www.surreycc.gov.uk

SURREY
COUNTY COUNCIL

Making Surrey a better place

Distribution Calibration

A case study using OmniTRANS

Richard Sweet

OmniTRANS user group 28 May 2010

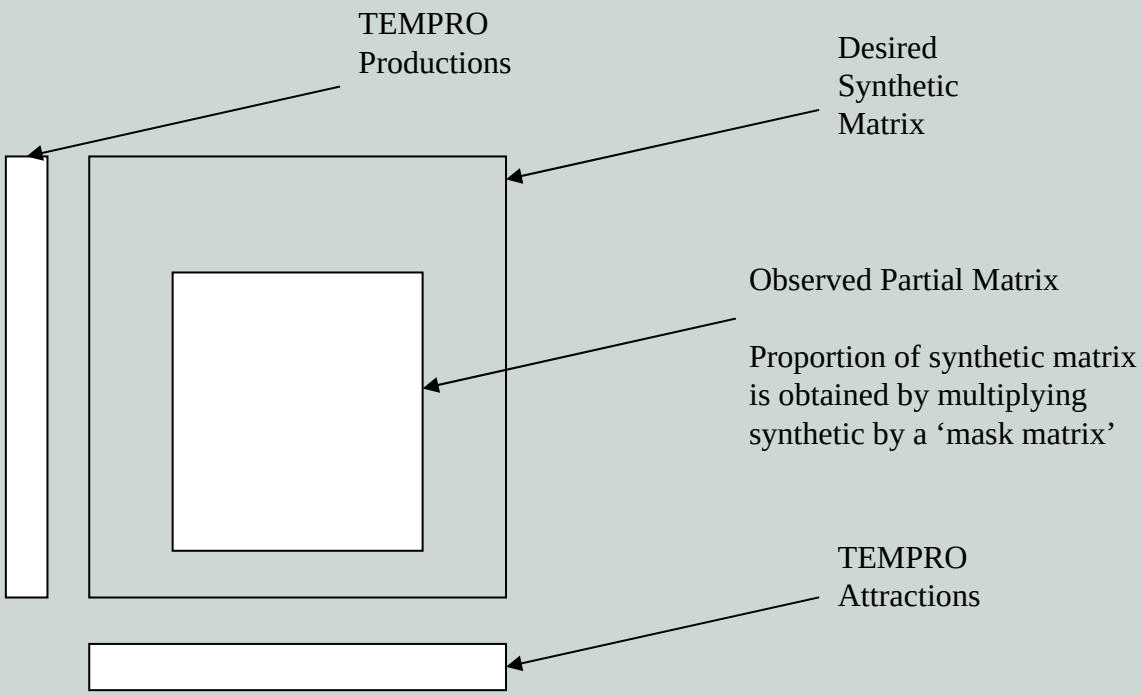
Calibration criteria

- Observed data:
  Partial matrix by purpose, time and mode derived from RSIs
  (preparation described at OT user group meeting Nov 2008)

