Biostatistics 140.754
Advanced Methods in Biostatistics IV

Jeffrey Leek

Assistant Professor
Department of Biostatistics
jleek@jhsph.edu

Lecture 7

Tip + Paper

Tip Learn a computer programming language other than R (Python, Perl, Java are popular options). R isn't built to handle big data sets, does loops slow, and doesn't deal with non-regular (read not tables) data sets really well. Many of the data sets of the future in a lot of disciplines will have these problems. Having a go-to language is a good idea. How do you learn? The best way is to start working on a project where you need the language, then get a book.

Paper of the Day:
"Phonemic Diversity Supports a Serial Founder Effect Model of Language Expansion from Africa"
http://www.sciencemag.org/content/332/6027/346.abstract

Outline For Today

- GEE!

Generalized Estimating Equations†

We move to semiparametric inference for vector outcomes, and in particular, **G**eneralized **E**stimating **E**quations

- Liang and Zeger (1986) developed GEE as a generalization of earlier quasi-likelihood techniques. GEE and QL have similar robustness properties
- We will make assumptions about the (true) mean model, and working assumptions about the covariance of the outcomes (notably within-cluster correlations)
- Familiar sandwich forms are used to provide estimates of the standard error

In many fields, GEE is the "industry standard" method for clustered data, notably longitudinal data. any serious statistical software will implement it.

GEE: mean model†

In addition to the between-cluster independence seen before, in a GEE "model" we assume a marginal mean model:

$$\mathbb{E}[Y_{ij}|X_{ij}] = g(X_{ij}^T \beta) \equiv \mu_{ij}, \quad \forall i, j$$

-with the usual $g(\cdot)$ link function

- "Marginal" means that we condition on X_{ij} **and nothing else**
- This is a notable restriction; "transition" models condition on "earlier" Y_{ij}; "latent variable" models condition on cluster-specific additional covariates b_i. $X_{ij'}$ may also matter
- If X_{ij} is the same for all j (e.g. treatment/placebo in a longitudinal clinical trial) or are fixed by design without regard to the other $\{Y_{ik}, j \neq k\}$, this conditioning does not reflect the true F; but we are still assuming $g(\cdot)$ is the right link function

Time-varying covariates (based on interim Y_{ij}) are not covered here.

GEE: mean model†

We are conditioning on the observed $\{X_{ij}\}$, i.e. frequentist inference considers replications where the covariates are those seen at the observed values [1]

- Similarly, statements about e.g. $\mathrm{Cov}[\hat{\boldsymbol{\beta}}]$ will now use $\mathrm{Cov}[\hat{\boldsymbol{\beta}}|X = x]$, i.e. we consider running the universe over and over, but using the same X values observed in our study.
- Describing the parameter is simpler than for nonparametrics; β_k is the difference/fold-change/odds-ratio in $\mathbb{E}[Y]$ per one-unit difference in covariate k, among observations with covariates that are otherwise identical [2].
- This is a **stronger** statement than just a "trend"

[1] ... though inferential properties carry over to replications where $\{X_{ij}\}$ is random.

[2] a.k.a. "keeping everything else fixed", although this phrase can mislead"

GEE: mean model†

It's very important to state that we are fitting a **marginal** mean model

- In general, this statement is an average over non-identical clusters (due to e.g. unmeasured variables)
- A marginal model *can* result from "integrating out" latent cluster-specific effects (recall non-collapsibility). Mean model parameters between Y and X need not be the same in these cases -which do you want?
- The assumed marginal mean model can be wrong (e.g. non-additivity, non-linearity on the specified scale) or unhelpful (e.g. linear in X but more interestingly linear in X at different levels of Z). We will discuss model-checking, later.

GEE: working assumptions†

The estimating equations in GEE generalize what we saw before:

$$\sum_{i=1}^{n} \frac{\partial g(X_i^T \beta)}{\partial \beta^T} \mathbf{V}_i^{-1} (Y_i - g(X_i^T \beta)) = \mathbf{0}_p$$

To evaluate these, we must specify some choice of $\{\mathbf{V}_i\}$. The standard choice of variance is that

$$(\mathbf{V}_i)_{jj} = \phi \mathbf{S}(\mu_{ij})$$

i.e. that the variance of every Y_{ij} is some specified form $\mathbf{S}(\cdot)$ of the mean, multiplied by dispersion parameter ϕ.

