Analysis of multivariate competing risks data

Klaus Holst & Thomas Scheike

March 3, 2020

Overview

- marginal modelling with standard errors cif,
- cause specific hazards
- cumulative incidence modelling
  - random effects simple cif
  - Luise model

When looking at multivariate survival data with the aim of learning about the dependence that is present, possibly after correcting for some covariates different approaches are available in the mets package

- Binary models and adjust for censoring with inverse probabilty of censoring weighting
- Bivariate surival models of Clayton-Oakes type
  - With regression structure on dependence parameter
  - With additive gamma distributed random effects
  - Special functionality for polygenic random effects modelling such as ACE, ADE ,AE and so forth.
- Plackett OR model model
  - With regression structure on OR dependence parameter
- Cluster stratified Cox

Typically it can be hard or impossible to specify random effects models with special structure among the parameters of the random effects. This is possible for our specification of the random effects models.

To be concrete about the model structure assume that we have paired binomial data $T_1, \delta_1, T_2, \delta_2, X_1, X_2$ where the censored survival responses are $T_1, \delta_1, T_2, \delta_2$ and we have covariates X_1, X_2.

The focus of this vignette is describe how to work on bivariate survival data using the addtive gamma-random effects models. We present two different ways of specifying different dependence structures.

The basic models assumes that each subject has a marginal on Cox-form

$$\lambda_0(t) \exp(X_{ki}^T \beta)$$

then two types of models can be considered.

- Univariate models with a single random effect for each cluster and with a regression design on the varince.
- Multivariate models with multiple random effects for each cluster.

The univariate models are then given a given cluster random effects Z_k with parameter θ the joint survival function is given by the Clayton copula and on the form

$$\psi(\theta, \psi^{-1}(\theta, S_1(t, X_{k1})) + \psi^{-1}(\theta, S_1(t, X_{k1}))$$

where ψ is the Laplace transform of a gamma distributed random variable with mean 1 and variance θ.

We then model the variance within clusters by a cluster specific regression design such that

$$\theta = z_j^T \alpha$$

where z is the regression design (specified by theta.des in the software).

This model can be fitted using a pairwise likelihood or the pseudo-likelihood using either

- twostage
- twostageMLE

For the Multivariate models we are given a multivarite random effect each subject $(Z_1, ..., Z_d)$ with d random effects. The total random effect for each subject is then specified using a regression design on these random effects, with a regression vector v_j such that the total random effect is $\{v_1^T (Z_{1,\ldots,Zd})\}$. Each random effect has an associated parameter $(\lambda_1, ..., \lambda_d)$ and Z_j is Gamma distributed with

- mean $lambda_j / v_1^T \lambda$
- variance \($\lambda_j/(v_1^T \lambda)^2$}.

The key assumption to make the two-stage fitting possible is that

$$lamtot = v_j^T \lambda$$

with clusters.

The DEFAULT parametrization (var.par=1) uses the variances of the random effecs

$$\theta_j = \lambda_j / (v_1^T \lambda)^2$$

For alternative parametrizations one can specify how the parameters relate to λ_j with the argument var.par=0.

For both types of models the basic model assumptions are that given the random effects of the clusters the survival distributions within a cluster are independent and ' on the form

$$P(T > t|x, z) = exp(-Z \cdot Laplace^{-1}(lamtot^{-1}, S(t|x)))$$

with the inverse laplace of the gamma distribution with mean 1 and variance 1/lamtot.

Finally the parameters $(\lambda_1, ..., \lambda_d)$ are related to the parameters of the model by a regression construction M (d x k), that links the d λ parameters with the k underlying α parameters

$$\lambda = M\alpha$$

here using theta.des to specify these low-dimension association. Default is a diagonal matrix. This can be used to make structural assumptions about the variances of the random-effects as is needed for the ACE model for example. In software M is called theta.des

