R-code for Chapter 9: Comparing regular vine copula models

Claudia Czado

04 March, 2019

Contents

Required R-packages

- VineCopula
- rafalib
- TSP

Section 9.1 Akaike and Bayesian information criteria for regularvine copulas

Example 9.1 WINE3: Model selection based on Akaike and Bayesian information criteria

Read in data and set column names

```
reddata<-read.csv(file="winequality-red.csv",sep=";")
n<-length(reddata[,1])
colnames(reddata)<-c("acf","acv","acc","sugar","clor","sf","st","den","ph","sp",
                      "alc","quality")
acf<-reddata[,1]
```

```
acv<-reddata[,2]
acc<-reddata[,3]
```

Transform original data to copula data (udata) using ranks and then to marginal normalized data (zdata)

```
udata<-reddata
zdata<-reddata
for(i in 1:12){
udata[,i]<-rank(reddata[,i])/(n+1)
zdata[,i]<-qnorm(udata[,i])
}
wine3<-udata[,1:3]
```

Setting copula parameters using empirical pairwise Kendall's τ's and the copula families chosen in Example 4.3

```
theta.acfacv<-BiCopTau2Par(34,cor(acf,acv,method="kendall"))
theta.acfacc<-BiCopTau2Par(4,cor(acf,acc,method="kendall"))
theta.acvacc<-BiCopTau2Par(5,cor(acv,acc,method="kendall"))
theta<-c(theta.acfacv,theta.acfacc,theta.acvacc)
```

Create pseudo observations for first trees and find copula parameter using inversion of Kendall's τ

```
h13<-BiCopHfunc(udata[,1],udata[,3],family=4,par=theta.acfacc)$hfunc2
h23<-BiCopHfunc(udata[,2],udata[,3],family=5,par=theta.acvacc)$hfunc2
theta.12.3<-BiCopTau2Par(1,cor(h13,h23,method="kendall"))
h12<-BiCopHfunc(udata[,1],udata[,2],family=34,par=theta.acfacv)$hfunc2
h32<-BiCopHfunc(udata[,3],udata[,2],family=5,par=theta.acvacc)$hfunc2
theta.13.2<-BiCopTau2Par(4,cor(h12,h32,method="kendall"))
h21<-BiCopHfunc(udata[,2],udata[,1],family=34,par=theta.acfacv)$hfunc2
h31<-BiCopHfunc(udata[,3],udata[,1],family=4,par=theta.acfacc)$hfunc2
theta.23.1<-BiCopTau2Par(5,cor(h21,h31,method="kendall"))
```

PCC1 estmation

```
fam=c(4,5,1)
theta12=theta[1]
theta13=theta[2]
theta23=theta[3]
theta312<-c(theta13,theta23,theta.12.3)
RVM1<-C2RVine(order=c(3,1,2),family = fam,par=theta312)
fam.name<-c(BiCopName(4),BiCopName(5),BiCopName(1))
r3.itau.seq.pcc1<-RVineSeqEst(wine3,RVM1,method="itau")
r3.mle.seq.pcc1<-RVineSeqEst(wine3,RVM1,method="mle")
r3.mle.pcc1<-RVineMLE(wine3,RVM1)
```

```
## iter   10 value -816.862869
## final  value -816.915732
## converged
```

PCC2 estmation

```
fam=c(34,5,4)
theta213<-c(theta12,theta23,theta.13.2)
RVM2=C2RVine(order=c(2,1,3),family = fam,par=theta213)
r3.itau.seq.pcc2<-RVineSeqEst(wine3,RVM2,method="itau")
r3.mle.seq.pcc2<-RVineSeqEst(wine3,RVM2,method="mle")
r3.mle.pcc2<-RVineMLE(wine3,RVM2,se=TRUE)
```

```
## final  value -772.584081
## converged
```

PCC3 estmation

```
fam=c(34,4,5)
fam.name<-c(BiCopName(34),BiCopName(5),BiCopName(4))
theta123<-c(theta12,theta13,theta.23.1)
RVM3<-C2RVine(order=c(1,2,3),family = fam,par=theta123)
r3.itau.seq.pcc3<-RVineSeqEst(wine3,RVM3,method="itau")
r3.mle.seq.pcc3<-RVineSeqEst(wine3,RVM3,method="mle")
r3.mle.pcc3<-RVineMLE(wine3,RVM3,se=TRUE)
```

```
## iter   10 value -785.719881
## final  value -785.719919
## converged
```

