R-code for 'A time-heterogeneous D-vine copula model for unbalanced and unequally spaced longitudinal data'

Md Erfanul Hoque, Elif F. Acar and Mahmoud Torabi

This script generates a random samples of repeated measurement data from the time-heterogeneous D-vine model (HET–P) presented in Section 4 of *A time-heterogeneous D-vine copula model for unbalanced and unequally spaced longitudinal data* by Hoque, Acar and Torabi (2020) and illustrates the associated R functions that implement HET–P model to estimate the model parameters and conditional quantiles.

Load required packages

```
library(here) # to specify the folder where something is located in
library(VineCopula) # required to work with D-vine copulas
```

Simulation Example: Time-heterogeneous D-vine copula under DGP4

```
# Data example used for DGP 4 (HET--P) in manuscript (Section 4).
n <- 250 # sample size
d <-  5 # dimension

# Decay Parameter values
beta0 <- c(0, 0.9,0.53,0.34,0.06, 0,0,0.68,0.21,0.15,
           0,0,0,0.27,0.09, 0,0,0,0,0.11, 0,0,0,0,0)
beta0 <- matrix(beta0, d, d)

beta <- c(0,1.4,0.82,0.63,0.21, 0,0,0.94,0.78,0.42,
          0,0,0,0.69,0.28, 0,0,0,0,0.36, 0,0,0,0,0)
beta <- matrix(beta, d, d)

# Generate time variable
```

```r
set.seed(20192020)
Time <- t(sapply(1:n, function(i) Longt.Sim(d)))

# Calculate Kendall's Tau
Tau.Array <- sapply(1:n, function(i) LDVine.Tau(beta0,beta, Time[i,]),
                                                simplify="array")
# Copula families in D-vine matrix
Family <- c(0, 1, 1, 4, 4,
            0, 0, 5, 3, 3,
            0, 0, 0, 5, 5,
            0, 0, 0, 0, 1,
            0, 0, 0, 0, 0)
Family <- matrix(Family, nrow=d, ncol=d)

# Calculate copula parameters
Par.Array <- sapply(1:n, function(i) LDVine.Par(Family, Tau=Tau.Array[,,i]),
                                                simplify="array")
# Generate 5 dimensional balanced data
set.seed(20192020)
Full.data <- data.frame(t(sapply(1:n, function(i)  LDVine.Sim(d, Family,
                                                    Par.Array[,,i]))))
colnames(Full.data) <- c("U1","U2","U3","U4","U5")

# Save the generated data in "OUTPUT" folder
save(Full.data,file=here::here("DATA","DGP4-n250.Rdata"))

#############################################################################
# Obtain unbalanced data from the balanced data
#############################################################################
prob <- c(20, 20, 15, 40)/100 # prob of d_i, d_i=2,3,4,5

# Sample lengths of observations
set.seed(12345)
n.event <- sample(x = 2:d, size = n, replace = TRUE, prob = prob)

# Unbalanced/reduced data
U.RM <- matrix(NA, n, d)

# Reduced data length
for (i in 1:n){
  U.RM[i, 1:n.event[i]] <- unlist(Full.data[i, 1:n.event[i]])
}

## Proportion of individual with at least j meaurements
```

```
ind.num<- c()
for (i in 1:d){
  ind.num[i]=sum(!is.na(U.RM[,i]))/n
}
ind.num
```

```
## [1] 1.000 1.000 0.796 0.548 0.364
```

```
# Save the unbalanced data in "OUTPUT" folder
save(U.RM,file=here("DATA","Unbal-DGP4-n250.Rdata"))
```

D-vine copula model fitting

We fit a time-homogeneous D-vine copula model (HOM–P) and a time-heterogeneous D-vine copula model (HET–P) to the data.

Load copula data to fit D-vine copula models

```
U.data<- U.RM # unobserved measurements are denoted by 'NA'
dim(U.data)
```

```
## [1] 250   5
```

```
head(U.data) # show the copula data structure
```

```
##            [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.63389481 0.8145352 0.5331264        NA        NA
## [2,] 0.16161333 0.2109845 0.3687877 0.1712888        NA
## [3,] 0.85435886 0.8424680 0.8660496        NA        NA
## [4,] 0.75163683 0.7789428 0.6565303 0.7254135        NA
## [5,] 0.05365216 0.1176500        NA        NA        NA
## [6,] 0.95097744 0.9270849 0.6660275 0.8910279 0.7487819
```