To complete the working covariance matrix, we require a working choice of the correlation matrix $\mathbf{R}$ where

$$\mathbf{V}_i = \phi \mathrm{diag}\{\mathbf{S}(\mu_{ij})^{1/2}\} \mathbf{R}_i \mathrm{diag}\{\mathbf{S}(\mu_{ij})^{1/2}\}$$

GEE: working assumptions†

Some commonly-used correlation structures for $\mathbf{R}_i$ are:

Independence:

$$\boldsymbol{R}_i = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Exchangeable:

$$\boldsymbol{R}_i = \begin{bmatrix} 1 & \alpha & \alpha & \cdots & \alpha \\ \alpha & 1 & \alpha & \cdots & \alpha \\ \alpha & \alpha & 1 & \cdots & \alpha \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha & \alpha & \alpha & \cdots & 1 \end{bmatrix}$$

GEE: working assumptions†

Some commonly-used correlation structures for $\mathbf{R}_i$ are:

Auto-regressive (AR1):

$$\boldsymbol{R}_i = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{m-1} \\ \alpha & 1 & \alpha & \cdots & \alpha^{m-2} \\ \alpha^2 & \alpha & 1 & \cdots & \alpha^{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha^{m-1} & \alpha^{m-2} & \alpha^{m-3} & \cdots & 1 \end{bmatrix}$$

Unstructured: (symmetric)

$$\boldsymbol{R}_i = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1m} \\ \alpha_{21} & 1 & \alpha_{23} & \cdots & \alpha_{2m} \\ \alpha_{31} & \alpha_{32} & 1 & \cdots & \alpha_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \alpha_{m3} & \cdots & 1 \end{bmatrix}$$

GEE: estimating equations†

For now, we assume that ϕ and/or any α parameters are known. In particular, this lets us evaluate the estimating equations, at any $\hat{\beta}$, and consequently to find $\hat{\boldsymbol{\beta}}$, the root of the estimating equations.

- The EEs we used for nonparametric work set (for convenience) $\mathbf{V}_i = \mathbf{I}_{n_i}$ - actually common in GEE work.
- We will see that (for correct marginal models)
  - The resulting $\hat{\boldsymbol{\beta}}$ is consistent and asymptotically Normal ($n \to \infty$) under mild regularity conditions (robustness)
  - We get better $\hat{\boldsymbol{\beta}}$ when $\mathbf{V}_i \propto \mathrm{Cov}[Y_i|X_i]$ (efficiency)

GEE: estimating equations†

For consistency; note that by assumption for any $\{\mathbf{V}_i^{-1}\}$ we have

$$\mathbb{E}_{X_i}\left[\mathbf{D}_i^T\mathbf{V}_i^{-1}\mathbb{E}_{Y_i|X_i}[Y_i-\mu_i(\beta)]\right]=\mathbf{0}_p$$

where μ_i denotes the vector of means $\mathbf{D}_i$ is their matrix of derivatives with respect to β.

Asymptotic Normality of $\hat{\beta}$ holds because we get it from an average over n similarly-distributed variables; the usual sandwich $\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}$ formula determines the asymptotic variance.

For the $\mathbf{A}$ matrix, we consider

$$\mathbb{E}\left[\left(\frac{\partial\mathbf{D}^T}{\partial\beta}\right)\mathbf{V}^{-1}(Y-\mu)+\mathbf{D}^T\left(\frac{\partial\mathbf{V}^{-1}}{\partial\beta}\right)(Y-\mu)-\mathbf{D}^T\mathbf{V}^{-1}\mathbf{D}\right]$$

Using the assumptions of the marginal mean model, only the right hand term here is nonzero. But $\mathbf{B}$ does not simplify in this way:

$$\mathbf{B}=\mathbb{E}[\mathbf{D}^T\mathbf{V}^{-1}(Y-\mu)(Y-\mu)^T\mathbf{V}^{-1}\mathbf{D}]$$

