Background

GLM with clustered data

A fixed effects approach

Göran Broström

Department of Statistics

Umeå University

SE–901 87 Umeå, Sweden

Poisson or Binomial data with the following properties

- A large data set,
- partitioned into many relatively small groups,
- and where members within groups have something in common,

The problem

- the number of parameters tend to increase with sample size.
- This fact causes the standard assumptions underlying asymptotic results to be violated.

Solutions

There are (at least) two possible solutions to the problem,

1. a random intercepts model, and
2. a fixed effects model, with
   - asymptotics replaced by simulation.
Packages in R

- The package *Matrix* has `lmer`,
- the MASS package has `glmmPQL`,
- Jim Lindsey’s `glmm` in his `repeated` package,
- Myles’ and Clayton’s `GLMMGibbs` for fitting mixed models by Gibbs sampling.
- Adding to that `glmmML` and `glmmboot` in the package `glmmML`.

Data structure

- $n$ clusters of sizes $n_i$, $i = 1, \ldots, n$.
- For each cluster $i$, $i = 1, \ldots, n$, observe responses $(y_{i1}, \ldots, y_{in_i})$ and vectors of explanatory variables $(x_{i1}, \ldots, x_{in_i})$, where $x_{ij}$ are $p$-dimensional vectors with
  - the first element identically equal to unity,
  - corresponding to the mean value of the random intercepts.
- The random part, $u_i$ of the intercepts are normal with mean zero and variance $\sigma^2$, and it is assumed that $u_1, \ldots, u_n$ are independent.

The conditional distribution

given the random intercepts $\beta_1 + u_i$, $i = 1, \ldots, n$:

$$
\Pr(Y_{ij} = y_{ij} \mid u_i; x) = P(\beta x_{ij} + u_i, y_{ij}),
$$

- Bernoulli distribution
  - logit link,
  $$
P(x, y) = \frac{e^{xy}}{1 + e^x}, \quad y = 0, 1; \quad -\infty < x < \infty,
$$
- cloglog link
  $$
P(x, y) = (1 - \exp(-e^x))^y \exp(-y e^x), \quad y = 0, 1; \quad -\infty < x < \infty,
$$
- Poisson distribution with log link
  $$
P(x, y) = \frac{e^{xy}}{y!} e^{-e^x}, \quad y = 0, 1, 2, \ldots; \quad -\infty < x < \infty
$$

Likelihood function

In the fixed effects model (and in the conditional random effects model), the likelihood function is

$$
L((\beta, \gamma); y, x) = \prod_{i=1}^{n} \prod_{j=1}^{n_i} P(\beta x_{ij} + \gamma_i, y_{ij}).
$$

The log likelihood function is

$$
\ell((\beta, \gamma); y, x) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \log P(\beta x_{ij} + \gamma_i, y_{ij}),
$$
Tests of cluster effect

Testing is performed via a simple bootstrap (glmmboot). Under the null hypothesis of no grouping effect,
- the grouping factor can be randomly permuted without changing the probability distribution (the conditional approach), or
- a parametric bootstrap approach: simulate observations from the fitted model under the null hypothesis (the unconditional approach).

The score vector

The partial derivatives wrt $\beta_m$, $m = 1, \ldots, p$, of the log likelihood function are:

$$U_m(\beta, \gamma) = \frac{\partial}{\partial \beta_m} \ell((\beta, \gamma); y, x)$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{n_i} x_{ijm} G(\beta x_{ij} + \gamma_i, y_{ij}), \quad m = 1, \ldots, p.$$  

where

$$G(x, y) = \frac{\partial}{\partial x} \log P(x, y) = \frac{\partial}{\partial x} \frac{P(x, y)}{P(x, y)}$$

Cluster components of the score

The partial derivatives wrt $\gamma_i$, $i = 1, \ldots, n$, are

$$U_{p+i}(\beta, \gamma) = \frac{\partial}{\partial \gamma_i} \ell((\beta, \gamma); y, x)$$
$$= \sum_{j=1}^{n_i} G(\beta x_{ij} + \gamma_i, y_{ij}), \quad i = 1, \ldots, n.$$  

