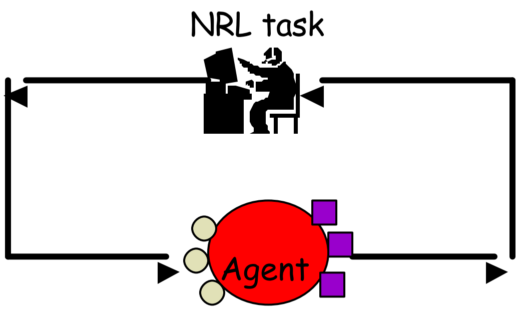
Tracking the evolution of learning on a complex visualmotor task

Devika Subramanian
Rice University

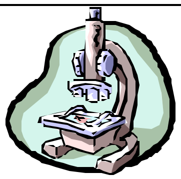
Supported by the Office of Naval Research

The context: training submarine pilots



Track the evolution of a human learning a visualmotor task with a significant strategic component, and alter training protocol to improve the speed and efficacy of that learning.

Goal of project

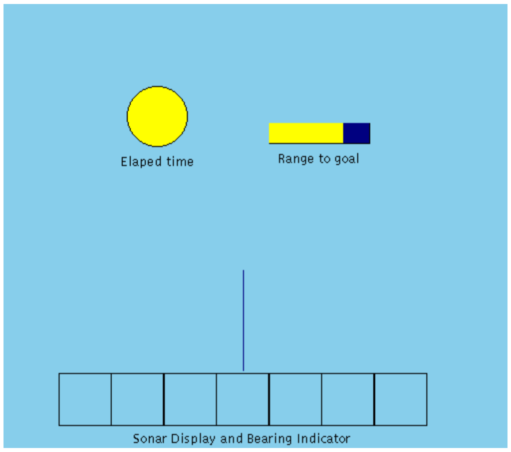


- Construct computational models of human learning based on performance data gathered during task learning.
  - Models will be used to diagnose problems in learning and aid in the design of training protocols that help humans achieve high levels of competence on the task.
- A computational microscope for training: can we infer cognitive constructs from objective performance data?.

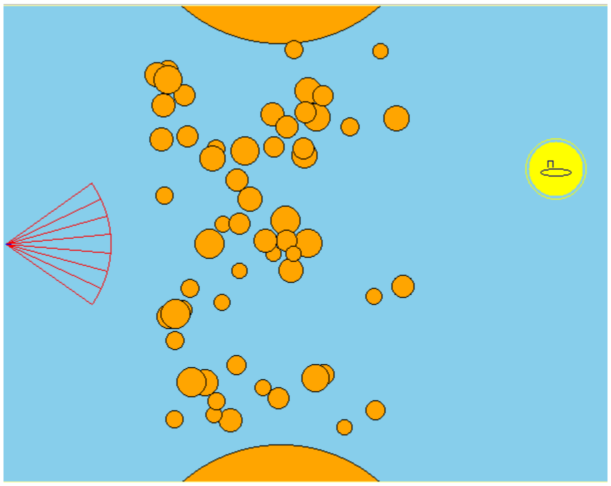
Outline

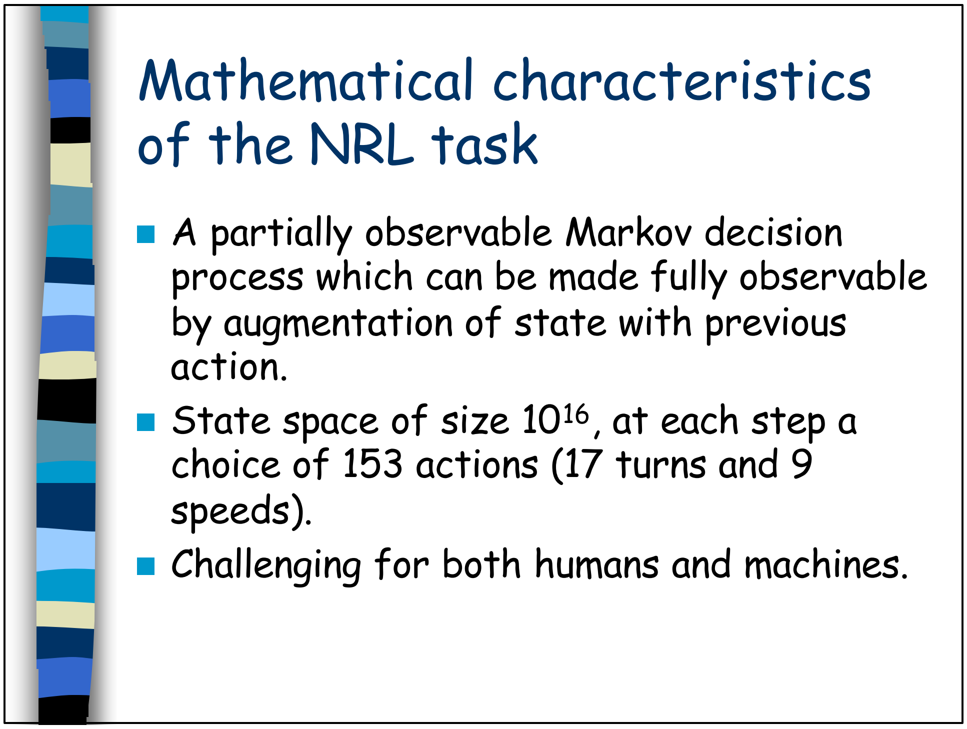
- The NRL Navigation Task
- Challenges in modeling human learning
- Understanding the task: optimal player
- A hybrid model for human learning
- Understanding the task: reinforcement learner
- High-fidelity models for human learning

The NRL Navigation Task



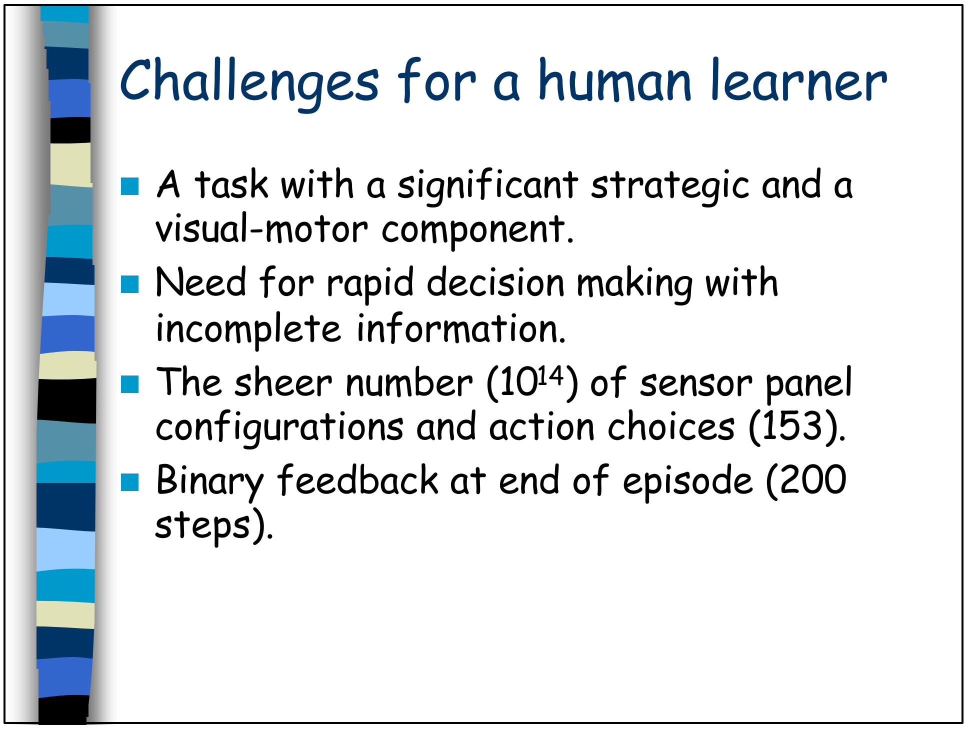
The NRL Navigation Task





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^{16}, at each step a choice of 153 actions (17 turns and 9 speeds).
- Challenging for both humans and machines.



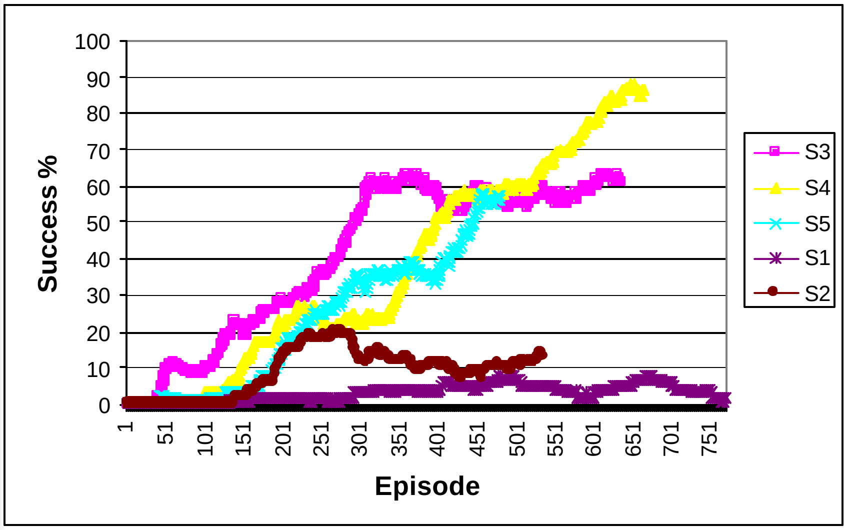
Challenges for a human learner

- 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).

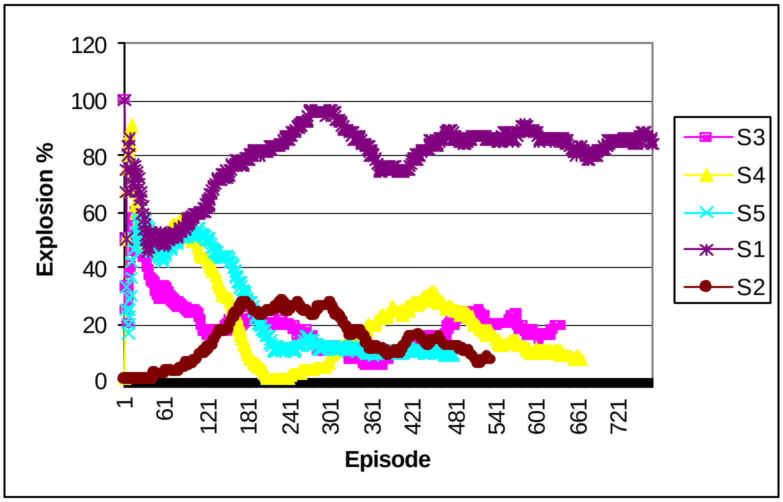
Experiments on human subjects

- Conducted at San Diego with ASL eyetracker.
- 5 subjects, five one-hour sessions each.
- 60 mines, small mine drift, small sensor noise.
- Collected visualmotor data, verbal protocols and eyetracker data.

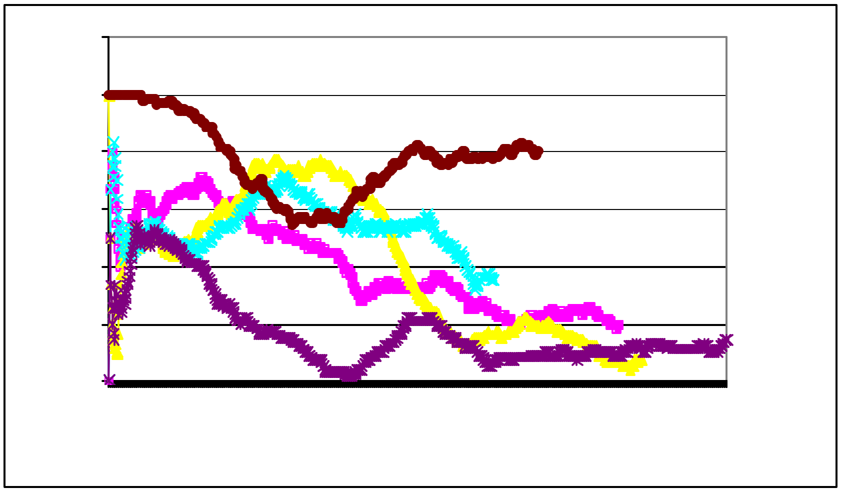
Learning curves (success)

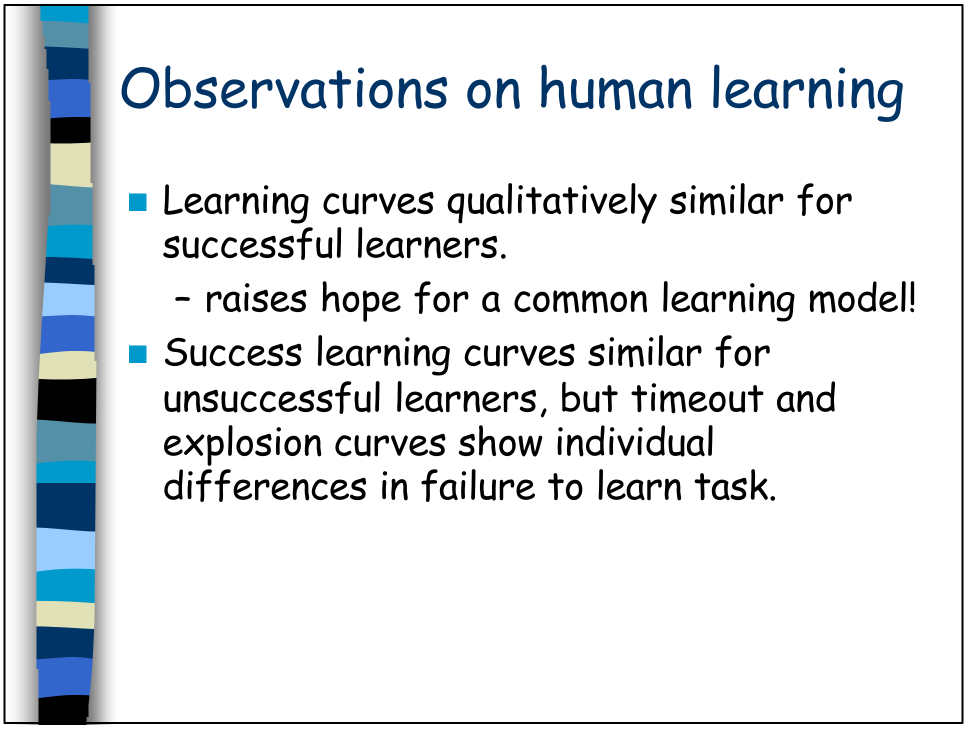


Learning curves (explosions)



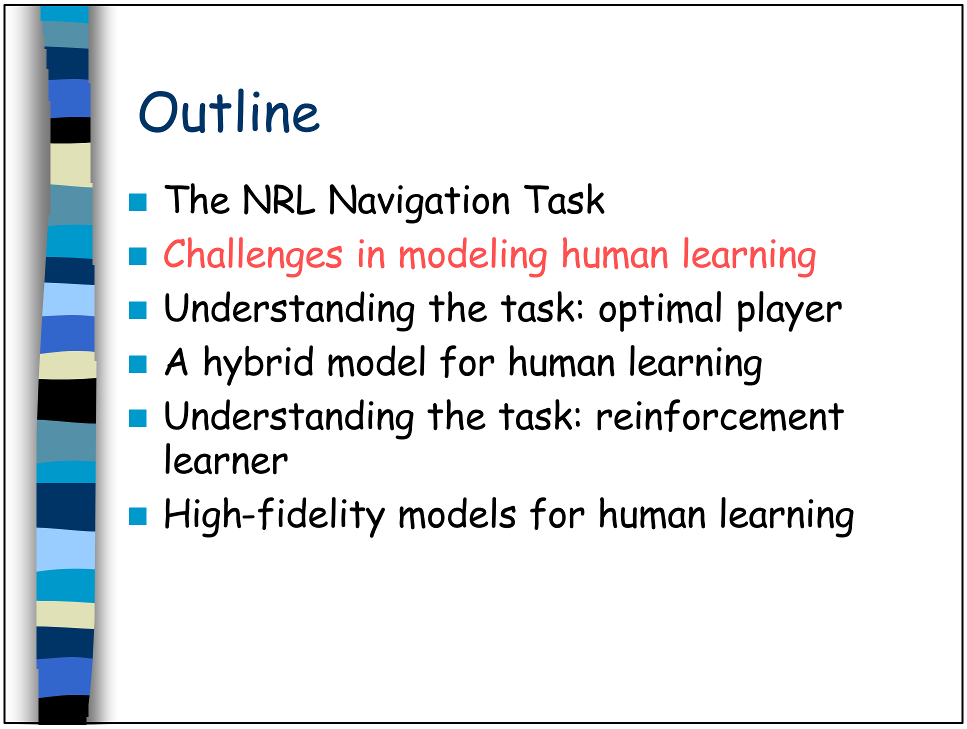
Learning curves (timeouts)





Observations on human learning

- Learning curves qualitatively similar for successful learners.
  - raises hope for a common learning model!
- Success learning curves similar for unsuccessful learners, but timeout and explosion curves show individual differences in failure to learn task.