Collecting results (df, loglik, AIC, BIC) for fitted PCC1, PCC2, PCC3

```
vine.out<-function(fit=fit.rv,data=wine3,digits=4){
df<-sum(abs(fit$par)>0)+sum(fit$par2>0)
out<-round(c(fit$logLik,df, fit$AIC,fit$BIC),digits)
names(out)<-c("loglik","par","AIC","BIC")
out
}

out.table.pcc123<-rbind(
  vine.out(fit=r3.itau.seq.pcc1,data=wine3),
vine.out(fit=r3.mle.seq.pcc1,data=wine3),
vine.out(fit=r3.mle.pcc1$RVM,data=wine3),
vine.out(fit=r3.itau.seq.pcc2,data=wine3),
vine.out(fit=r3.mle.seq.pcc2,data=wine3),
vine.out(fit=r3.mle.pcc2$RVM,data=wine3),
vine.out(fit=r3.itau.seq.pcc3,data=wine3),
vine.out(fit=r3.mle.seq.pcc3,data=wine3),
vine.out(fit=r3.mle.pcc3$RVM,data=wine3))
row.names(out.table.pcc123)<-c("PCC1.itau.seq","PCC1.mle.seq","PCC1.mle.joint",
                               "PCC2.itau.seq","PCC2.mle.seq","PCC2.mle.joint",
                               "PCC3.itau.seq","PCC3.mle.seq","PCC3.mle.joint")
out.table.pcc123
```

```
##                   loglik par       AIC       BIC
## PCC1.itau.seq  807.8791   3 -1609.758 -1593.627
## PCC1.mle.seq   816.8276   3 -1627.655 -1611.524
## PCC1.mle.joint 816.9157   3 -1627.832 -1611.700
## PCC2.itau.seq  763.9325   3 -1521.865 -1505.734
## PCC2.mle.seq   763.4292   3 -1520.858 -1504.727
## PCC2.mle.joint 772.5841   3 -1539.168 -1523.037
## PCC3.itau.seq  770.2684   3 -1534.537 -1518.406
## PCC3.mle.seq   784.0528   3 -1562.105 -1545.974
## PCC3.mle.joint 785.7199   3 -1565.440 -1549.308
```

Example 9.2 WINE7: Model selection based on AIC and BIC

Read in data and set column names

```
reddata<-read.csv(file="winequality-red.csv",sep=";")
n<-length(reddata[,1])
colnames(reddata)<-c("acf","acv","acc","sugar","clor","sf","st","den","ph","sp",
                     "alc","quality")
reddata7<-reddata[,c(1,2,3,5,7,8,9)]
```

Transform original data to copula data (udata) using ranks and then to marginal normalized data (zdata)

```
udata<-reddata
zdata<-reddata
for(i in 1:12){
udata[,i]<-rank(reddata[,i])/(n+1)
zdata[,i]<-qnorm(udata[,i])
}
udata7<-udata[,c(1,2,3,5,7,8,9)]
```

Fitting R-vines and Gauss R-vines using sequential estimation with restricted pair copula family set

```
fit.rv=RVineStructureSelect(udata7, familyset=c(1,5,6,4),
selectioncrit="AIC",
indeptest=FALSE, level=0.05)
fit.rv.ind=RVineStructureSelect(udata7, familyset=c(1,5,6,4),
 selectioncrit="AIC",
indeptest=TRUE, level=0.05)
fit.Gauss=RVineStructureSelect(udata7, familyset=1, selectioncrit="AIC",
                               indeptest=FALSE, level=0.05)
fit.Gauss.ind=RVineStructureSelect(udata7, familyset=1, selectioncrit="AIC",
                              indeptest=TRUE, level=0.05)
```

Fitting C-vines using sequential estimation with restricted pair copula family set

```r
fit.cv=RVineStructureSelect(udata7, familyset=c(1,5,6,4), selectioncrit="AIC",
                            indeptest=FALSE, level=0.05,type="CVine")
fit.cv.ind=RVineStructureSelect(udata7, familyset=c(1,5,6,4), selectioncrit="AIC",
                                indeptest=TRUE, level=0.05,type ="CVine")
```

Fitting D-vines using sequential estimation with restricted pair copula family set

