Model Evaluation

ACT-R, IMPRINT, and Matlab
Comparisons, Parameter Optimizations, and Opportunities

- 'Unified test problems'
- RADAR
- Parameter optimization
- Radial Basis Functions (RBFs)
- Summarizing observations

Keystroke entry task, and RADAR
Some modeling results
Genetic algorithms / Simulated annealing
Opportunity to remove stochastic 'noise'

Bengt Fornberg
University of Colorado, Boulder
Department of Applied Mathematics

Presentation at annual MURI meeting, held at Carnegie Mellon University, Pittsburgh, August 14, 2009.
Modeling Systems: Designed for:

ACT-R: cognitive modeling
IMPRINT: human and system performances in military tasks
Matlab: science and engineering applications

Key issues highlighted in recent MURI presentations:

Accuracy and Speed comparisons on Keystroke Entry task:
- Accuracies equivalent: Expected: IMPRINT vs. Matlab (near line-by-line conversion, identical algorithm) Less expected: ACT-R vs. IMPRINT (Matlab)
- Speed: 10,000 times advantage with Matlab (on comparable hardware)

Parameter optimizations: Local vs. Global optimization
Genetic Algorithms and Simulated Annealing

New issues:
- RADAR task now implemented in Matlab (no-tone counting only): Speed, Accuracy, Parameter Optimizations
- Ability to extract LARGE amounts of information from relatively little experimental data
- Opportunity with Radial Basis Functions
Unified Test Problems

Tasks implemented in all three systems (ACT-R, IMPRINT, and Matlab) in order to allow most direct guides to key characteristics

1. **Keystroke Data Entry task:**
   Type on a number key pad a series of 4-digit numbers (e.g. 1395) as quickly and accurately as possible when they are presented; terminate with pressing the "Enter" key

2. **RADAR:**
   - TASK, ACT-R: Best, Gonzales, Young, Healy, and Bourne (2007)
   - IMPRINT: Buck-Gengler, Work in progress
   RADAR task combines:
   - Mapping type: Targets and distractors from same or different character sets
   - Load level: Number of items in target set, Number of items to look at to see if target
   - Tone counting: Concurrent secondary task using auditory modality
Codes for modeling the RADAR task:

- The ACT-R and the IMPRINT codes were developed separately.
- The Matlab code (so far, 'no-tone' case only) is mathematically equivalent to the IMPRINT code. Implemented by B. Fornberg and B. Raymond, with help from C. Buck-Gengler.

Two main differences compared to Keystroke Data Entry task:

- Stochastic features enter in the RADAR task in such a way that Matlab's array processing features are no longer practical to apply,
- General programming style of Matlab (shared with Fortran, C/C++) allows for particularly simple and effective code structure; nested loops instead of many interacting modules.

Roughly offsetting advantages/disadvantages - Matlab speed again about 10,000 times faster than ACT-R / IMPRINT on equivalent hardware.

The Matlab code is extraordinarily compact and readable - about 100 lines only (plus about 30 lines for entering parameters)
Example of output of Matlab RADAR model

Means per block for 3 different performance measures (total of 48 data points) used to optimize model performance.

In the sample display to the right:

<table>
<thead>
<tr>
<th>Model_RT</th>
<th>Response times (for hits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6368</td>
<td>0.9076</td>
</tr>
<tr>
<td>0.6246</td>
<td>0.9707</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment_RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6169</td>
</tr>
<tr>
<td>0.6176</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RMSE_RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0497</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model_hits</th>
<th>Proportion of hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9944</td>
<td>0.9889</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment_hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9667</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RMSE_hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0192</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model_FA</th>
<th>Proportion of false alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1722</td>
<td>0.1024</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment_FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1663</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RMSE_FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0532</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0407</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Elapsed time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1642 seconds</td>
</tr>
</tbody>
</table>

Rows: Two Sessions; Columns: Eight Blocks

These show average over: 12 Subjects, each performing 20 Shifts, containing 7 Frames

Very high level of stochastic fluctuations: RMSE = 0.0407. If averaged over 1000 runs: RMSE = 0.0369.
Parameter Optimization

The model has about 30 parameters

We want to optimize the model with respect to 16 parameters in four categories:

1. 4 parameters controlling the decision times during visual search;
2. 4 parameters controlling the probability that a statistical subject will respond that a target has been detected on a trial containing a target;
3. 5 parameters controlling the false alarm rate, as a function of block type;
4. 3 parameters controlling the rate at which statistical subjects learn to avoid false alarms.