Fit time-homogeneous D-vine copula model (HOM–P) to copula data

```
# Calculate Kendall's tau values
Tau.Array.True <- sapply(1:n, function(i) LDVine.Tau(beta0, beta,
                                                     Time[i,]), simplify="array")
Tau=apply(Tau.Array.True,1:2,mean) #average tau

# Calculate copula parameter values
Par <- matrix(0, d, d)
```

```r
Par2 = matrix(0,nrow=d,ncol=d)
 for (i in 2:d) {
   for (j in 1:(i-1)) {
     Par[i, j] <- BiCopTau2Par(Family[i, j], Tau[i, j])
   }
 }
# D-vine matrix with fixed order
 elements <- 1:d
  Mat <- diag(1:d)
  for (i in 2:d){
    Mat[d+2-i, 1:(d+1-i)] <- elements[i:d]
  }

# Construct the RVM matrix to fit HOM-P
RVM <- VineCopula::RVineMatrix(Mat, Family, Par, Par2)

# Fit HOM-P with fixed order 1:5
Dvine.HOM = RVineSeqEst(data = U.data, RVM=RVM)
```

```
## Warning: In RVineSeqEst: Some of the data are NA. Only pairwise complete
## observations are used.
```

```r
# The above warning only confirms that the data are unbalanced.
Dvine.HOM
```

```
## D-vine copula with the following pair-copulas:
## Tree 1:
## 2,1  Gumbel (par = 4.21, tau = 0.76)
## 3,2  Clayton (par = 3.3, tau = 0.62)
## 4,3  Frank (par = 10.45, tau = 0.68)
## 5,4  Gaussian (par = 0.8, tau = 0.59)
##
## Tree 2:
## 3,1;2  Gumbel (par = 1.3, tau = 0.23)
## 4,2;3  Clayton (par = 1.18, tau = 0.37)
## 5,3;4  Frank (par = 3.37, tau = 0.34)
##
## Tree 3:
## 4,1;3,2  Gaussian (par = 0.36, tau = 0.24)
## 5,2;4,3  Frank (par = 2.13, tau = 0.22)
##
## Tree 4:
## 5,1;4,3,2  Gaussian (par = 0.23, tau = 0.15)
##
## ---
```

```
## 1 <-> V1,   2 <-> V2,   3 <-> V3,   4 <-> V4,   5 <-> V5
```

```
# Copula log-likelihood value for HOM-P model
Cloglik_HOMByPair<- sum(Dvine.HOM$pair.logLik);Cloglik_HOMByPair
```

```
## [1] 580.3503
```

```
# Store the copula parameters and families for quantile prediction
par<-Dvine.HOM$par
parset.HOM <-list(c(par[5,1],par[5,2],par[5,3],par[5,4]),
                  c(par[4,1],par[4,2],par[4,3]),
                  c(par[3,1],par[3,2]),
                  c(par[2,1]))
fam<-Dvine.HOM$family
familyset.HOM <- list(c(fam[5,1],fam[5,2],fam[5,3],fam[5,4]),
                      c(fam[4,1],fam[4,2],fam[4,3]),
                      c(fam[3,1],fam[3,2]),
                      c(fam[2,1]))
```

