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.**

Reference

Algorithm AS 274

Least Squares Routines to Supplement those of Gentleman

By Alan J. Miller†

CSIRO Division of Mathematics and Statistics, Melbourne, Australia

(Received November 1988. Final revision June 1991)

Source: *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 41, No. 2 (1992), pp. 458-478

Published by: Wiley for the Royal Statistical Society

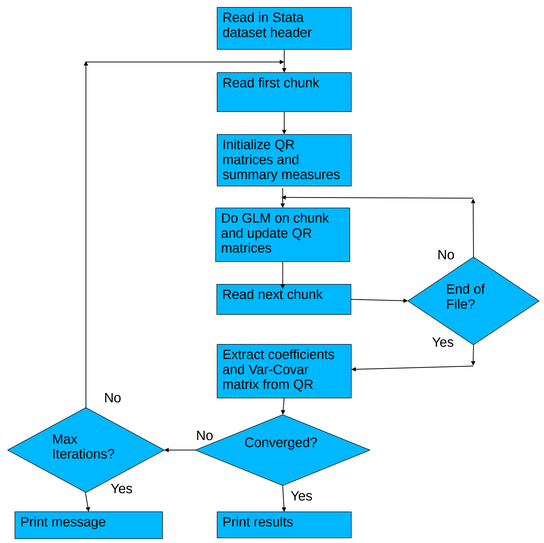
Stable URL: https://www.jstor.org/stable/2347583

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



Features

- **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

Time to complete a GLM with two independent variables using 2013 Nationwide Readmission Database (378 bytes per observation)

Model	Dataset Size (N)	Chunk Size (N)	Stata bigglm[1]	Stata glm
Logit	30,000	5,000	23.94 sec	0.45 sec
	300,000	100,000	227.42 sec	3.82 sec
	14,319,948 (5.4 GB RAM)	1,000,000[2,3]	10,870.15 sec	214.75 sec
Poisson	30,000	5,000	15.60 sec	0.54 sec
	300,000	100,000	152.48 sec	4.86 sec
	14,319,948	1,000,000[2,3]	7,264.62 sec	402.12 sec

1. Tolerance 10^{-8} unless otherwise noted
2. Tolerance increased to 10^{-6} to achieve convergence before max iterations reached
3. Also tested chunk sizes of 100K, 500K, 2000K, and 4000K, with similar results

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)