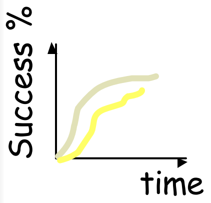


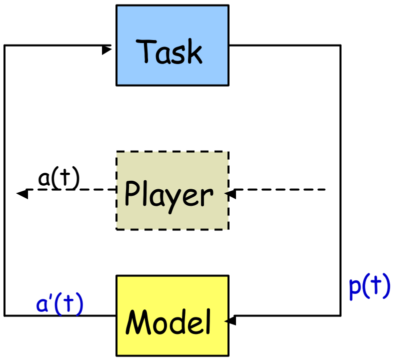
Outline

- The NRL Navigation Task
- Challenges in modeling human learning
- Understanding the task: optimal player
- A hybrid model for human learning
- Understanding the task: reinforcement learner
- High-fidelity models for human learning

Building Representative Models

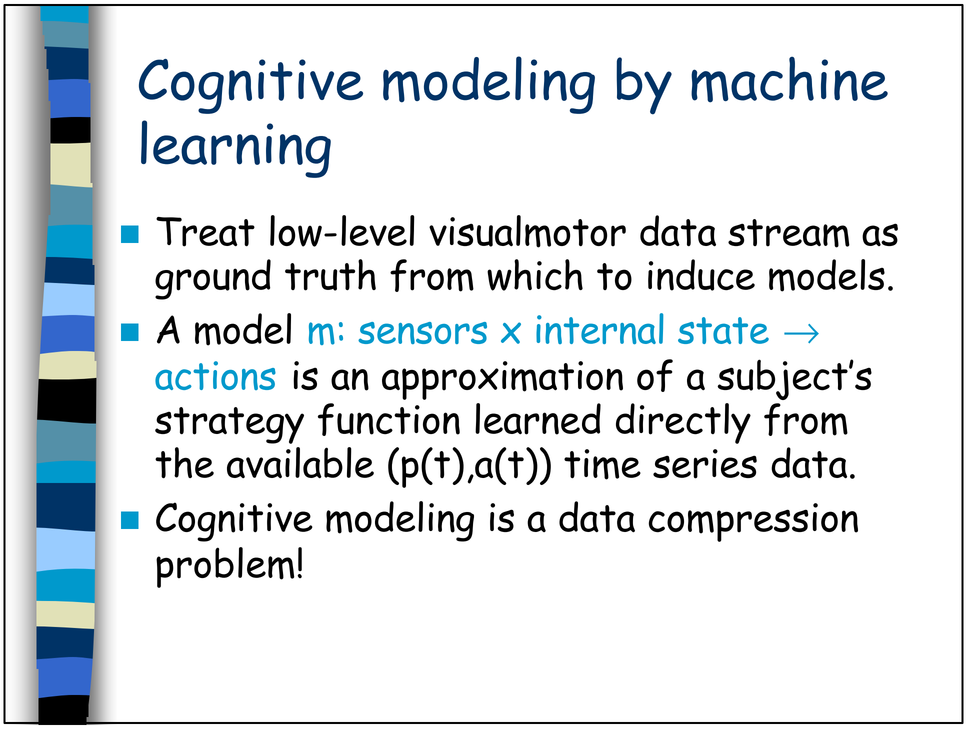
- Behavioral equivalence (similarity in learning curves)





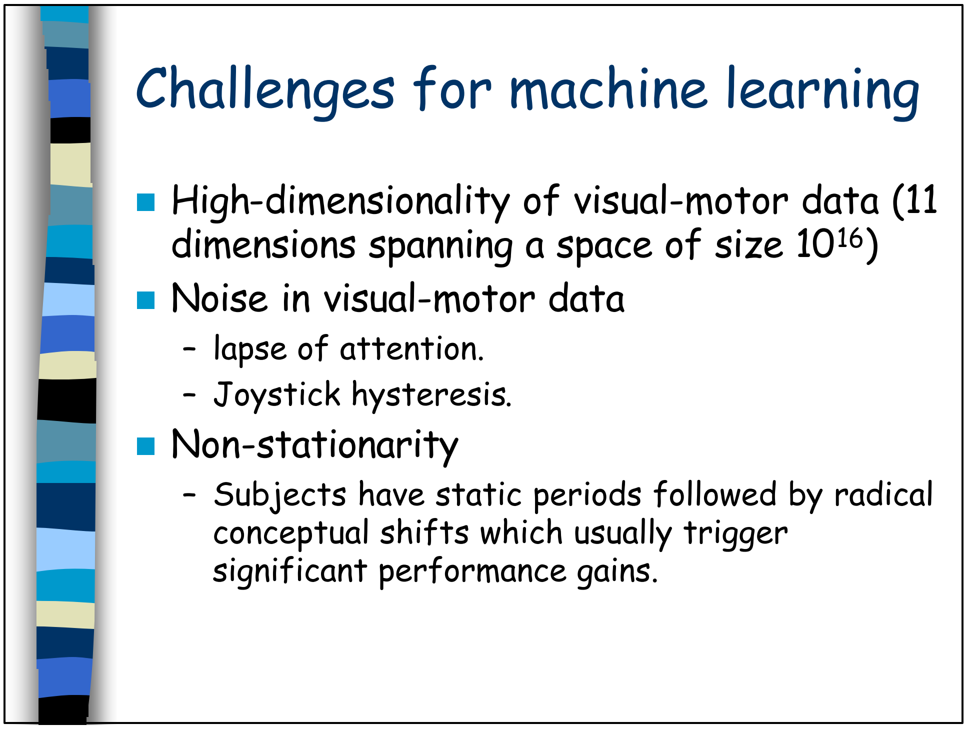
Challenges in modeling learning

- It is not possible to gather objective data from subjects about their strategy; game doesn't allow useful verbalizations during play, and post-play explanations are often inaccurate and incomplete.



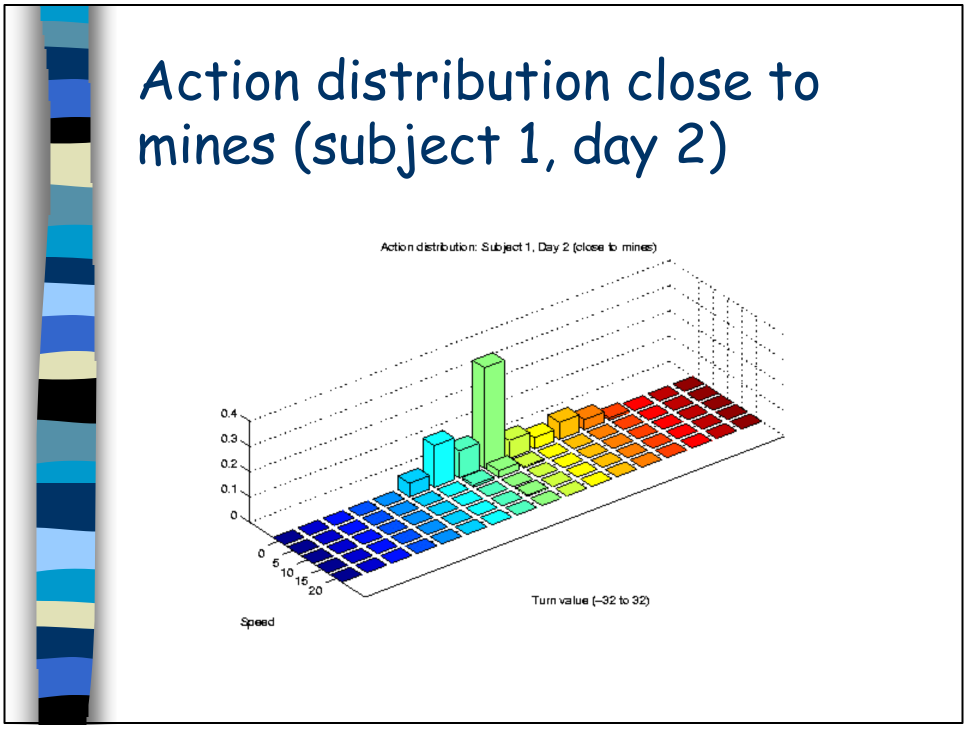
Cognitive modeling by machine learning

- Treat low-level visualmotor data stream as ground truth from which to induce models.
- A model m: sensors x internal state → actions is an approximation of a subject's strategy function learned directly from the available (p(t),a(t)) time series data.
- Cognitive modeling is a data compression problem!

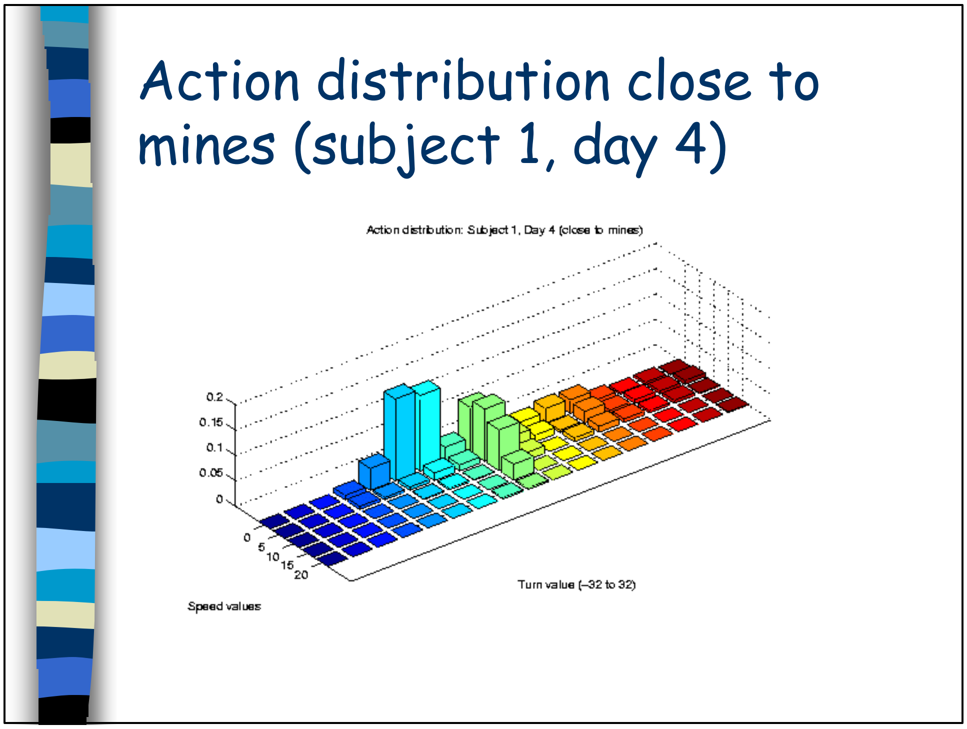


Challenges for machine learning

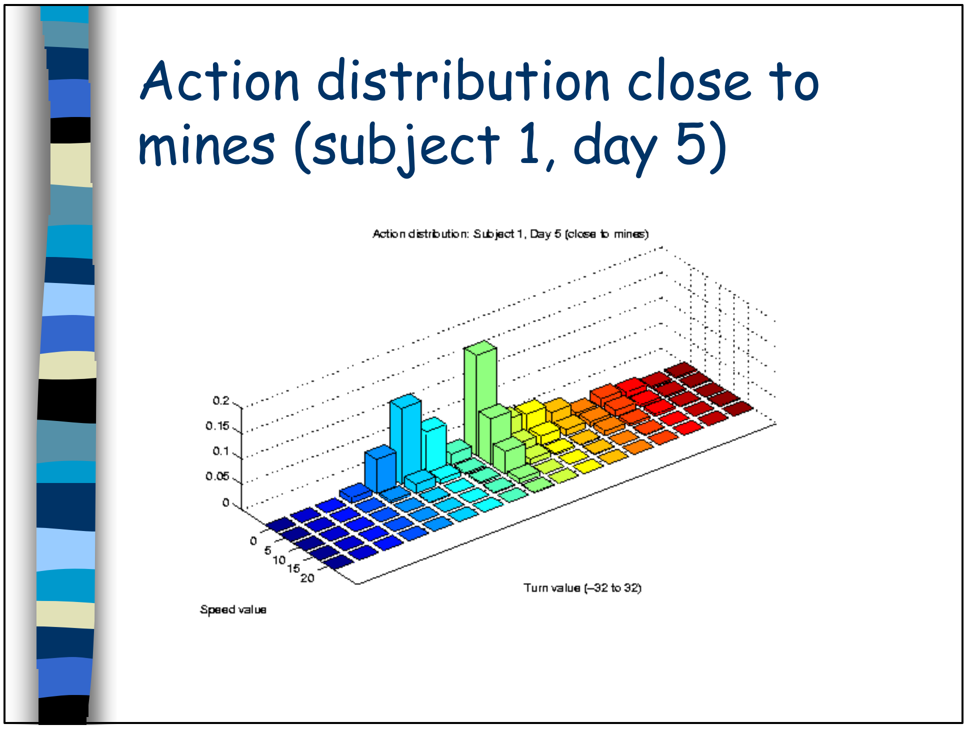
- High-dimensionality of visual-motor data (11 dimensions spanning a space of size 10^{16})
- Noise in visual-motor data
  - lapse of attention.
  - Joystick hysteresis.
- Non-stationarity
  - Subjects have static periods followed by radical conceptual shifts which usually trigger significant performance gains.



