A simple, predictive, architectural model of categorization in the wild

Christian Lebiere & Eric Biefeld

Human-Computer Interaction Institute

Carnegie Mellon University

cl@cmu.edu; biefeld@cs.cmu.edu

Plan

- Round 3 of AMBR project
- Model overview
- Model methodology
- Model description
- Results: the BBN graphs
- Additional analysis
- Parameter sensitivity analysis
- Modifications
- Rule Learning

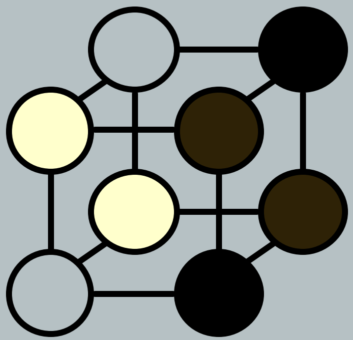
AMBR III

- Showcase and challenge the state of the art
- Targeted behavior in third phase: learning
- Paradigm: category learning (Shepard et al, 1961)
- Primary learning task integrated with secondary ATC task of transferring planes between airspaces
- Impact of workload on learning performance (not!)
- Limited data-fitting: learning transfer data withheld
- Competitors: ACT-R, Cognet, D-COG2, EPIC-Soar

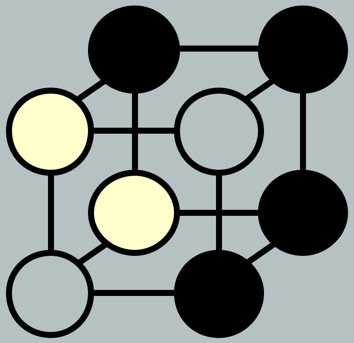
Categorization Task

- 3 category structures out of a possible 6
- Categorize 8 stimuli into accept/reject
- Stimuli varied on 3 binary-valued dimensions