GEE: sandwich estimates†

I hope you'd have guessed these (robust) estimates

$$\begin{aligned}\hat{\mathbf{A}} &= (1/n)\sum_{i=1}^{n}\mathbf{D}_i^T\mathbf{V}_i^{-1}\mathbf{D}_i \\ \hat{\mathbf{B}} &= (1/n)\sum_{i=1}^{n}\mathbf{D}_i^T\mathbf{V}_i^{-1}(Y_i-\mu_i)(Y_i-\mu_i)^T\mathbf{V}_i^{-1}\mathbf{D}_i\end{aligned}$$

where everything is evaluated at the point estimates for all relevant parameters, and for inference we use

$$\hat{\mathrm{Cov}}[\hat{\boldsymbol{\beta}}] = \hat{\mathbf{A}}^{-1}\hat{\mathbf{B}}\hat{\mathbf{A}}^{-1}/n$$

... some authors/code use summation (not averages) in $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ which removes the $1/n$ term in the resulting sandwich estimate.

Asymptotically-justified confidence intervals result, in the usual way, without further assumptions. Wald tests of e.g., $H_0 : \beta_j = 0$ also use familiar machinery.

Example: measles data†

Sherman and Le Cessie (1997, on the class site) studied number of cases of measles in preschool children, in 15 counties in the US, between 1985 and 1991.

For each county the annual number of preschoolers with measles was recorded, as well as factors possibly related to measles incidence, like immunization rate, and density of preschoolers per county.

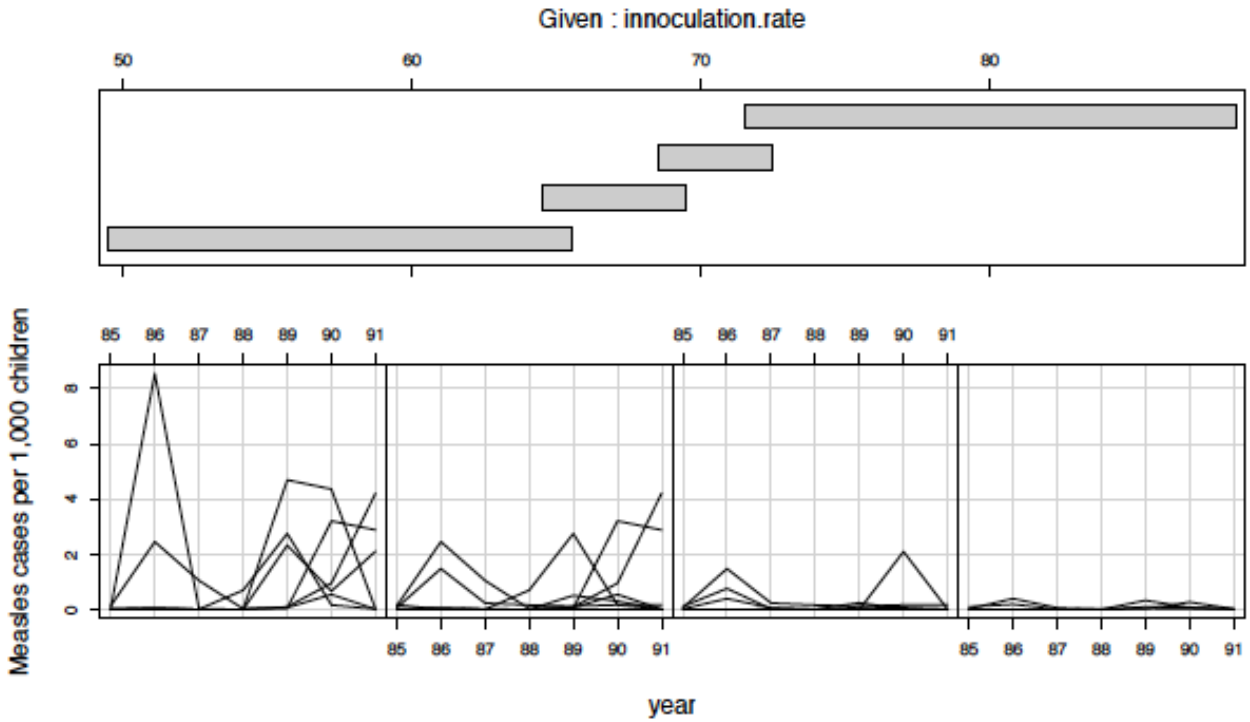
The data are annual for each county. We know:

- Number of cases of measles
- Immunization rate (percentage, fixed over time)
- Total number of preschoolers

We want to know the association between immunization rate and measles incidence.