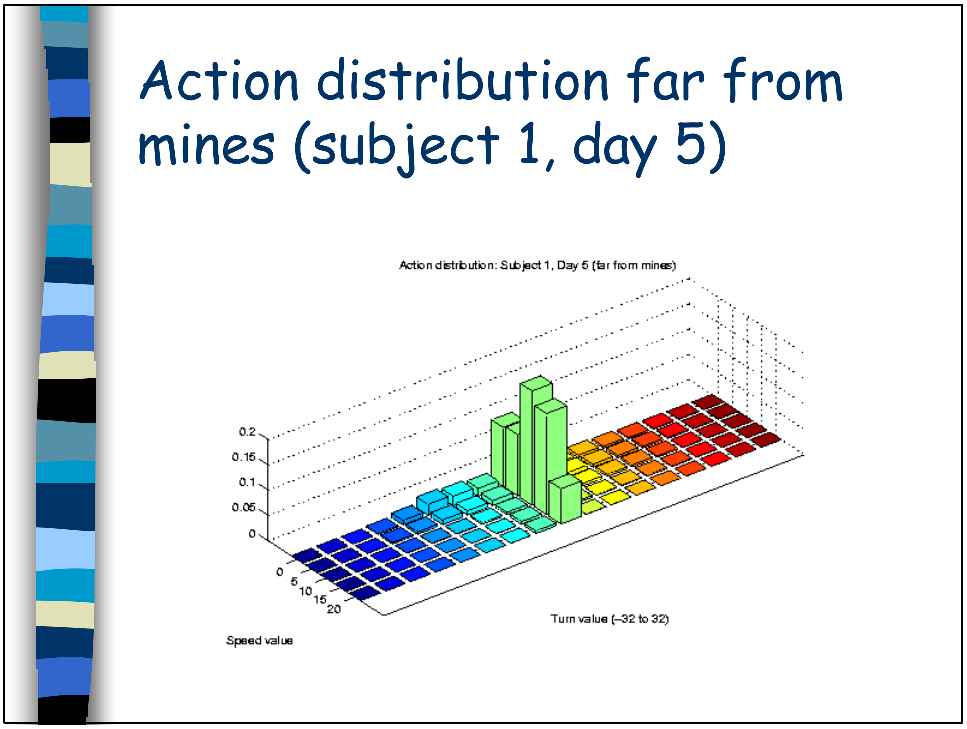
Action distribution close to mines (subject 1, day 2)



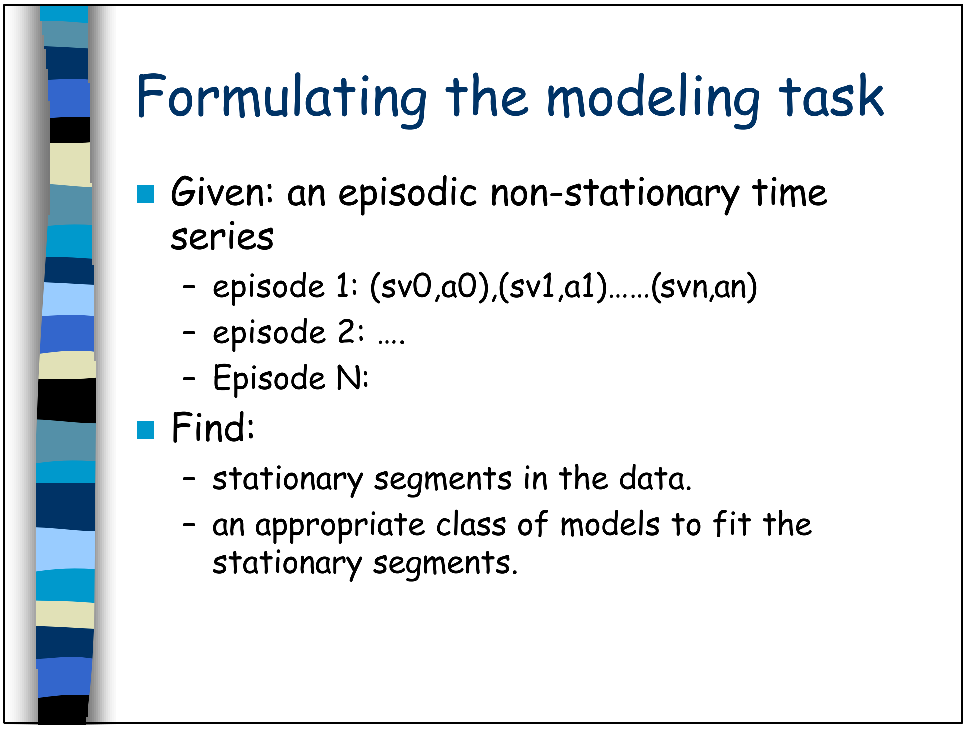
Action distribution close to mines (subject 1, day 4)



Action distribution close to mines (subject 1, day 5)

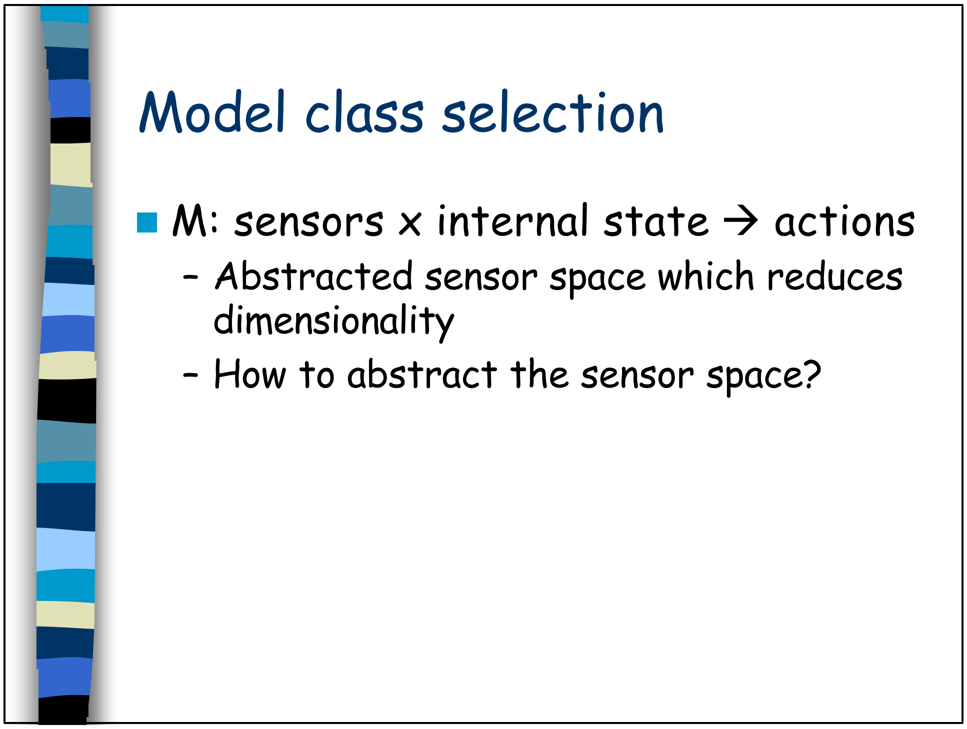


Action distribution far from mines (subject 1, day 5)



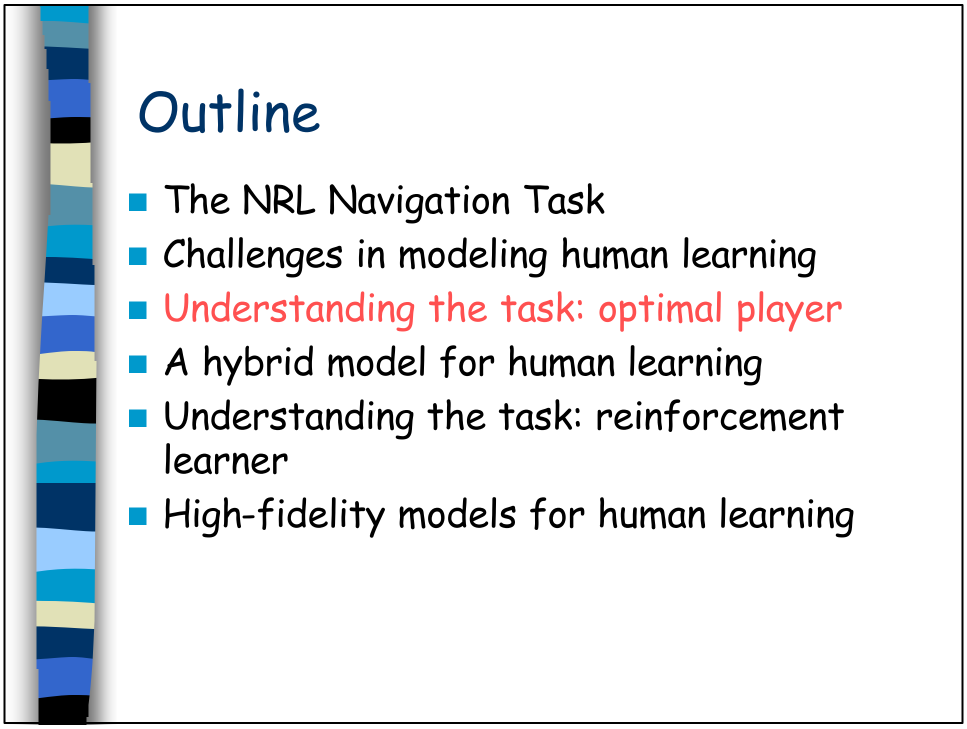
Formulating the modeling task

- Given: an episodic non-stationary time series
  - episode 1: (sv0,a0),(sv1,a1)……(svn,an)
  - episode 2: ….
  - Episode N:
- Find:
  - stationary segments in the data.
  - an appropriate class of models to fit the stationary segments.



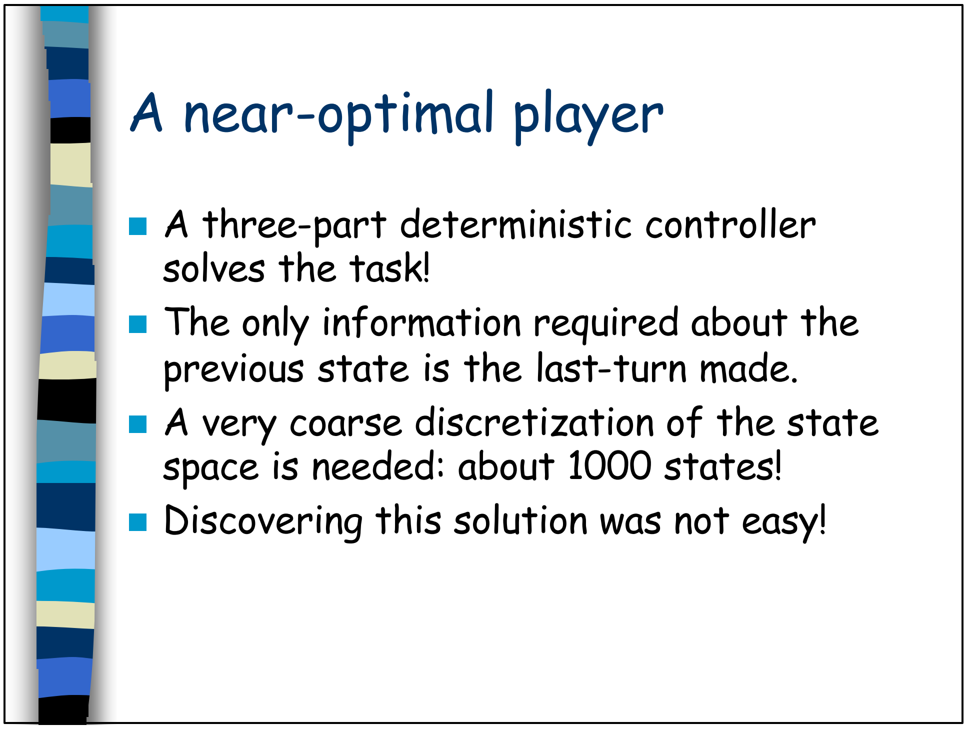
Model class selection

- M: sensors x internal state → actions
  - Abstracted sensor space which reduces dimensionality
  - How to abstract the sensor space?



Outline

- The NRL Navigation Task
- Challenges in modeling human learning
- Understanding the task: optimal player
- A hybrid model for human learning
- Understanding the task: reinforcement learner
- High-fidelity models for human learning

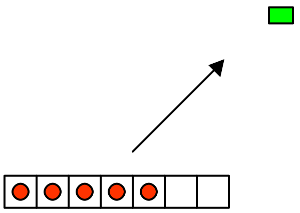


A near-optimal player

- A three-part deterministic controller solves the task!
- The only information required about the previous state is the last-turn made.
- A very coarse discretization of the state space is needed: about 1000 states!
- Discovering this solution was not easy!

Part 1: Seek Goal

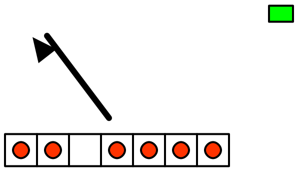
There is a clear sonar in the direction of the goal.



If the sonar in the direction of the goal is clear, follow it at speed of 20, unless goal is straight ahead, then travel at speed 40.

Part 2: Avoid Mine

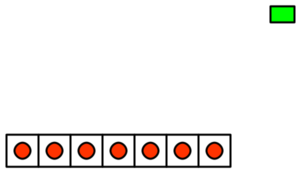
There is a clear sonar but not in the direction of the goa



Turn at zero speed to orient with the first clear sonar counted from the middle outward. If middle sonar is clear, move forward with speed 20.

Part 3: Gap Finder

There are no clear sonars.



If the last turn was non-zero, turn again by the same amount, else initiate a soft turn by summing the right and left sonars and turning in the direction of the lower sum.

Performance of optimal player

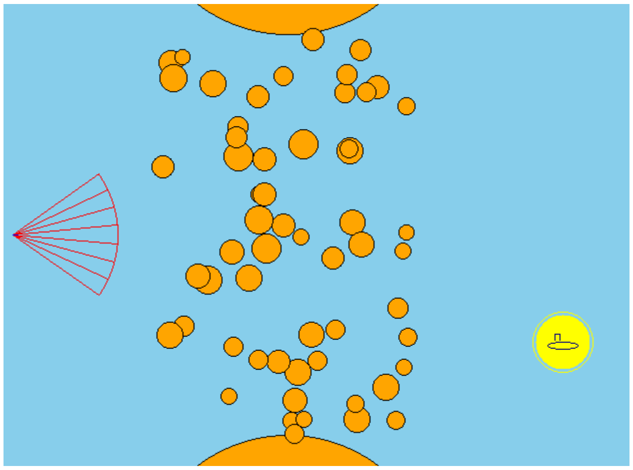
Player	Success %	Behavior
Opt. player	99.7%	Baseline
Opt. player – Part 3	79.9%	Oscillates/times out
Opt. player – Part 2	98.3%	Times out
Opt. player – Part 1	7.3%	Never gets to goal/times out
Part 1	50.1%	Aggressive goal seeker/blows up

Mine density = 60
All results reported for 10,000 episodes

Properties of optimal player

- Reflects task decomposition found in human players.
  - However, sub-goals are very coupled and this coupling is what is hard for humans to learn.
- Key to success:
  - state space partitioning; threshold of 50 on sonar value makes right compromise between succeeding, timing out and blowing up.
  - turning at zero speeds.
  - turning consistently in a given direction to find gap in mines.