Since the traveling salesman problem is a NP hard problem, the package TSP provides different choices as solution. To be consistent with the book we fix the order to the one utilized there, which is given by

den acf ph acc acv st clor

6 1 7 3 2 5 4

```r
d = dim(udata7)[2]
M = 1 - abs(TauMatrix(udata7))
hamilton = insert_dummy(TSP(M),label="cut")
sol = solve_TSP(hamilton,method="repetitive_nn")
order = cut_tour(sol,"cut")
order<-c(6,1,7,3,2,5,4)
names(order)<-c("den","acf","ph","acc","acv","st","clor")
DVM= D2RVine(order,family=rep(0,d*(d-1)/2),par=rep(0,d*(d-1)/2))
fit.dv=RVineCopSelect(data=udata7,familyset=c(1,5,6,4),indeptest=FALSE,
                      level=0.05,Matrix=DVM$Matrix,selectioncrit="AIC")
fit.dv.ind=RVineCopSelect(data=udata7,familyset=c(1,5,6,4),indeptest=TRUE,
                          level=0.05,Matrix=DVM$Matrix,selectioncrit="AIC")
```

Fitting R-vines, Gaussian R-vines, C-vines and D-vines using joint MLE

```r
fit.mle=RVineMLE(udata7, fit.rv)
```

```
## iter   10 value -2527.262525
## iter   20 value -2527.296219
## final  value -2527.297156
## converged
```

```r
fit.mle.ind=RVineMLE(udata7, fit.rv.ind)
```

```
## iter   10 value -2521.146096
## final  value -2521.169311
## converged
```

```r
fit.mle.Gauss=RVineMLE(udata7, fit.Gauss)
```

```
## final  value -2271.796370
## converged
```

```r
fit.mle.Gauss.ind=RVineMLE(udata7, fit.Gauss.ind)
```

```
## final  value -2267.147957
## converged
```

```r
fit.cv.mle=RVineMLE(udata7, fit.cv)
```

```
## iter   10 value -2496.333280
## iter   20 value -2496.413547
## iter   30 value -2496.428978
## iter   40 value -2496.522471
## iter   50 value -2496.642744
## final  value -2496.644184
## converged
```

```
fit.cv.mle.ind=RVineMLE(udata7, fit.cv.ind)
```

```
## iter   10 value -2453.953098
## iter   20 value -2454.161703
## iter   30 value -2454.169972
## iter   40 value -2454.194084
## iter   50 value -2454.391388
## iter   60 value -2454.452799
## iter   70 value -2454.457011
## final  value -2454.457049
## converged
```

```
fit.dv.mle=RVineMLE(udata7, fit.dv)
```

```
## iter   10 value -2449.239360
## iter   20 value -2449.475296
## iter   30 value -2451.560037
## iter   40 value -2452.636435
## iter   50 value -2452.649270
## iter   60 value -2452.659293
## iter   70 value -2452.668590
## iter   80 value -2452.684993
## iter   90 value -2452.688507
## iter  100 value -2452.691139
## iter  110 value -2452.710200
## iter  120 value -2452.737637
## final  value -2452.739344
## converged
```

```
fit.dv.mle.ind=RVineMLE(udata7, fit.dv.ind)
```

```
## iter   10 value -2447.687098
## iter   20 value -2447.955469
## iter   30 value -2449.450627
## iter   40 value -2450.757271
## iter   50 value -2451.062637
## iter   60 value -2451.097156
## iter   70 value -2451.103529
## iter   80 value -2451.141415
## iter   90 value -2451.188271
## final  value -2451.189605
## converged
```

Collecting results (df, loglik, AIC, BIC) for fitted R-vines

```
vine.out<-function(fit=fit.rv,data=udata7,digits=2){
df<-sum(abs(fit$par)>0)+sum(fit$par2>0)
out<-round(c(fit$logLik,df, fit$AIC,fit$BIC),digits)
names(out)<-c("loglik","par","AIC","BIC")
out
}

out.table<-rbind(
vine.out(fit=fit.rv,data=udata7),
vine.out(fit=fit.mle$RVM,data=udata7),
vine.out(fit=fit.rv.ind,data=udata7),
vine.out(fit=fit.mle.ind$RVM,data=udata7),
vine.out(fit=fit.Gauss,data=udata7),
vine.out(fit=fit.mle.Gauss$RVM,data=udata7),
vine.out(fit=fit.Gauss.ind,data=udata7),
vine.out(fit=fit.mle.Gauss.ind$RVM,data=udata7))
row.names(out.table)<-c("R-vine-seq","R-vine-mle",
                        "R-vine-ind-seq","R-vine-ind-mle",
                        "Gauss-seq","Gauss-mle",
                         "Gauss-ind-seq","Gauss-ind-mle")
```