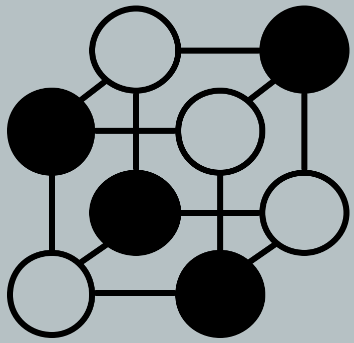
○ - Accept ● - Reject



Type 1



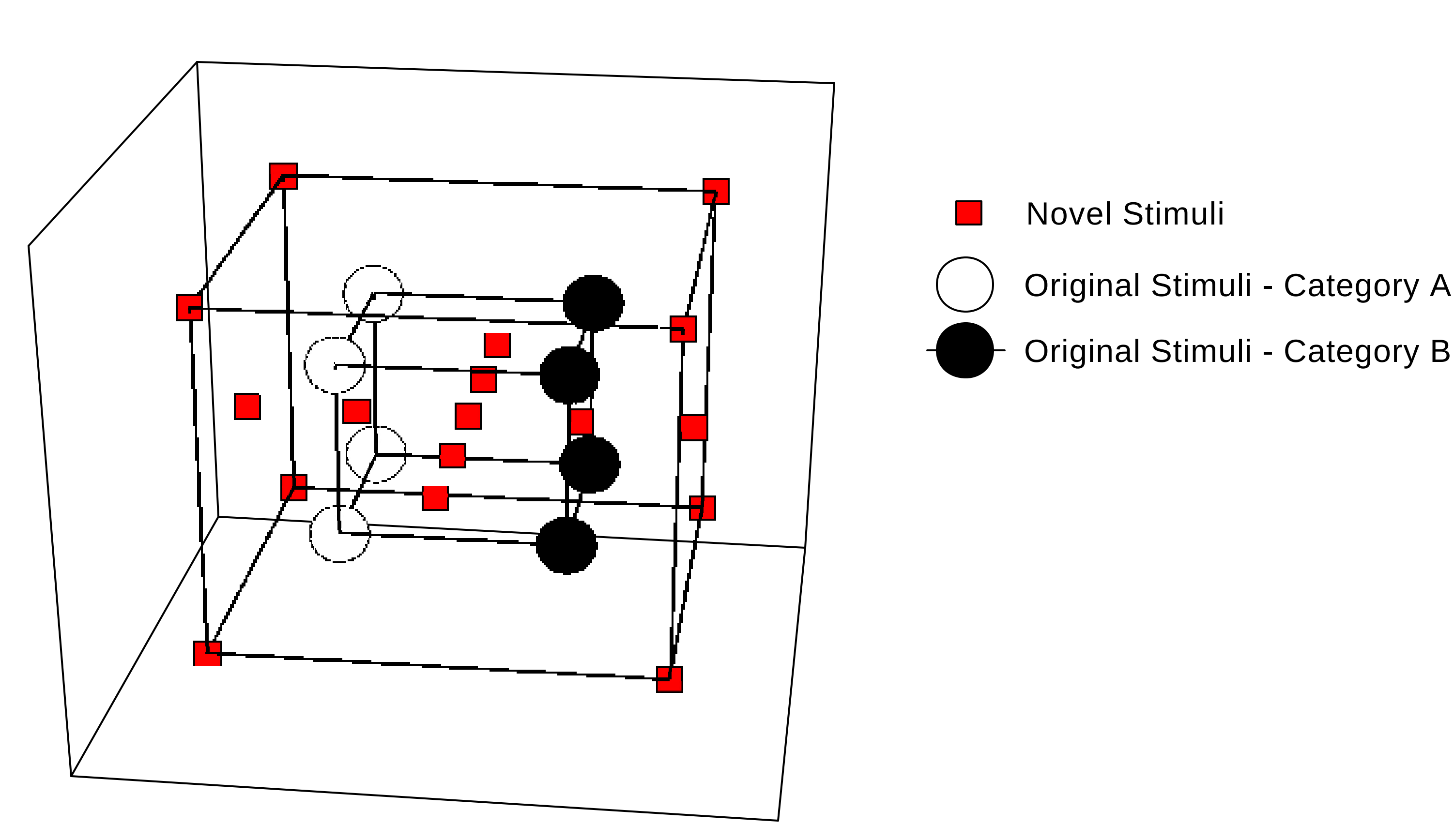
Type 3



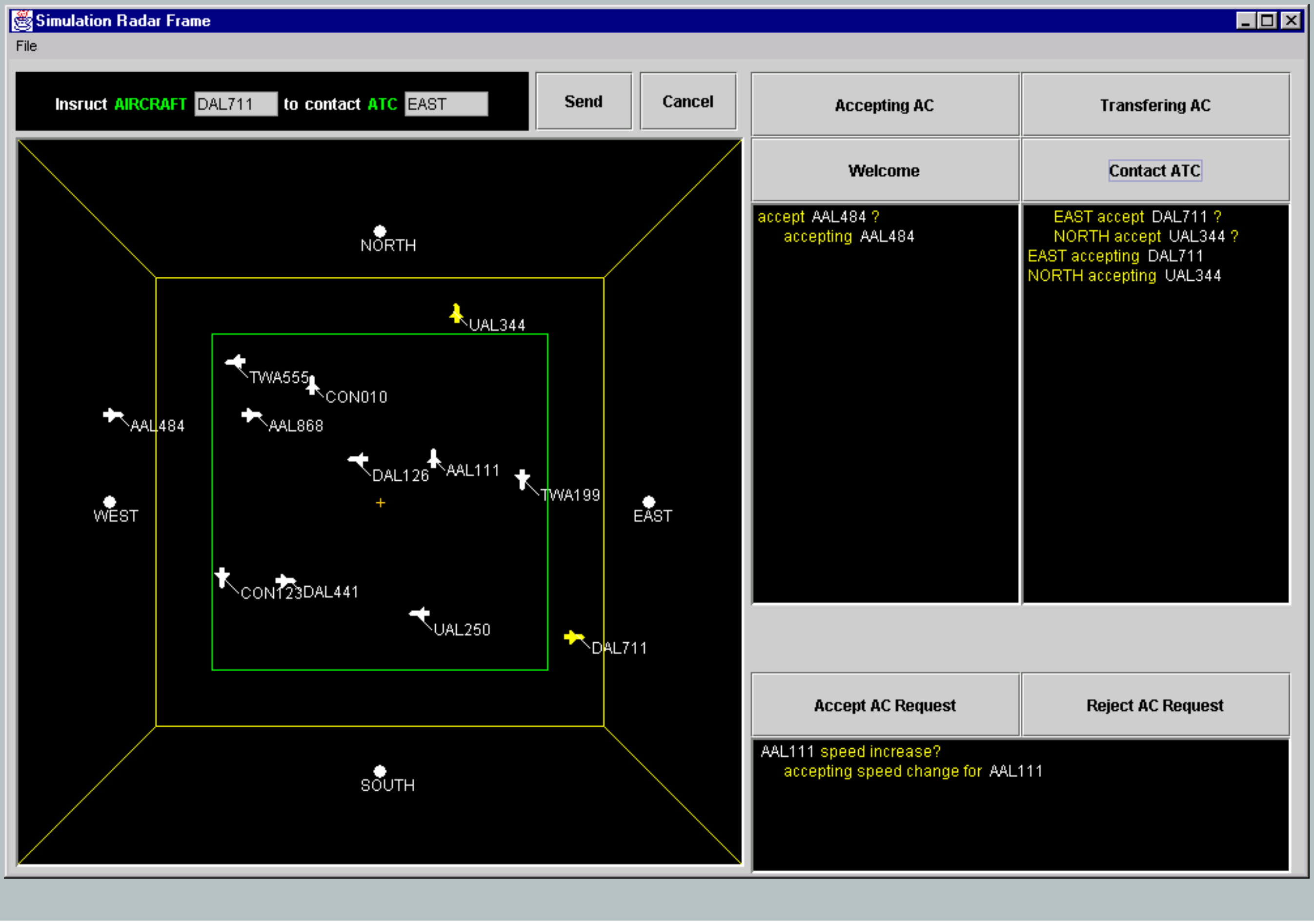
Type 6

Transfer Phase

8 training stimuli and 17 transfer stimuli


🟥 Novel Stimuli
○ Original Stimuli - Category A
● Original Stimuli - Category B

Interface



Insruct AIRCRAFT DAL711 to contact ATC EAST

```
accept AAL484 ?
    accepting AAL484
```

```
    EAST accept DAL711 ?
    NORTH accept UAL344 ?
EAST accepting DAL711
NORTH accepting UAL344
```

```
AAL111 speed increase?
    accepting speed change for AAL111
```

Quote of the Day

“What do we mean by a correct (appropriate) model for an assemblage of data? The generally accepted criterion, I believe, is that the model is necessary and sufficient for prediction of the data.”

W. K. Estes, “Traps in the route to models and decision”

Psychonomic Society Keynote Address, 2001

Model Overview

- The model is an instance-based model (e.g. Logan, 1988). No “rules”.
- Gradual switch from general strategy to knowledge retrieval (expertise).
- Retrieval of instances is architecturally constrained by theory of declarative memory and (almost) parameter free. It is also necessary. Is it sufficient?
- Rule learning (NOT production rules), ie. hypothesis generation and testing, is a complex skill w/ lots of degrees of freedom and individual differences.
- Instances are acquired more easily and straightforwardly than rules (time!).
- Hard rules are seldom followed or known or even exist in even mildly complex situations (next slide) because of complexity, uncertainty, changes, and approximations and can be very difficult to acquire if at all (scaling).
- Instances together with generalization can be ‘soft” rules (e.g. language)

Some instance-based models

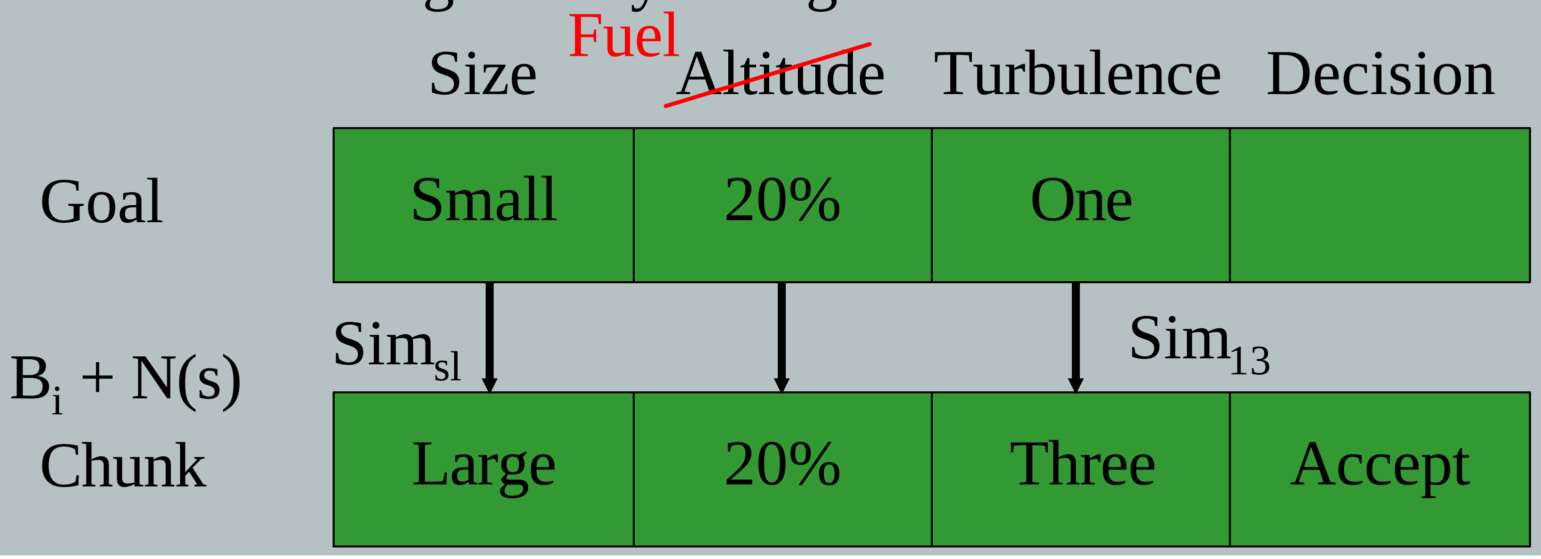
- Control Problems
  - Sugar Factory (Lebiere, Wallach & Taatgen, 1998; Wallach & Lebiere, 2002)
  - Transportation Task (Wallach & Lebiere, 2002)
- Game Playing
  - Paper Rock Scissors (Lebiere & West, 1999; West & Lebiere, 2001)
  - Backgammon (Sanner, Anderson, Lebiere & Lovett, 2000)
  - 2x2 Games (Bracht, Lebiere & Wallach, 1998; Lebiere, Wallach & West, 2001)
- Decision Making
  - Pipes (Lerch, Gonzalez & Lebiere, 1999; Gonzalez, Lerch & Lebiere, subm)
  - Melioration, Committee, Centipede (Lebiere & Shang, 2002)

Methodology

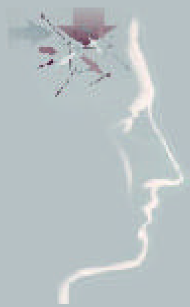
- The model does not specifically generate rules, but one could interpret the set of instances as “virtual” rules the way “hard” rules are extracted out of other “soft” systems such as neural networks. There are “quantum” issues of representing uncertainty in modeling.
- A standalone version of the categorization model was developed for rapid evaluation. The only (roughly) estimated parameter was the retrieval threshold, for which no architectural default exists. Other parameters were either architectural defaults or inherited from AMBR I, as were the color and process goals. A new decision goal was added.

