Fitting generalized linear models when the dataset exceeds available memory

Joseph K. Canner, MHS, Krisztian Sebestyen, MS
Johns Hopkins School of Medicine, Department of Surgery, Baltimore, MD

**Introduction**
- Random Access Memory (RAM) capacity has increased and RAM prices have decreased since Stata was first released.
- However, increases in the size of data sets can still exceed available memory, especially on personal laptops and desktops.
- There is a need for statistical tools that can read small chunks of data from disk, perform calculations on those chunks, accumulate intermediate results, and produce final results equivalent to those obtained if the data were all in memory.
- Mathematical methods have been available for many years to fit a generalized linear model (GLM) by updating the Q-R or Cholesky decomposition matrices with small chunks of data.
- Thomas Lumley’s R command bigglm uses Fortran functions published by Alan J. Miller in 1992 as Algorithm AS 274.

**Syntax**
Simplified version of Stata GLM, just need to specify chunk size:
```
syntax namelist using/ [, CHUNKsize(integer 5000)]
link(string) family(string) VQuasi(string)
```

**Flowchart**
- Reads Stata and ODBC
- Family options:
  - Gaussian
  - Poisson
  - Gamma
  - Inverse Gaussian
  - Negative Binomial
  - Power
  - Quasi-<family>
- Link options:
  - Identity
  - Log
  - Logit
  - Probit
  - Negative binomial
  - Inverse
  - Square root
  - Power

**Challenges**
- Stata use in x/y using <filename> is not very efficient for reading chunks; had to read in data using Mata (but still not as fast as use <filename>)
- Very slow: one full pass through the data set for each iteration and model also has a hard time converging (perhaps could use a higher convergence tolerance for model-building?)
- Converting Fortran to Mata (particularly instances where procedure parameters that are vectors are passed by reference with implicit size)

**Benchmarks**

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset Size (N)</th>
<th>Chunk Size (N)</th>
<th>Stata bigglm</th>
<th>Stata glm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logit</td>
<td>300,000</td>
<td>5,000</td>
<td>23.94 sec</td>
<td>0.45 sec</td>
</tr>
<tr>
<td>Poisson</td>
<td>14,319,948 (≈ 4.6 GB RAM)</td>
<td>1,000,000</td>
<td>10,870.15 sec</td>
<td>214.75 sec</td>
</tr>
<tr>
<td>Poisson</td>
<td>300,000</td>
<td>5,000</td>
<td>15.60 sec</td>
<td>0.54 sec</td>
</tr>
<tr>
<td>Poisson</td>
<td>14,319,948</td>
<td>1,000,000</td>
<td>7,264.62 sec</td>
<td>402.12 sec</td>
</tr>
</tbody>
</table>

**Next Steps**
- Automatic missing data handling
- Thoroughly test all family and link combinations
- Make ODBC option more general
- Use C for chunk input?
- Explore other methods with different tradeoffs between speed and accuracy

**Acknowledgements**
Richard Gates (StataCorp) provided some code which informed the early stages. This code was provided in answer to a Statalist question in 2005: “data set larger than RAM” (https://www.stata.com/statalist/archive/2005-11/msg00374.html)