On the usage of the `geepack`

Søren Højsgaard and Ulrich Halekoh

geepack version 1.2-0 as of 2014-09-13

1 Introduction


```
> library(geepack)
> citation("geepack")
```

To cite geepack in publications use:


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)
```

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
1.820063      1.764176

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

Scale Link: identity
Estimated Scale Parameters: [1] 1.362202

Correlation: Structure = ar1 Link = identity
Estimated Correlation Parameters:
  alpha
0.5504676

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
1       2          2 0.4171669  1.5808820
2       1          1 0.9721636 11.9801894
3       1          1 0.5504676  0.5808820
4       1          1 0.4171669  1.5808820
5       2          1 2.3225641  7.1547640
6       1          1 0.5837173  0.6757944

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

Fitting the model as before gives
```r
> 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
    1.650260   1.835375

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

Scale Link: identity
Estimated Scale Parameters: [1] 1.324928

Correlation: Structure = ar1   Link = identity
Estimated Correlation Parameters:
 alpha
     0.5148134

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):

```r
> ## 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
    1.586836   1.863810

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

Scale Link: identity
Estimated Scale Parameters: [1] 1.321515

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:
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)
```

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)
```

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
      1.775079  1.805614

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

Scale Link: identity
Estimated Scale Parameters: [1] 1.335053

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

Number of clusters: 6  Maximum cluster size: 5