Where optimal player loses



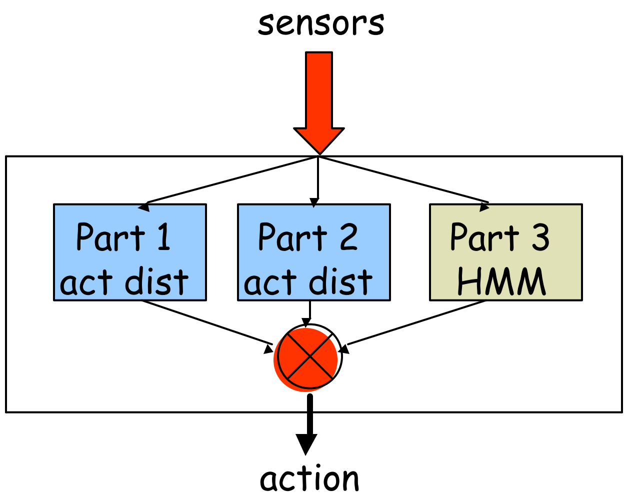
Outline

- The NRL Navigation Task
- Challenges in modeling human learning
- Understanding the task: optimal player
- A hybrid model for human learning
- Understanding the task: reinforcement learner
- High-fidelity models for human learning

A modeling approach

- Abstraction of sensor space
  - view sensors through prism of equivalence classes defined by a near-optimal policy for task.
  - Model subject's policy as stochastic map from abstracted sensor space to actions.
- Advantage
  - deviations from optimal can be the basis for directed training of subjects.
- Disadvantage
  - humans may not adopt anything close to the conceptualization needed for optimal play.

A hybrid model



A very small number of parameters is sufficient to capture subject. Can acquire subject model online!

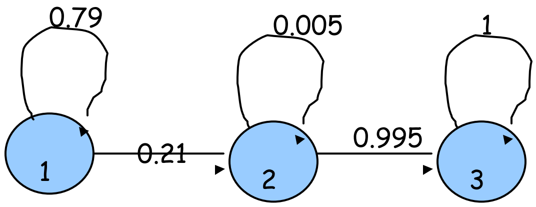
Model extraction algorithm

- To find stationary subsequences, segment the data using KL divergence measure on all distributions
  - chunk data into uniform segments.
  - for each segment, compute Part 1, 2 and 3 distributions
  - for each Part, compute KL divergence, and identify change points as those segment boundaries where measure changes significantly.
- Given a stationary subsequence of visual-motor data
  - learn Part 1 and Part 2 conditional action distributions from data (a counting process)
  - obtain action sequences in Part 3 and learn HMM.

Evolution of gap finding strategy

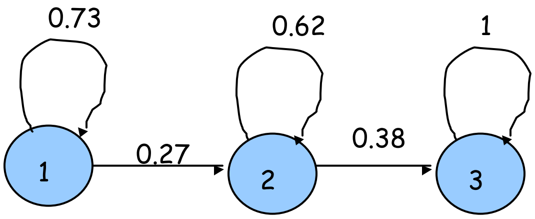
- Subject Col: episodes 45-67 and episodes 68-90 on day 2.
- Subject learns to turn in place.
- HMM models for gap finding.

Pre-shift gap finding strategy



0	0	0.84	0.00	0.36
right	0	0.03	0.003	0.33
left	0	0.05	0.00	0.31
other		0.08	0.997	0.00

Post-shift gap finding strategy



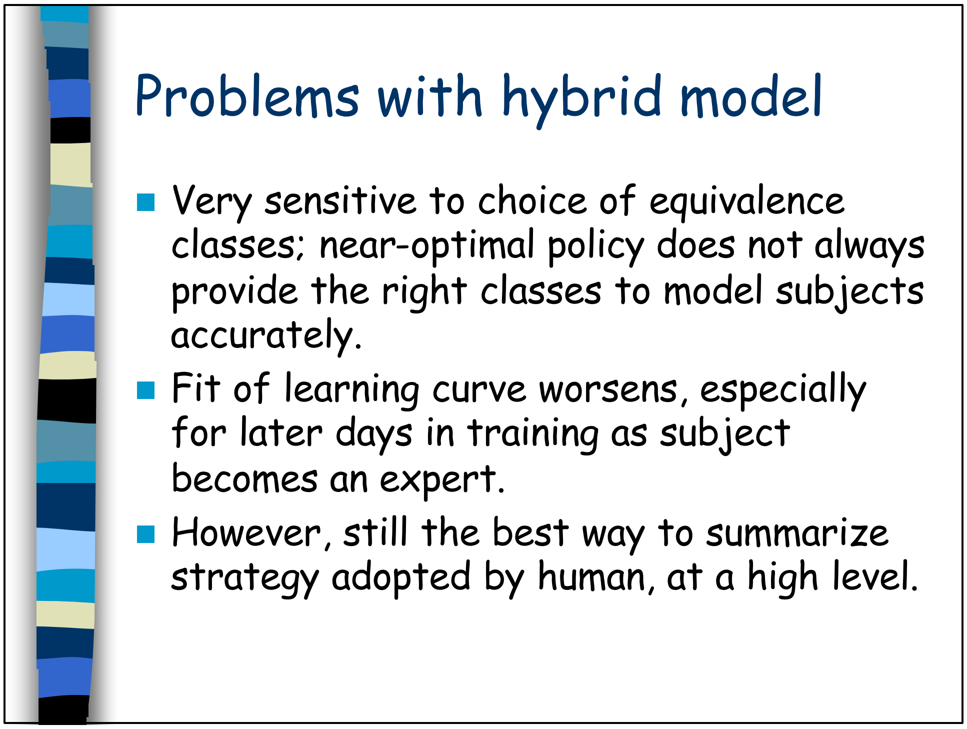
0	0	0.82	0.05	0.37
right	0	0.024	0.00	0.553
left	0	0.025	0.95	0.07
other		0.131	0.00	0.00

Results

Pre-shift	Successes	Explosions	Timeouts	Total episodes
Col	0	12	11	23
Model	0	17	6	23

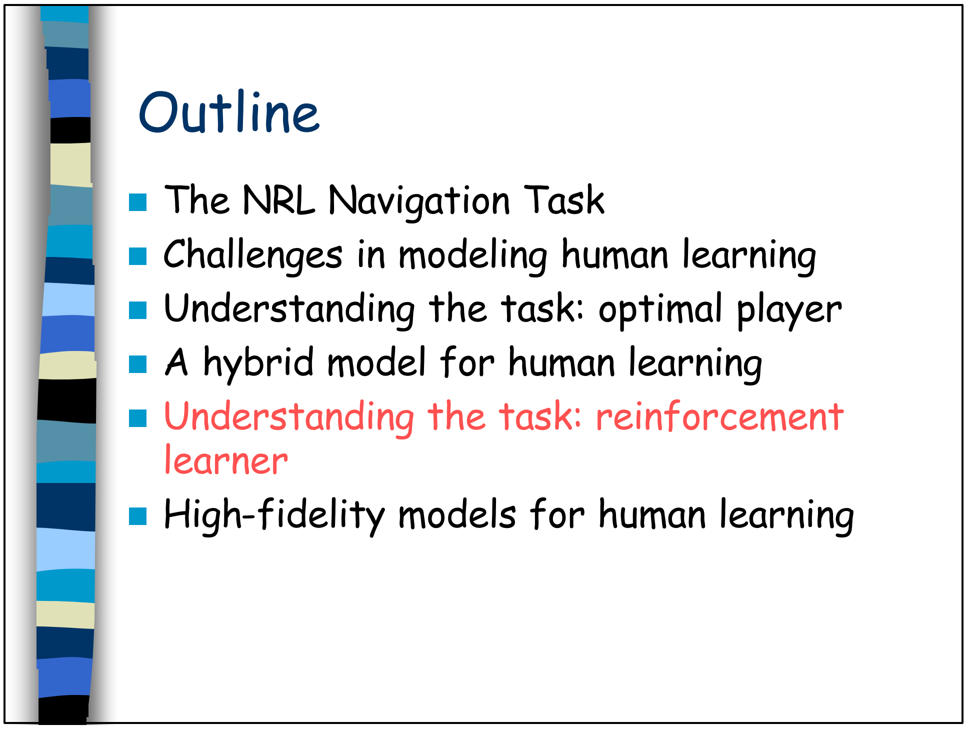
Post-shift	Successes	Explosions	Timeouts	Total episodes
Col	0	2	13	15
Model	0	4	11	15

A better fit than using C4.5.



Problems with hybrid model

- Very sensitive to choice of equivalence classes; near-optimal policy does not always provide the right classes to model subjects accurately.
- Fit of learning curve worsens, especially for later days in training as subject becomes an expert.
- However, still the best way to summarize strategy adopted by human, at a high level.



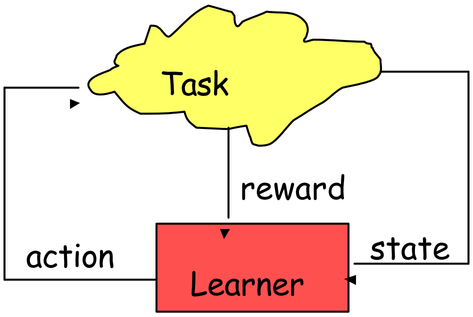
Outline

- The NRL Navigation Task
- Challenges in modeling human learning
- Understanding the task: optimal player
- A hybrid model for human learning
- Understanding the task: reinforcement learner
- High-fidelity models for human learning

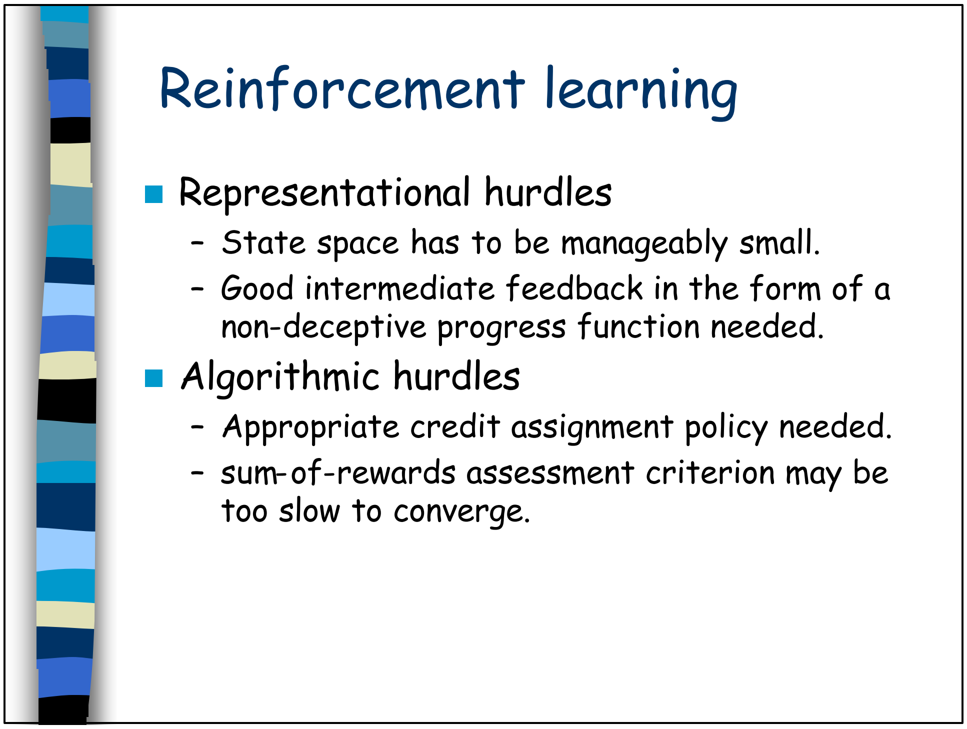
Machine learning of NRL task

- What does it take to get machines to learn task?
- Can machine learners achieve higher levels of competence?
- How does the sample complexity of machine learning compare with humans?
- Can we use machine learning to improve human learning?

Reinforcement learning

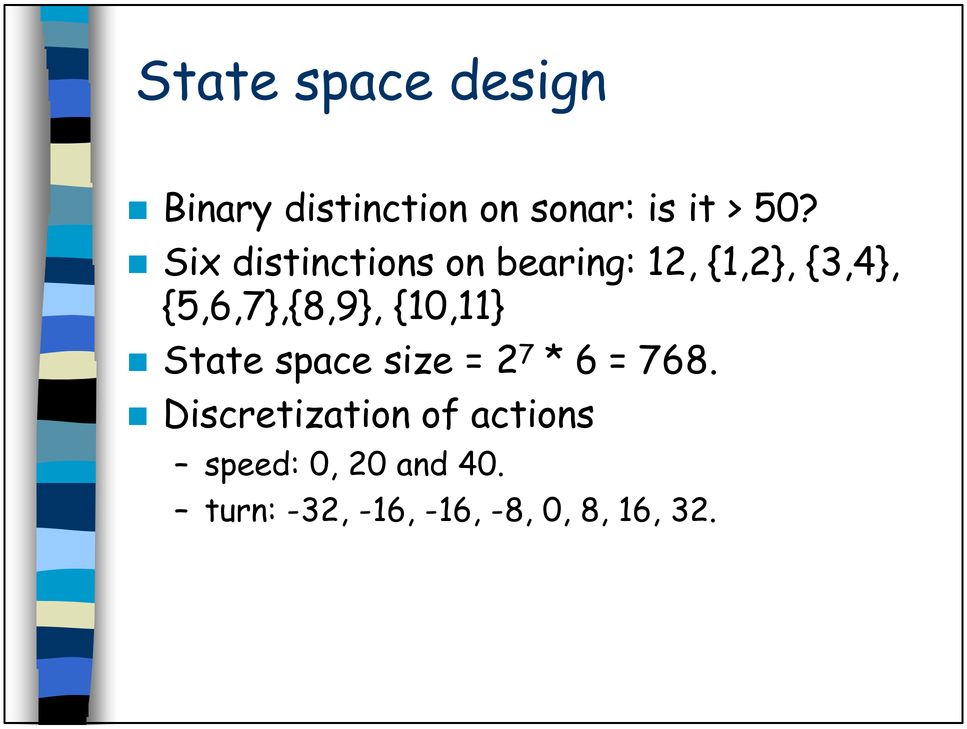


$s_1,a_1,r_1,s_2,a_2,r_2,$……….



Reinforcement learning

- Representational hurdles
  - State space has to be manageably small.
  - Good intermediate feedback in the form of a non-deceptive progress function needed.
- Algorithmic hurdles
  - Appropriate credit assignment policy needed.
  - sum-of-rewards assessment criterion may be too slow to converge.



State space design

- Binary distinction on sonar: is it > 50?
- Six distinctions on bearing: 12, {1,2}, {3,4}, {5,6,7},{8,9}, {10,11}
- State space size = 2^7 * 6 = 768.
- Discretization of actions
  - speed: 0, 20 and 40.
  - turn: -32, -16, -16, -8, 0, 8, 16, 32.

The progress function

r(s,a,s') = 0 if s' is a state where player hits mine.
= 1 if s' is a goal state
= 0.5 if s' is a timeout state

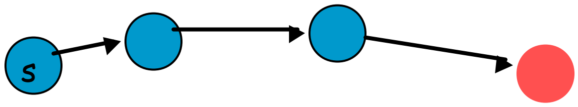
= 0.75 if s is a Part3 state and s' is a Part1 or Part2 state
= 0.5 + Dsum of sonars/1000 if s' is a Part3 state
= 0.5 + Drange/1000 + abs(bearing - 6)/40 otherwise

Credit assignment policy

- Penalize the last action alone in a sequence which ends in an explosion.
- Penalize all actions in sequence which ends in a timeout.

Simplification of value estimation

- Estimate the average local reward for each action in each state.
  - A big change from learning sum-of-rewards from each state.

