

Discrimination for 2 classes based on distributions of features

2022 April 24

Step 1. Scale variables and split into training and holdout sets

```
# Forest Cover data set from www.kaggle.com
# Selected subset of variables:
# elev = Elevation,
# hydrologyh = HorizontalDistanceToHydrology,
# hydrologyv = VerticalDistanceToHydrology,
# roadway = HorizontalDistanceToRoadways,
# firept = HorizontalDistanceToFirePoints
# soil= 1-SoilType1-SoilType2-SoilType3-SoilType4-SoilType5-SoilType6-SoilType10-SoilType17
# as binary (=0 for one of types 1,2,3,5,6,10,17)
# cover = CoverType
load("fcsubset.RData")
print(names(fc4))
#> [1] "elev"          "hydrologyh" "hydrologyv" "roadway"      "firept"
#> [6] "soil"          "cover"

labels=names(fc4)

fc12 = subset(fc4,cover==1 | cover==2)
print(dim(fc12))
#> [1] 4320      7

# Scale variables before fitting univariate densities
fc12$elev = fc12$elev/1000
fc12$hydrologyh = fc12$hydrologyh/100 # also jitter the 0s
fc12$hydrologyv = fc12$hydrologyv/100
fc12$roadway = fc12$roadway/1000
fc12$firept = fc12$firept/1000

# Jiggle the zeros with a small positive number in order to fit gamma
# or generalized gamma density to hydrologyh
iz=which(fc12$hydrologyh<=0.35)
set.seed(123)
nz=length(iz)
fc12$hydrologyh[iz]=fc12$hydrologyh[iz]+runif(nz,0,0.05)
print(summary(fc12))
#>      elev      hydrologyh      hydrologyv      roadway
#> Min.   :2.169   Min.    : 0.000031   Min.    :-1.4600   Min.    :0.030
#> 1st Qu.:2.897   1st Qu.: 1.080000   1st Qu.: 0.0600   1st Qu.:1.233
#> Median :3.038   Median : 2.370000   Median : 0.2700   Median :2.200
#> Mean   :3.025   Mean    : 2.798708   Mean    : 0.4431   Mean    :2.503
#> 3rd Qu.:3.178   3rd Qu.: 4.020000   3rd Qu.: 0.6500   3rd Qu.:3.536
#> Max.   :3.675   Max.    :13.430000   Max.    : 5.5400   Max.    :6.890
```

```

#>      firept      soil      cover
#> Min. :0.030 Min. :0.0000 Min. :1.0
#> 1st Qu.:1.130 1st Qu.:1.0000 1st Qu.:1.0