We consider K independent clusters, with n_k subject within each cluster. For each cluster we are given a set of independent random effects $V = (V_1, \ldots, V_m)^T$. We let $(V_1, \ldots, V_m)^T$ be independent Gamma distributed with $V_l \sim \Gamma(\eta_l, \nu_l), l = 1, \ldots, p$ independent gamma distributed random variables such that $E(V_l) = \eta_l/\nu$ and $Var(V_l) = \eta_l/\nu^2$. %%Let $\nu = (\nu_1, \ldots, \nu_p)$. The $\eta = (\eta_1, \ldots, \eta_m)$ parameters are given such that $\eta = D\theta$. Letting the rows in the matrix be denoted as $Q_i, \ldots, Q_m$. %%%As is commonly done [1] [1]; and

To facilitate our two-stage construction we also assume that $\nu = Q_i^T\eta$ for all $i = 1, \ldots, n_k$ such that $Q_i^T V$ is also Gamma distributed with $\Gamma(1, \nu)$, that is has variance ν^{-1} and mean 1. We get back to specific models where this is the case, but this assumption is often reasonable and needed [2] [2]; and

Let $\Psi(\eta_l, \nu, \cdot)$ denote the Laplace transform of the Gamma distribution $\Gamma(\eta_l, \nu)$, and let its inverse be $\Psi^{-1}(\eta_l, \nu, \cdot)$. For simplicity we also assume that η is the same across clusters.

Assume that the marginal survival distribution for subject i within cluster k is given by $S_{X_{k,i}}(t)$ given covariates $X_{k,i}$.

Now given the random effects of the cluster V_k and the covariates$X_{k,i}$ $i = 1, \ldots, n_k$ we assume that subjects within the cluster are independent with survival distributions

$$\exp(-(Q_{k,i}V_k)\Psi^{-1}(\nu, \nu, S_{X_{k,i}}(t))).$$

A consequence of this is that the hazards given the covariates $X_{k,i}$ and the random effects V_k are given by

$$\lambda_{k,i}(t; X_{k,i}, V_{k,i}) = (Q_{k,i}V_k)D_3\Psi^{-1}(\nu, \nu, S_{X_{k,i}}(t))D_t S_{X_{k,i}}(t) \qquad (1)$$

where D_t and D_3 denotes the partial derivatives with respect to t and the third argument, respectively.

Further, we can express the multivariate survival distribution as

$$\begin{aligned} S(t_1, \ldots, t_m) &= \exp(-\sum_{i=1}^{m}(Q_i V)\Psi^{-1}(\eta_l, \nu_l, S_{X_{k,i}}(t_i))) \\ &= \prod_{l=1}^{p}\Psi(\eta_l, \eta, \sum_{i=1}^{m} Q_{k,i}\Psi^{-1}(\eta, \eta, S_{X_{k,i}}(t_i))). \end{aligned} \qquad (2)$$

In the case of considering just pairs, we write this function as $C(S_{k,i}(t), S_{k,j}(t))$.

In addition to survival times from this model, we assume that we independent right censoring present $U_{k,i}$ such that the given V_k and the covariates$X_{k,i}$ $i = 1, \ldots, n_k$ $(U_{k,1}, \ldots, U_{k,n_k})$ of $(T_{k,1}, \ldots, T_{k,n_k})$, and the conditional censoring distribution do not depend on V_k. We can also express this via counting processes $N_{k,i}(t) = I(T_{k,i} < t, T_{k,i} < U_{k,i})$ and with at risk indicators $Y_{k,i}(t) = I(T_{k,i} > t, U_{k,i} > t)$, and the censoring indicators $\delta_{k,i} = I(T_{k,i} < U_{k,i})$.

%%%Due to the marginal specification we can estimate apply the two-stage approach %%%as in [3]. We return to this in the next section.

[3]; and

One consequence of the model strucure is that the Kendall's can be computed for two-subjects (i, j) across two clusters "1" and "2" as

$$E\left(\frac{(Q_{1i}V_1 - Q_{1j}V_2)(Q_{2i}V_1 - Q_{2j}V_2)}{(Q_{1i}V_1 + Q_{2i}V_2)(Q_{1j}V_1 + Q_{2j}V_2)}\right) \tag{3}$$

under the assumption that that we compare pairs with equivalent marginals ($S_{X_{1,i}}(t) = S_{X_{2,i}}(t)$ and $S_{X_{1,j}}(t) = S_{X_{2,j}}(t)$) and that $S_{X_{1,i}}(\infty) = S_{X_{1,j}}(\infty) = 0$. %%%We return to another charactetrization %%%of the dependence via the cross hazards ratio. Here we also use that η is the same across clusters. The Kendall's tau would be the same for (**??**) due to the same additive structure for the frailty terms, and the random effects thus have the same interpretation in terms of Kendall's tau.