Fit time-heterogeneous D-vine copula model (HET–P) to copula data

To fit the HET–P model, we first calculate time interval between two measurements. The fitted model gives the estimated values of the decay parameters for each pair copula.

```
family.set <- familyset.HET<- familyset.HOM
# Creating the Time variable for the analysis of HET-P
Visits <- data.frame(Time)
colnames(Visits)<-c("visit 1","visit 2","visit 3","visit 4","visit 5")
head(Visits)
```

```
##      visit 1   visit 2   visit 3   visit 4   visit 5
## 1 0.08467748 0.2955051 0.5651674 0.7908657 0.8729746
## 2 0.09827619 0.2086252 0.5393502 0.6561231 0.8350379
## 3 0.16025226 0.3524889 0.5305249 0.7222311 0.9257016
## 4 0.12370777 0.3470433 0.4869784 0.7179654 0.9194771
## 5 0.03441657 0.2606775 0.4518959 0.6713634 0.8304718
## 6 0.14975638 0.3770949 0.4375150 0.7656387 0.9912695
```

```
## BEGIN fitted model
# Calculate the time intervals for Tree 1
t <- sapply(1:(d-1), function(j) Visits[,j+1] - Visits[,j])

# Estimation of model parameters of pair copulas in Tree 1
fit<- sapply(1:(d-1), function(j) LBiCopFit(u1=U.data[,j], u2=U.data[,j+1],
                        x=t[,j],family=family.set[[1]][j],repar="pearson",
                        start = c(0.01,0.01)))
```

```
fit.beta.T1<- sapply(1:(d-1), function(j) c(fit[1,j],fit[2,j]))
ll.T1<-sum(unlist(fit[3,])) #loglikelihood value in Tree 1

# Estimate copula parameters using fitted copulas
tau.T1<- sapply(1:(d-1), function(j) LDVine.rho2tau(unlist(fit.beta.T1[,j]),
                                                    t[,j]))
par.T1<- sapply(1:(d-1), function(j) BiCopTau2Par(family.set[[1]][j],
                                                  tau.T1[,j]))
# Obtain the pseudo-conditional marginals for Tree 2
vv<-list()
for (j in 1:(d-2)){
  vv[[j]] <- BiCopHfunc2(u1=U.data[,j], u2=U.data[,j+1],par.T1[,j],
                                        family = family.set[[1]][j])
  vv[[(d-2)+j]] <- BiCopHfunc2(u1=U.data[,j+2], u2=U.data[,j+1],par.T1[,j+1],
                                        family =family.set[[1]][j+1])
}
# Calculate the time intervals for Tree 2
t<- sapply(1:(d-2), function(j) Visits[,j+2] - Visits[,j])

# List of available conditional data needed for estimation in Tree 2
condata1 <- list(vv[[1]],vv[[4]],vv[[2]],vv[[5]],vv[[3]],vv[[6]])

# Estimation of model parameters of pair copulas in Tree 2
fit<- sapply(1:(d-2), function(j) LBiCopFit(u1 = condata1[[2*j-1]],
               u2 = condata1[[2*j]], x=t[,j],family=family.set[[2]][j],
               repar="pearson",start = c(0.01,0.01)))

fit.beta.T2<- sapply(1:(d-2), function(j) c(fit[1,j],fit[2,j]))
ll.T2<-sum(unlist(fit[3,])) #loglikelihood value in Tree 2

# Estimate copula parameters using fitted copulas
tau.T2<- sapply(1:(d-2), function(j) LDVine.rho2tau(unlist(fit.beta.T2[,j]),
                                                    t[,j]))
par.T2<- sapply(1:(d-2), function(j) BiCopTau2Par(family.set[[2]][j],
                                                  tau.T2[,j]))
# Obtain the pseudo-conditional marginals for Tree 3
vvv<-list()
for (j in 1:(d-3)){
  vvv[[j]] <- BiCopHfunc2(u1=condata1[[2*j-1]], u2=condata1[[2*j]],
                                  par.T2[,j],family = family.set[[2]][j])
  vvv[[(d-3)+j]] <- BiCopHfunc2(u1=condata1[[2*j+2]], u2=condata1[[2*j+1]],
                                  par.T2[,j+1],family = family.set[[2]][j+1])
}
# Calculate the time intervals for Tree 3
```

```
t<- sapply(1:(d-3), function(j) Visits[,j+3] - Visits[,j])

# List of available conditional data needed for estimation in Tree 3
condata2 <- list(vvv[[1]],vvv[[3]],vvv[[2]],vvv[[4]])

# Estimation of model parameters of pair copulas in Tree 3
fit<- sapply(1:(d-3), function(j) LBiCopFit(u1 = condata2[[2*j-1]],
                u2 = condata2[[2*j]], x=t[,j], family=family.set[[3]][j],
                repar="pearson",start = c(0.01,0.01)))

fit.beta.T3<- sapply(1:(d-3), function(j) c(fit[1,j],fit[2,j]))
ll.T3<-sum(unlist(fit[3,])) #loglikelihood value in Tree 3

# Estimate copula parameters using fitted copulas
tau.T3<- sapply(1:(d-3), function(j) LDVine.rho2tau(unlist(fit.beta.T3[,j]),
                                                    t[,j]))
par.T3<- sapply(1:(d-3), function(j) BiCopTau2Par(family.set[[3]][j],
                                                  tau.T3[,j]))
# Obtain the pseudo-conditional marginals for Tree 4
vvvv<-list()
for (j in 1:(d-4)){
  vvvv[[j]] <- BiCopHfunc2(u1=condata2[[2*j-1]], u2=condata2[[2*j]],
                                  par.T3[,j],family = family.set[[3]][j])
  vvvv[[(d-4)+j]] <- BiCopHfunc2(u1=condata2[[2*j+2]], u2=condata2[[2*j+1]],
                                  par.T3[,j+1],family = family.set[[3]][j+1])
}

# Calculate the time intervals for Tree 4
t<- sapply(1:(d-4), function(j) Visits[,j+4] - Visits[,j])

# List of available conditional data needed for estimation in Tree 4
condata3 <- list(vvvv[[1]],vvvv[[2]])

# Estimation of model parameters of pair copulas in Tree 4
fit<- sapply(1:(d-4), function(j) LBiCopFit(u1 = condata3[[2*j-1]],
               u2 = condata3[[2*j]], x=t[,j], family=family.set[[4]][j],
               repar="pearson",start = c(0.01,0.01)))

fit.beta.T4<- sapply(1:(d-4), function(j) c(fit[1,j],fit[2,j]))
ll.T4<-sum(unlist(fit[3,])) #loglikelihood value in Tree 4

# Estimate copula parameters using fitted copulas
tau.T4<- sapply(1:(d-4), function(j) LDVine.rho2tau(unlist(fit.beta.T4[,j]),
                                                    t[,j]))
```

```
par.T4<- sapply(1:(d-4), function(j) BiCopTau2Par(family.set[[4]][j],
                                                  tau.T4[,j]))
## END fitted model

# Store the parameter estimates
beta.fit <- list(fit.beta.T1,fit.beta.T2,fit.beta.T3,fit.beta.T4)

# Store the copula parameters and families for quantile prediction
par.fit.HET <- list(par.T1,par.T2,par.T3,par.T4)
familyset.HET <- family.set

# Copula log-likelihood value for HET-P model
Cloglik_HETByPair <- ll.T1+ll.T2+ll.T3+ll.T4; Cloglik_HETByPair
```

```
## [1] 596.8212
```