Declarative Knowledge

- (8) chunks created from past goals to solve decision problem
- They contain four slots describing problem and decision
- Decision process favors retrieval (if possible) over guessing, with the former gradually taking over from the latter



	Size	~~Altitude~~ Fuel	Turbulence	Decision
Goal	Small	20%	One	
$B_i + N(s)$ Chunk	Large	20%	Three	Accept



Procedural Knowledge

- The secondary task is handled by the productions for the color and process goals from AMBR I.
- One goal (unit task) is added to handle the decision making task.
- The model consists of a total of 19 production rules applicable to the 3 goals.
- Productions are small and hopefully straightforward, encoding little more than the structure of the task.
- The key is the retrieve production, which performs a partial match to memory for which all decisions chunks qualify.

Color Goal

Color-Target-Detection

IF the goal is to detect a color aircraft and *aircraft* is a color aircraft
THEN note *aircraft*

Color-Target-Acquisition

IF the goal is to detect a color *aircraft* and the color of aircraft is *color*
THEN note *color*

Color-Magenta-Action

IF the goal is to detect a color *aircraft* at *position* and its color is magenta
THEN push a goal to make a decision for *aircraft* at *position*

Color-Action

IF the goal is to detect a color *aircraft* and its color is *color*
and there is a memory associating *color* with *action*
THEN push a goal to process *action* for *aircraft* at *position*

Wait

IF the goal is to detect a color aircraft
THEN wait

Decision Goal

Target-altitude/turbulence/size (3)

IF the goal is to make a decision for *aircraft* and no altitude/turbulence/size is known
THEN encode the altitude/turbulence/size of *aircraft* in the goal

Remember-decision

IF the goal is to make a decision for *aircraft* of *altitude, turbulence* and *size* and there is a memory of a *decision* for an aircraft of *altitude, turbulence* and *size*
THEN select *decision*

Guess-decision

IF the goal is to make a decision for *aircraft* of *altitude, turbulence* and *size*
THEN randomly decide between accept-altitude and reject-altitude

Subgoal-process

IF the goal is to make a *decision* for *aircraft* at *position*
THEN push the goal to process *decision* for *aircraft* at *position*

Feedback

IF the goal is to make a *decision* and *feedback* indicates *correct*
THEN change decision to *correct* and pop the goal

Wait-for-feedback

IF the goal is to make a decision and a *decision* has been made
THEN wait for feedback

Process Goal

Answer-other-requests

IF the goal is to process an *action* other than altitude-request and the step is select
THEN push the button corresponding to the *action* and change the step to controller

Answer-altitude-requests

IF the goal is to process an altitude-request *action* and the step is select
THEN push the button corresponding to the *action* and change the step to target

Skip-welcome-controller

IF the goal is to process a welcome action and the step is controller
THEN change the step to target

Click-controller

IF the goal is to process an *aircraft* and the step is controller
THEN select the *controller* associated with *aircraft* and change the step to target

Click-target

IF the goal is to process an *aircraft* at *position* and the step is target
THEN select the *aircraft* at *position* and change the step to send

Click-send

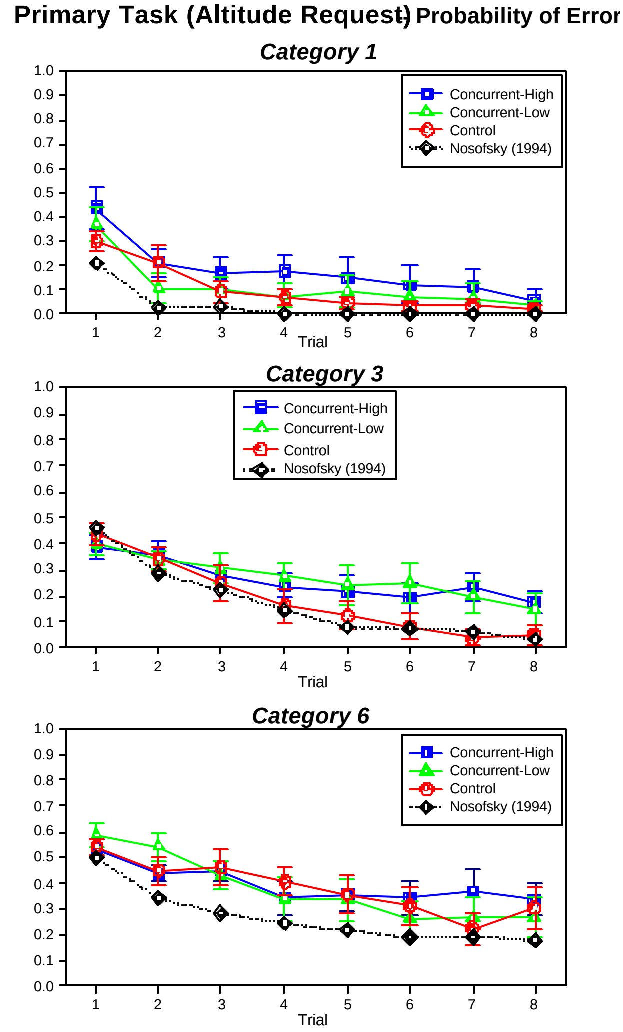
IF the goal is to process an action and the step is send
THEN push the send button and pop the goal

Parameters

- Retrieval latency factor **F**: 1.0 (architectural default)
- Decay rate **d**: 0.5 (architectural default)
- Mismatch Penalty **mp**: 1.5 (architectural default)
- Activation noise **s**: 0.25 (personal default)
- Retrieval threshold ?: -1.0 (no general default)
- Similarities: linear in the [0,1] interval (general default)
- Priors for color-action chunks: same as AMBR I
- Action times (efforts) for productions: same as AMBR I

Lack of Interaction

Lack of interaction between tasks is the default prediction. If no work on the categorization task happens between instances of categorization, then it doesn't matter what else happens at that time, i.e. what the workload is for the other task. This is important because it allows for a modular design of models around independent goals (unit tasks) that don't interfere with each other when put together (composability). Of course, some time devoted to other tasks could be used for rehearsal of instances and elaboration of rules, so some effect is possible.