Clusters stratified Cox models

Show how efficient the stratified Cox is with GOF and all

```
library(mets)
 data(diabetes)
 margph <- phreg(Surv(time,status)~treat+strata(id),data=
     diabetes)
```

```
library(mets)
gg <- gof (margph)

par(mfrow=c(2,2))
plot(gg)
```

Univariate plackett model twostage models

```
library(mets)
 data(diabetes)

 # Marginal Cox model with treat as covariate
 margph <- phreg(Surv(time,status)~treat+cluster(id),data=
     diabetes)
 # Clayton-Oakes, MLE
 fitco1<-twostageMLE(margph,data=diabetes,theta=1.0)
 summary(fitco1)

```

```
# Plackett model
mph <- phreg(Surv(time,status)~treat+cluster(id),data=
    diabetes)
fitp <- survival.twostage(mph,data=diabetes,theta=3.0,Nit
    =40,
      clusters=diabetes$id,var.link=1,model="plackett")
summary(fitp)

# Clayton-Oakes
fitco2 <- survival.twostage(mph,data=diabetes,theta=0.0,
    detail=0,
        clusters=diabetes$id,var.link=1,model="clayton.oakes
            ")
summary(fitco2)
fitco3 <- survival.twostage(margph,data=diabetes,theta=1.0,
    detail=0,
        clusters=diabetes$id,var.link=0,model="clayton.oakes
            ")
summary(fitco3)

# without covariates but with stratafied
marg <- phreg(Surv(time,status)~+strata(treat)+cluster(id),
    data=diabetes)
fitpa <- survival.twostage(marg,data=diabetes,theta=1.0,
       clusters=diabetes$id,score.method="optimize")
summary(fitpa)

fitcoa <- survival.twostage(marg,data=diabetes,theta=1.0,
    clusters=diabetes$id,
        model="clayton.oakes")
summary(fitcoa)


# Piecewise constant cross hazards ratio modelling
d <- subset(simClaytonOakes(2000,2,0.5,0,stoptime=2,left=0),
    !truncated)
udp <- piecewise.twostage(c(0,0.5,2),data=d,score.method="
    optimize",
            id="cluster",timevar="time",
            status="status",model="clayton.oakes",silent=0)
summary(udp)
```

Univariate gamma (clayton-oakes) model twostage models

Looking at the data

```
library(mets)
data(diabetes)

# Marginal Cox model with treat as covariate
margph <- phreg(Surv(time,status)~treat+cluster(id),data=
    diabetes)
# Clayton-Oakes, MLE
fitco1<-twostageMLE(margph,data=diabetes,theta=1.0)
summary(fitco1)

# Plackett model
```

```
mph <- phreg(Surv(time,status)~treat+cluster(id),data=
    diabetes)
fitp <- survival.twostage(mph,data=diabetes,theta=3.0,Nit
    =40,
      clusters=diabetes$id,var.link=1,model="plackett")
summary(fitp)

# Clayton-Oakes
fitco2 <- survival.twostage(mph,data=diabetes,theta=0.0,
    detail=0,
        clusters=diabetes$id,var.link=1,model="clayton.oakes
            ")
summary(fitco2)
fitco3 <- survival.twostage(margph,data=diabetes,theta=1.0,
    detail=0,
        clusters=diabetes$id,var.link=0,model="clayton.oakes
            ")
summary(fitco3)

# without covariates but with stratafied
marg <- phreg(Surv(time,status)~+strata(treat)+cluster(id),
    data=diabetes)
fitpa <- survival.twostage(marg,data=diabetes,theta=1.0,
       clusters=diabetes$id,score.method="optimize")
summary(fitpa)

fitcoa <- survival.twostage(marg,data=diabetes,theta=1.0,
    clusters=diabetes$id,
        model="clayton.oakes")
summary(fitcoa)


# Piecewise constant cross hazards ratio modelling
d <- subset(simClaytonOakes(2000,2,0.5,0,stoptime=2,left=0),
    !truncated)
udp <- piecewise.twostage(c(0,0.5,2),data=d,score.method="
    optimize",
            id="cluster",timevar="time",
            status="status",model="clayton.oakes",silent=0)
summary(udp)
```

