Comparing three penalty functions – Cross Entropy, Quadratic and Linear Loss – in SAM balancing and splitting applications

Wolfgang Britz, University Bonn
Presentation at the 23rd Annual Conference on Global Economic Analysis "Global Economic Analysis Beyond 2020"
Why?

• Old interest in comparing cross-entropy and quadratic loss
  – differences with regard to estimates
  – solution behavior: accuracy, speed, suitable solvers

• CGEBox comprises routine to split larger data bases (e.g. more details for agri-food, 57x40 => 80x40), requires stable and performant solution:
  – what penalty function?
  – what solvers work best?
Two parts

1. Small didactic SAM balancing example with controlled data generation: are cross entropy (CE) and quadratic loss (QL) different? Compare to linear loss (LL, minimizing absolute differences)

2. Split exercises of larger GTAP data sets: compare penalty functions and solvers, with regard to accuracy and solution time

=> Recommendations for similar exercises
The SAM balancing example

• Monte-Carlo exercise with 100 Draws:
  – 30x30 SAM, entries from n~(0,100), cutoff at 1
  – use RAS remove imbalances
  – add weight noise n~(0,σ²), σ = {0.1,0.5,1,2,5}
  – use QL, CE (with different assumed normal errors) and LL to balance the distorted SAM
  – w/wo non-negativity of estimates imposed

• Coded in GAMS, 100x5x3x2 = 3.000 problems solved with CONOPT4
SAM balancing example, key findings

- CE and QL give identical results
- QL ~100 times faster than CE, LL slightly faster than QL
- Assuming different (but uniform) variances a priori for CE has no impact
- QL and CE give better estimates $\iff$ assume correct distribution of error terms
- QL maximizes the posterior log likelihood under normally distributed errors (Highest Posterior Density), seems not really different from entropy criterion here
- Sign preserving approach does not change much
The split example

- Empirical example splitting differently detailed GTAP data sets
  - Split „Other Food Processing“ to „concentrate feed“ and „rest“ based on data from FABIO MRIO
  - Data: 35x10, 47x10, 57x10, 57x24, filtered to 0.01%, 0.001%, 0.0001% => 12 test cases
  - True distribution of error terms unknown: QL and CE not by construction “right” choice
  - Same standard error (QL,CE) or same relative absolute error (LL) assumed, i.e. **no weights for split factors**
The split example

• **Constraints** of problem all **linear**:
  – Various **exhaustion** and **market balancing** conditions
  – Includes **bounds**, not only non-negativity conditions on estimates
  – Includes also **inequalities**, e.g. ensure that factor subsidies do not exceed value at market prices

• **Three penalty functions**: CE, QL and LL, tested under **three solvers**: CONOPT4, GUROBI, CPLEX

• RAS no candidate (bounds, inequalities ...)
The solvers

• CPLEX and GUROBI:
  – specialized solvers for MIP and LP/QP problems
  – free academic licenses, otherwise quite expensive
  – tuned for massive parallelism

• CONOPT4:
  – general purpose gradient based solver
  – moderate license fees, not free for academic use
  – workhorse in CGEBox to solve CGE model and various other problems, sole candidate for CE
  – some parallelism
Split example: some observations

• CE supports imply upper and lower bounds on estimates, here large *a priori* standard error required for feasibility

• Problems comprise between ~25k and ~100k non-zeros: moderately sized compared to some other split exercises

• All solvers had at least some cases where they did not find a fully optimal solution

• Some tweaks such as turning scaling off required to make solvers work on certain model classes
Split examples: Scaling behavior

QL and CE in CONOPT4 require far more time (close to 1.000 instead of 1 sec)

CPLEXD scales extremely well for LP and QP case
Average relative diff to best solver/algorithm combination

Specialized LP/QP solvers clearly outperform the general purpose NLP solver for LL/QP and offer extremely high accuracy for LL.

CONOPT4 requires 500 times longer in average on QP

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Max. Imbalance</th>
<th>Sum of Imbalances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Solver</td>
<td>Total</td>
<td>Solver</td>
</tr>
<tr>
<td>LL</td>
<td>CPLEXD</td>
<td>0,81</td>
<td>0,46</td>
</tr>
<tr>
<td>LL</td>
<td>GUROBI</td>
<td>0,95</td>
<td>0,16</td>
</tr>
<tr>
<td>LL</td>
<td>CONOPT4</td>
<td>5,16</td>
<td>0,94</td>
</tr>
<tr>
<td>QL</td>
<td>CPLEXD</td>
<td>1,32</td>
<td>2,18</td>
</tr>
<tr>
<td>QL</td>
<td>GUROBI</td>
<td>6,90</td>
<td>12,27</td>
</tr>
<tr>
<td>QL</td>
<td>CONOPT4</td>
<td>505</td>
<td>26,83</td>
</tr>
<tr>
<td>CE</td>
<td>CONOPT4</td>
<td>466</td>
<td>24,81</td>
</tr>
</tbody>
</table>
Take home messages

• CE and QL give identical results under normally distributed errors, QL easier to code and faster
• Specialized LP/QP solver outperform by far general purpose solvers on LP/QP problems
• If errors not known to be normally distributed, LL good choice: high accuracy and extremely fast to solve (also with NLP solvers)
• Compared to RAS, constrained optimization gives far more control (bounds, inequalities)
And finally ...

- **CGEBox** is open-source and open-access
  https://www.ilr.uni-bonn.de/em/rsrch/cgebox/cgebox_e.htm
  (including code for filter and split)

- CGEBox download includes the FABIO database by Bruckner et al. to split up GTAP to
  more agri-food detail

- Code for SAM balancing available from the author for own experiments