On the usage of the geepack

Søren Højsgaard

November 7, 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)
> 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 -0.34108882  0.4383566
2      1         2  0.37299536  2.1330306
3      1         3  3.15943643 10.7664282
4      1         4  3.52809222 10.0405388
5      1         5  5.46090430 13.0903519
6      2         1  0.74989532  4.0921461
7      2         2  1.33010806  2.3767854
8      2         3  3.73483782  8.4791778
9      2         4  3.88581008  8.8664917
10     2         5  4.86544924 10.4001096
11     3         1  0.44974452  0.9157016
12     3         2  0.07736249  3.0465948
```

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
  0.9560893   2.0777361

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

Scale Link:                  identity
Estimated Scale Parameters:  [1] 2.095242

Correlation:  Structure = ar1    Link = identity
Estimated Correlation Parameters:
    alpha
0.3938552

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  0.3729954  2.1330306
4     1         4  3.5280922 10.0405388
1     1         1 -0.3410888  0.4383566
3     1         3  3.1594364 10.7664282
5     1         5  5.4609043 13.0903519
9     2         4  3.8858101  8.8664917
```

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
  0.9950998   2.0531706

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

Scale Link:                  identity
Estimated Scale Parameters:  [1] 2.092175

Correlation:  Structure = ar1    Link = identity
Estimated Correlation Parameters:
     alpha
0.06148152

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
   1.011533    2.048190

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

Scale Link:                  identity
Estimated Scale Parameters:  [1] 2.092081

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
  0.9560893   2.0777361

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

Scale Link:                  identity
Estimated Scale Parameters:  [1] 2.095242

Correlation:  Structure = ar1    Link = identity
Estimated Correlation Parameters:
    alpha
0.3938552

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
   1.004214    2.038155

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

Scale Link:                  identity
Estimated Scale Parameters:  [1] 2.093476

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

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