Example: measles data†

The measles data via `coplot`; what trend do you notice?



Example: measles data†

For number of measles cases y_{ij} and immunization rate $x_{ij} = x_i$ to fit a line of the form

$$y_{.j} = \text{n.children}_{ij} \times e^{\beta_0 + \beta_1 x_{ij}}$$

... it's sane to use these estimating equations:

$$\sum_{i=1}^{n} X_i (Y_i - \mu_i) = \mathbf{0}_2$$

where $\mu_{ij} = \exp(\log(\text{n.children}_{ij}) + \beta_0 + \beta_1 X_{ij})$

The term $\log(\text{n.children}_{ij})$ is called an offset, think of it as a covariate whose coefficient is fixed to be 1, regardless of the other data.

Similarly to the previous example, we can solve these EEs using `glm()`; the solution is $\hat{\boldsymbol{\beta}} = \{-0.458, -0.108\}$.

Example: measles data†

For the measles data, we could assume:

$$\mathbb{E}[Y_{ij}|X_{ij} = x_{ij}] = \text{n.children}_{ij} \times e^{\beta_0 + \beta_1 x_{ij}}$$

Now, we are making a semi-parametric assumption; comparing observations where x_{ij} differs by $\Delta(x)$, the expected rates of measles cases differ by a factor of $\exp(\beta_1)\Delta(x)$, for any $\Delta(x)$. In particular, the expected rate of measles cases differs by $\exp(\beta_1)$ for each one-unit difference in x_{ij}.

The estimating equations are exactly those we saw on the previous page.

Example: measles data†

Coding this is slightly simpler than the non-parameteric version; there is no need to work out second derivatives in $\hat{\mathbf{A}}$.

```
measles <- read.table("measlesdata.txt",header=T)
measles$county <- as.character(measles$county)

# Same helper function as before
sum.list <- function(l1){
l <- length(l1); out <- l1[[1]]
for(i in 2:l){out <- out + l1[[i]]}
out}

glm1 <- glm(cases ~ rate + offset(log(children)), data=measles, family=poisson)
measles$muhat <- fitted(glm1)
# i.e.  hte point estimates - these include the offset

allAi <- by(measles, measles$county, simplify=F,function(data){ ni <- dim(data)[1]
Xi <- cbind(rep(1,ni),data$rate)
Di <- Xi * data$muhat
Vi <- diag(data$muhat)
# could use crossprod t(Di)%*% solve(Vi) %*%Di
})

Ahat <- sum.list(allAi)
```

Example: measles data†

We can similarly code $\hat{\mathbf{B}}$, and get the sandwich estimate

```
allBi <- by(measles, measles$county,simplify=F,function(data){
ni <- dim(data)[1]
Xi <- cbind(rep(1,ni),data$rate)
Di <- Xi * data$muhat
Vi <- diag(data$muhat)
resid <- data$cases - data$muhat
t(Di) %*% solve(Vi) %*% resid %*% t(resid) %*% solve(Vi) %*% Di
})
Bhat <- sum.list(allBi)


sand1 <- solve(Ahat) %*% Bhat %*% solve(Ahat)
```

- Where did ϕ go? (Hint: count the $\mathbf{V}$ terms in the formulae)
- Within each cluster, these calculations should be familiar from the univariate case
- `crossprod()` is faster, `sum.list()` is clunky
- Premature optimization is the root of all evil (Donald Knuth)

Example: measles data†

Finally the answers:

```
> coeff(glm1)
(Intercept)   rate
-0.4581488 -0.1081273
> sqrt(diag(sand1))
[1] 1.01237231   0.01589211
> coeff(glm1) + sqrt(diag(sand)) %o% qnorm(c(0.25,0.975))
[,1]  [,2]
[1,]   -2.4423621   1.52606448
[2,]   -0.1392755   -0.07697913
```

- Comparing measles incidence where the `rate` differs by one percentage point, the estimated log-rate ratio is -0.11 (-0.14, -0.08). The estimates RR is 0.90 (0.87, 0.93)
- A naïve standard error, assuming independence, estimates the standard error to be 0.002 (Poisson model-based) or 0.021 (robust)

Example: measles data†

A non-trivial factor in the success of GEE is the availability of (free) software that implements it, at little hassle to the user.