Multivariate gamma twostage models

```
library(mets)

# structured random effects model additive gamma ACE
# simulate structured two-stage additive gamma ACE model
data <- simClaytonOakes.twin.ace(2000,2,1,0,3)
out <- twin.polygen.design(data,id="cluster")
pardes <- out$pardes
pardes
des.rv <- out$des.rv
head(des.rv)
aa <- phreg(Surv(time,status)~x+cluster(cluster),data=data,
    robust=0)
```

```
ts <- survival.twostage(aa,data=data,clusters=data$cluster,
    detail=0,
        theta=c(2,1),var.link=0,step=0.5,
        random.design=des.rv,theta.des=pardes)
summary(ts)
```

```
library(mets)

set.seed(1000)
source("mets/R/sim.clayton.oakes.R")
data <- simClaytonOakes.family.ace(8000,2,1,0,3)
head(data)
data$number <- c(1,2,3,4)
data$child <- 1*(data$number==3)
out <- ace.family.design(data,member="type",id="cluster")
out$pardes
head(out$des.rv)

aa <- aalen(Surv(time,status)~+1,data=data,robust=0)
pa <- phreg(Surv(time,status)~+1+cluster(cluster),data=data)

# additive gamma models with and without pair call
# make ace random effects design

# simple random effects call
ts0 <- twostage(aa,data=data,clusters=data$cluster,
   detail=1,var.par=1,var.link=0,
   theta=c(2,1),
   random.design=out$des.rv,theta.des=out$pardes)
summary(ts0)

ts00 <- twostage(pa,data=data,clusters=data$cluster,
   detail=1,var.par=1,var.link=0,
   theta=c(2,1),
   random.design=out$des.rv,theta.des=out$pardes)
summary(ts00)


checkderiv=0
if (checkderiv==1) {
ts0 <- twostage(aa,data=data,clusters=data$cluster,
   detail=1,numDeriv=1,Nit=0,var.par=1,
   theta=log(c(2,1)/9),var.link=1,step=1.0,
   random.design=out$des.rv,theta.des=out$pardes)
ts0$score
ts0$score1

ts0 <- twostage(aa,data=data,clusters=data$cluster,
   detail=1,numDeriv=1,Nit=0,var.par=1,
   theta=c(2,1)/9,var.link=0,step=1.0,
   random.design=out$des.rv,theta.des=out$pardes)
ts0$score
ts0$score1


ts0 <- twostage(aa,data=data,clusters=data$cluster,
   detail=1,numDeriv=1,Nit=0,var.par=0,
```

```
   theta=log(c(2,1)),var.link=1,step=1.0,
   random.design=out$des.rv,theta.des=out$pardes)
ts0$score
ts0$score1

ts0 <- twostage(aa,data=data,clusters=data$cluster,
   detail=1,numDeriv=1,Nit=0,var.par=0,
   theta=c(2,1),var.link=0,step=1.0,
   random.design=out$des.rv,theta.des=out$pardes)
ts0$score
ts0$score1

}


# now specify fitting via specific pairs

# first all pairs
mm <- familycluster.index(data$cluster)
head(mm$familypairindex,n=10)
pairs <- matrix(mm$familypairindex,ncol=2,byrow=TRUE)
tail(pairs,n=12)
# make all pairs and pair specific design and pardes
# same as ts0 but pairs specified
ts <- twostage(aa,data=data,clusters=data$cluster,
      theta=c(2,1),var.link=0,step=1.0,
      random.design=out$des.rv,
      theta.des=out$pardes,pairs=pairs)
summary(ts)

ts <- twostage(pa,data=data,clusters=data$cluster,
      theta=c(2,1),var.link=0,step=1.0,
      random.design=out$des.rv,
      theta.des=out$pardes,pairs=pairs)
summary(ts)


# random sample of pairs
ssid <- sort(sample(1:48000,20000))