Collecting results (df, loglik, AIC, BIC) for fitted C-vines

```
out.cv.table<-rbind(
vine.out(fit=fit.cv,data=udata7),
vine.out(fit=fit.cv.mle$RVM,data=udata7),
vine.out(fit=fit.cv.ind,data=udata7),
vine.out(fit=fit.cv.mle.ind$RVM,data=udata7))
row.names(out.cv.table)<-c("C-vine-seq","C-vine-mle", "C-vine-ind-seq"
                           ,"C-vine-ind-mle")
```

Collecting results (df, loglik, AIC, BIC) for fitted D-vines

```
out.dv.table<-rbind(
vine.out(fit=fit.dv,data=udata7),
vine.out(fit=fit.dv.mle$RVM,data=udata7),
vine.out(fit=fit.dv.ind,data=udata7),
vine.out(fit=fit.dv.mle.ind$RVM,data=udata7))
row.names(out.dv.table)<-c("D-vine-seq","D-vine-mle",
                           "D-vine-ind-seq", "D-vine-ind-mle")
```

Table 9.2: WINE7: Estimated log-likelihood, number of parameters (# par), AIC RV and BIC RV for all fitted models and estimation methods (seq=sequential estimation, mle=maximum likelihood, ind= independence test used to allow for independence pair copula family).

```
out.all<-rbind(out.table,out.cv.table,out.dv.table)
out.all
```

```
##                 loglik par     AIC     BIC
## R-vine-seq      2525.3  21 -5008.6 -4895.7
## R-vine-mle     2527.3  21 -5012.6 -4899.7
## R-vine-ind-seq 2519.5  18 -5003.0 -4906.2
## R-vine-ind-mle 2521.2  18 -5006.3 -4909.6
## Gauss-seq      2271.8  21 -4501.6 -4388.6
## Gauss-mle      2271.8  21 -4501.6 -4388.7
## Gauss-ind-seq  2267.0  18 -4497.9 -4401.2
## Gauss-ind-mle  2267.2  18 -4498.3 -4401.5
## C-vine-seq     2489.4  21 -4936.8 -4823.9
## C-vine-mle     2496.6  21 -4951.3 -4838.4
## C-vine-ind-seq 2446.5  19 -4855.1 -4752.9
## C-vine-ind-mle 2454.5  19 -4870.9 -4768.8
## D-vine-seq     2445.2  21 -4848.3 -4735.4
## D-vine-mle     2452.7  21 -4863.5 -4750.6
## D-vine-ind-seq 2443.5  20 -4846.9 -4739.4
## D-vine-ind-mle 2451.2  20 -4862.4 -4754.8
```

Section 9.3 Vuong test for comparing different regular vine copula models

Example 9.3: Model comparison of selected vine copula models for the extended WINE data set.

Table 9.3: WINE7: Comparing different fitted vine copula models using the asymptotic Vuong test in (9.8) with no adjustment.

```r
vuong.p<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.cv.mle$RVM)$p.value,
RVineVuongTest(udata7, fit.mle$RVM,fit.dv.mle$RVM)$p.value,
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.Gauss$RVM)$p.value,
RVineVuongTest(udata7, fit.cv.mle$RVM,fit.dv.mle$RVM)$p.value,
RVineVuongTest(udata7,fit.cv.mle$RVM,fit.mle.Gauss$RVM)$p.value,
RVineVuongTest(udata7, fit.dv.mle$RVM,fit.mle.Gauss$RVM)$p.value
)
vuong.stat<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.cv.mle$RVM)$statistic,
RVineVuongTest(udata7, fit.mle$RVM,fit.dv.mle$RVM)$statistic,
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.Gauss$RVM)$statistic,
RVineVuongTest(udata7, fit.cv.mle$RVM,fit.dv.mle$RVM)$statistic,
RVineVuongTest(udata7, fit.cv.mle$RVM,fit.mle.Gauss$RVM)$statistic,
RVineVuongTest(udata7, fit.dv.mle$RVM,fit.mle.Gauss$RVM)$statistic
)
vuong.table<-round(cbind(vuong.stat,vuong.p),digits=2)
rownames(vuong.table)<-c("rv-cv","rv-dv","rv-Gauss","cv-dv",
"cv-Gauss","dv-Gauss")
vuong.table
```

```
##          vuong.stat vuong.p
## rv-cv          1.82    0.07
## rv-dv          4.00    0.00
## rv-Gauss       7.83    0.00
## cv-dv          2.10    0.04
## cv-Gauss       6.41    0.00
## dv-Gauss       6.41    0.00
```