We consider the gee package (Carey, Lumley, and Ripley). The syntax is similar to that of `glm()`, but an `id` must be specified, indicating cluster membership.

```
> library(gee) > gee1 <- gee(cases ~ rate + offset(log(children)),data=measles,id=county,
family = "poisson", corstr="independence")
Beginning Cgee S-function (#) geeformula.q.4.13 98/01/27
running glm to get initial regression estimate
(Intercept)  rate
-0.458149   -0.108127
```

Similar to our implementation, output from `glm()` is used for parts of the calculation.

Example: measles data†

```
> summary(gee1) GEE: GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
gee S-function, version 4.13 modified 98/01/27 (1998)
Model:
Link:  Logarithm
Variance to Mean Relation:  Poisson
Correlation Structure:  Independent
Coefficients:
Estimate Naive S.E. Naive z Robust S.E. Robust z
(Intercept) -0.458149 1.5370065 -0.298079 1.0123723 -0.45255
rate -0.108127 0.0240767 -4.490956 0.0158922 -6.80379
Estimated Scale Parameter:  155.89
Number of Iterations:  1 # nothing to update, for corstr=indep't

> sqrt(diag(gee1$robust.variance))
(Intercept) rate
1.0123723 0.0158922
> mean((fitted(glm1) - measles$cases)^2/fitted(glm1))*105/(105-2)

[1] 155.89
```

The scale parameter ϕ is the standard bias-corrected Pearson estimator (see QL notes).

- A wide range of `family` options are supported
- Do read the documentation, in particular that "data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula". This means that `id = c(1,1,0,0,1,1,0,0)` gets you four clusters, each with two observations.
- Observations with relevant missing values are omitted. See earlier discussion on informative cluster size- but also watch out if using `gee()` for several regressions, that have different covariates.
- Get rid of the irritating `Cgee` message by making a copy of the `gee()` function, and removing its `message()` call.

geepack†

Also be aware of `geepack`, which features a `glmgee()` function.

- Has the same syntax as `gee()`
- It calls function `geese()`, which in turn calls `geese.fit()`
- Slightly different internal calculations, e.g. dispersion parameters, sandwich estimate, which are not (quite) those given here
- Output is designed to mimic that of `glm()`, which is good if you are intricately familiar with that setup
- Permits use of `anova()` .. should you want that
- Messes around with `options(‘‘digits’’)`, on some machines
- Can crash `R` (!) - see documentation

GEE: working correlation†

The inverse-weighting matrix $\mathbf{V}_i$ is known as the "working" covariance matrix. This terminology emphasizes that exact knowledge of the true covariance is not required; we get consistency, asymptotic Normality and valid sandwich-based intervals regardless of the choice.

(If we additionally assume a variance structure, further simplification occurs in the sandwich - that is what our presentation of QL did for the univariate case. But this is not standard GEE practice; robustness to variance assumptions is attractive)

Following the same principles as Aitken/Gauss-Markov/Godambe and Heyde, to get efficiency we should choose each "working" $\mathbf{V}_i$ to be as close as possible to proportional to the true variance-covariance matrix of elements in each Y_i.

How much difference does it actually make?

GEE: choice of working correlation†

Asymptotic relative efficiency, (relative to the "right" choice) for $n_i = 10$ (Liang and Zeger 1986, Table 1)

		Working correlation		
Trur Corr'n	α	Indep	Exch	AR1
Exch	0.3	0.99	1.00	0.95
	0.7	0.99	1.00	0.72
AR	0.3	0.97	0.97	1.00
	0.7	0.88	0.88	1.00

- Minor correlation $\Longrightarrow$ choice doesn't matter
- Large correlation $\Longrightarrow$ large efficiency loss possible

The same caveats apply as with univariate regression; an extra 10% of efficiency may not change practical inference based on very precise/imprecise estimates. The choice of working correlation structure should be based on external information - "picking the best" induces the "wiggle room" problem.

GEE: choice of working correlation†

Fitzmaurice (1995) showed the answer depends on covariate design (here r = within-subject correlation, T = n_i).

