Nonparametric estimation of state occupation probabilities from multistate models with current-status data

Samuel Anyaso-Samuel

September 11, 2022

Introduction

In the first part of this vignette, we explain the utility of the `mspack2` package for the nonparametric estimation of state occupation probabilities (SOP) of a multistate model with current-status data. In the second part of the vignette, we describe the pseudo-value regression for estimating the effects of covariates on the SOP. Throughout this article, we focus on the case where the current-status data are cluster-correlated and the cluster sizes are informative. The methods described here can be easily adapted to the scenario where the data are uncorrelated.

The `SOP` function in the `mspack2` package depends on two packages which users may not have installed. These packages can be installed running the following

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("graph")
BiocManager::install("Rgraphviz")
```

First, we load the required packages

```
library(mspack2)
library(graph)
library(Rgraphviz)
library(isotone)
library(KernSmooth)
library(geeM)
```

Three-state tracking model

We simulate clustered transition times from a three-state tracking model where all subjects begin at state 1 and eventually end up in state 3.

```
Nodes <- c("1","2","3")
Edges <- list("1"=list(edges=c("2")),
              "2"=list(edges=c("3")),
              "3"=list(edges=NULL))
tree <- new("graphNEL", nodes=Nodes, edgeL=Edges, edgemode="directed")
plot(tree)
```

For the clusters, $i = 1, ..., m$ and subjects, $j = 1, ..., n_i$, the cluster-correlated exit times from state 1 are simulated from a lognormal AFT model with two covariates: a cluster-level exposure covariate denoted by Z_1 and a subject-level continuous covariate denoted by Z_2. The exit times from state 2 are simulated by a transformation which ensures that the exit times from state 2 are greater than the exit times from state 1. Lastly, we simulate the clustered inspection times from a Weibull distribution shape=3 and scale=5. The simulated data comprise the triplet $(C_{ij}, S(C_{ij}), \mathbf{Z_{ij}})$ where C_{ij} is the inspection time, $S(C_{ij})$ denotes the state occupied at the time of inspection, and $\mathbf{Z_{ij}}$ denotes the covariate vector. Moreover, we simulate cluster sizes n_i from a Poisson distribution where the mean depends on Z_1 and a cluster-specific random effects term. In the package, we provide an example of the simulated data with $m = 200$ clusters where the cluster sizes n_i are informative.

```
head(CSdata)
#>   cID id     time state Z1        Z2 csize
#> 1   1  1 5.552977     3  1 0.9316346     3
#> 2   1  2 4.075666     3  1 1.1635907     3
#> 3   1  3 5.059416     1  1 0.9576991     3
#> 4   2  4 3.513745     3  1 1.0237992     2
#> 5   2  5 1.852104     1  1 0.8982381     2
#> 6   3  6 4.498849     3  1 0.9889437     2
```

Estimating the SOP

Now, we obtain the nonparametric marginal estimates of the occupation probability of the three states. We estimate the probabilities using the `SOP` function, this function has the capability of also estimating the conditional probabilities given a continuous covariate. Further, if the argument `weight="ICS"`, then the function uses inverse cluster size reweighting to adjust for informative cluster size.

```
# initial probabilities for each state
start.probs <- c(1, 0, 0)
names(start.probs) <- nodes(tree)