Workload Predictions

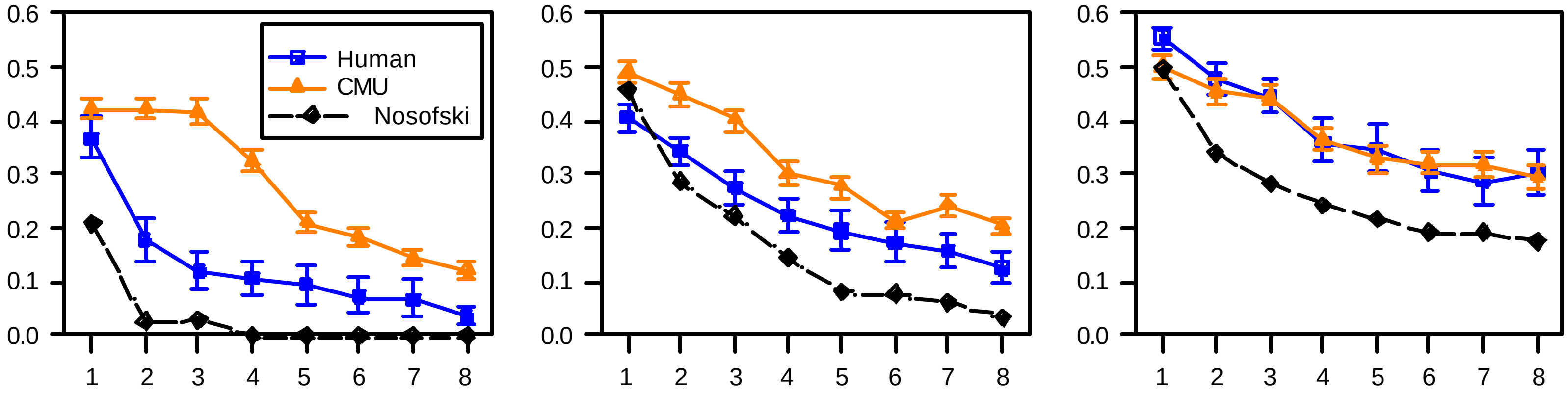
- Definition: $$\mathit{Workload} = c \cdot \frac{\sum_{\mathit{criticaltasks}} \mathit{TimeonTask}}{\mathit{TotalTime}}$$

- The same definition and scaling parameter was used as in AMBR I. No free parameters were introduced or estimated. The definition is grounded in the architectural concept of goal. The only generalization needed from AMBR I is the addition of the decision goal to the list of critical unit tasks.

Errors on Primary Task

- The good: Excellent fit to category 6
- The bad: Slow by a constant factor on category 3
- The ugly: Delayed learning for category 1
- Reason: wrong RT, lack of rehearsal, rule, primacy?

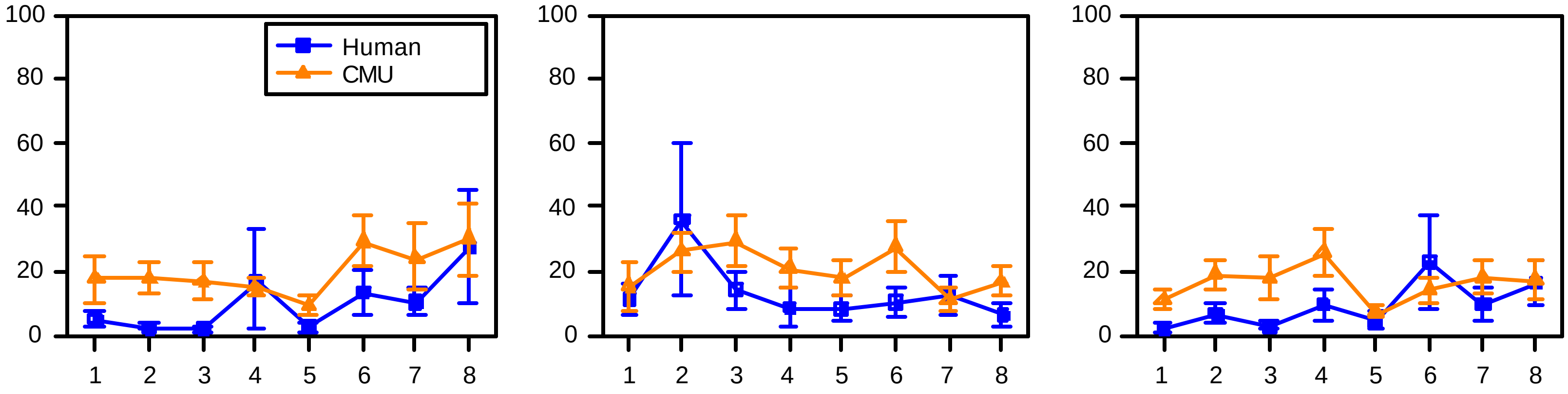
CMU



Penalty for Secondary Task

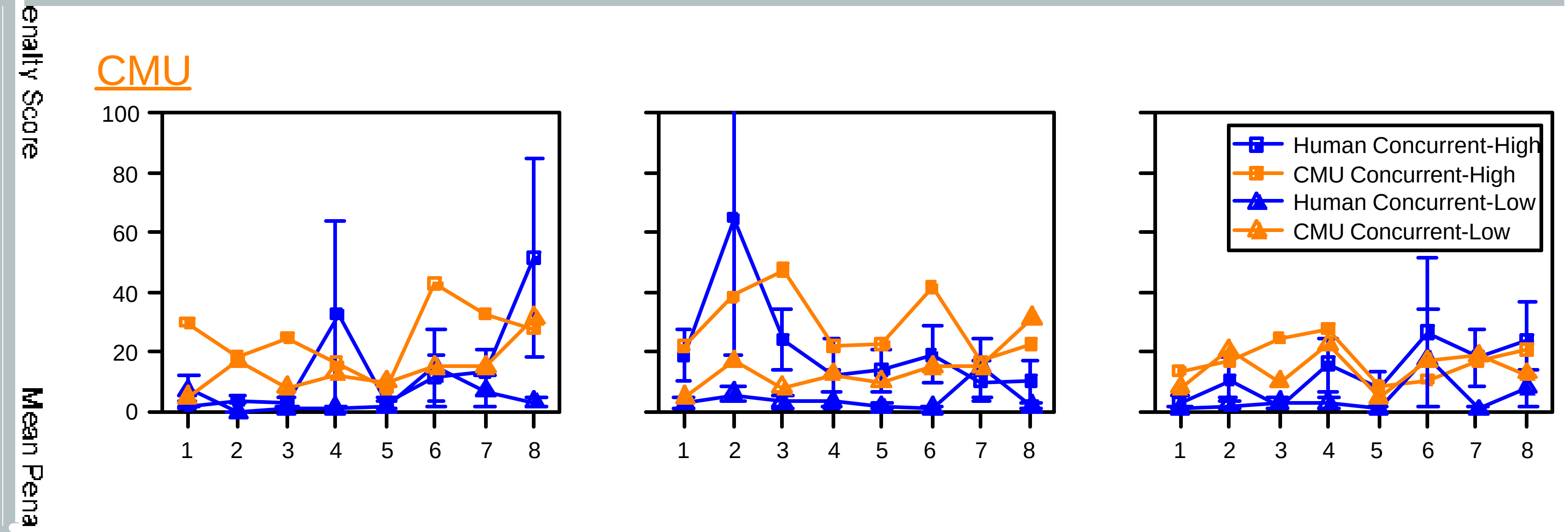
- Sources of secondary task errors:
  - Lack of time to accomplish task in timely manner
  - Errors in retrieving mapping color-action chunks
    - Same source and constraints as primary errors