Computational aspects

- A profiling approach reduces an optimizing problem in high dimensions to a problem consisting of
  - solving several one-variable equations followed by optimization in low dimensions.
With profiling

Setting $U_{p+i}(\beta, \gamma) = 0$ defines $\gamma$ implicitly as functions of $\beta$,

$$\gamma_i = \gamma_i(\beta), \ i = 1, \ldots, n$$

$$F(\beta, \gamma_i(\beta)) = \sum_{j=1}^{n_i} G(\beta x_{ij} + \gamma_i(\beta), y_{ij}) = 0, \ i = 1, \ldots, n.$$ 

From

$$\frac{\partial}{\partial \beta_m} F(\beta, \gamma_i(\beta)) = \frac{\partial \gamma_i}{\partial \beta_m} \frac{\partial F}{\partial \gamma_i} + \frac{\partial F}{\partial \beta_m} = 0$$

we get

Profile log likelihood

Replacing $\gamma$ by $\gamma(\beta)$ gives the profile log likelihood $\ell^{(P)}$:

$$\ell^{(P)}(\beta; y, x) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \log P(\beta x_{ij} + \gamma_i(\beta), y_{ij}),$$

as a function of $\beta$ alone.

Profile score

$$\frac{\partial \gamma_i(\beta)}{\partial \beta_m} = -\frac{\partial F}{\partial \beta_m} \frac{\partial F}{\partial \gamma_i} = -\frac{\sum_{j=1}^{n_i} x_{ijm} H(\beta x_{ij} + \gamma_i, y_{ij})}{\sum_{j=1}^{n_i} H(\beta x_{ij} + \gamma_i, y_{ij})}, \ i = 1, \ldots, n; \ m = 1, \ldots, p.$$ 

which is needed when calculating the score corresponding to the profile likelihood.

Profile partial derivatives

The partial derivatives wrt $\beta_m, m = 1; \ldots, p$, of the log profile likelihood function becomes:

$$U^{(P)}_m(\beta) = \frac{\partial}{\partial \beta_m} \ell^{(P)}(\beta; y, x)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n_i} \left( x_{ijm} + \frac{\partial \gamma_i(\beta)}{\partial \beta_m} \right) G(\beta x_{ij} + \gamma_i(\beta), y_{ij})$$

$$= U_m(\beta, \gamma(\beta)) + \sum_{i=1}^{n} \frac{\partial \gamma_i(\beta)}{\partial \beta_m} \sum_{j=1}^{n_i} G(\beta x_{ij} + \gamma_i(\beta), y_{ij})$$

$$= U_m(\beta, \gamma(\beta)),$$

Thus we get back the unprofiled partial derivatives.
Profile hessian

\[ -I_{ms}^{(P)}(\beta) = \frac{\partial}{\partial \beta_s} U_m(\beta, \gamma(\beta)) \]
\[ = \sum_{i=1}^{n} \sum_{j=1}^{n_i} x_{ijm} \left( x_{ij} + \frac{\partial \gamma_i(\beta)}{\partial \beta_s} \right) H(\beta x_{ij} + \gamma_i(\beta), y_{ij}) \]
\[ = -I_{ms}(\beta, \gamma(\beta)) \]
\[ - \sum_{i=1}^{n} \sum_{j=1}^{n_i} x_{ijm} H_{ij} \sum_{j=1}^{n_i} x_{ij} H_{ij}, \]
\[ m, s = 1, \ldots, p. \]

where

\[ H_{ij} = H(\beta x_{ij} + \gamma_i(\beta), y_{ij}), \quad j = 1, \ldots n_i; \quad i = 1, \ldots, n. \]

At the maximum

Justifying the use of the profile likelihood:

**Theorem 1 (Patefield)** The inverse hessians from the full likelihood and from the profile likelihood for \( \beta \) are equal when

\[ (\gamma, \beta) = (\hat{\gamma}, \hat{\beta}). \]

