On the usage of the **geepack**

Søren Højsgaard and Ulrich Halekoh

**geepack** version 1.3-2 as of 2020-12-17

---

Contents

1 Introduction 1

2 Citing **geepack** 1

3 Simulating a dataset 2

4 Using the **waves** argument 3

5 Using a fixed correlation matrix and the **zcor** argument 5

6 When do GEE’s work best? 6

---

1 Introduction

This note contains a few extra examples. We illustrate the usage of the **waves** argument and the **zcor** argument together with a fixed working correlation matrix for the **geeglm()** function.

2 Citing **geepack**

The primary reference for the **geepack** package is


[https://www.jstatsoft.org/article/view/v015i02](https://www.jstatsoft.org/article/view/v015i02)
To cite `geepack` in publications use:


To see these entries in BibTeX format, use `print(citation, bibtex=TRUE)`, `toBibtex(.)`, or set `options(citation.bibtex.max=999)`.

If you use `geepack` in your own work, please do cite the above reference.

### 3 Simulating a dataset

To illustrate the usage of the `waves` argument and the `zcor` argument together with a fixed working correlation matrix for the `geeglm()` we simulate some data suitable for a regression model.

```r
> 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 1.0129037 4.748418
 2       1        2 2.8372421 4.702714
 3       1        3 2.1081591 5.992704
 4       1        4 3.0879299 6.654721
 5       1        5 4.8892114 12.329905
 6       2        1 0.6006967 2.612684
 7       2        2 3.2557060 7.672846
 8       2        3 0.9603749 2.284175
 9       2        4 3.8575990 9.324752
 10      3        1 2.0969177 7.743479
 11      3        2 2.5670947 6.927915
 12      3        3 2.5670947 6.927915
```

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
    2.058697    1.874905

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

Scale Link: identity
Estimated Scale Parameters: [1] 2.50573

Correlation: Structure = ar1 Link = identity
Estimated Correlation Parameters:
    alpha
          0.5576376

Number of clusters: 6  Maximum cluster size: 5

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

4 Using the waves argument

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

```r
> 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
 3      1          3 2.108159  5.992704
 5      1          5 4.889211 12.329905
 4      1          4 3.087930  6.654721
 1      1          1 1.012904  4.748418
 2      1          2 2.837242  4.702714
10     2          5 4.853958 12.852980
```

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
2.028442 1.830476
Degrees of Freedom: 30 Total (i.e. Null); 28 Residual
Scale Link: identity
Estimated Scale Parameters: [1] 2.528465
Correlation: Structure = ar1 Link = identity
Estimated Correlation Parameters:
alpha
0.5350554
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
1.747519 1.953259
Degrees of Freedom: 30 Total (i.e. Null); 28 Residual
Scale Link: identity
Estimated Scale Parameters: [1] 2.482827
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:
5 Using a fixed correlation matrix and the \texttt{zcor} argument

Suppose we want to use a fixed working correlation matrix:

\begin{verbatim}
> 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,] 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
\end{verbatim}

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

\begin{verbatim}
> zcor <- fixed2Zcor(cor.fixed, id=simdatPerm$idvar, waves=simdatPerm$timeorder)
> zcor
[1,] 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
[2,] 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
[3,] 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
[4,] 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
[5,] 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
\end{verbatim}

Notice that \texttt{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 \texttt{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
1.87526060  1.91911095

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

Scale Link: identity
Estimated Scale Parameters: [1] 2.486674

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

Number of clusters: 6    Maximum cluster size: 5

6 When do GEE’s work best?

GEEs work best when you have relatively many relatively small clusters in your data.