# estimates the SOP
res <- SOP(data.type='cluster-correlated',dat=CSdata, tree=tree,
           start.probs=start.probs, ngrid=1000, weight='ICS', pavY=TRUE)
```

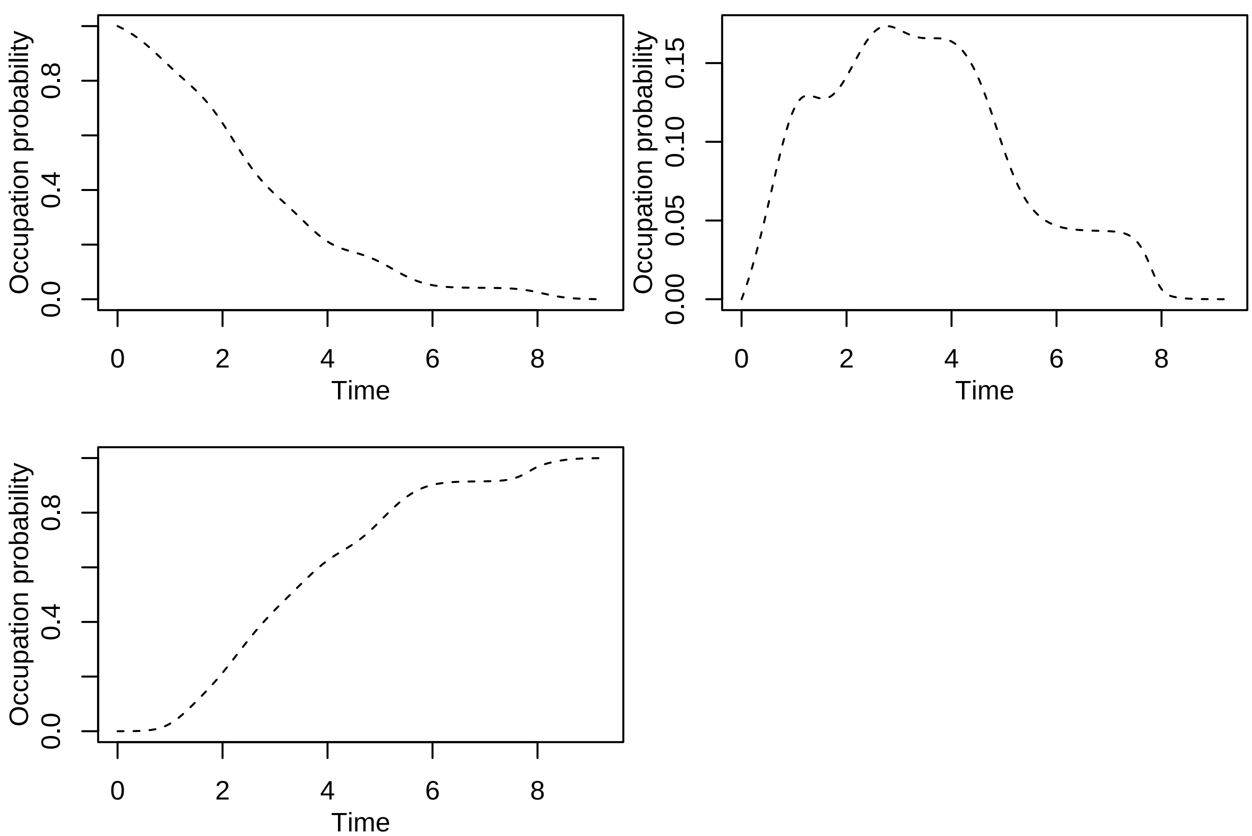
The plots of the estimated state occupation probabilities are shown below.

```
par(mfrow=c(2,2), mar=c(4,3,0.3,0.1))
et <- as.numeric(res[, c("time")])

plot(et, res[, c("p1")], type="n",mgp=c(2,1,0), xlab="Time", ylab="Occupation probability")
lines(et, res[, c("p1")], lty=2)

plot(et, res[, c("p2")], type="n",mgp=c(2,1,0), xlab="Time", ylab="Occupation probability")
lines(et, res[, c("p2")], lty=2)

plot(et, res[, c("p3")], type="n",mgp=c(2,1,0), xlab="Time", ylab="Occupation probability")
lines(et, res[, c("p3")], lty=2)
```



If the estimates of the SOP are desired at certain time points, then the argument `cutoffs` should be used in the `SOP` function.

Pseudo-value regression

The pseudo-value approach is a flexible method for performing covariate inference with cluster-correlated current status data. The procedure proceeds in two steps: (i) marginal estimation of SOP, and (ii) fitting the estimating equations based on pseudo-value responses.

Let $\widehat{\pi}_\ell(t)$ denote the marginal occupation probability estimate for state ℓ, the jackknife pseudo-values are based on the marginal estimates of the SOP, they are defined by

$$Y_{ij}(t) = n \cdot \widehat{\pi}_\ell(t) - (n-1)\widehat{\pi}_{\ell,-ij}(t),\ i = 1, ..., m;\ j = 1, ..., n_i$$

where $n = \sum_{i=1}^m n_i$, and $\widehat{\pi}_{\ell,-ij}(t)$ is obtained by omitting the ith subject. The $Y_{ij}(t)$ are now used as the responses in a marginal model to estimate the effects of available covariates. Readily available software (e.g the `geem` function from the `geeM` package) can be used to estimate the covariate effects. Suppose we desire inference at $t = 3$, we perform pseudo-value regression by running.