Preparation for R

\[ \ell^{(P)}(\beta) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \log P(\beta x_{ij} + \gamma_i(\beta), y_{ij}), \]
\[ U_m^{(P)}(\beta) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} x_{ijm} G(\beta x_{ij} + \gamma_i(\beta), y_{ij}), \]
\[ m = 1, \ldots, p. \]

For fixed \( \beta \), \( \gamma_i(\beta) \) is found by solving

\[ \sum_{j=1}^{n_i} G(\beta x_{ij} + \gamma_i, y_{ij}) = 0, \]

with respect to \( \gamma_i, i = 1, \ldots, n. \)

The maximization is performed by optim, via the C function vmmin, available as an entry point in the C code of R.

Implementation in R

- Implemented in the package *glmML* in R.
- Covers three cases,
  1. Binomial with logit link,
  2. Binomial with cloglog link,
  3. Poisson with log link.
- The function is *glmmboot*,
- Testing of cluster effect is done by simulation (a simple form of bootstrapping).
  - conditionally, or
  - unconditionally.
Binomial with logit link

\[ P(x, y) = \frac{\exp(xy)}{1 + \exp(x)}, \]
\[ G(x, y) = y - P(x, 1). \]

We get \((\gamma_1, \ldots, \gamma_n)\) by solving the equations

\[ \sum_{j=1}^{n_i} y_{ij} = \sum_{j=1}^{n_i} \frac{\exp(\beta x_{ij} + \gamma_i)}{1 + \exp(\beta x_{ij} + \gamma_i)} \]

for \(i = 1, \ldots, n\) (using the C version of \texttt{uniroot}).

Special cases: \(\sum y_{ij} = 0\) or \(n_i\); giving \(\gamma_i = -\infty\) or \(+\infty\), respectively.

- Corresponding cluster can be thrown out.
- (Should be used in \texttt{glm}?)

Binomial with cloglog link

\[ P(x, y) = (1 - \exp(-\exp(x))^y \exp(- (1 - y) \exp(x))), \]
\[ G(x, y) = \frac{\exp(x)}{P(x, 1)} \{y - P(x, 1)\} \]

We get \((\gamma_1, \ldots, \gamma_n)\) by solving the equations

\[ \sum_{j=1}^{n_i} y_{ij} = n_i - \sum_{j=1}^{n_i} \exp(-\exp(\beta x_{ij} + \gamma_i)) \]

for \(i = 1, \ldots, n\) (using the C version of \texttt{uniroot}).

Special cases: \(\sum y_{ij} = 0\) or \(n_i\); \(\gamma_i = -\infty\) or \(+\infty\), respectively.

- Corresponding cluster can be thrown out.

Poisson with log link

\[ P(x, y) = e^{xy}/y! \exp(-\exp(x)) \]
\[ G(x, y) = y - e^x \]

We get \((\gamma_1, \ldots, \gamma_n)\) from

\[ \sum_{j=1}^{n_i} y_{ij} = e^{\gamma_i} \sum_{j=1}^{n_i} \exp(\beta x_{ij}), \quad i = 1, \ldots, n, \]

giving

\[ \gamma_i = \log \left\{ \frac{\sum_j y_{ij}}{\sum_j \exp(\beta x_{ij})} \right\}, \quad i = 1, \ldots, n. \]

- Special case: \(\sum y_{ij} = 0\), giving \(\gamma_i = -\infty\).

Simulation

Model:

\[ P(Y_{ij} = 1 \mid \gamma_i) = 1 - P(Y_{ij} = 0 \mid \gamma_i) \]
\[ = \frac{e^{\gamma_i}}{1 + e^{\gamma_i}}, \quad j = 1, \ldots, 5; \quad i = 1, \ldots, n, \]

where \(\gamma_1, \ldots, \gamma_n\) are \textit{iid} \(N(0, \sigma)\).