# take some of all
tsd <- twostage(aa,data=data,clusters=data$cluster,
      theta=c(2,1)/10,var.link=0,step=1.0,
      random.design=out$des.rv,iid=1,
         theta.des=out$pardes,pairs=pairs[ssid,])
summary(tsd)

# same analyses but now gives only data that is used in the
    relevant pairs
ids <- sort(unique(c(pairs[ssid,])))

pairsids <- c(pairs[ssid,])
pair.new <- matrix(fast.approx(ids,c(pairs[ssid,])),ncol=2)
head(pair.new)

# this requires that pair.new refers to id's in dataid
    (survival, status and so forth)
# random.design and theta.des are constructed to be the
```

```
    array 3 dims via individual specfication from
    ace.family.design
dataid <- dsort(data[ids,],"cluster")
outid <- ace.family.design(dataid,member="type",id="cluster"
    )
outid$pardes
head(outid$des.rv)

tsdid <- twostage(aa,data=dataid,clusters=dataid$cluster,
      theta=c(2,1)/10,var.link=0,step=1.0,
      random.design=outid$des.rv,iid=1,
      theta.des=outid$pardes,pairs=pair.new)
summary(tsdid)
coef(tsdid)
coef(tsd)
# same as tsd


# now direct specification of random.design and
    theta.design
# rather than taking the rows of the des.rv for the
    relevant pairs
# can make a pair specific specification of random effects

pair.types <- matrix(dataid[c(t(pair.new)),"type"],byrow=T,
    ncol=2)
head(pair.new)
head(pair.types)

# here makes pairwise design , simpler random.design og
    pardes, parameters
# stil varg, varc
# mother, child, share half rum=c(1,1,0) ruc=c(1,0,1),
# thetadesmcf=rbind(c(0.5,0),c(0.5,0),c(0.5,0),c(0,1))
#
# father, child, share half ruf=c(1,1,0) ruc=c(1,0,1),
# thetadescf=rbind(c(0.5,0),c(0.5,0),c(0.5,0),c(0,1))
#
# child, child, share half ruc=c(1,1,0) ruc=c(1,0,1),
# thetadesmf=rbind(c(0.5,0),c(0.5,0),c(0.5,0),c(0,1))
#
# mother, father, share 0 rum=c(1,0) ruf=c(0,1),
# thetadesmf=rbind(c(1,0),c(1,0),c(0,1))

theta.des <- array(0,c(4,2,nrow(pair.new)))
random.des <- array(0,c(2,4,nrow(pair.new)))
# random variables in each pair
rvs <- c()
for (i in 1:nrow(pair.new))
{
    if (pair.types[i,1]=="mother" & pair.types[i,2]=="father
        ")
    {
    theta.des[,,i] <- rbind(c(1,0),c(1,0),c(0,1),c(0,0))
    random.des[,,i] <- rbind(c(1,0,1,0),c(0,1,1,0))
    rvs <- c(rvs,3)
    } else {
    theta.des[,,i] <- rbind(c(0.5,0),c(0.5,0),c(0.5,0),c
        (0,1))
```

```
158     random.des[,,i] <- rbind(c(1,1,0,1),c(1,0,1,1))
159     rvs <- c(rvs,4)
160     }
161 }
162 # 3 rvs here
163 random.des[,,7]
164 theta.des[,,7]
165 # 4 rvs here
166 random.des[,,1]
167 theta.des[,,1]
168 head(rvs)
169
170 tsdid2 <- twostage(aa,data=dataid,clusters=dataid$cluster,
171       theta=c(2,1)/10,var.link=0,step=1.0,
172       random.design=random.des,
173       theta.des=theta.des,pairs=pair.new,pairs.rvs=rvs)
174 summary(tsdid2)
175 tsd$theta
176 tsdid2$theta
177 tsdid$theta
178
179
180 # simpler specification via kinship coefficient for each
      pair
181
182 kinship <- c()
183 for (i in 1:nrow(pair.new))
184 {
185 if (pair.types[i,1]=="mother" & pair.types[i,2]=="father")
      pk1 <- 0 else pk1 <- 0.5