CMU



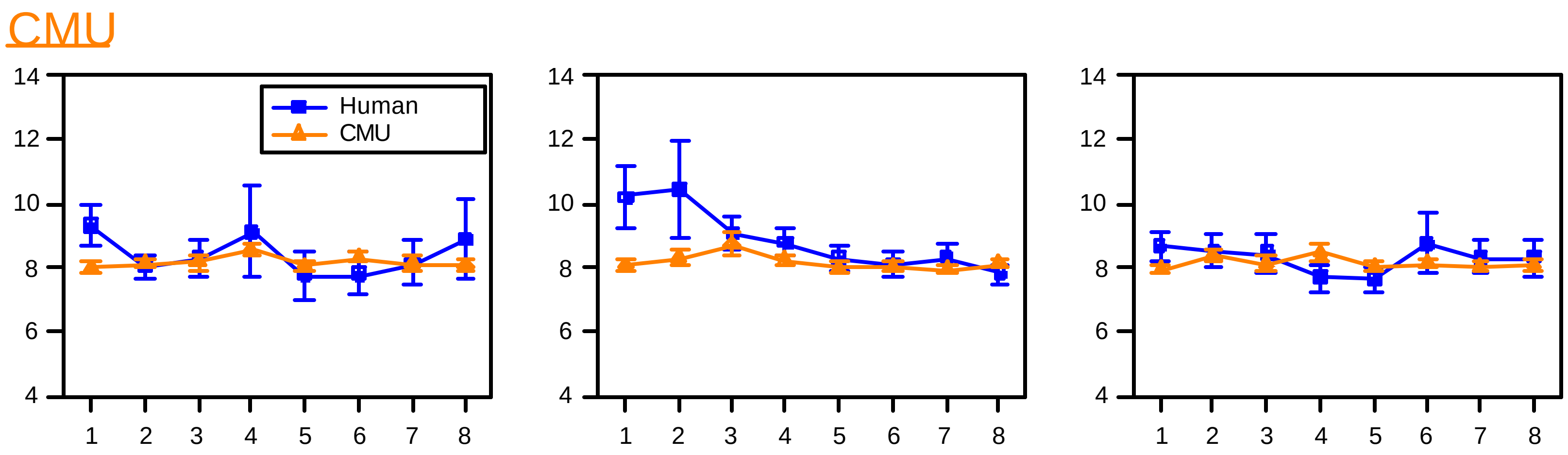
Penalty per Condition

- Significant effect of condition
- More opportunities
- More time constraints



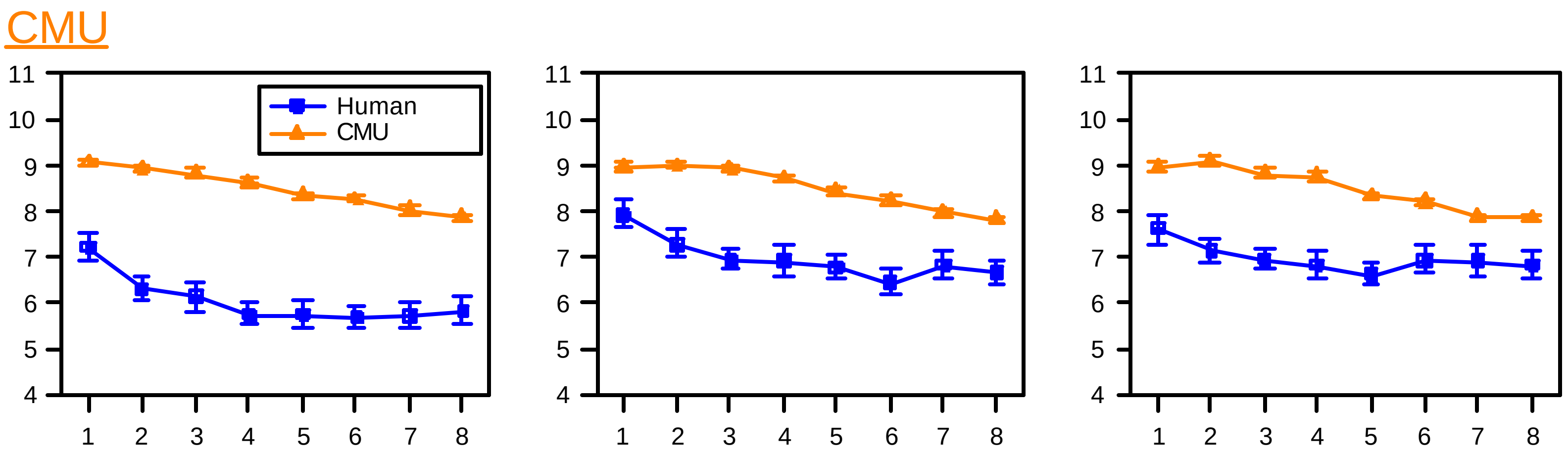
Secondary RT

- Direct prediction from AMBR I
- No significant speed up with practice
- No sensitivity to primary task category



Primary RT

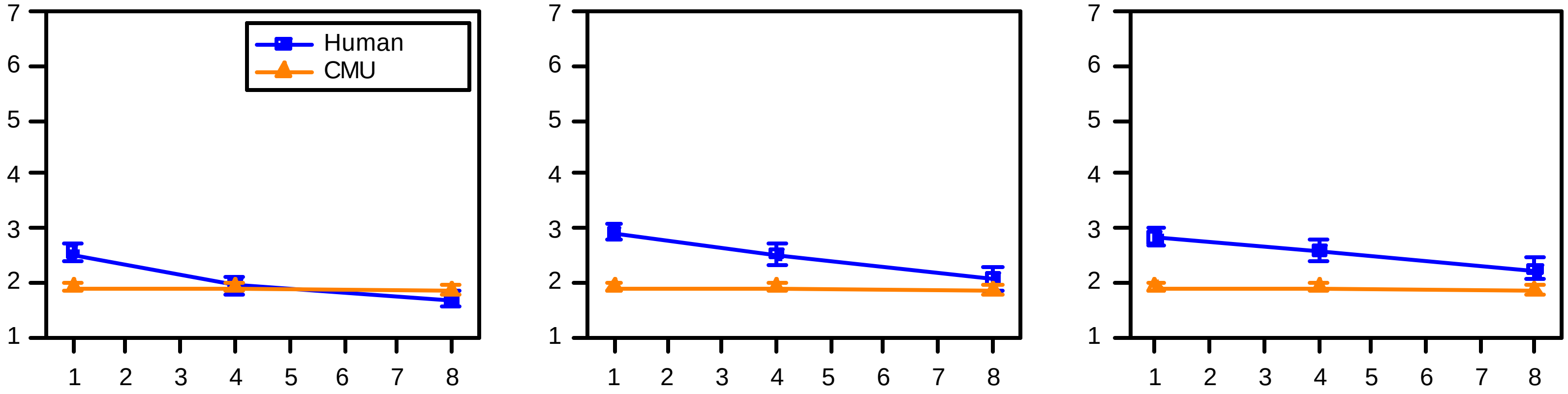
- About correct speedup with practice (~1 second)
- Systematic overestimate of average RT
- Seems logical that primary RT > secondary RT
- Higher priority for primary task would lead to lower RT
- No significant effect of category on RT level -> blending?