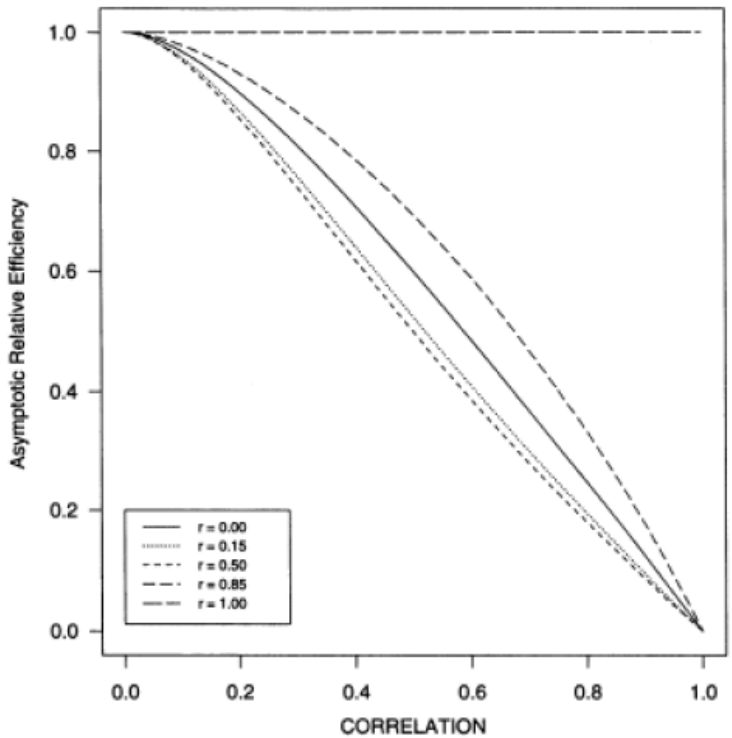


Figure 2. Asymptotic efficiency of the Exchangeable GEE estimator, relative to the Independence estimator, for selected values of the intra-cluster correlation for the covariate (and when $T = 5$).

Large ICC $\Longrightarrow$ big efficiency loss

GEE: choice of working correlation†

Cluster size also matters (Mancl and Leroux 1996)