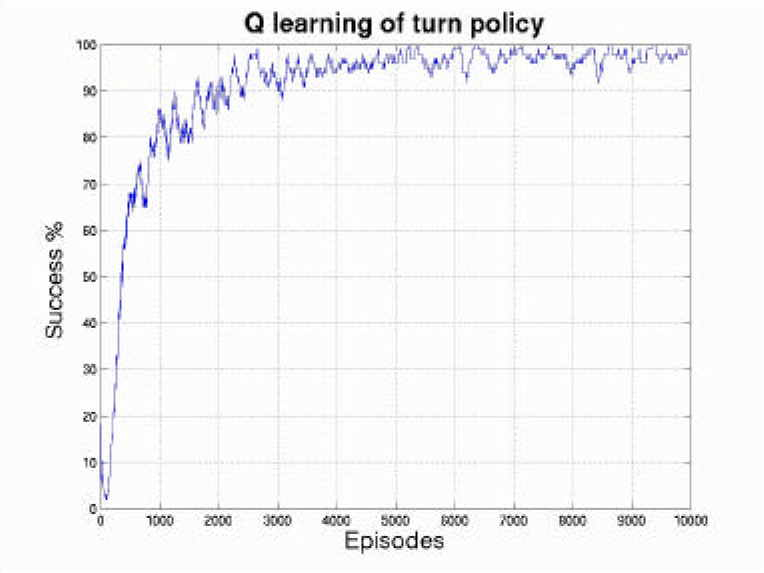


Q(s,a) is the sum of rewards from s to terminal state. Here we only maintain local reward at state s.

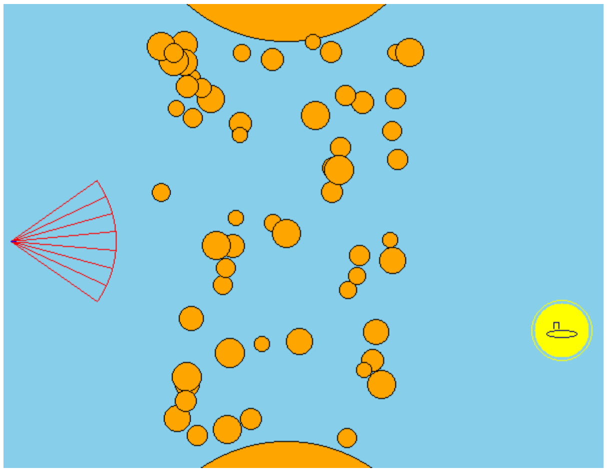
Staged learning

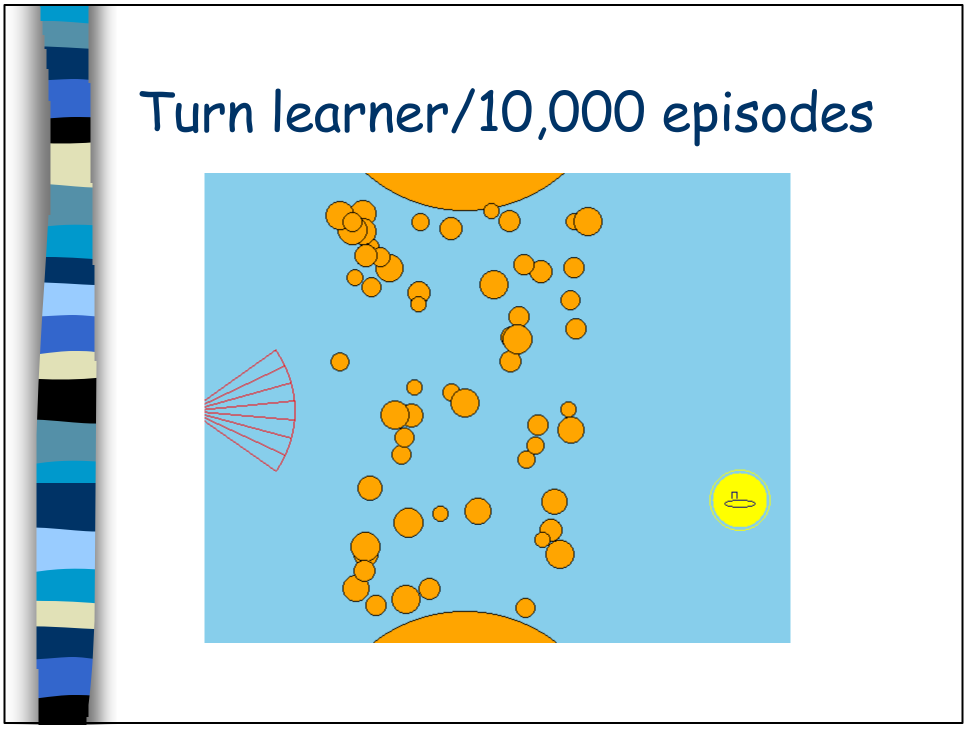
- First learn turns alone, with speed supplied by near-optimal player.
- Next learn both turn and speed.
- Differences in two learners suggest new protocols for training humans.

Results of learning turns

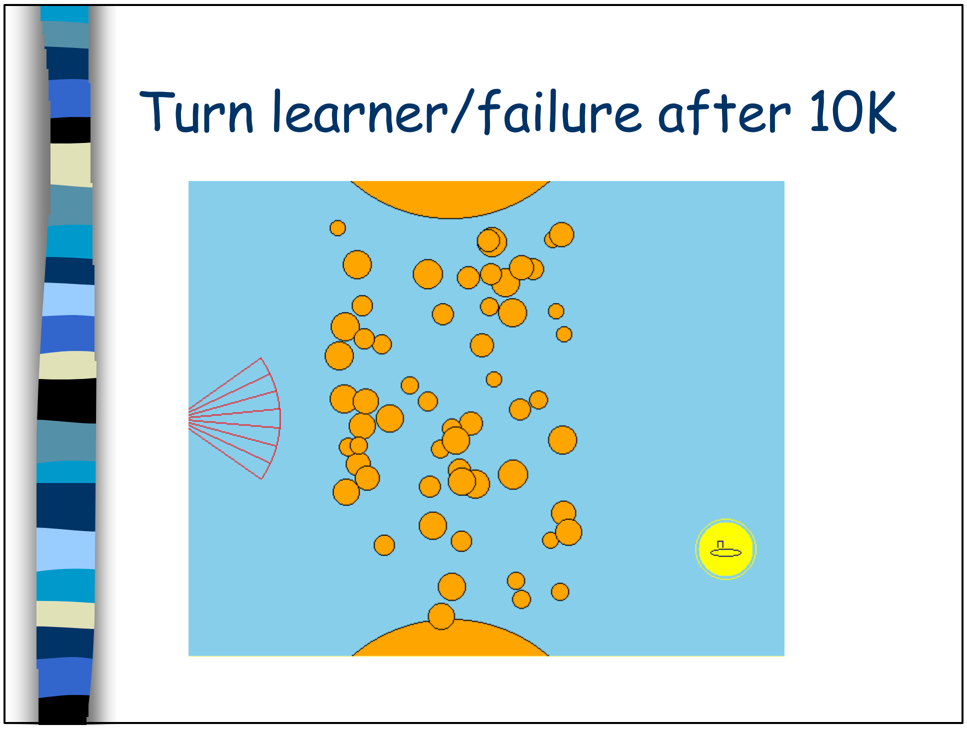


Turn learner/600 episodes





Turn learner/10,000 episodes

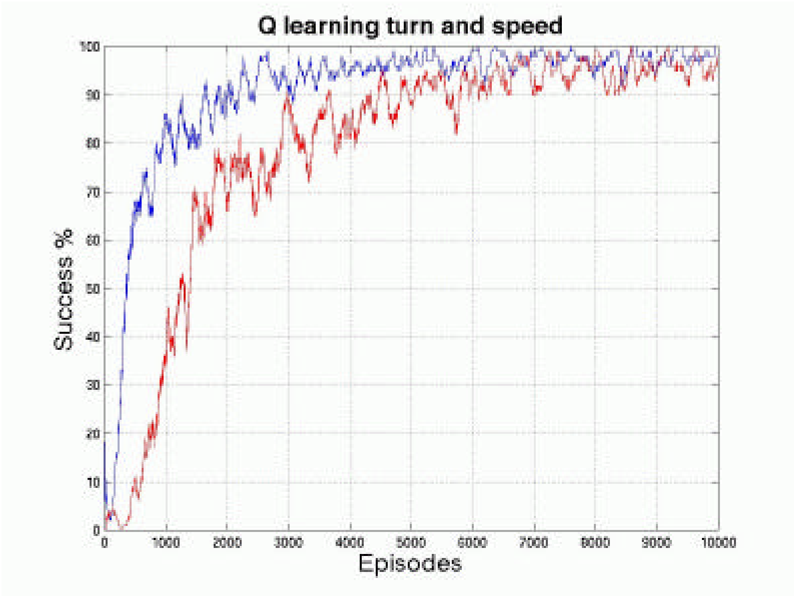


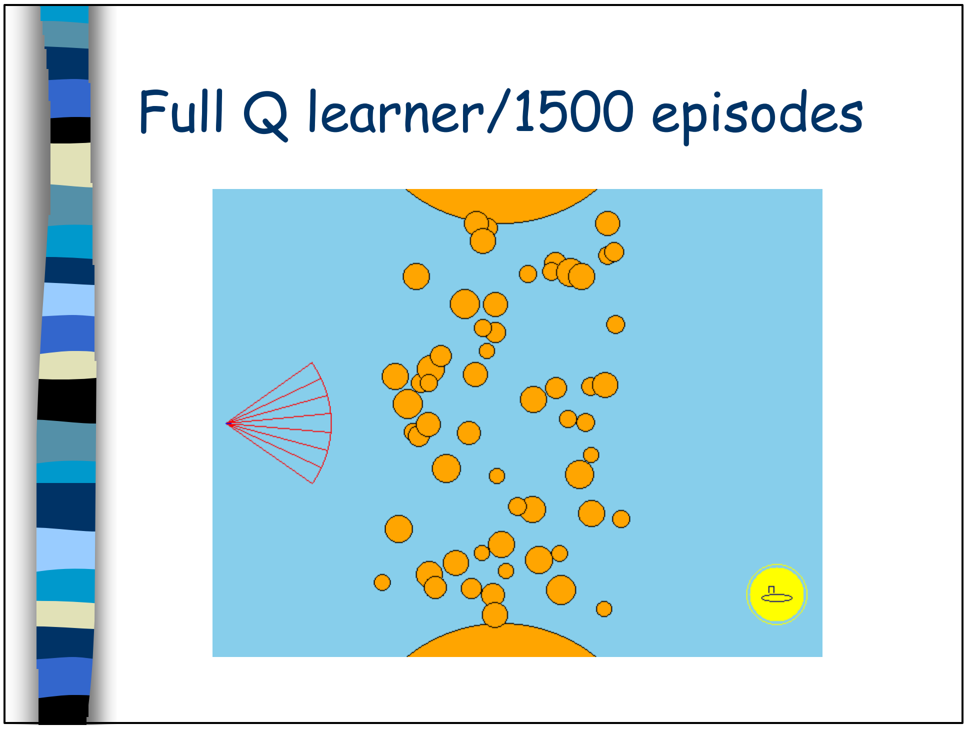
Turn learner/failure after 10K

Learning of complete policy

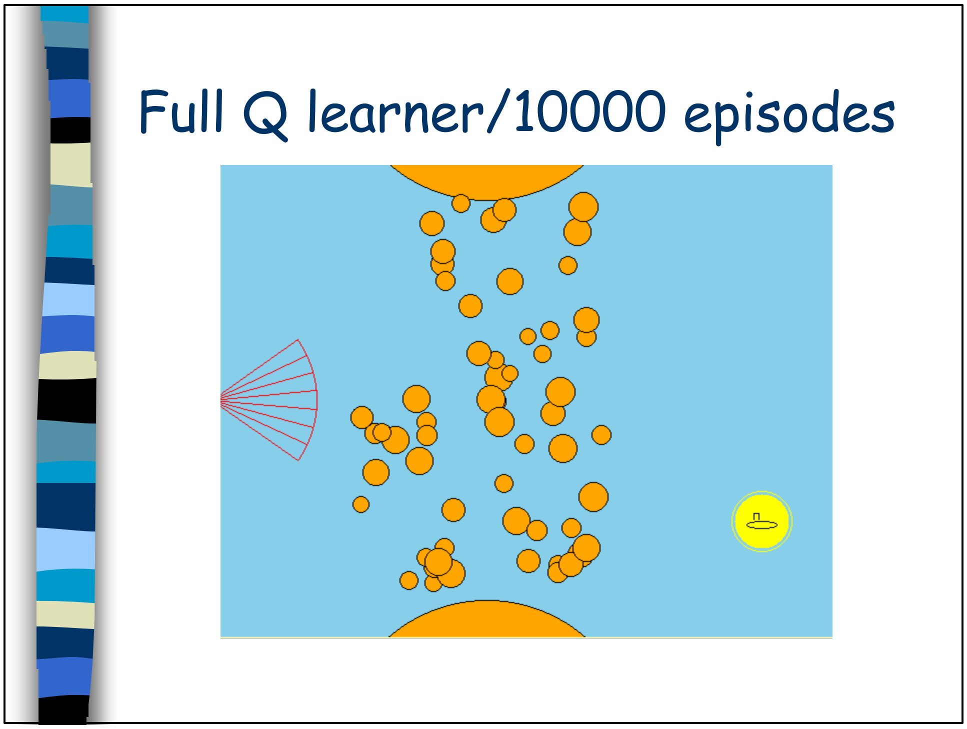
- Estimate of average local reward not a perfect substitute for global sum-of-rewards.
- Make action choice based on estimated local reward weighted by the global measure of wins/(wins+timeouts) from that state.
- Optimistic initialization of q values.

Results of learning complete policy



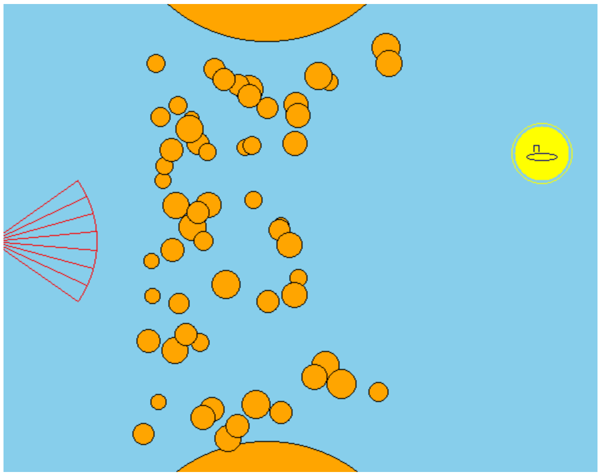


Full Q learner/1500 episodes

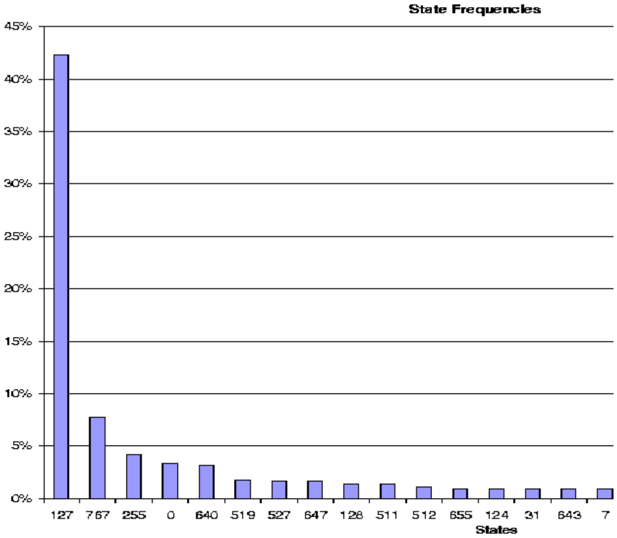


Full Q learner/10000 episodes

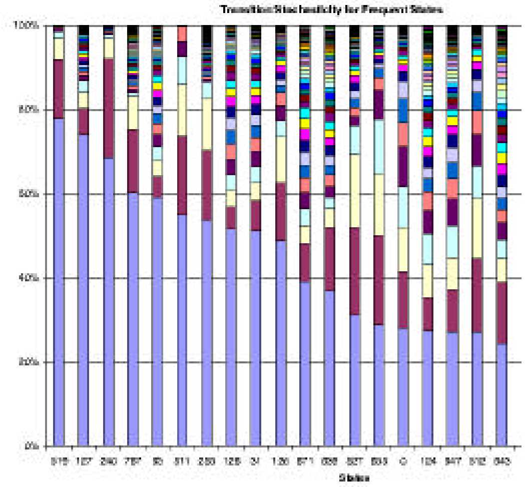
FullQ learner/failure after 10K



Why learning takes so long



Effect of discretization



Lessons from machine learning

- Why task is hard: most frequently occurring state occurs 45% of time, all others are less than 5%.
- Long sequence of moves makes credit assignment hard.
- Staged learning makes task easier; and might help humans acquire task easier.
- Need for a locally non-deceptive reward function to speed up training. Can giving progress function as hints to human players help?

