GEEs in R

Lee McDaniel, Nick Henderson, Paul Rathouz

10/24/2012

Longitudinal data

Characteristics:

- Take multiple observations of K patients
- Call observations on one patient a cluster
- Correlation within clusters
- Independence between clusters

Example (from some website):

Student Name	2001 Score	2002 Score	2003 Score	2004 Score
Mike	339	350	361	366
Jasmine	332	343	350	351
Thomas	360	380	400	420

Generalized estimating equations: setup

K multivariate observations of responses
$Y_i = (Y_{i1}, \ldots, Y_{it}, \ldots, Y_{in_i})$
Let $E(Y_{it}) = \mu_{it}$.
Let $g(\cdot)$ be a monotone link function and

$$g(\mu_{it}) = \eta_{it} = X_{it}^T \beta.$$

Also assume

$$\mathrm{var}(Y_{it}) = \phi a_{it} = \phi a(\mu_{it}).$$

Link function options in R: identity, log, exponential, logit, probit, and so on.

We don't want to be limited by the link function options!

Setup, continued

First, denote $\boldsymbol{\mu}_i = (\mu_{i1}, \ldots, \mu_{in_i})^T$.
Next, denote $A_i = \text{diag}(a(\boldsymbol{\mu}_i))$
Finally, let

$$V_i = \phi A_i^{\frac{1}{2}} R_i(\boldsymbol{\alpha}) A_i^{\frac{1}{2}}$$

where $R_i(\boldsymbol{\alpha})$ is a working correlation matrix.

Independence structure

All within cluster observations are independent, as well as all between cluster observations

$$R_i(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exchangeable structure

Correlation between any two observations is always α

$$R_i(\alpha) = \begin{bmatrix} 1 & \alpha & \alpha & \alpha \\ \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha \\ \alpha & \alpha & \alpha & 1 \end{bmatrix}$$

AR-1 structure

Correlation between any two observations is $\alpha^{|i-j|}$

$$R_i(\alpha) = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 \\ \alpha & 1 & \alpha & \alpha^2 \\ \alpha^2 & \alpha & 1 & \alpha \\ \alpha^3 & \alpha^2 & \alpha & 1 \end{bmatrix}$$

M-dependence structure

if $M = 1$, it will look like

$$R_i(\alpha) = \begin{bmatrix} 1 & \alpha & 0 & 0 \\ \alpha & 1 & \alpha & 0 \\ 0 & \alpha & 1 & \alpha \\ 0 & 0 & \alpha & 1 \end{bmatrix}$$

If $M = 2$, then

$$R_i(\alpha) = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & 0 \\ \alpha_1 & 1 & \alpha_1 & \alpha_2 \\ \alpha_2 & \alpha_1 & 1 & \alpha_1 \\ 0 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}$$

Unstructured structure

Each entry is allowed to be different (still symmetric)

$$R_i(\alpha) = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} \\ \alpha_{14} & \alpha_{24} & \alpha_{34} & 1 \end{bmatrix}$$

GEEs: solutions

Estimating equations are:

$$\sum_{i=1}^{K} X_i^T \Delta_i A_i V_i^{-1} S_i = 0 \tag{1}$$

where

$$\Delta_i = \text{diag}\left(\frac{dg^{-1}(\boldsymbol{\eta}_i)}{d\boldsymbol{\eta}_i}\right)$$

and

$$S_i = Y_i - \mu_i.$$

In the future, let

$$D_i = A_i \Delta_i X_i$$

Example: linear regression

Consider the case: $g(x) = x$, $a(x) = 1$, $R_i(\alpha) = I$

This corresponds to plain vanilla linear regression

Then, $\Delta_i = A_i = I$ and $V_i = \phi I$

So the estimating equation becomes

$$\sum_{i=1}^{K} X_i^T \Delta_i A_i V_i^{-1} S_i = \sum_{i=1}^{K} \sum_{t=1}^{n_i} X_{it}^T (\phi I)^{-1} (Y_{it} - X_{it}\beta) = 0$$

A little algebra and putting everything in matrix notation gives

$$X^T X \beta = X^T Y$$

Solutions, continued

Liang and Zeger recommend calculating by

$$\beta^{(k+1)} = \beta^{(k)} - \left\{ \sum_{i=1}^{K} D_i^T V_i^{-1} D_i \right\}^{-1} \left\{ \sum_{i=1}^{K} D_i^T V_i^{-1} S_i \right\} \qquad (2)$$

with a robust variance estimate of

$$\left\{ \sum_{i=1}^{K} D_i^T V_i^{-1} D_i \right\}^{-1} \left\{ \sum_{i=1}^{K} D_i^T V_i^{-1} S_i S_i^T V_i^{-1} D_i \right\} \left\{ \sum_{i=1}^{K} D_i^T V_i^{-1} D_i \right\}^{-1}$$

More solutions

Replace $\sum_{i=1}^{K}$ with bigger matrices. Define all these things:

$$X = [X_1 \quad X_2 \quad \cdots \quad X_K]^T,$$

$$\mu = \begin{bmatrix} \mu_1^T & \mu_2^T & \cdots & \mu_K^T \end{bmatrix}^T$$

$$A = \operatorname{diag}(a(\mu)) \qquad \Delta = \operatorname{diag}\left(\frac{dg^{-1}(\eta)}{d\eta}\right)$$

$$S = \begin{bmatrix} S_1^T & S_2^T & \cdots & S_K^T \end{bmatrix}^T$$

and $R(\alpha)$ is a block diagonal matrix with each $R_i(\alpha)$ as blocks.