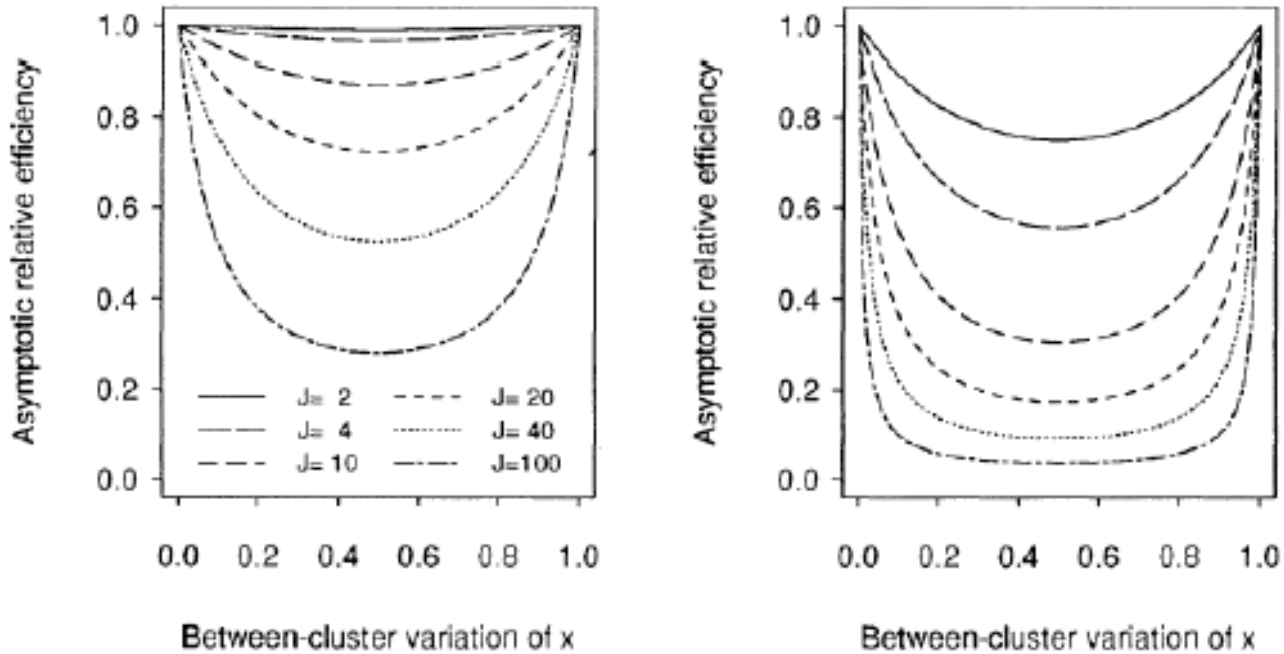


Figure 1. Asymptotic relative efficiency of independence to exchangeable for the case of constant weights, equal cluster sizes (J), and common pairwise correlation between responses of (a) $\rho_y = 0.1$ and (b) $\rho_y = 0.5$.

- Small clusters $\Longrightarrow$ possibly minor efficiency loss
- Large clusters $\Longrightarrow$ possibly major efficiency loss

GEE: estimating α †

Except for independence working correlation assumptions, unknown parameters α have to be estimated.

However, as with "plug-in" and operations seen earlier, it turns out that [3] any consistent estimator $\hat{\alpha}$ can be "plugged in" for this unknown, and (in large samples) the behavior of $\boldsymbol{\beta}$ is the same as if α were known. Showing this formally will happen in your stat theory class, but the result is not magic. Informally the argument is

- For known $\mathbf{R}(\alpha)$, estimator $\hat{\boldsymbol{\beta}}_\alpha$ is consistent for $\boldsymbol{\beta}$
- Estimating α in any (sane) way will give a $\hat{\alpha}$ that is consistent for some value α'
- If $\hat{\alpha} \to \alpha'$ sufficiently quickly, $\hat{\boldsymbol{\beta}}_{\hat{\alpha}'}$ is consistent for $\boldsymbol{\beta}$

Note focusing on consistency alone upsets some people (e.g. Drum and McCullagh) - & in low power settings this is fair.

[3] basically! ... mild regularity conditions apply

GEE: estimating α †

The consistency result motivates finding straightforward estimates of α that we can incorporate into a fitting procedure for both $\hat{\beta}$ and $\hat{\alpha}$.

Liang and Zeger (1986) proposed the following moment based estimators of α, making use of Pearson residuals:

$$e_{ij} = \frac{Y_{ij} - g(X_{ij}^T \hat{\beta})}{\sqrt{V(g(X_{ij}^T \hat{\beta}))}}$$

Exchangeable

$$\hat{\alpha} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{n_i(n_i - 1)} \sum_{j=j'} e_{ij} e_{ij'}$$

i.e. averaging the products of all pairs of residuals where the pairs are formed within clusters.

GEE: estimating α †

Autoregressive (AR1):

$$\hat{\alpha} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{(n_i - 1)} \sum_{j \leq n_i - 1} e_{ij} e_{i,j+1}$$

i.e. averaging the products of all pairs of residuals, where the pairs are "neighbors" within cluster.

Unstructured:

$$\hat{\alpha}_{jj'} = \frac{1}{n} \sum_{i=1}^{n} e_{ij} e_{ij'}$$

i.e. averaging the products of all pairs of residuals at positions j and j', over all clusters.

.... together with independence, which has no α to estimate, these working correlation matrices are "built in" to standard GEE software. For others ... roll up your sleeves.

GEE: fitting algorithm†

The GEE fitting algorithm:

1. Start with initial $\hat{\beta}^0$, obtained from e.g. a univariate analysis - typically by using `glm()`
2. (a) Calculate Pearson residuals, (b) calculate $\hat{\alpha}$, based on current $\hat{\boldsymbol{\beta}}$ and $\{\mu_i\}$
3. Update $\hat{\boldsymbol{\beta}}$, by setting:

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} + \left(\sum_{i=1}^{n} \mathbf{D}_i^T \mathbf{V}_i(\hat{\alpha})^{-1} \mathbf{D}_i \right)^{-1} \left(\sum_{i=1}^{n} \mathbf{D}_i^T \mathbf{V}_i(\hat{\alpha})^{-1} (Y_i - \mu_i) \right)$$

Repeat 2 and 3 until convergence. This algorithm generalizes the Fisher scoring method you saw earlier. It is one (good) way to solve simultaneously for $\hat{\alpha}$ and $\hat{\beta}$.