Outline

- The NRL Navigation Task
- Challenges in modeling human learning
- Understanding the task: optimal player
- A hybrid model for human learning
- Understanding the task: reinforcement learner
- High-fidelity models for human learning

Direct models

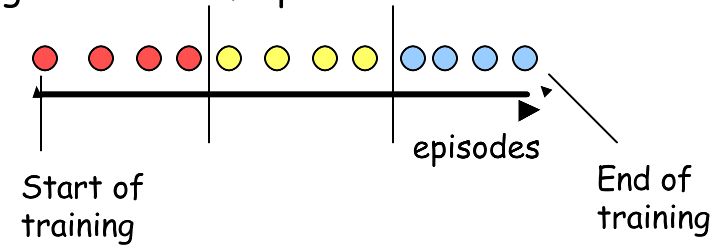
- How well can stateless stochastic models of the form m:sensors → P(actions) match subject learning curves?
  - Associate with every observed sensor configuration, the distribution of actions taken by the player at that configuration.
- Advantage:
  - no need to abstract sensor space.
  - Model construction 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 the subject.

Model construction

- Segmentation of episodic data



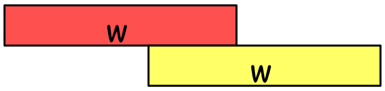
- Fitting models of the form sensors → P(actions) on the stationary segments.

Model Derivative

- dm/dt =

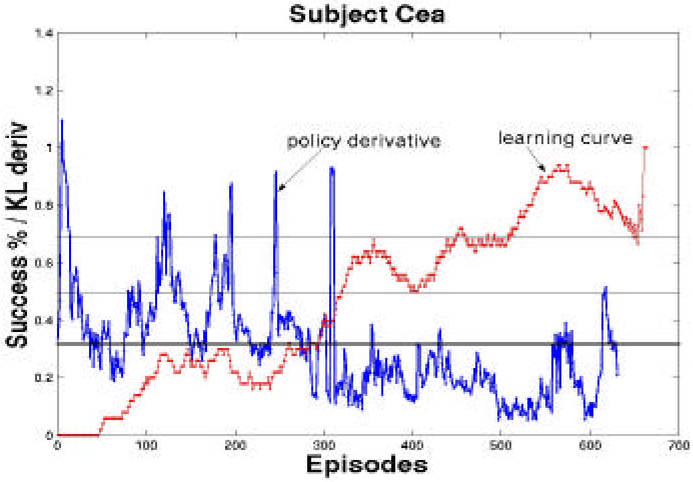
$$\frac{KLdiv(\Pi(i+w-s, i+2w-s), \Pi(i, i+w))}{w}$$

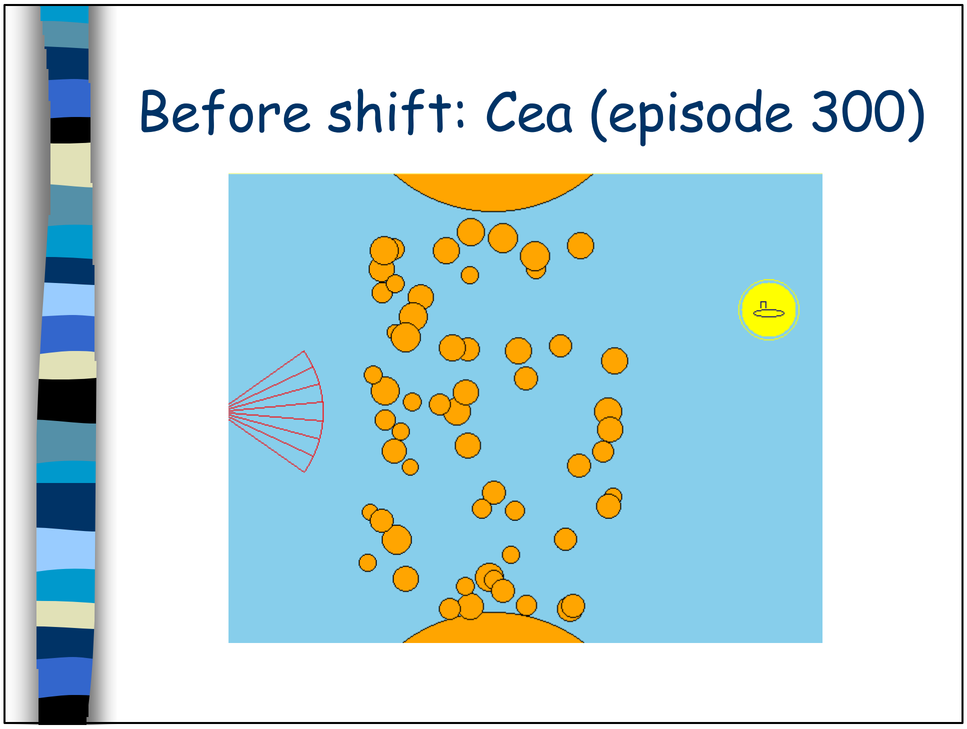
- empirical optimum: w = 20, s = 5
- Computed by Monte Carlo sampling (stabilizes after 5% of entries are sampled)



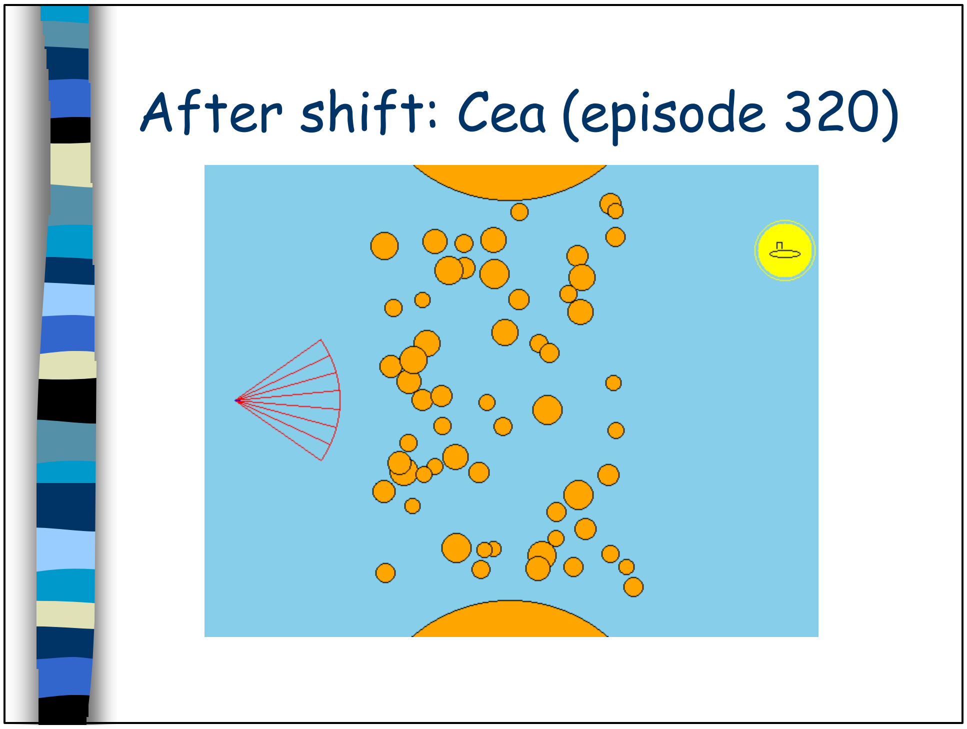
Overlap = s

Model derivative for Cea

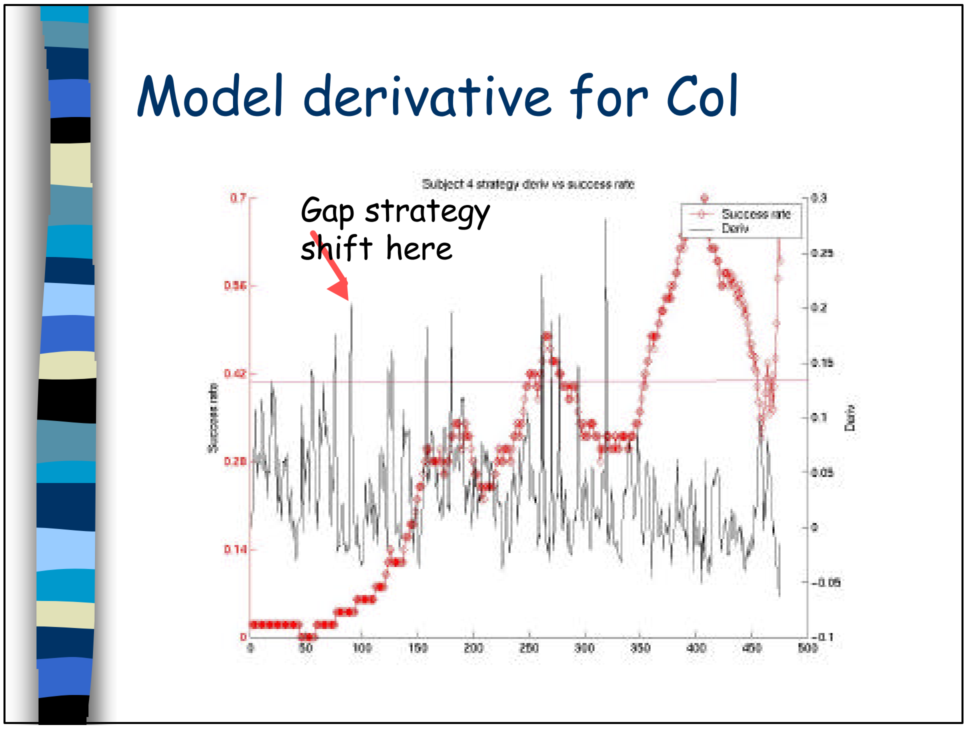




Before shift: Cea (episode 300)

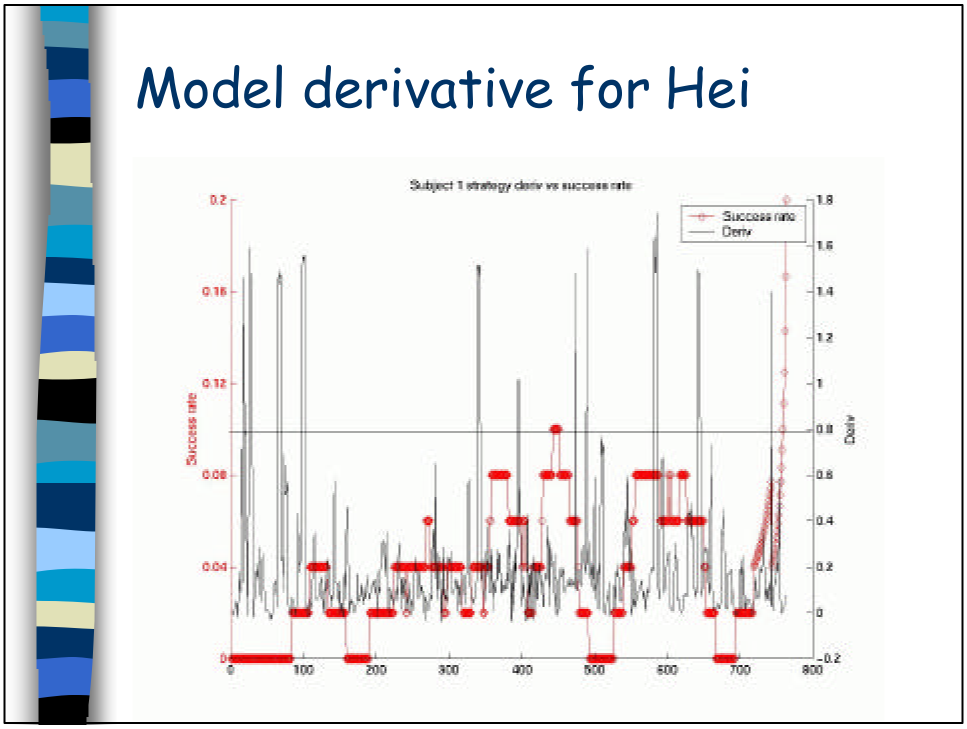


After shift: Cea (episode 320)



Model derivative for Col

Gap strategy shift here



Model derivative for Hei

How humans learn

- Subjects have relatively static periods of action policy choice punctuated by radical shifts.
- Successful learners have conceptual shifts during the first part of training; unsuccessful ones keep trying till the end of the protocol!

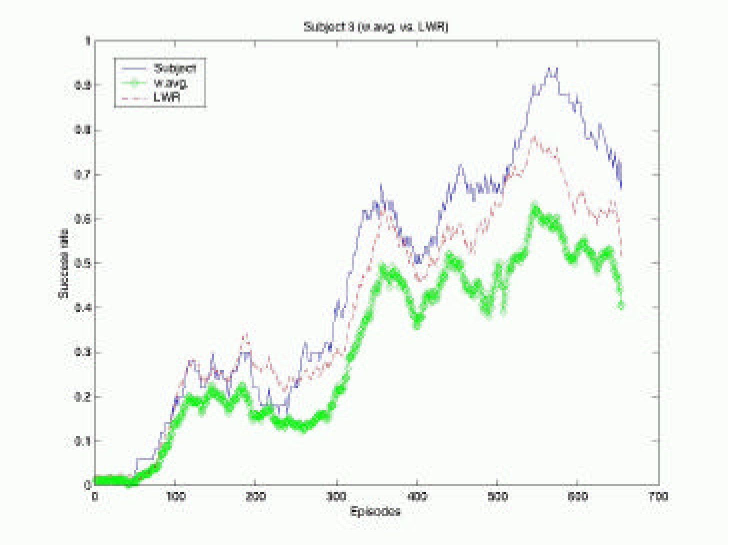
How model is used

- To compute action a associated with current sensor configuration s
  - take 100 neighbors of s in lookup table.
  - Compute weighted average of the actions taken by these neighbors, OR
  - perform locally weighted regression (LWR) on these 100 (s,a) pairs.