Workload Ratings

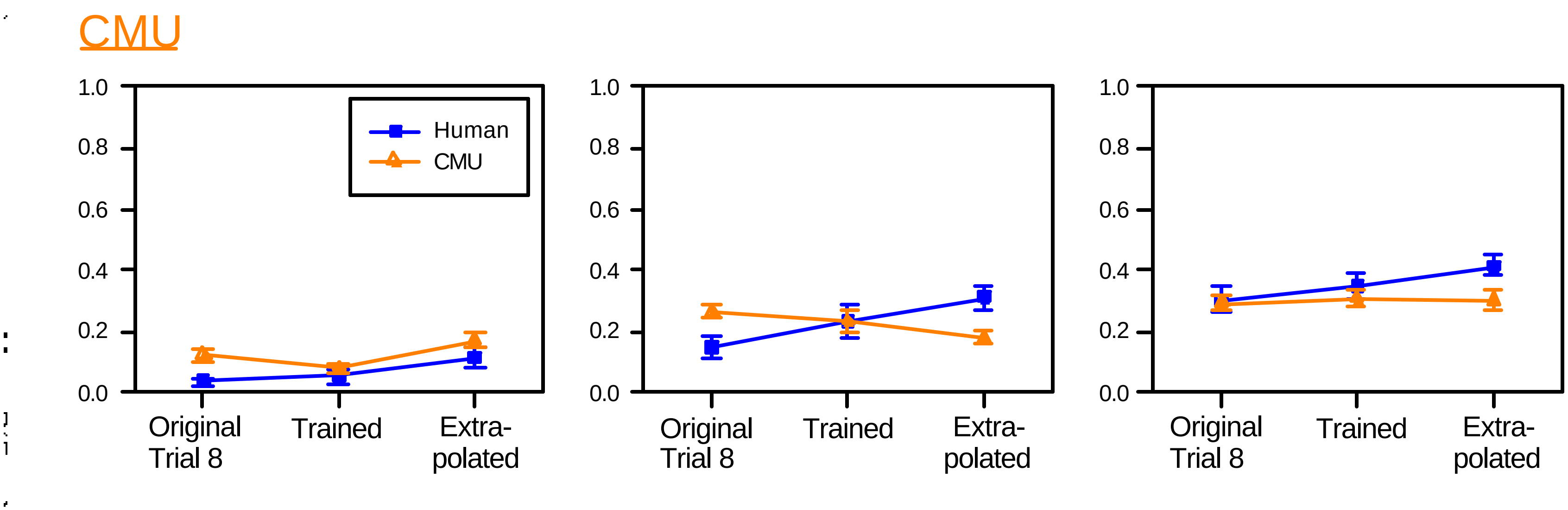
- Direct prediction of general workload from AMBR I
- Some speed up with practice from faster retrieval
- Very little sensitivity to primary task category
- Factor in (in)direct impact of categorization failures?

CMU



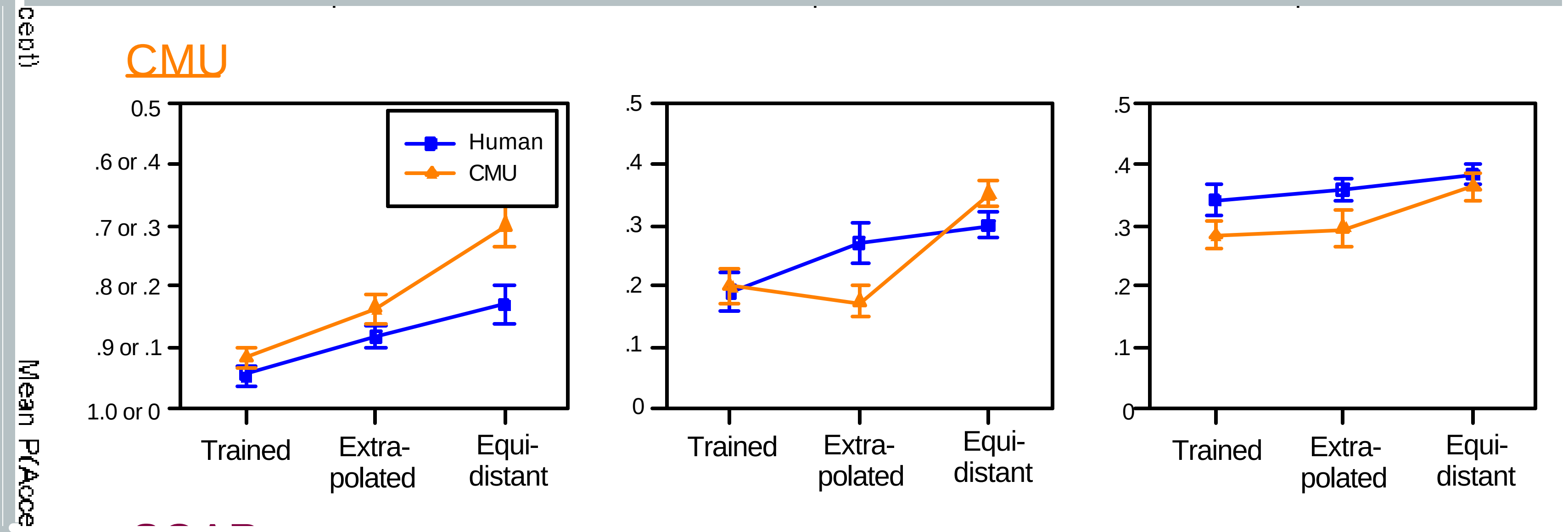
Transfer Errors

- No degrees of freedom!
- Excellent fit to trained instances (better than trial 8!)
- Extrapolated: conflicting effects of similarity distance break down to a wash: non-additive similarity function, guessing, or perhaps the role of associative strengths?

