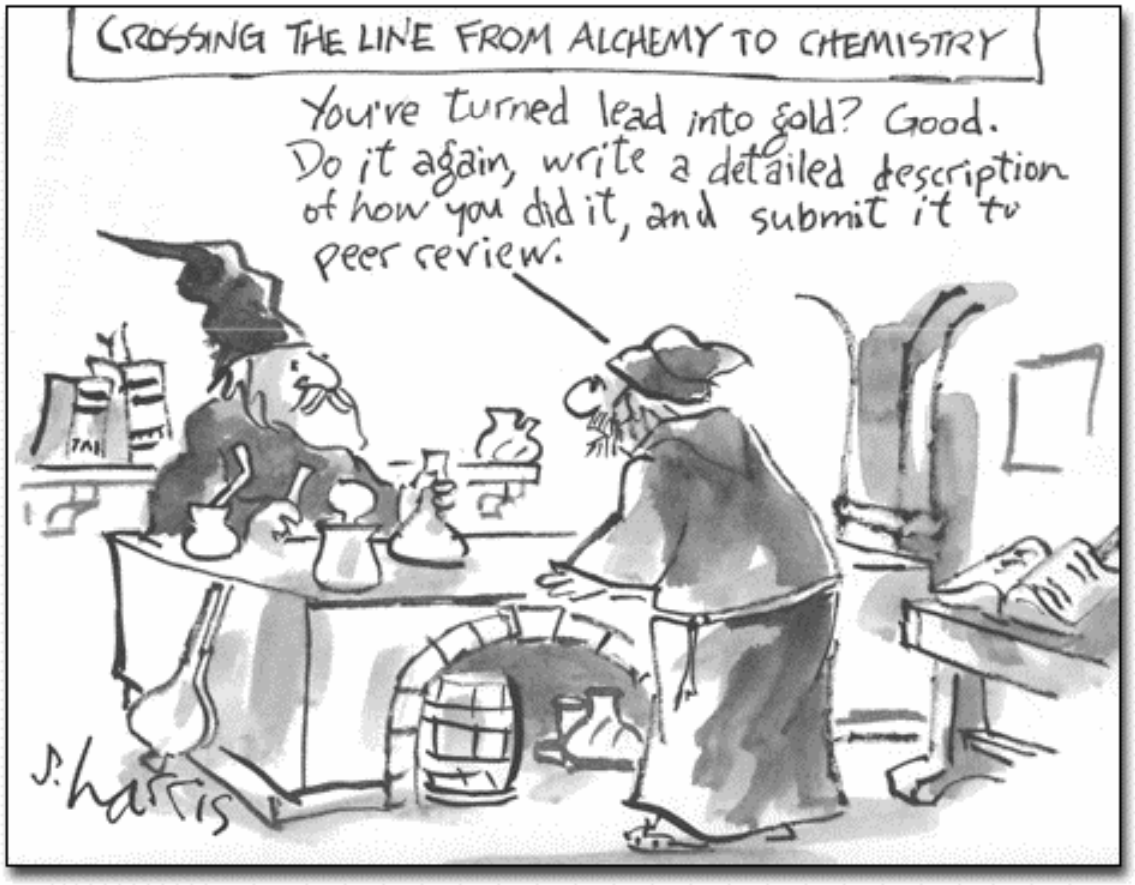
The future

- Mr. Gates scoffed at the notion, advanced by some, that the computer industry was a mature business of waning opportunity. In one question-and-answer session, a student asked if there could ever be another technology company as successful as Microsoft.
- "If you invent a breakthrough in artificial intelligence, so machines can learn, that is worth 10 Microsofts."

  --- B. Gates,

  NY Times March 1, 2004

Scientific drivers



Science Magazine