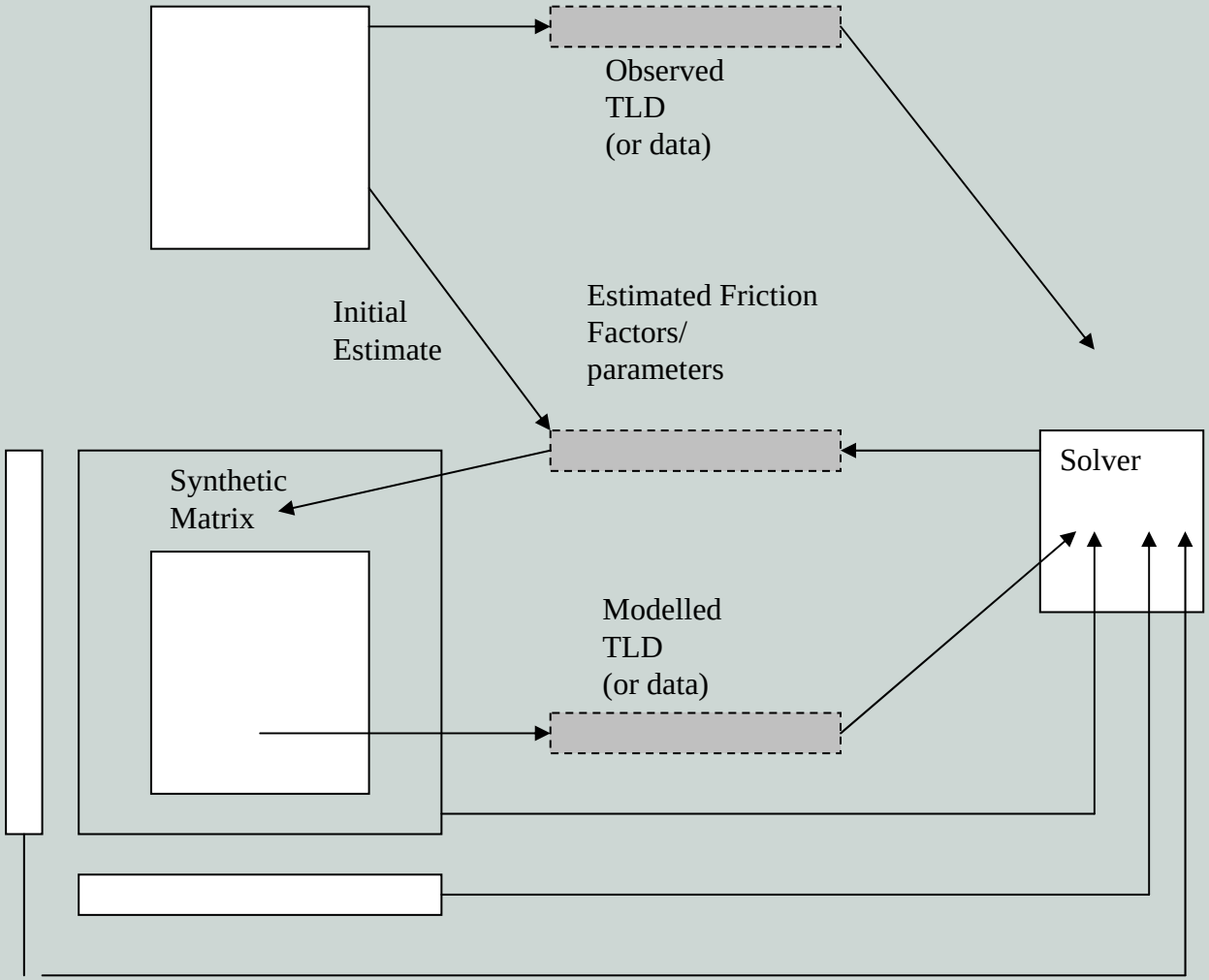
- Origin and destination (or Production and Attraction) NTEM data
  (for whole distribution matrix) from TEMPRO

- Aim: to obtain distribution parameters that when used with the given Origin and destinations, provide the best match to the observed data – in this case a partial matrix.

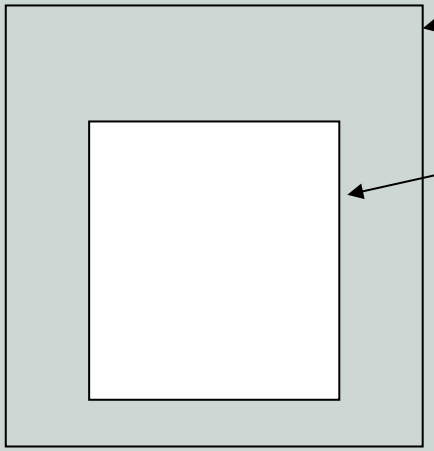
Distribution data schematic



Calibration schematic



Modelled Partial matrix



Desired Synthetic Matrix

Modelled Partial Matrix

Proportion of synthetic matrix is obtained by multiplying synthetic by a ‘mask matrix’

Mask matrix is obtained by either

• Calculating a screenline (intercept) matrix over all RSI sites

• Skimming with a link attribute of 1 for links containing RSI sites, and 0 otherwise.

The two methods should give identical results

Solver

- Aim: to obtain distribution parameters that when used with the given Origin and destinations, provide the best match to the observed data – in this case a partial matrix.

- A variety of solution methods may be used

- Preliminary calibration performed by matching the modelled Trip Length Distribution with the observed (both for the partial matrix) by calculation Friction Factors FF^k for each interval k of the TLD.

FF Solver

- Uses method of tri-proportion – an extended version of Furness
- Simple and quick to implement in Ruby ☺
- Can represent an arbitrary functional form ☺
- Does not give statistical estimates ☹
- Constrained to the aggregate intervals chosen for the TLD ☹

$$t_{ij} = O_i A_i D_j B_j FF^k, \text{where k} = \left| c_{ij} / \text{cost\_interval} \right|$$

$$B_j = D_j / \sum_i t_{ij}, \quad A_i = O_i / \sum_j t_{ij}, \quad FF^k = ObsTLD^k / EstTLD^k$$

- Could use maximum likelihood or least squares
  - gives statistical estimates and allows disaggregate data

Conclusions

- Calibration should model the observed data, but within the framework of the forecast data structure
- Possible to implement this in Ruby efficiently
- OmniTRANS is good at managing the data structures
- OmniTRANS is poor at calculating screenline matrix over many sites and cannot yet calculate user specified skims

The Future

- Maximum LL solver might be implemented reasonably efficiently in Ruby using the supplied matrix library
- Might be able to do this efficiently using a ‘contraction mapping’ to reduce the dimensionality of the attractions (columns)
- This method would iterate between classical Newton iterations over the parameters and Furness over the attractions