Fundamental question:
Do the 48 data points (means of experimental data) provide enough information to separate the influences of the 16 parameters from each other?
Global vs. local Parameter Optimization

2-D Example: Find the GLOBAL maximum value of elevation in the Bryce Canyon amphitheater

- Search for LOCAL maximum is trivial: Start anywhere and steps upwards until one reaches a top.
- Is there anything more effective than exhaustive search to locate a GLOBAL maximum?
- Is this illustration above at all relevant to the task at hand?
Second question first: We will encounter highly irregular functions

We want to minimize the RMSE by searching a space defined by the 16 free parameters.
Below is a plot of RMSE when we vary just 2 of the 16 parameters over their 'reasonable range'.

Analogy not perfect:
- What we see is stochastic simulation noise, which varies from run to run.
- We will be working in a lot more than 2-D.

Can we reduce or eliminate the noise?
- Can we find deterministic equations in terms of means, deviations, etc.? A long story...
  Answer: probably NOT effectively.
- Fit model output with Radial Basis Functions (RBFs), then optimize?
  Answer: Excellent opportunity (novel in the field; will be pursued).
Effective approaches for global optimization

Two major strategies for a much more efficient 'exploration' of the high-dimensional space than exhaustive search:

- Simulated annealing (SA)
- Genetic algorithms (GA)

The main challenges are:

- To explore vast spaces effectively (here 16-D, rather than 2-D in the illustrations)
- To find global minima without getting trapped in any local minima.

Both strategies (SA and GA):

- Borrow their concept from processes in nature (formation of a crystal, and evolution of species, respectively),
- Are effective for functions both without and with 'stochastic noise' present in the data,
- Are available in a Matlab 'tool box'; can be invoked by just 3 - 4 lines of extra programming.

Experience from Unified Test Problem 1 - key stroke entry task:

Both approaches (GA and SA) worked well, but possibly with a slight edge in favor of GA. Hence, we only use GA in the present RADAR task.
Example of RADAR parameter optimization results with GA

The Matlab version of the RADAR model (like the IMPRINT model) has about thirty parameters describing cognitive factors.

- Select a subset of key parameters: We have chosen a total of 16 parameters in 4 categories.
- Optimize in each group separately, in order to improve agreement between model and experiment.

Each optimization is made up of 20 GA runs with population size 40, run through 10-30 generations.
Radial Basis Functions

Topic discussed at length at the Training MURI kickoff meeting, November 4, 2005 (in Washington, DC; well before we had any actual models or data to work with). Notes below from that presentation:

Brief history of RBF

1970  RBF proposed for drawing contour lines on maps from scattered elevation measurements (R.L. Hardy)

1985  Methodology proven to be robust for any number of points in any number of dimensions. Intense work on RBF starts at several centers.

1993  First industrial use of RBF for modeling/data mining (Exxon and Shell)

2000+ Literature exceeds 2000 research papers, 4 books published, Fast computational algorithms emerging, Vast number of applications

What about modeling data in, say, 6-D?

- RBF methodology capable of doing automatically what splines do in 1-D.
- Permits multidimensional visualization
- Can be used for interpolation or regression (to filter out noise)
- Comparatively robust also for extrapolation

Present idea is to re-represent the model output by RBFs before optimization with GA.
Some thoughts about opportunities, and what realistically can be achieved

Stochastic vs. fully deterministic modeling

- If stochastic quantity does not influence subsequent actions (such as eye movement time, response time), we can use distributions functions effectively - no need for 'Monte-Carlo' approach.
- If it influences a chain of further actions that may or may not be taken, one gets in very few steps far too many possibilities for effective non-stochastic modeling. This most certainly becomes best choice.
- Stochastic modeling can be followed up by RBF-based 'de-noising' for improved optimization.

Parameter optimization: Ability to extract VAST amounts of information from relatively little data

In present example, 48 data values - sufficed for extensive optimization (here of 16 parameters - this could easily have been extended). All results were found to be in good agreement with hand derived values.

Opportunity: To introduce and then optimize parameters that are NOT directly measurable. This might allow assessment of cognitive variables that are more abstract or general (e.g. situation awareness, attentional allocation, or motivation).

Computational scaling to much larger / more complex scenarios

We are far from any present hardware (or software) limitations:
- Matlab some 10,000 times faster than ACT-R / IMPRINT on both 'Unified Test Problems'.
- Transfer of code from PC to large-scale supercomputer might offer another factor of 10,000.

Visualization

Very advanced multivariate visualization has been developed for many applications. Attempts should be made to apply these capabilities to psychological data: experimental as well as simulated.