Matrix-izing it all

Using new matrices, solve (2) by

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} - \left\{ D^T V^{-1} D \right\}^{-1} \left\{ D^T V^{-1} S \right\} \tag{3}$$

with $V = \phi A^{\frac{1}{2}} R(\boldsymbol{\alpha}) A^{\frac{1}{2}}$.
Variance estimate is now

$$\left\{ D^T V^{-1} D \right\}^{-1} \left\{ D^T V^{-1} S^T J S V^{-1} D \right\} \left\{ D^T V^{-1} D \right\}^{-1}$$

where J is block diagonal with ones.

Block diagonals

Two big block diagonals are used:

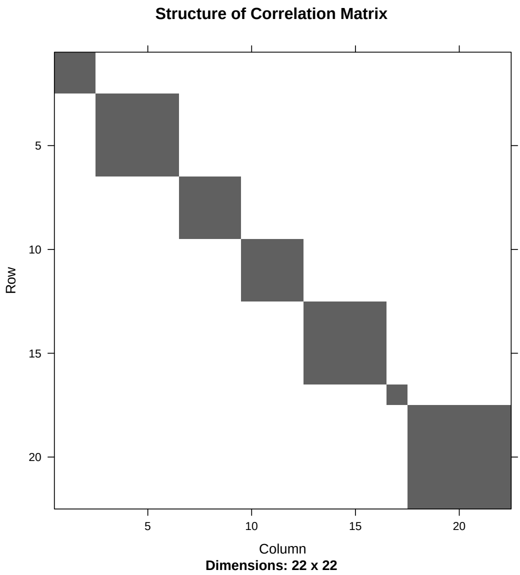
$$R(\alpha) = \begin{bmatrix} R_1(\alpha) & & & \\ & R_2(\alpha) & & \\ & & \ddots & \\ & & & R_K(\alpha) \end{bmatrix}$$

$$J = \begin{bmatrix} J_{n_1} & & & \\ & J_{n_2} & & \\ & & \ddots & \\ & & & J_{n_K} \end{bmatrix}$$

where J_n is a $n \times n$ matrix of ones

Memory allocation for these is the single most expensive thing

Block diagonal structure



Matrix housekeeping

Let $N = \sum_{i=1}^{K} n_i$.

- X is $N \times p$
- A and Δ are $N \times N$
- $R(\alpha)$ is $N \times N$ block diagonal
- J has same structure as $R(\alpha)$, but with only 1 and 0
- S, μ and η are $N \times 1$.

The Matrix package in R

Sparse matrix storage:

$$\begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

represented as

```
Formal class 'dgTMatrix' [package "Matrix"] with 6 slots
  ..@ i : int [1:2] 0 3
  ..@ j : int [1:2] 0 3
  ..@ Dim : int [1:2] 4 4
  ..@ Dimnames:List of 2
  .. .. $ : NULL
  .. .. $ : NULL
  ..@ x : num [1:2] 1 1
  ..@ factors : list()
```

Matrix package

- So we can build really large matrices and multiply them!
- However, we still have the problem of V^{-1}
- $V = \phi A^{\frac{1}{2}} R(\boldsymbol{\alpha}) A^{\frac{1}{2}}$ is block diagonal (sparse, symmetric)
- Inverse of block diagonal is block diagonal of inverses
- R doesn't know this
- The real problem is $R(\alpha)^{-1}$

Analytic inverses

Analytic inverses are worked out for

- AR-1
- Independence
- Exchangeable

However, this requires that we have no skipped observations!
R package geepack does not have this limitation

Numeric inverses

For everything else, we have to use the solve function in R. Consider unstructured matrix

$$R_i(\alpha) = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} \\ \alpha_{14} & \alpha_{24} & \alpha_{34} & 1 \end{bmatrix}$$

Then the full $R(\alpha)$ matrix will be roughly K blocks of this matrix

But say we have clusters of size 2,3, and 4 with no skipped observations (only dropouts)

Numeric inverses, continued

In addition to the previous $R_i(\alpha)$, we have

$$\begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} \\ \alpha_{12} & 1 & \alpha_{23} \\ \alpha_{13} & \alpha_{23} & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & \alpha_{12} \\ \alpha_{12} & 1 \end{bmatrix}$$

and that's it!
So we only need to solve a 2×2, a 3×3, and a 4×4 matrix, then reassemble appropriately for the full $R(\alpha)$ matrix.

Much easier than $N \times N$ (largest N we use is about 300k)

Speed comparison

$N = 296{,}218$, clusters of size 2 or 3 (birth outcomes data)

Independence		
	Averaged Elapsed (s)	Relative To Min
geeM	3.666	1.000
gee	11.634	3.173
geepack	10.940	2.984
AR-1		
	Averaged Elapsed (s)	Relative To Min
geeM	6.594	1.000
gee	30.590	4.639
geepack	25.578	3.879
Unstructured		
	Averaged Elapsed (s)	Relative To Min
geeM	4.820	1.000
gee	19.462	4.038
geepack	31.418	6.518

Speed comparison, comments

- Our package is much faster on large data
- Maybe a little slower on small data
- We use tons of memory
- Using large matrices $\Rightarrow$ time/memory tradeoff