Table 9.4: WINE7: Comparing full and reduced by independence tests vine copula models (models ending with ind) using the Vuong test with Akaike and Schwarz corrections.

```r
vuong.p.ind.aic<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.ind$RVM)$p.value.Akaike,
RVineVuongTest(udata7,fit.mle.Gauss$RVM,fit.mle.Gauss.ind$RVM)$p.value.Akaike,
RVineVuongTest(udata7,fit.cv.mle$RVM,fit.cv.mle.ind$RVM)$p.value.Akaike,
RVineVuongTest(udata7,fit.dv.mle$RVM,fit.dv.mle.ind$RVM)$p.value.Akaike
)

vuong.p.ind<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.ind$RVM)$p.value,
RVineVuongTest(udata7,fit.mle.Gauss$RVM,fit.mle.Gauss.ind$RVM)$p.value,
RVineVuongTest(udata7,fit.cv.mle$RVM,fit.cv.mle.ind$RVM)$p.value,
RVineVuongTest(udata7,fit.dv.mle$RVM,fit.dv.mle.ind$RVM)$p.value
)


vuong.p.ind.schwarz<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.ind$RVM)$p.value.Schwarz,
RVineVuongTest(udata7,fit.mle.Gauss$RVM,fit.mle.Gauss.ind$RVM)$p.value.Schwarz,
RVineVuongTest(udata7,fit.cv.mle$RVM,fit.cv.mle.ind$RVM)$p.value.Schwarz,
RVineVuongTest(udata7,fit.dv.mle$RVM,fit.dv.mle.ind$RVM)$p.value.Schwarz
)
vuong.stat.ind<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.ind$RVM)$statistic,
RVineVuongTest(udata7,fit.mle.Gauss$RVM,fit.mle.Gauss.ind$RVM)$statistic,
RVineVuongTest(udata7,fit.cv.mle$RVM,fit.cv.mle.ind$RVM)$statistic,
RVineVuongTest(udata7,fit.dv.mle$RVM,fit.dv.mle.ind$RVM)$statistic
)

vuong.stat.ind.aic<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.ind$RVM)$statistic.Akaike,
RVineVuongTest(udata7,fit.mle.Gauss$RVM,fit.mle.Gauss.ind$RVM)$statistic.Akaike,
RVineVuongTest(udata7,fit.cv.mle$RVM,fit.cv.mle.ind$RVM)$statistic.Akaike,
RVineVuongTest(udata7,fit.dv.mle$RVM,fit.dv.mle.ind$RVM)$statistic.Akaike
)
vuong.stat.ind.schwarz<-c(
RVineVuongTest(udata7, fit.mle$RVM,fit.mle.ind$RVM)$statistic.Schwarz,
RVineVuongTest(udata7,fit.mle.Gauss$RVM,fit.mle.Gauss.ind$RVM)$statistic.Schwarz,
RVineVuongTest(udata7,fit.cv.mle$RVM,fit.cv.mle.ind$RVM)$statistic.Schwarz,
RVineVuongTest(udata7,fit.dv.mle$RVM,fit.dv.mle.ind$RVM)$statistic.Schwarz
)
vuong.table.ind<-round(cbind(vuong.stat.ind,vuong.p.ind,vuong.stat.ind.aic,vuong.p.ind.aic,
                        vuong.stat.ind.schwarz,vuong.p.ind.schwarz),digits=2)
rownames(vuong.table.ind)<-c("rv-rv.ind","Gauss-Gauss.ind","cv-cv.ind","dv-dv.ind")
colnames(vuong.table.ind)<-c("stat","p","stat.aic","p.aic","stat.schwarz","p.schwarz")
vuong.table.ind
```

```
##                 stat    p stat.aic p.aic stat.schwarz p.schwarz
## rv-rv.ind       1.61 0.11     0.82  0.41        -1.29      0.20
## Gauss-Gauss.ind 1.49 0.14     0.53  0.60        -2.06      0.04
## cv-cv.ind       3.07 0.00     2.92  0.00         2.53      0.01
## dv-dv.ind       0.87 0.38     0.31  0.76        -1.20      0.23
```