

On the usage of the **geepack**

Søren Højsgaard

November 16, 2011

1 Introduction

The **geepack** package for generalized estimating equations is described in Halekoh, U., Højsgaard, S., Yan, J. (2006). The package **geepack** for generalized estimating equations. Journal of Statistical Software. 15, 2. If you use **geepack** in your own work, please do cite the above reference.

This note contains a few extra examples. We illustrate the usage of a the **waves** argument and the **zcor** argument together with a fixed working correlation matrix for the **geeglm()** function. To illustrate these features we simulate some data suitable for a regression model.

```
> library(geepack)
> timeorder <- rep(1:5, 6)
> tvar      <- timeorder + rnorm(length(timeorder))
> idvar <- rep(1:6, each=5)
> uu      <- rep(rnorm(6), each=5)
> yvar  <- 1 + 2*tvar + uu + rnorm(length(tvar))
> simdat <- data.frame(idvar, timeorder, tvar, yvar)
> head(simdat,12)
```

	idvar	timeorder	tvar	yvar
1	1	1	1.8449631	4.432342
2	1	2	3.1749567	7.629293
3	1	3	3.1925733	6.679165
4	1	4	4.3811518	9.440102
5	1	5	2.9371403	8.162482
6	2	1	2.0575328	3.968900
7	2	2	1.8507987	2.721638
8	2	3	2.1536808	4.897881
9	2	4	3.1341779	4.894287
10	2	5	5.6353729	9.910881
11	3	1	0.9225695	3.907438
12	3	2	2.2135960	5.588491

Notice that clusters of data appear together in **simdat** and that observations are ordered (according to **timeorder**) within clusters.

We can fit a model with an AR(1) error structure as

```

> mod1 <- geeglm(yvar~tvar, id=idvar, data=simdat, corstr="ar1")
> mod1

Call:
geeglm(formula = yvar ~ tvar, data = simdat, id = idvar, corstr = "ar1")

Coefficients:
(Intercept)      tvar
    1.084046    1.981558

Degrees of Freedom: 30 Total (i.e. Null);  28 Residual

Scale Link:
Estimated Scale Parameters:  [1] 1.390683

Correlation: Structure = ar1    Link = identity
Estimated Correlation Parameters:
      alpha
0.5638696

Number of clusters: 6    Maximum cluster size: 5

```

This works because observations are ordered according to time within each subject in the dataset.

2 Using the waves argument

If observations were not ordered according to cluster and time within cluster we would get the wrong result:

```

> set.seed(123)
> ## library(doBy)
> simdatPerm <- simdat[sample(nrow(simdat)),]
> ## simdatPerm <- orderBy(~idvar, simdatPerm)
> simdatPerm <- simdatPerm[order(simdatPerm$idvar),]
> head(simdatPerm)

  idvar timeorder   tvar   yvar
2     1         2 3.174957 7.629293
4     1         4 4.381152 9.440102
1     1         1 1.844963 4.432342
3     1         3 3.192573 6.679165
5     1         5 2.937140 8.162482
9     2         4 3.134178 4.894287

```

Notice that in `simdatPerm` data is ordered according to subject but the time ordering within subject is random.

Fitting the model as before gives

```

> mod2 <- geeglm(yvar~tvar, id=idvar, data=simdatPerm, corstr="ar1")
> mod2

Call:
geeglm(formula = yvar ~ tvar, data = simdatPerm, id = idvar,
      corstr = "ar1")

Coefficients:
(Intercept)      tvar
    0.9366466    2.0317212

Degrees of Freedom: 30 Total (i.e. Null);  28 Residual

Scale Link:
Estimated Scale Parameters:  [1] 1.359835

Correlation: Structure = ar1    Link = identity
Estimated Correlation Parameters:
      alpha
0.5719275

Number of clusters: 6    Maximum cluster size: 5

