A Key Deliverable!

Some elements adapted from H. Taylor (2001)
Recall: Dealing with Data Gradients

- Often we don’t have reliable information on some parameters, but do have other data
  - Some parameters may not be observable, but some closely related observable data is available
  - Sometimes the data doesn’t have the detailed breakdown needed to specifically address one parameter
    - Available data could specify sum of a bunch of flows or stocks
    - Available data could specify some function of several quantities in the model (e.g. prevalence)
- Some parameters may implicitly capture a large set of factors not explicitly represented in model
- There are two big ways of dealing with this: manually “backing out”, and automated calibration
Recall: Calibration: “Triangulating” from Diverse Data Sources

- Calibration involves “tuning” values of less well known parameters to best match observed data
  - Often try to match against *many* time series or pieces of data at once
  - Idea is trying to get the software to answer the question: “What must these (less known) parameters be in order to explain all these different sources of data I see”

- Observed data can correspond to complex combination of model variables, and exhibit “emergence”

- Frequently we learn from this that our model structure just can’t produce the patterns!
Recall: Calibration: A Bit of the How

- Calibration uses a (global) optimization algorithm to try to adjust unknown parameters so that it automatically matches an arbitrarily large set of data.
- The data (often in the form of time series) forms constraints on the calibration.
- The optimization algorithm will run the model many (minimally, thousands, typically 100K or more) times to find the “best” match for all of the data.
Recall: Required Information for Calibration

• Specification of what to match (and how much to care about each attempted match)
  – Involves an “error function” ( “penalty function”, “energy function”) that specifies “how far off we are” for a given run (how good the fit is)
  – Alternative: specify “payoff function” ( “objective function”)

• A statement of what parameters to vary, and over what range to vary them (the “parameter space”)

• Characteristics of desired tuning algorithm
  – Single starting point of search?
Recall: Example Global Optimization Algorithm

• Starts at random position, tries to improve match (minimize error) by
  – Adjusting parameters
  – Running Model
  – Recording error function

• Keeps on improving until reaches “local minimum” in error of fit
  – May add some randomness to knock out of local minima
Running Calibrations in Vensim:
(Under Model/Simulate Commands)
Payoff Definition

Payoff Definition. Edit the filename to save changes to a different control file.

Filename: GDMCalibration.vpd

Type: Calibration

Payoff Elements:
Total Weighted Absolute Value of All Discrepancy for Current Time/-1

Variable

Compare to

Weight

The weight should be positive for calibration. For policy optimizations, use a positive number when more is better and a negative number when less is better.

Ok

Cancel
The Pieces of the Elephant
Example Model of Underlying Process & Time Series It Must Match
Single Model Matches Many Data Sources
Example: Iteration & Calibration

From Sterman
SEIR Model vs. Data, Taiwan

Cumulative Cases, No Behavioral Response

Adapted from Sterman

From Sterman
Expanding the Boundary: Behavioral Feedbacks

From Sterman
Model vs. Data with Behavioral Feedback

Cumulative Cases

Adapted from Sterman
Pieces of the Elephant: STI
Hands on Model Use Ahead

Load Sample Model:
SIR Agent Based Calibration
(Via “Sample Models” under “Help” Menu)
An Optimization Experiment in AnyLogic

- Stops after 500 optimization iterations
- Varying these parameters
- Stops after best objective ceases to significantly improve
- **Caveat Modelor**: May prematurely terminate the optimization
Defining a Payoff Function

Caveat: Non-Analytic, Non-Concave

Computing discrepancy between (historic & model values at this point during the run)
How to interpolate ("fill in") between data points.
Stochastics in Agent-Based Models

• Recall that ABMs typically exhibit significant stochastics
  – Event timing within & outside of agents
  – Inter-agent interactions

• When calibrating an ABM, we wish to avoid attributing a good match to a particular set of parameter values simply due to chance

• To reliably assess fit of a given set of parameters, we need to repeatedly run model realizations
  – We can take the mean fit of these realizations
Distinction

• Replication/”Run”: One realization
  – Particular random number seed
• Iteration: Evaluation of a particular parameter set
  – This can contain many realizations (“replications”)
• Confusingly, the term “simulation” appears to sometimes be used for either of the above
Populating the Appropriate Datasets

Populates historic data up front from table fn

These datasets are within the experiment

Persist beyond the simulation

Saves away best simulation

Within in iteration

Retaining the Current value

After the realization (Simulation run)
Running Calibration in AnyLogic

Best payoff (objective) yet reached (lower is better)

Values of parameters being calibrated at best calibration thus far
Optimization Constraints – Tests on Legitimacy of Parameter Values
Optimization Requirements – Tests on Emergent results to Sense Validity
Enabling Multiple Realizations ("Replications","Runs") per Iteration
Fixed Number of Replications per Iteration

Specifies stopping Condition once minimum replications have been run. Indicates that the X% confidence interval around the mean is within “Error percent” of the iteration mean obtained as of the most recent replication.
Example

\[
\bar{x}_5 \left( 1 + \frac{e}{100} \right) \\
\bar{x}_5 = \sum_{r=1}^{5} \text{payoff}_r \\
\bar{x}_5 \left( 1 - \frac{e}{100} \right)
\]

\[
\bar{x}_{10} \left( 1 + \frac{e}{100} \right) \\
\bar{x}_{10} = \sum_{r=1}^{10} \text{payoff}_r \\
\bar{x}_{10} \left( 1 - \frac{e}{100} \right)
\]

\[
\bar{x}_{40} \left( 1 + \frac{e}{100} \right) \\
\bar{x}_{40} = \sum_{r=1}^{40} \text{payoff}_r \\
\bar{x}_{40} \left( 1 - \frac{e}{100} \right)
\]

After 5 replications

After 10 replications

After 40 replications

Terminates

x % (e.g. 80%) confidence
Interval for sample mean (average) of replications to this point

Minimum and maximum Observed values from replications

x % (e.g. 80%) confidence
Interval for sample mean (average) of replications to this point

Minimum and maximum Observed values from replications

x % (e.g. 80%) confidence
Interval for sample mean (average) of replications to this point

Minimum and maximum Observed values from replications
Automatic Throttling of Replications Based on Empirical Fractiles for the Average of the Differences between Best and Current
Enabling Random Variation Between Realizations (“Replications”)
Understanding Replications:
Report Results for Each Replication!
During First Several Realizations ("Replications", "Runs"), No Results Appear
Report on Iteration 1 Appears after a Count of Runs Equal to Replications per Iteration

Reports best payoff (objective) yet reached (lower is better), but from where did this number Come?
The reported payoff for the iteration is the average of the payoffs for each replication within the replication.
Average of Results for Replications is the Reported Score for the Iteration!

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>For this pa</td>
<td>3324</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>For this pa</td>
<td>3363</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>For this pa</td>
<td>8866</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>For this pa</td>
<td>2052</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>For this pa</td>
<td>2447</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>For this pa</td>
<td>3552</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>For this pa</td>
<td>6079</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>For this pa</td>
<td>6082</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>For this pa</td>
<td>4775</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>For this pa</td>
<td>2866</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>For this pa</td>
<td>4340.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In cell D14, the formula =AVERAGE(D3:D12) calculates the average of the values in columns D from row 3 to row 12.