```
## Step-1 - obtain the marginal SOP
res <- SOP(data.type='cluster-correlated',dat=CSdata, tree=tree,
           start.probs=start.probs, ngrid=1000, weight='ICS', pavY=TRUE, cutoffs=3)
res <- res[, !(names(res) %in% c("time"))]

## Compute the pseudo-values
covs <- CSdata[, !(names(CSdata) %in% c("times","state"))]
covs <- dplyr::distinct(covs,id,.keep_all=T)

ids <- sort(unique(CSdata$id)) # unique ids
n <- length(ids) # total number of observations

ps_vals <- matrix(0,nrow=n,ncol=ncol(covs)+ncol(res))
colnames(ps_vals) <- c(colnames(covs), colnames(res))

for(ijs in 1:n) {
  # specifies the ij-th observation to be omitted
  ij = ids[ijs]

  # removes the ij-th observation from the data
  temp_dat <- CSdata[which(CSdata[,"id"] != ij), ]

  # computes the leave-one-out statistic
  tmp0 <- SOP(data.type='cluster-correlated',dat=temp_dat, tree=tree,
              start.probs=start.probs, ngrid=1000, weight='ICS', pavY=TRUE, cutoffs=3)
  tmp0 <- tmp0[, !(names(tmp0) %in% c("time"))]

  # computes the jackknife pseudo-values
  pseudo <- n*res - (n-1)*tmp0
  pseudo <- matrix(pseudo, nrow = 1)
  names(pseudo) <- c("p1", "p2", "p3")
  rownames(pseudo) <- NULL
  ps_vals[ijs,] <- unlist(c(subset(covs, id==ij), pseudo))
}

## Step-2 - Fit the marginal models for state 1
ps_vals <- as.data.frame(ps_vals)
ps_vals$weights <- 1/ps_vals$csize

CWGEE <- geem(formula= p1 ~ Z1 + Z2, data=ps_vals, useP=T, weights = weights,
```

```r
              id="cID", family=gaussian(link="identity"), corstr="independence")
GEE <- geem(formula=p1 ~ Z1 + Z2, data=ps_vals, useP=T,
            id="cID", family=gaussian(link="identity"), corstr="independence")

# print results
summary(CWGEE)
#>              Estimates Model SE Robust SE    wald       p
#> (Intercept)    -0.2070   0.5016    0.7024 -0.2947 0.76820
#> Z1              0.5456   0.1411    0.2564  2.1280 0.03334
#> Z2              0.1918   0.4952    0.7199  0.2664 0.78990
#>
#>  Estimated Correlation Parameter:  0
#>  Correlation Structure:  independence
#>  Est. Scale Parameter:  0.9943
#>
#>  Number of GEE iterations: 2
#>  Number of Clusters:  200    Maximum Cluster Size:  81
#>  Number of observations with nonzero weight:  1389
summary(GEE)
#>              Estimates Model SE Robust SE   wald       p
#> (Intercept)     0.1130   0.2821    0.2329 0.4852 0.62750
#> Z1              0.3970   0.1036    0.1964 2.0210 0.04323
#> Z2              0.2003   0.2807    0.2321 0.8629 0.38820
#>
#>  Estimated Correlation Parameter:  0
#>  Correlation Structure:  independence
#>  Est. Scale Parameter:  2.352
#>
#>  Number of GEE iterations: 2
#>  Number of Clusters:  200    Maximum Cluster Size:  81
#>  Number of observations with nonzero weight:  1389
```