```

Likewise if clusters do not appear contiguously in data we also get the wrong result (the clusters are not recognized):

```
> ## simdatPerm2 <- orderBy(~timeorder, data=simdat)
> simdatPerm2 <- simdat[order(simdat$timeorder),]
> geeglm(yvar~tvar, id=idvar, data=simdatPerm2, corstr="ar1")

Call:
geeglm(formula = yvar ~ tvar, data = simdatPerm2, id = idvar,
       corstr = "ar1")

Coefficients:
(Intercept)      tvar
   0.650570    2.112266

Degrees of Freedom: 30 Total (i.e. Null);  28 Residual

Scale Link:              identity
Estimated Scale Parameters: [1] 1.339184

Correlation: Structure = ar1    Link = identity
Estimated Correlation Parameters:
alpha
    0

Number of clusters:   30    Maximum cluster size: 1
```

To obtain the right result we must give the `waves` argument:

```
> wav <- simdatPerm$timeorder
> wav

[1] 2 4 1 3 5 4 5 2 1 3 2 3 4 5 1 5 4 2 1 3 3 4 5 1 2 2 5 4 1 3

> mod3 <- geeglm(yvar~tvar, id=idvar, data=simdatPerm, corstr="ar1", waves=wav)
> mod3

Call:
geeglm(formula = yvar ~ tvar, data = simdatPerm, id = idvar,
       waves = wav, corstr = "ar1")

Coefficients:
(Intercept)      tvar
   1.084046    1.981558

Degrees of Freedom: 30 Total (i.e. Null);  28 Residual

Scale Link:              identity
Estimated Scale Parameters: [1] 1.390683

Correlation: Structure = ar1    Link = identity
Estimated Correlation Parameters:
alpha
0.5638696

Number of clusters:    6    Maximum cluster size: 5
```

3 Using a fixed correlation matrix and the `zcor` argument

Suppose we want to use a fixed working correlation matrix:

```

> cor.fixed <- matrix(c(1, 0.5, 0.25, 0.125, 0.125,
+ 0.5, 1, 0.25, 0.125, 0.125,
+ 0.25, 0.25, 1, 0.5, 0.125,
+ 0.125, 0.125, 0.5, 1, 0.125,
+ 0.125, 0.125, 0.125, 0.125, 1), 5, 5)
> cor.fixed

      [,1] [,2] [,3] [,4] [,5]
[1,] 1.000 0.500 0.250 0.125 0.125
[2,] 0.500 1.000 0.250 0.125 0.125
[3,] 0.250 0.250 1.000 0.500 0.125
[4,] 0.125 0.125 0.500 1.000 0.125
[5,] 0.125 0.125 0.125 0.125 1.000

```

Such a working correlation matrix has to be passed to `geeglm()` as a vector in the `zcor` argument. This vector can be created using the `fixed2Zcor()` function:

```

> zcor <- fixed2Zcor(cor.fixed, id=simdatPerm$idvar, waves=simdatPerm$timeorder)
> zcor

 [1] 0.125 0.500 0.250 0.125 0.125 0.500 0.125 0.250 0.125 0.125 0.125 0.125
[13] 0.125 0.500 0.125 0.125 0.125 0.500 0.250 0.250 0.250 0.125 0.125 0.500
[25] 0.500 0.125 0.250 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
[37] 0.500 0.500 0.250 0.250 0.500 0.125 0.250 0.250 0.125 0.125 0.125 0.125
[49] 0.125 0.500 0.125 0.125 0.500 0.250 0.125 0.125 0.125 0.125 0.500 0.250

```

Notice that `zcor` contains correlations between measurements within the same cluster. Hence if a cluster contains only one observation, then there will be generated no entry in `zcor` for that cluster. Now we can fit the model with:

```

> mod4 <- geeglm(yvar~tvar, id=idvar, data=simdatPerm, corstr="fixed", zcor=zcor)
> mod4

Call:
geeglm(formula = yvar ~ tvar, data = simdatPerm, id = idvar,
       zcor = zcor, corstr = "fixed")

Coefficients:
(Intercept)      tvar 
 0.9196243    2.0271640 

Degrees of Freedom: 30 Total (i.e. Null);  28 Residual

Scale Link:              identity
Estimated Scale Parameters: [1] 1.360649

Correlation: Structure = fixed   Link = identity
Estimated Correlation Parameters:
alpha:1
      1

Number of clusters: 6   Maximum cluster size: 5

```