Multilevel Emulation

Jonathan Cumming

May 24, 2007
Introduction

- Single level: we wish to emulate a complex simulator $f(x)$ in order to solve some interesting problems.
- Multi-level: instead of having a single simulator, we have two or more versions of $f(x)$, but we still want to solve the same problems!
- Different simulator versions arise from:
  - Grid complexity
  - Solver complexity
  - Complexity of the underlying mathematics
  - Other sources, e.g. stochastic models
- Coarse simulators quick but crude; finer simulators are slower but more accurate.
- The simulators all model the same system and so will be closely related (in some way...)
Multiscale Models

- We have a coarse model $f_0(x)$, and a fine model $f_1(x)$ and we want to use many runs of $f_0$ to glean some information about $f_1$, and use that information to inform how we emulate $f_1$.

- General strategy:
  1. **Coarse Model** – Standard emulation of $f_0$
  2. **Coarse-to-Fine I** – Try to assess the nature of the differences between $f_0$ and $f_1$
  3. **History Matching** – Using information from previous steps, perform a conservative history match (possibly repeat)
  4. **Coarse-to-Fine II** – Re-assess coarse-fine differences; use coarse emulator to inform beliefs about fine emulator
  5. **Fine Model** – Finally, use the fine model to solve some problems
The Coarse Model

- ‘True’ system value: \( y = f_1(x) + \eta(x) \)
- Coarse–fine: \( f_1(x) = f_0(x) + e \)
- Observational data: \( z = y + \gamma \)
- \( f_0 \) is quick to evaluate so we can generate a large space-filling design and do many evaluations
- Screen outputs – Using the model runs we can identify important outputs
- Emulate each selected output – stepwise OLS regression fitting with a final re-fit by GLS/maximum likelihood

\[
f_0(x) = \beta_0 g(x_A) + \epsilon_0(x_A) + \delta(x)
\]

- Diagnostics – check that the emulators are adequate
The Model

- Reservoir is a subterranean region of rock containing oil, gas and water. These liquids are pumped out over time, changing pressures and distribution of hydrocarbons.
- Model is based on a $48 \times 26 \times 50$ grid.
- 20 input variables – mainly multipliers for geological properties.
- Output variables - time series for each well in the reservoir.
- Fine model – 50 layers, 20mins per run; coarse model – 2 layers, 30sec per run.
A Minor Detour – Screening Outputs

- The hydrocarbon reservoir model produces a huge number of outputs
- We can’t emulate everything so we want to select a good subset to focus on
- Use PVs – select the variable which has maximum $\sum r_{ij}^2$, then eliminate the effects of selected variables by taking partial correlation and iterate
- Advantages over PCA:
  - PCA still needs all the variables! – problematic for history matching; involves a complex model discrepancy specification
  - Interpretation - what does it mean for $x_1$ to be a key input in the emulator of $0.2Y_1 - 1.7Y_2 + 0.8Y_3$?
## A Minor Detour – Results

<table>
<thead>
<tr>
<th>T</th>
<th>1 PC1</th>
<th>2 PC2</th>
<th>3 PC3</th>
<th>4 PC4</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 PVs</td>
<td>W4.pre</td>
<td>W1.oilrt</td>
<td>W3.bhp</td>
<td>W2.wattot</td>
</tr>
<tr>
<td>% Var</td>
<td>78 79</td>
<td>9 9</td>
<td>3 4</td>
<td>2 2</td>
</tr>
<tr>
<td>$R^2$</td>
<td>96 98</td>
<td>54 52</td>
<td>64 17</td>
<td>74 76</td>
</tr>
<tr>
<td>36 PVs</td>
<td>W4.bhp</td>
<td>W2.oilrt</td>
<td>W4.wct</td>
<td>W3.bhp</td>
</tr>
<tr>
<td>% Var</td>
<td>72 73</td>
<td>10 11</td>
<td>5 5</td>
<td>3 3</td>
</tr>
<tr>
<td>$R^2$</td>
<td>93 93</td>
<td>47 8</td>
<td>32 22</td>
<td>27 13</td>
</tr>
<tr>
<td>55 PVs</td>
<td>WINJ3.resrt</td>
<td>W2.oiltot</td>
<td>WINJ4.watrt</td>
<td>W1.watrt</td>
</tr>
<tr>
<td>% Var</td>
<td>53 56</td>
<td>14 14</td>
<td>11 11</td>
<td>5 6</td>
</tr>
<tr>
<td>$R^2$</td>
<td>88 92</td>
<td>97 78</td>
<td>75 73</td>
<td>46 48</td>
</tr>
</tbody>
</table>
History Matching

- First, do some runs of the coarse and fine simulators at the same location to determine \( E[f_0(x) - f_1(x)]^2 \)
- We want to rule out values, \( x \), of the parameter space where it is unlikely that \( f_1(x) \) is near true system value (or it’s surrogate measurement, \( z \)). We construct an *implausibility* measure:

\[
I(x) = \frac{(z - E[f_1(x)])^2}{\text{Var}[z - f_1(x)]}
\]

- Evaluating this function over a reasonably fine grid in our input space allows us to identify regions in which it is infeasible for the simulator to match the data.
History Matching II
We finally have an emulator for our coarse simulator and a notion of the behaviour of the difference between the two simulators.

All we need now to move to the fine emulator is to specify our beliefs linking the key components of the emulators, e.g. coefficients

$$\beta_0 = \Lambda_0 \beta^* + \xi_0, \quad \beta_1 = \Lambda_1 \beta^* + \xi_1$$

Combine the information from our coarse emulator with our coarse-fine differences to obtain a partial prior specification for our fine emulator.

All that remains is to design and evaluate some additional runs of the fine simulator and perform a Bayes linear update.
We finally have an emulator for our fine simulator!

Now we can perform diagnostic tests on the fine simulator to make sure that it holds up.

We can also seek to determine whether we have missed any variables that are active on the fine model, but were inactive in our coarse model.

Being satisfied with the fine emulator, we can return to history matching and perform a more definitive analysis.
In summation

- In order to build a multi-level emulator we need
  - To apply standard emulation methods to the coarse simulator
  - Knowledge of the behaviour of the differences between the two simulators
  - A belief specification describing the linkage of the two simulators
  - Design, diagnostic, screening and history matching methodologies
- Currently, we have implemented a simplified version of many aspects of this methodology