Evaluation protocol

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

Results: w.avg. vs. LWR

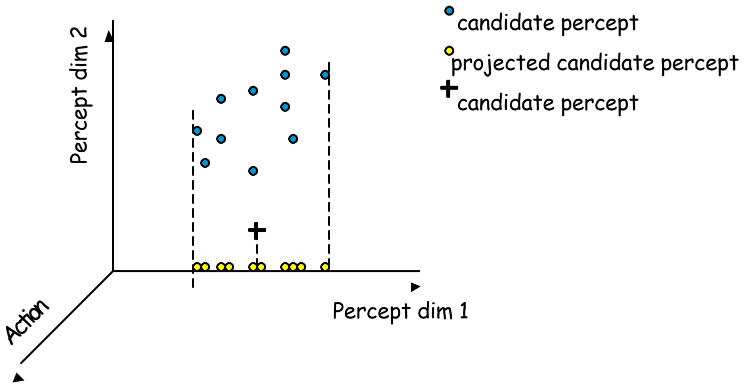


LWR is worse: why?

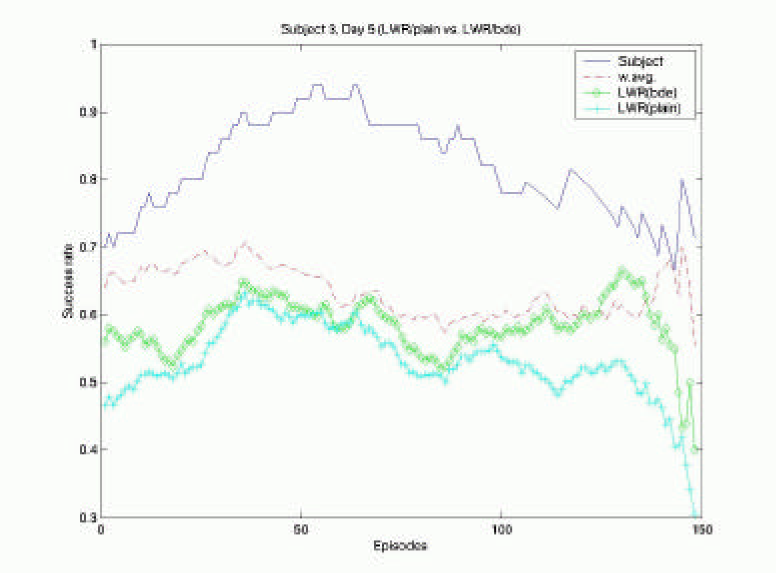
- LWR performs worse than w.avg.
  - data sparsity implies otherwise
- Reason: LWR extrapolates often
  - shown by timeout record

Biased dimension elimination

- Projecting out dimensions to force interpolation



Results: use of bde with LWR



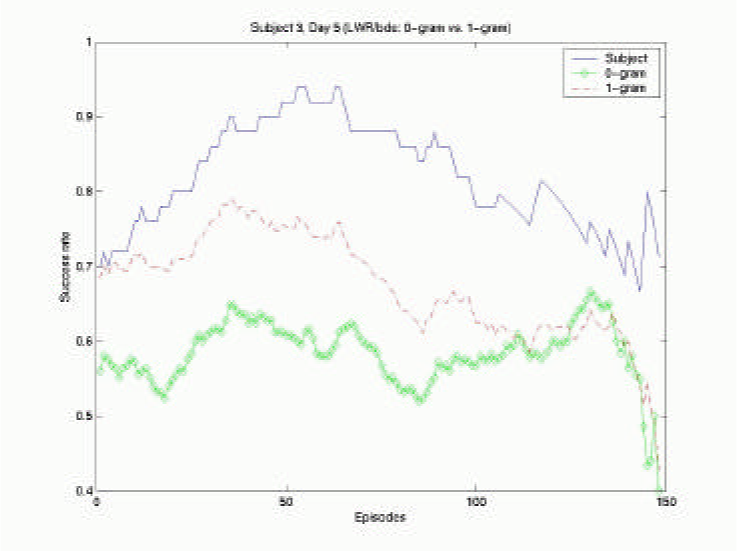
Richer models: internal state

- Remember past k actions

$$f_k = < p_t, a_{t-1}, \ldots, a_{t-k} >$$

- K-gram models: experimented with k=1, 2, 3

Results: 1-gram models

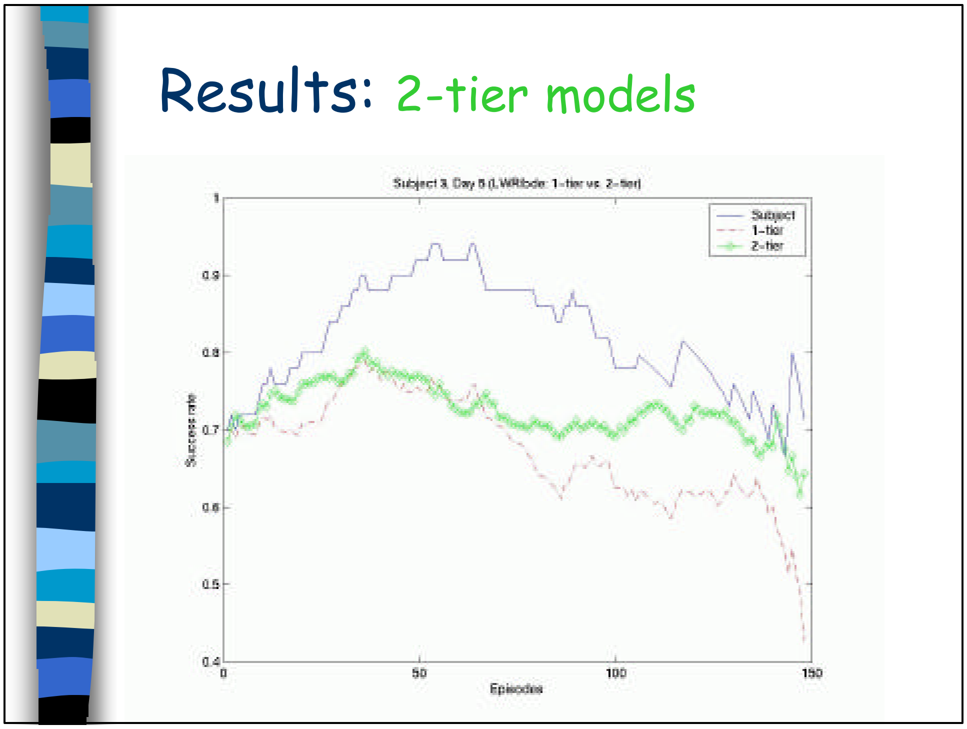


Increasing state preferentially

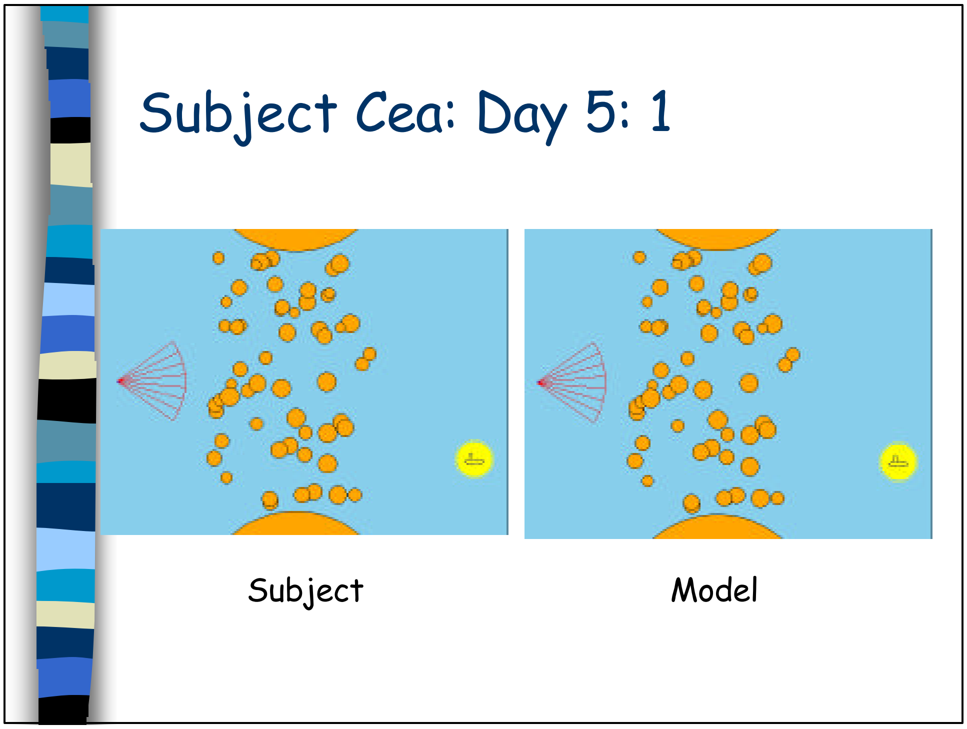
- Failure to control explosions
  - Close-mine situation:

    #(sonars < k units) > n
  - local optimum: (k, n) = (3, 4)
- Two-tier model
  - tier 1: in far-mine, use 1-gram model
  - tier 2: in close-mine, use