Hypothesis: \(\sigma = 0\).
Simulation specifications

- \( \sigma = 0, 0.5 \).
- \( n = 5, 50, 500 \).
- Four methods:
  - glmmboot, unconditional and conditional,
  - glmmML,
  - glm (naively?).

Null model (\( \sigma = 0 \)); 5 clusters

Null model (\( \sigma = 0 \)); 50 clusters

Null model (\( \sigma = 0 \)); 500 clusters
<table>
<thead>
<tr>
<th>Clustering ($\sigma = 0.5$); 5 clusters</th>
<th>Clustering ($\sigma = 0.5$); 50 clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Boot</strong></td>
<td><strong>Boot</strong></td>
</tr>
<tr>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
</tr>
<tr>
<td><strong>Boot, conditional</strong></td>
<td><strong>Boot, conditional</strong></td>
</tr>
<tr>
<td><img src="image3.png" alt="Graph" /></td>
<td><img src="image4.png" alt="Graph" /></td>
</tr>
<tr>
<td><strong>ML</strong></td>
<td><strong>ML</strong></td>
</tr>
<tr>
<td><img src="image5.png" alt="Graph" /></td>
<td><img src="image6.png" alt="Graph" /></td>
</tr>
<tr>
<td><strong>glm</strong></td>
<td><strong>glm</strong></td>
</tr>
<tr>
<td><img src="image7.png" alt="Graph" /></td>
<td><img src="image8.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

| GLM with clustered data – p. 29        | GLM with clustered data – p. 30         |

<table>
<thead>
<tr>
<th>Timings, 5 clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image9.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

```r
> system.time(glmmboot(y ~ 1, cluster = cluster, + data = timing, conditional = FALSE, boot = 2000))
[1] 0.06 0.00 0.06 0.00 0.00

> system.time(glmmboot(y ~ 1, cluster = cluster, + data = timing, conditional = TRUE, boot = 2000))
[1] 0.044 0.000 0.044 0.000 0.000

> system.time(glm(y ~ factor(cluster), data = timing))
[1] 0.013 0.000 0.012 0.000 0.000

> system.time(glm(y ~ factor(cluster), data = timing, family = binomial))
[1] 0.008 0.000 0.008 0.000 0.000
```
Timings, 50 clusters

> system.time(glmmboot(y ~ 1, cluster = cluster, data = timing, conditional = FALSE, boot = 2000))
[1] 0.529 0.000 0.529 0.000 0.000

> system.time(glmmboot(y ~ 1, cluster = cluster, data = timing, conditional = TRUE, boot = 2000))
[1] 0.376 0.000 0.376 0.000 0.000

> system.time(glm(y ~ factor(cluster), data = timing, family = binomial))
[1] 0.047 0.002 0.061 0.000 0.000

Timings, 500 clusters

> system.time(glmmboot(y ~ 1, cluster = cluster, data = timing, conditional = FALSE, boot = 2000))
[1] 5.208 0.000 5.214 0.000 0.000

> system.time(glmmboot(y ~ 1, cluster = cluster, data = timing, conditional = TRUE, boot = 2000))
[1] 3.713 0.003 3.719 0.000 0.000

> system.time(glm(y ~ factor(cluster), data = timing, family = binomial))
[1] 27.840 0.593 28.434 0.000 0.000

glm vs. glmmboot(boot = 0)

<table>
<thead>
<tr>
<th>No. of clusters</th>
<th>glm</th>
<th>glmmboot</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.008</td>
<td>0.007</td>
</tr>
<tr>
<td>25</td>
<td>0.019</td>
<td>0.008</td>
</tr>
<tr>
<td>100</td>
<td>0.182</td>
<td>0.011</td>
</tr>
<tr>
<td>500</td>
<td>28.434</td>
<td>0.031</td>
</tr>
<tr>
<td>1000</td>
<td>223.288</td>
<td>0.056</td>
</tr>
</tbody>
</table>

Conclusion: Profiling is numerically very efficient.