Model Evaluation

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

- 'Unified test problems'
- Computational speeds
- Some modeling results
- Parameter optimization
- Radial Basis Functions (RBFs)
- Summarizing observations

Data Entry, and RADAR
Genetic algorithms / Simulated annealing
Opportunity to remove stochastic 'noise'

Bengt Fornberg
University of Colorado, Boulder
Department of Applied Mathematics

in collaboration with, and presented by

Bill Raymond
University of Colorado, Boulder
Department of Psychology and Neuroscience

Presentation at MURI meeting, held in Denver, April 16, 2010.
Comparison procedure:

<table>
<thead>
<tr>
<th>Modeling Tasks</th>
<th>Computational Systems:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Entry</td>
<td>ACT-R cognitive modeling</td>
</tr>
<tr>
<td>RADAR</td>
<td>IMPRINT human and system performances in military tasks</td>
</tr>
<tr>
<td>Matlab</td>
<td>Matlab science and engineering applications</td>
</tr>
</tbody>
</table>

Key previous results:

- Model fits to Equivalent data: IMPRINT vs. Matlab (since same algorithm)
- Speed: 10,000 times advantage for Matlab over other computational systems
- Parameter optimization: Global optimization of numerous parameters therefore became practical.

New results:

- RBFs accelerate parameter optimization.
- RBFs provide greatly enhanced possibilities for parameter space visualization.

Some comments on the choice of Matlab for comparisons:

- Simplicity of code writing and code optimization (very effective profiling system)
- Advanced multi-parameter optimization techniques available in just few (5-10) lines of extra code
- Ease of porting to parallel computing environments (again, only 5-10 extra lines of code).

There are alternative programming environments to Matlab, and Matlab might not be the fastest. However, if that is the case, our main conclusions would only be demonstrated more strongly.
Optimization using Radial Basis Functions (RBFs)

RBFs discussed at length at the Training MURI kickoff meeting, November 4, 2005 (in Washington, DC).
- RBFs allow fast regularization and evaluation of multivariate functions
- We used RBFs to approximate the objective functions over our models' parameter spaces
- Parameter space optimizations speeded up by factor around 500.

Visualization problematic already for a function of three variables

Varying two parameters at a time around a single point gives three 2-D slices:

Most of the 3-D parameter space has not been explored with these slices.

Specific application:

In the IMPRINT/Matlab model of the Data Entry task, we have five main parameters to be varied in order to optimize by minimizing the objective function (RMSE of difference between model and experiment)

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(cognitive)</th>
<th>(physical)</th>
<th>(rep learn)</th>
<th>(left penalty)</th>
<th>(fatigue)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognitive learning</td>
<td>[-0.20, 0.00]</td>
<td>[-0.10, 0.00]</td>
<td>[0.00, 0.30]</td>
<td>[1.00, 1.50]</td>
<td>[0.00, 0.10]</td>
</tr>
<tr>
<td>Motoric learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repetition priming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left-hand penalty</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cognitive slowdown</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

'Reasonable ranges':

- Cognitive learning: [-0.20, 0.00]
- Motoric learning: [-0.10, 0.00]
- Repetition priming: [0.00, 0.30]
- Left-hand penalty: [1.00, 1.50]
- Cognitive slowdown: [0.00, 0.10]

'Hand derived' values:

- Cognitive learning: -0.045
- Motoric learning: -0.015
- Repetition priming: 0.050
- Left-hand penalty: 1.125
- Cognitive slowdown: 0.0125
In order to visualize the 5-D parameter space, we can vary two parameters at a time from hand-derived values, which gives 10 slices.

Partial visualization over the 5-D parameter space using the original Data Entry model

Computational speed is a critical issue:

The figures display a total of 4,410 surface points.

Each surface point takes 48 minutes in IMPRINT but only 0.17 seconds in Matlab.

This is still completely insufficient to illustrate the objective function in the full 5-D parameter space.
The RBF approximation can be used to regularize and evaluate the objective function.

Partial visualization using RBF approximation of the objective function:

Once the RBF approximation has been created, each surface point now takes 0.0003 seconds to evaluate.

Stochastic noise has been eliminated.

Speed gain in evaluating objective function:
Over Matlab: 560
Over IMPRINT: 5,000,000

This speed gain is crucial for improved visualization.
Effective visualization requires fast and noise-free evaluations, as offered by RBFs

Illustration of how a 2-variate function can be visualized in 1-D:

'Slider' (bottom right) can be dragged by mouse to display different slices (dashed in left plot)

Similarly, functions in $n$ - $D$ can be represented by RBFs and shown as 2 - D surfaces with $n$ - 2 'sliders'

Novel way to visually explore multi-variate data - It overcomes the traditional limitation of no higher than 2-D data when displaying on a 2-D computer screen.
Some concluding thoughts and future opportunities

Computational scaling to much larger and more complex scenarios
We have realized major improvements in computational efficiencies:
- Matlab can be some 10,000 times faster than ACT-R / IMPRINT.
- RBF approximations over the parameter space can both remove stochastic 'noise' and add a further factor of 100 - 1,000 in evaluation speed.

Opportunities: We are still far from any present hardware (or software) limitations:
- Transfer of code from PC to large-scale supercomputer might offer another factor of 10,000. In some programming environments, this can be done with addition of just a few lines of code.
- Numerical tasks should, through appropriate interfaces, be handled by systems specifically designed for them.

Parameter optimization
- It has been found to be possible to optimize many parameters using relatively little data.
- Although all parameter optimization results (Data Entry - 5 parameters and RADAR - 16 parameters) were found to be fairly close to hand-derived values, optimization can now be done in a much more automated way.

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

Visualization
Opportunities:
- 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.
- RBFs will be particularly useful in dynamic visualization thanks to their ability to quickly render noise-free versions of multivariate functions.