Implementation of Conditional Quantile Prediction

We select a subject with 5 measurements. We pretend that the selected subject has only 4 measurements rather than 5 measurements and predict the conditional 5%-, 50%- and 95%-quantiles in copula scale for its fifth measurement based on the first four measurements.

```
# id of selected individual and corresponding vector of measurements
id.sel <- 15
U.sel <- U.data[id.sel,]
U.sel
```

```
## [1] 0.9082963 0.8458989 0.9831575 0.9218821 0.9621467
```

```
time.sel <- Visits[id.sel,]

# Quantile levels to be estimated
alpha.level <- c(0.05, 0.5, 0.95)
```

For the D-vine copula models HOM–P and HET–P, we use the copula data to calculate the D-vine quantile of the fifth measurement.

```
udata = rbind(U.sel[1:4])
```

Time-homogeneous D-vine copula model (HOM–P)

```
# Calculate the quantiles on the coupla scale
u_quant.p5 <-  DvineQuant (p = alpha.level[1], u = udata,
                     family.set = familyset.HOM, par.set = parset.HOM)
```

```
u_quant.p50 <- DvineQuant (p = alpha.level[2], u = udata,
                     family.set = familyset.HOM, par.set = parset.HOM)
u_quant.p95 <- DvineQuant (p = alpha.level[3], u = udata,
                     family.set = familyset.HOM, par.set = parset.HOM)
U.quant.HOM<-c(u_quant.p5,u_quant.p50,u_quant.p95)
U.quant.HOM
```

```
## [1] 0.7937391 0.9470042 0.9914879
```

```
# For a given marginal distribution, we can easily transform the estimated
# quantiles from the copula scale to the original scale
```

Time-heterogeneous D-vine copula model (HET–P)

```
# Estimated copula parameters from fitted HET-P model
parset.HET <- sapply(1:(d-1), function(j) par.fit.HET[[j]][id.sel,])

# Calculate the quantiles on the copula scale
u_quant.p5 <- DvineQuant (p = alpha.level[1], u = udata,
                        family.set = family.set, par.set = parset.HET)
u_quant.p50 <- DvineQuant (p = alpha.level[2], u = udata,
                        family.set = family.set, par.set = parset.HET)
u_quant.p95 <- DvineQuant (p = alpha.level[3], u = udata,
                        family.set = family.set, par.set = parset.HET)
U.quant.HET<-c(u_quant.p5,u_quant.p50,u_quant.p95)
U.quant.HET
```

```
## [1] 0.8365408 0.9578474 0.9934379
```

```
# For a given marginal distribution, we can easily transform the estimated
# quantiles from the copula scale to the original scale

# Save the copula prediction results in "OUTPUT" folder
U.quant<-rbind(U.quant.HOM,U.quant.HET)
rownames(U.quant)<-c("HOM","HET")
colnames(U.quant)<-c("5%","50%","95%")
U.quant
```

```
##            5%       50%       95%
## HOM 0.7937391 0.9470042 0.9914879
## HET 0.8365408 0.9578474 0.9934379
```

```
U.sel
```

```
## [1] 0.9082963 0.8458989 0.9831575 0.9218821 0.9621467
```

```
write.table(round(U.quant,2),file=here("OUTPUT","copula.quant.txt"))
write.table(U.sel,file=here("OUTPUT","copula.data.txt"))
```