$$f_7 =< p_t, p_t - p_{t-1,} a_{t-1}, a_{t-3}, a_{t-5}, a_{t-7} >$$



Results: 2-tier models

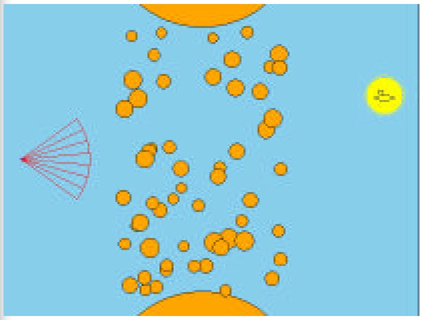


Subject Cea: Day 5: 1

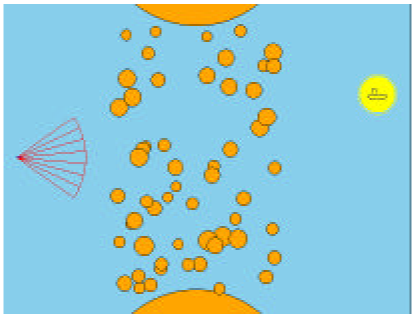
Subject

Model

Subject Cea: Day 5: 2

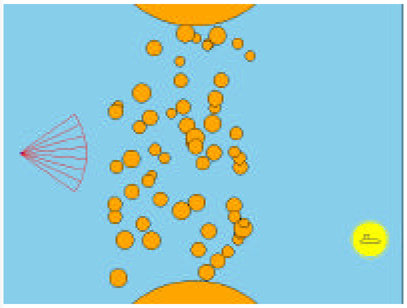


Subject

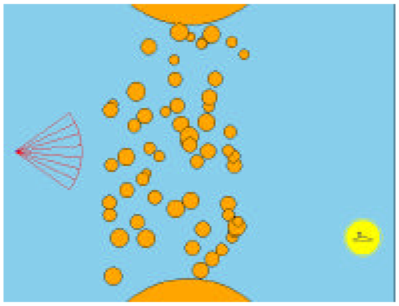


Model

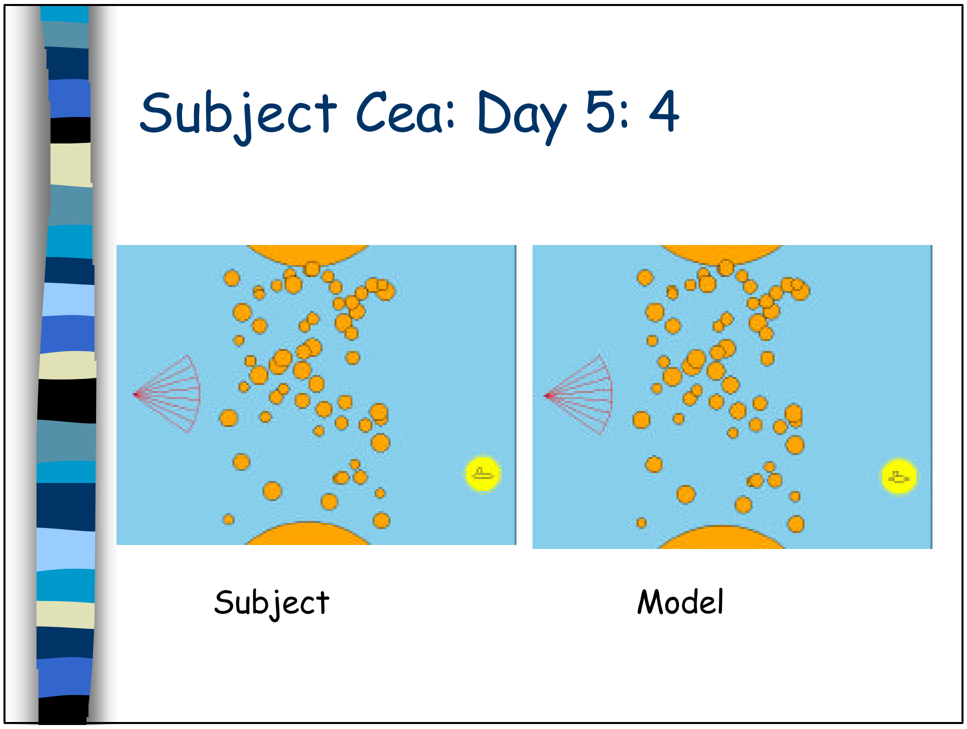
Subject Cea: day 5: 3



Subject



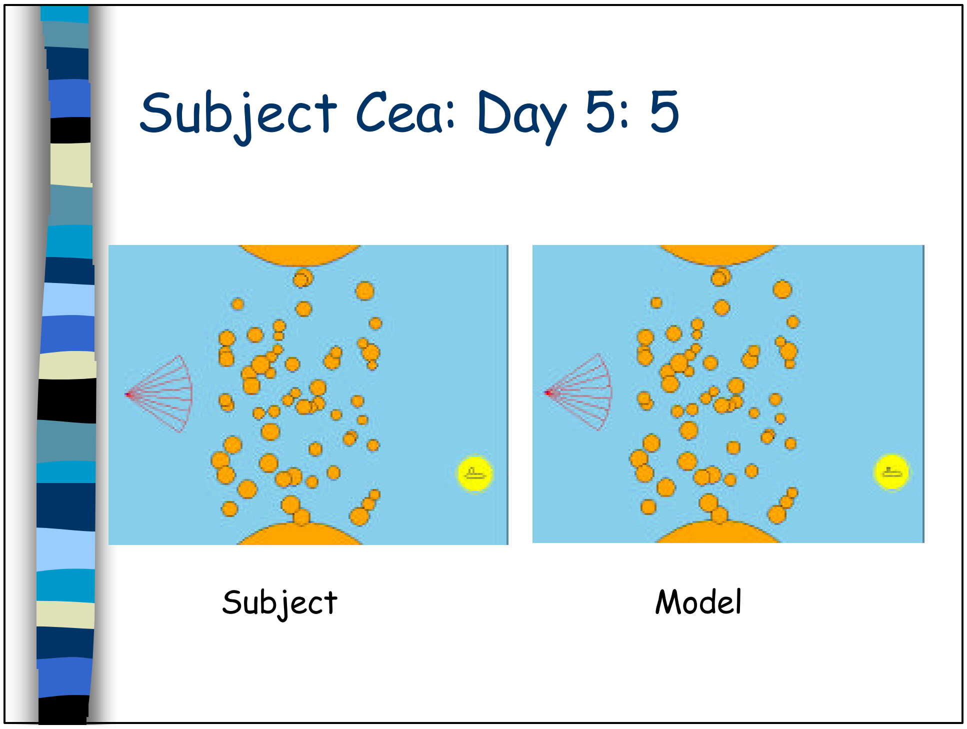
Model



Subject Cea: Day 5: 4

Subject

Model

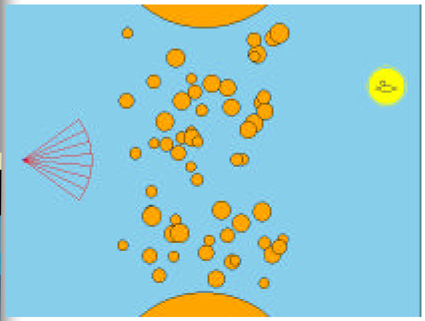


Subject Cea: Day 5: 5

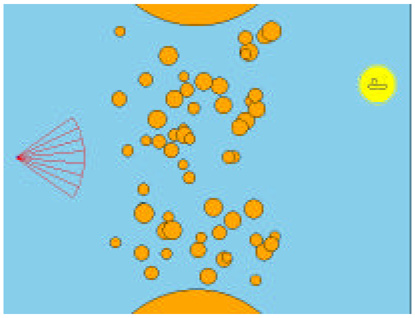
Subject

Model

Subject Cea: Day 5: 6

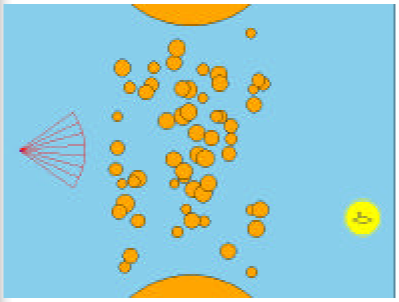


Subject

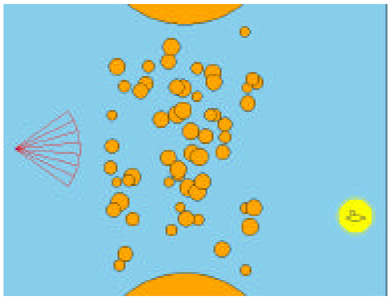


Model

Subject Cea: Day 5: 7

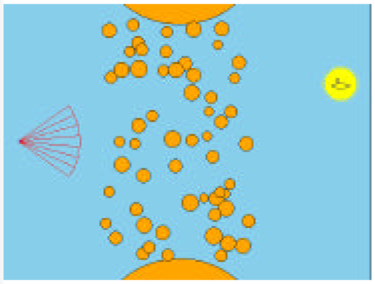


Subject

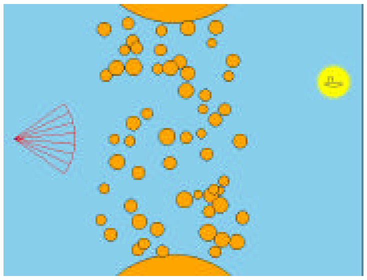


Model

Subject Cea: Day 5: 8

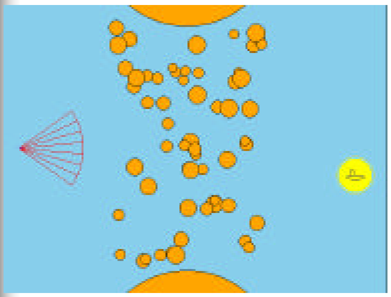


Subject

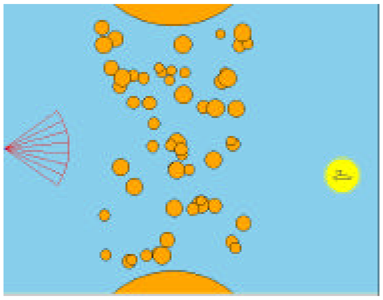


Model

Subject Cea: Day 5: 9

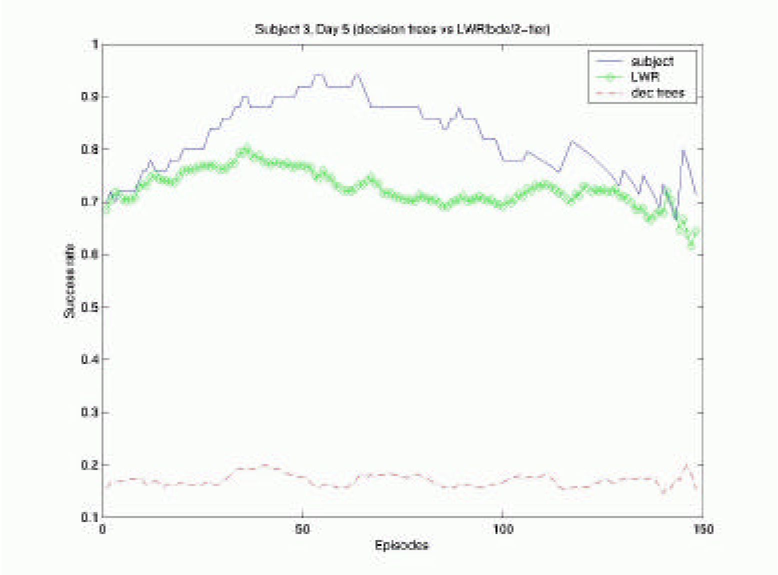


Subject



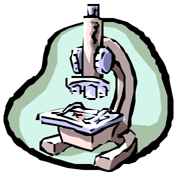
Model

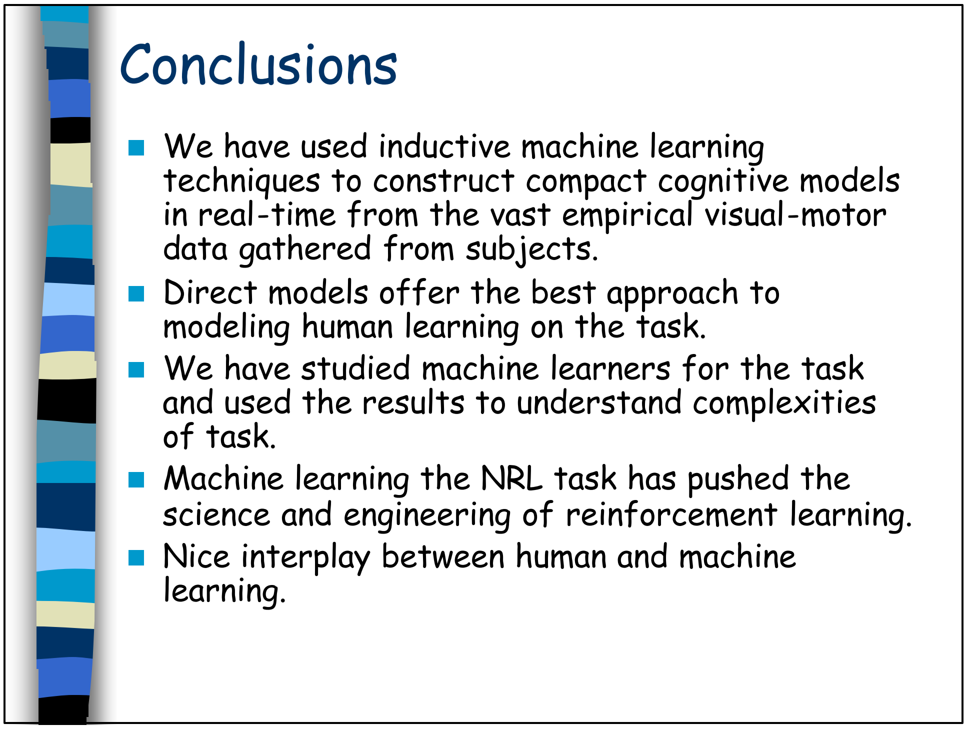
Comparison with global methods



Result summary

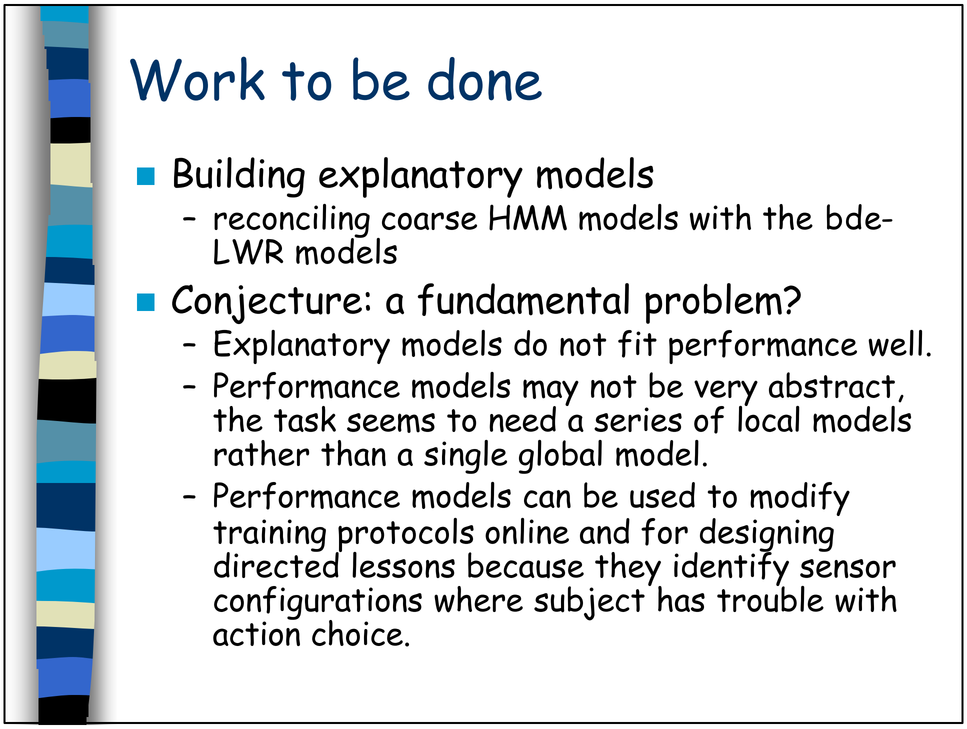
- We can model subjects on the NRL task in real-time, achieving excellent fits to their learning curves, using the technique of 1-gram/bde-LWR/2-tier on the available visual-motor data stream.





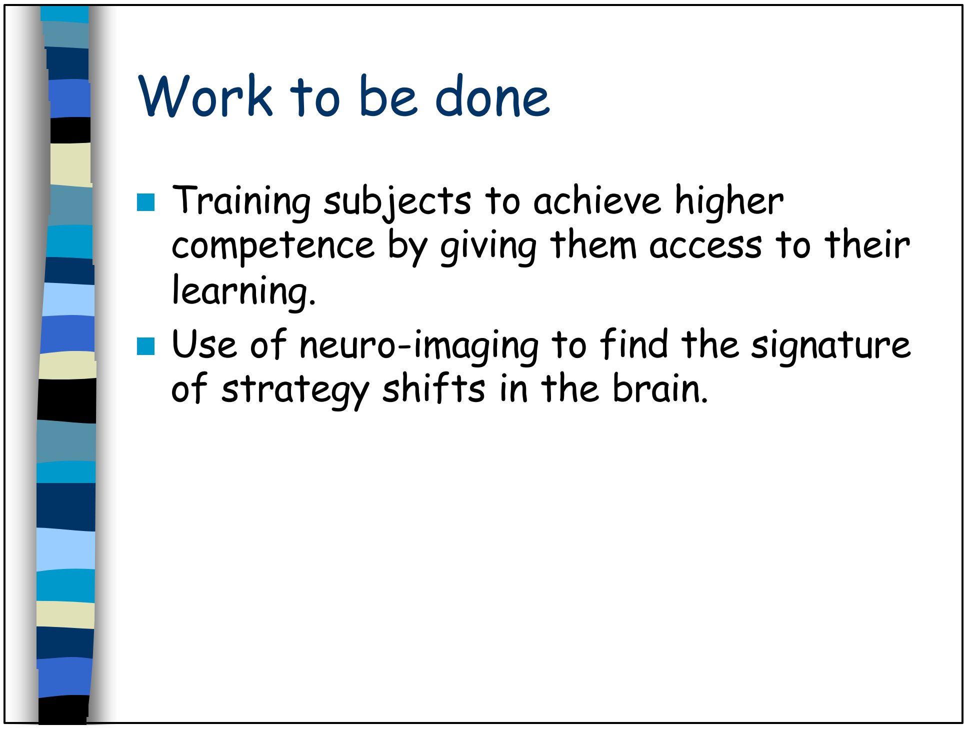
Conclusions

- We have used inductive machine learning techniques to construct compact cognitive models in real-time from the vast empirical visual-motor data gathered from subjects.
- Direct models offer the best approach to modeling human learning on the task.
- We have studied machine learners for the task and used the results to understand complexities of task.
- Machine learning the NRL task has pushed the science and engineering of reinforcement learning.
- Nice interplay between human and machine learning.



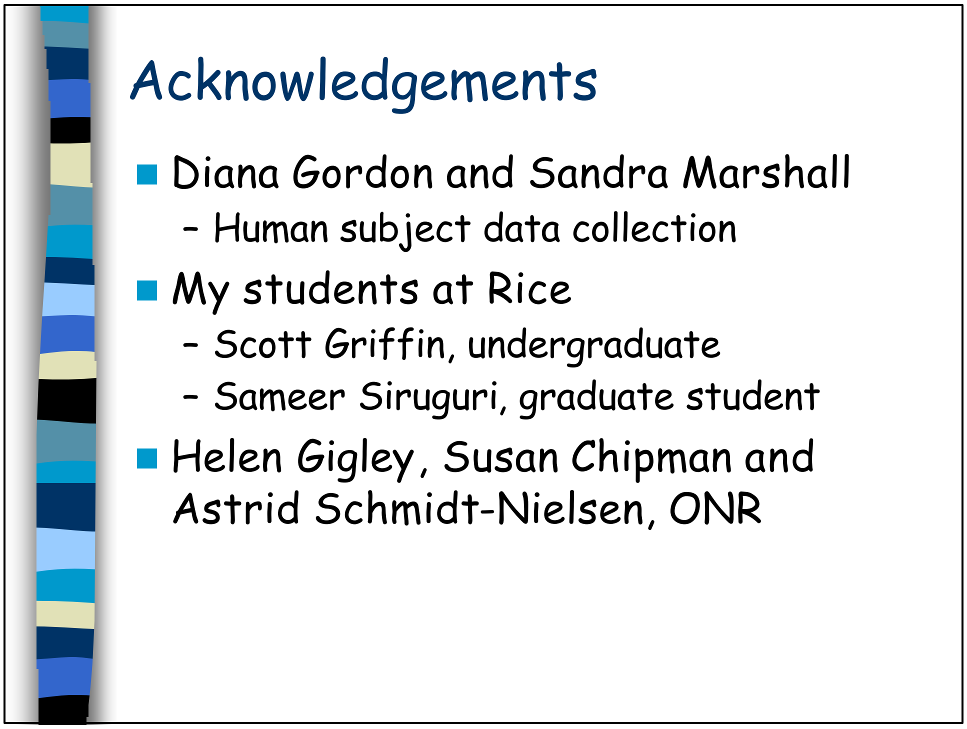
Work to be done

- Building explanatory models
  - reconciling coarse HMM models with the bde-LWR models
- Conjecture: a fundamental problem?
  - Explanatory models do not fit performance well.
  - Performance models may not be very abstract, the task seems to need a series of local models rather than a single global model.
  - Performance models can be used to modify training protocols online and for designing directed lessons because they identify sensor configurations where subject has trouble with action choice.



Work to be done

- Training subjects to achieve higher competence by giving them access to their learning.
- Use of neuro-imaging to find the signature of strategy shifts in the brain.



Acknowledgements

- Diana Gordon and Sandra Marshall
  - Human subject data collection
- My students at Rice
  - Scott Griffin, undergraduate
  - Sameer Siruguri, graduate student
- Helen Gigley, Susan Chipman and Astrid Schmidt-Nielsen, ONR