186 kinship <- c(kinship,pk1)
187 }
188 head(kinship,n=10)
189
190 out <- make.pairwise.design(pair.new,kinship,type="ace")
191 names(out)
192 # 4 rvs here , here independence since shared component has
      variance 0 !
193 out$random.des[,,9]
194 out$theta.des[,,9]
195
196
197 tsdid3 <- twostage(aa,data=dataid,clusters=dataid$cluster,
198       theta=c(2,1)/10,var.link=0,step=1.0,
199       random.design=out$random.design,
200       theta.des=out$theta.des,pairs=pair.new,pairs.rvs=out$
          ant.rvs)
201 summary(tsdid3)
202 coef(tsdid3)
203
204 # same as above tsdid2
205
206
207 # simple models, test for pairs structure
208
209 library(mets)
210
211 ts0 <- twostage(aa,data=data,clusters=data$cluster,
```

```
    detail=0,numDeriv=1,Nit=10,
    theta=c(0.17),var.link=0,step=1.0)
summary(ts0)
ts0$score; ts0$score1
ts0$Dscore; ts0$hess

mm <- familycluster.index(data$cluster)
head(mm$familypairindex,n=10)
pairs <- matrix(mm$familypairindex,ncol=2,byrow=TRUE)
head(pairs,n=12)
tail(pairs,n=12)
dim(pairs)

cc <- cluster.index(data$cluster)

ts0 <- twostage(aa,data=data,clusters=data$cluster,
    detail=1,Nit=0,
    theta=ts0$theta,var.link=0,pairs=pairs)
summary(ts0)



library(mets)

set.seed(100)
data <- simClaytonOakes.family.ace(8000,2,1,0,3)
head(data)
data$number <- c(1,2,3,4)
data$child <- 1*(data$number==3)

# make ace random effects design
out <- ace.family.design(data,member="type",id="cluster")
out$pardes
head(out$des.rv)

# makes marginal model (same for all)
aa <- aalen(Surv(time,status)~+1,data=data,robust=0)


mm <- familycluster.index(data$cluster)
head(mm$familypairindex,n=10)
pairs <- matrix(mm$familypairindex,ncol=2,byrow=TRUE)
head(pairs,n=12)
tail(pairs,n=12)
dim(pairs)
#

ts0 <- twostage(aa,data=data,clusters=data$cluster,
     detail=1,Nit=10,
    theta=c(0.2),var.link=0,step=1.0)
summary(ts0)

ts0 <- twostage(aa,data=data,clusters=data$cluster,
     detail=1,Nit=10,numDeriv=1,
    theta=c(0.2),var.link=0,step=1.0,pairs=pairs)
summary(ts0)
ts0$score
```

```
ts0$score1

ts0 <- twostage(aa,data=data,clusters=data$cluster,
    detail=1,Nit=10,
   theta=c(0.2),var.link=0,step=1.0,model="plackett")
summary(ts0)

ts0 <- twostage(aa,data=data,clusters=data$cluster,
    detail=1,Nit=10,
   theta=c(0.2),var.link=0,step=1.0,model="plackett",pairs=
       pairs)
summary(ts0)



theta.des <- model.matrix(~x1,data=data)

ts0 <- twostage(aa,data=data,clusters=data$cluster,
    detail=1,Nit=10,theta.des=theta.des,
   theta=c(0.2),var.link=0,step=1.0)
summary(ts0)

ts0 <- twostage(aa,data=data,clusters=data$cluster,
    detail=1,Nit=10,theta.des=theta.des,
   theta=c(0.2),var.link=0,step=1.0,pairs=pairs)
summary(ts0)

ts0 <- twostage(aa,data=data,clusters=data$cluster,
    detail=1,Nit=10,theta.des=theta.des,
   theta=c(0.2),var.link=0,step=1.0,model="plackett")
summary(ts0)

ts0 <- twostage(aa,data=data,clusters=data$cluster,
    detail=1,Nit=10,theta.des=theta.des,
   theta=c(0.2),var.link=0,step=1.0,model="plackett",pairs=
       pairs)
summary(ts0)
```