On the usage of the `geepack`

Søren Højsgaard

January 27, 2010

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

   idvar timeorder       tvar      yvar
1      1         1  1.5040657  5.241757
2      1         2  2.7744803  7.116769
3      1         3  3.6255102  8.043483
4      1         4  4.4472244 11.253735
5      1         5  5.3685155 13.814593
6      2         1  0.8200200  2.453322
7      2         2  0.3141491  1.041775
8      2         3  2.5373702  5.878853
9      2         4  2.7017999  6.214465
10     2         5  4.2511560  9.622072
11     3         1 -0.3265481  1.723997
12     3         2  2.5634857  7.430754
```

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 
   1.367698    2.132899 

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

Scale Link:                   identity
Estimated Scale Parameters:  [1] 1.220929

Correlation:  Structure = ar1    Link = identity 
Estimated Correlation Parameters:
    alpha 
0.5490498 

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 observatios 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)
> head(simdatPerm)

  idvar timeorder     tvar      yvar
2     1         2 2.774480  7.116769
4     1         4 4.447224 11.253735
1     1         1 1.504066  5.241757
3     1         3 3.625510  8.043483
5     1         5 5.368515 13.814593
9     2         4 2.701800  6.214465
```

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 
   1.645094    2.044937 

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

Scale Link:                   identity
Estimated Scale Parameters:  [1] 1.226304

Correlation:  Structure = ar1    Link = identity 
Estimated Correlation Parameters:
    alpha 
0.5836987 

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

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

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

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

Coefficients:
(Intercept)        tvar
   1.410261    2.093953

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

Scale Link:                   identity
Estimated Scale Parameters:  [1] 1.211901

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
   1.367698    2.132899

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

Scale Link:                   identity
Estimated Scale Parameters:  [1] 1.220929

Correlation:  Structure = ar1    Link = identity
Estimated Correlation Parameters:
    alpha
0.5490498

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
   1.427171    2.075243

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

Scale Link:                   identity
Estimated Scale Parameters:  [1] 1.214241

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

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