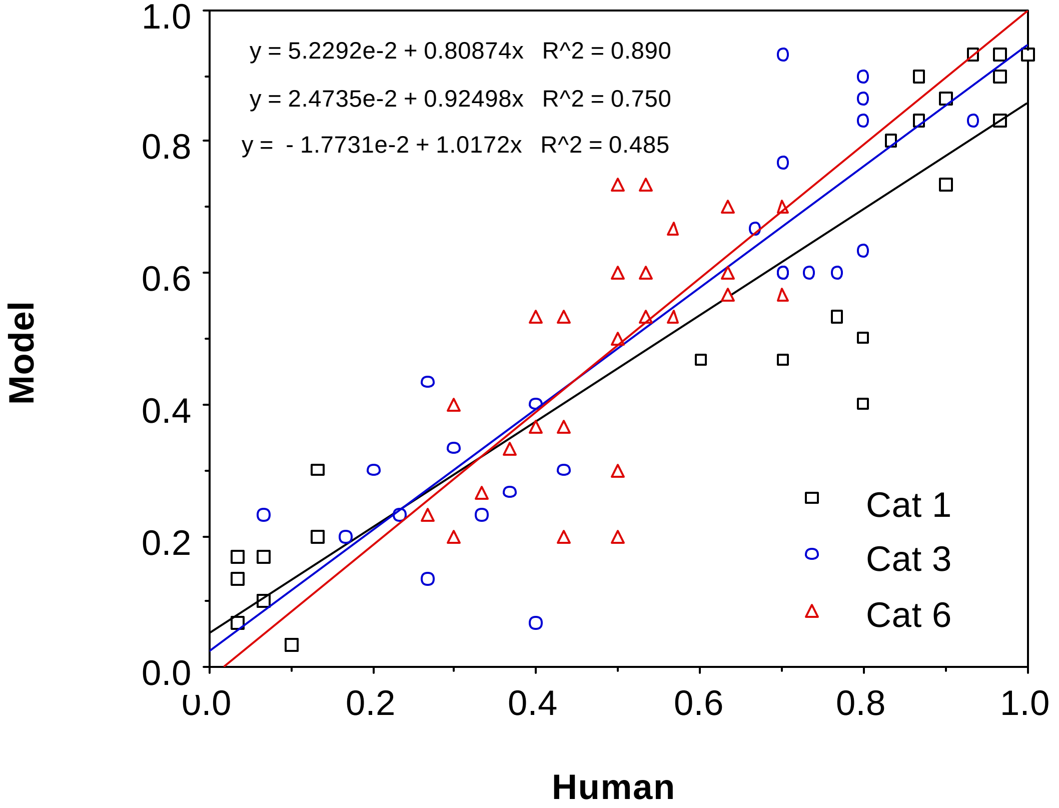


Transfer Consistency

- Good for Cat 3 and 6, less extreme on Category 1
- Explanation: lack of rule, too much noise, … ?
- Different take on (lack of) feedback in transfer phase might produce more consistent, i.e. extreme answers



Prediction per Category Type



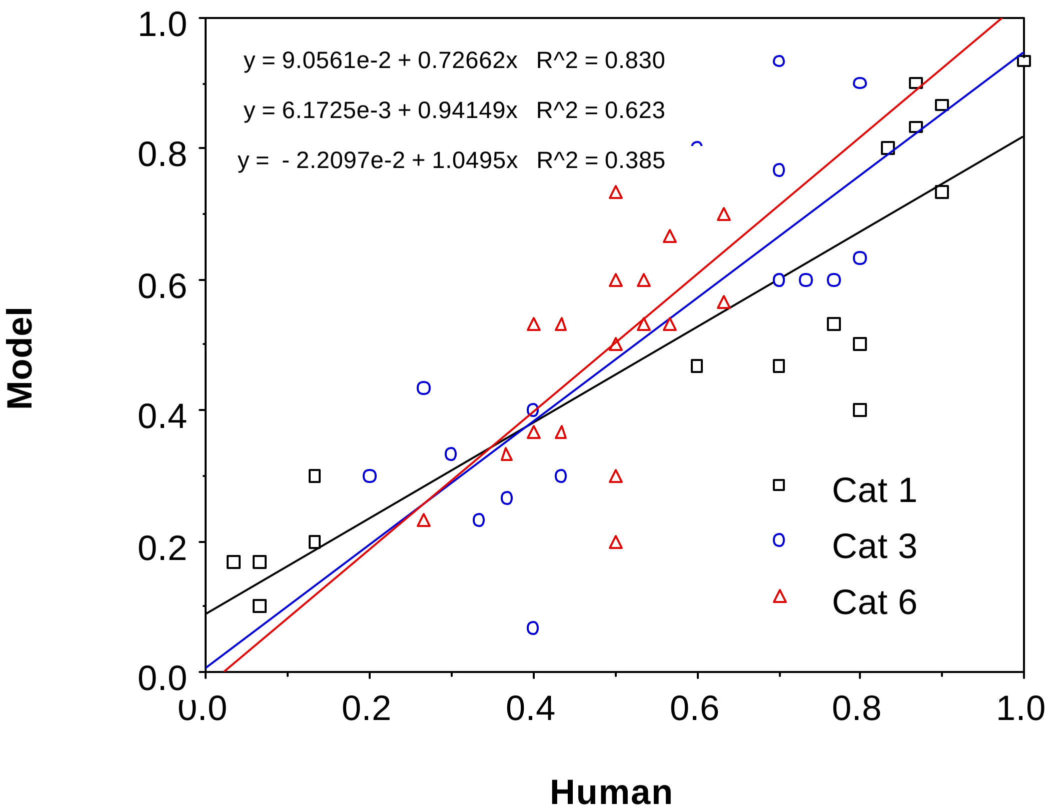
- Parameter-free prediction (similarities, order mostly)
- Best for 6, good for 3, off for 1: opposite of R^2!
- Better measure: RMSE

Cat 1 = 14.1%

Cat 3 = 13.4%

Cat 6 = 12.5%

Prediction for Transfer Stimuli



- When training stimuli are removed, all the patterns remain, even slightly stronger
- RMSE:

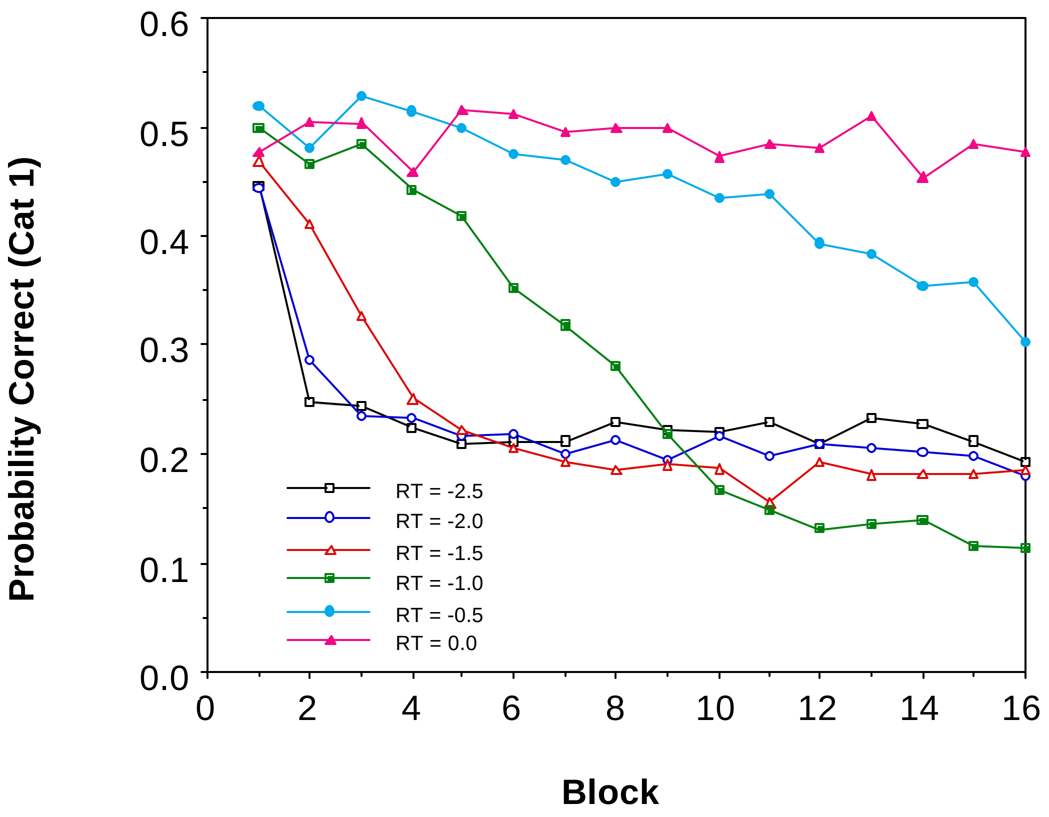
Cat 1 = 16.7%

Cat 3 = 15.1%

Cat 6 = 12.2%

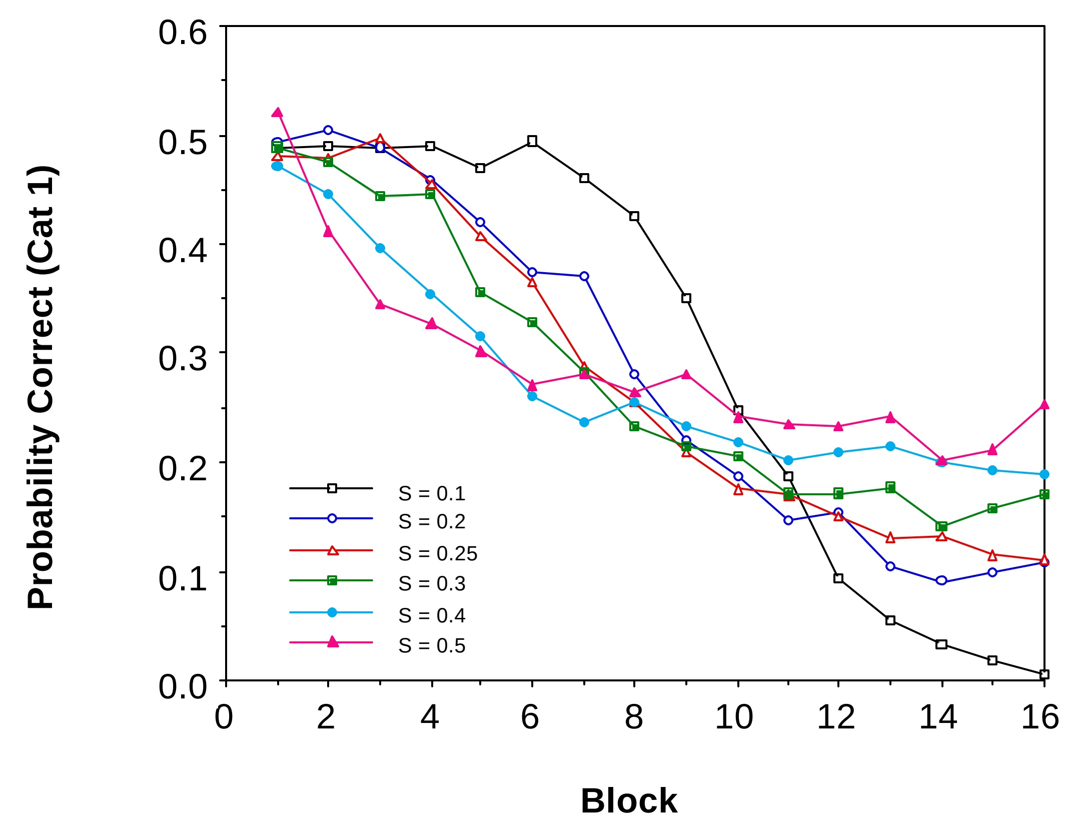
- Actually better for Cat 6!

Parameter Sensitivity: RT



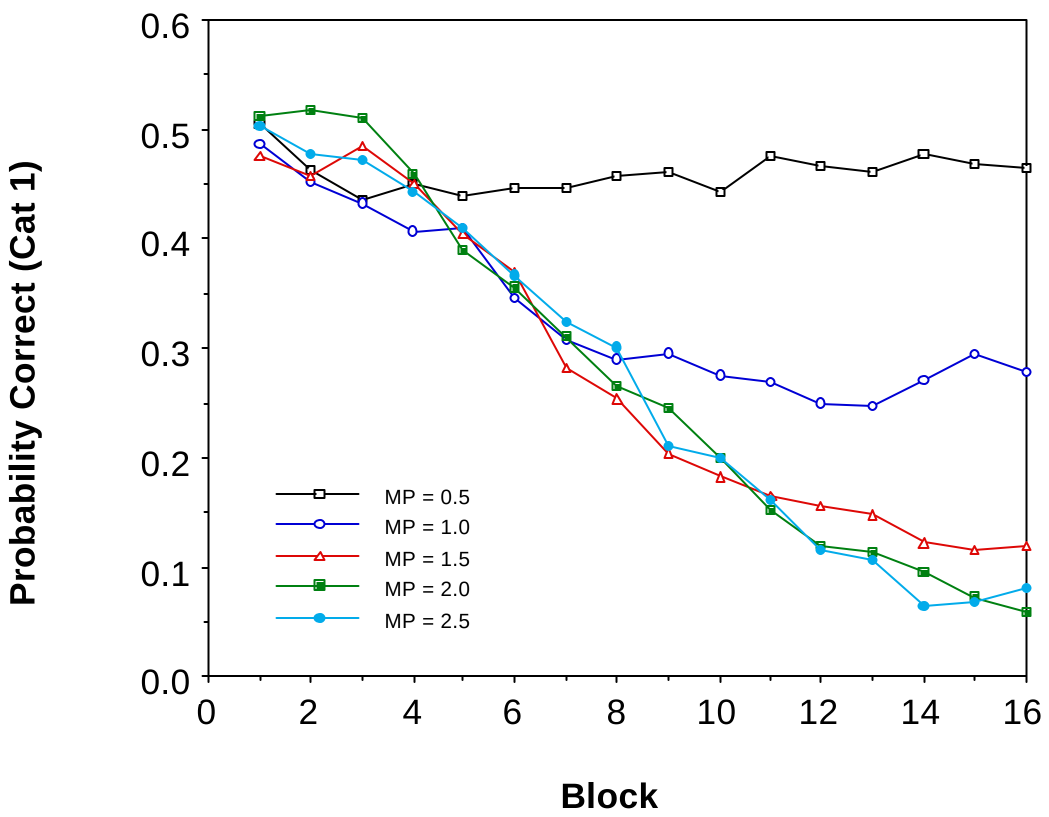
- RT = -1.0 is optimal!
- Seen in other analyses (e.g. Lebiere, 1998; Lebiere & Wallach, 2001)
- Lower RT more compatible with slope of human data but then other parameter (e.g. MP) needs to be adjusted.

Parameter Sensitivity: S



- S = 0.25 optimal for some training amount
- Optimal noise depends upon amount of training; suggests simulated annealing (BM)
- See cognitive arithmetic (Lebiere, 1998)

Parameter Sensitivity: MP



- MP = 1.5 is a good tradeoff of maximum generalization with satisfactory correctness
- As for noise, different values of MP are optimal for various amounts of training.
- Learn similarity with practice?

Modifications

- Fix the bug in the feedback of the transfer phase, leading to no or even positive reinforcement of answers, rather than negative as currently the case.
- RTs for the classification task could be lowered by increasing the prioritization of that task.
- An exponential (non-linear) similarity measure would increase errors for extrapolation stimuli in the transfer phase.
- Blending for grouping effects (e.g. category effect on RT).
- Add a rule learning skill model to make people happy.

Rule Learning

- Could use an existing ACT-R model (Anderson & Betz, 2002), or for that matter implement other models in ACT-R.
- Bases for rule creation:
  - Random spawning of possible rules with utility learning to sort out which are useful (also applicable below).
  - Difference between current problem and retrieved instance as basis for which dimensions matter.
  - Retrieval inversion (with blending?) as memory-based estimate of which dimensions correlate with categories.