#> Median :1.834 Median :1.0000 Median :1.5
#> Mean :2.075 Mean :0.9662 Mean :1.5
#> 3rd Qu.:2.642 3rd Qu.:1.0000 3rd Qu.:2.0
#> Max. :6.993 Max. :1.0000 Max. :2.0

# Random split into training and holdout sets
nn=nrow(fc12)
ntrain=round(nn * 0.75) # 3240
set.seed(447); ii=sample(1:nn)
train = fc12[1:ntrain,]
train1 = subset(train,cover==1)
print(nrow(train1))
#> [1] 1620
train2 = subset(train,cover==2)
print(nrow(train2))
#> [1] 1620
holdo = fc12[(ntrain+1):nn,]; print(dim(holdo))
#> [1] 1080 7

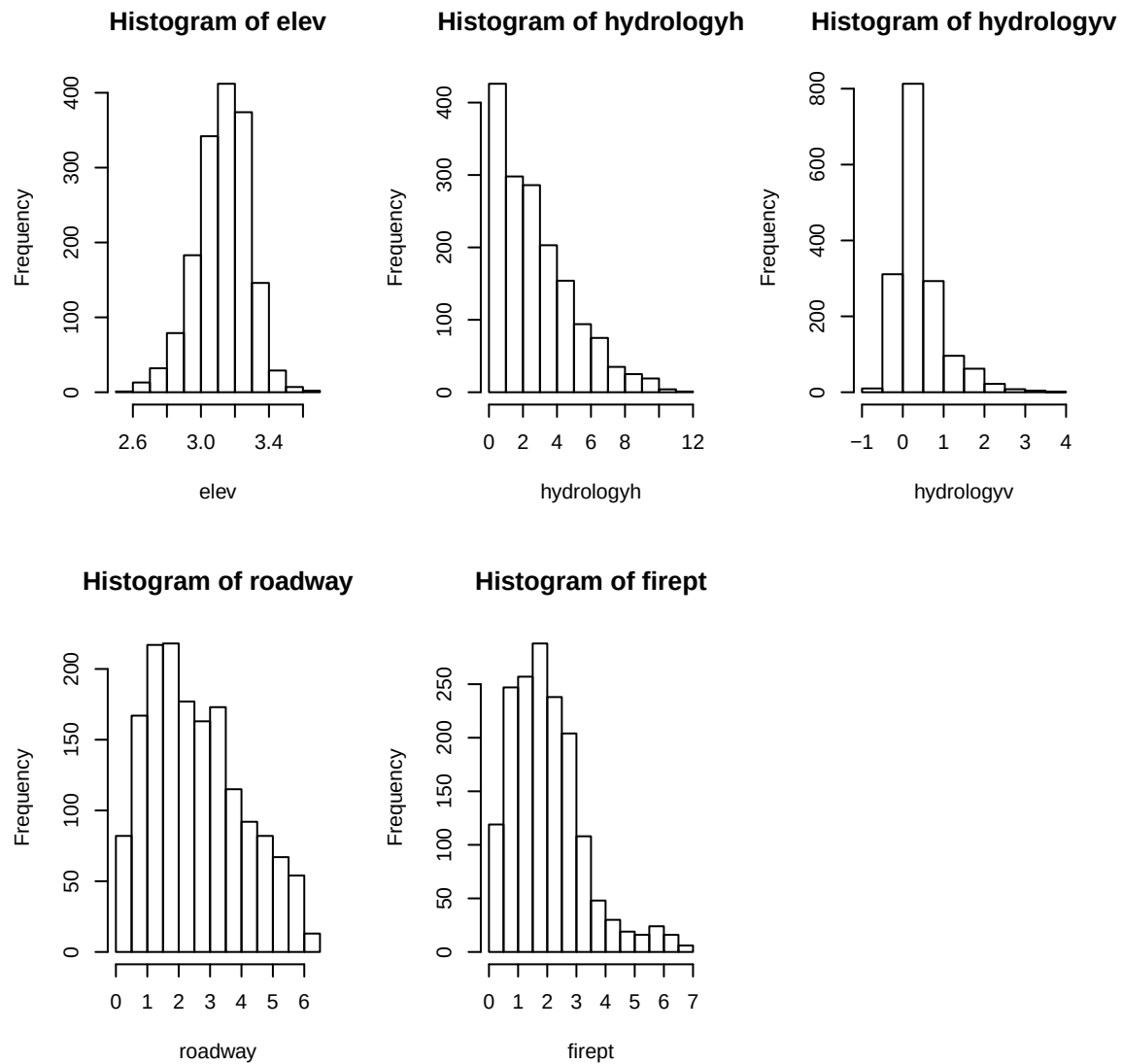
print(summary(train1))
#>      elev      hydrologyh      hydrologyv      roadway
#> Min. :2.525 Min. : 0.002292 Min. : -0.8300 Min. :0.108
#> 1st Qu.:3.039 1st Qu.: 0.900000 1st Qu.: 0.0400 1st Qu.:1.380
#> Median :3.144 Median : 2.340000 Median : 0.2400 Median :2.351
#> Mean :3.129 Mean : 2.750284 Mean : 0.4134 Mean :2.569
#> 3rd Qu.:3.233 3rd Qu.: 4.020000 3rd Qu.: 0.6100 3rd Qu.:3.558
#> Max. :3.675 Max. :11.290000 Max. : 3.6300 Max. :6.414
#>      firept      soil      cover
#> Min. :0.030 Min. :0.0000 Min. :1
#> 1st Qu.:1.092 1st Qu.:1.0000 1st Qu.:1
#> Median :1.839 Median :1.0000 Median :1
#> Mean :2.005 Mean :0.9914 Mean :1
#> 3rd Qu.:2.639 3rd Qu.:1.0000 3rd Qu.:1
#> Max. :6.853 Max. :1.0000 Max. :1

print(summary(train2))
#>      elev      hydrologyh      hydrologyv      roadway
#> Min. :2.169 Min. : 0.000031 Min. : -1.4600 Min. :0.030
#> 1st Qu.:2.804 1st Qu.: 1.270000 1st Qu.: 0.1000 1st Qu.:1.110
#> Median :2.934 Median : 2.420000 Median : 0.3200 Median :2.014
#> Mean :2.926 Mean : 2.917298 Mean : 0.4823 Mean :2.429
#> 3rd Qu.:3.046 3rd Qu.: 4.102500 3rd Qu.: 0.6900 3rd Qu.:3.414
#> Max. :3.413 Max. :13.430000 Max. : 5.5400 Max. :6.890
#>      firept      soil      cover
#> Min. :0.042 Min. :0.0000 Min. :2
#> 1st Qu.:1.173 1st Qu.:1.0000 1st Qu.:2
#> Median :1.841 Median :1.0000 Median :2
#> Mean :2.153 Mean :0.9457 Mean :2
#> 3rd Qu.:2.616 3rd Qu.:1.0000 3rd Qu.:2
#> Max. :6.686 Max. :1.0000 Max. :2

```

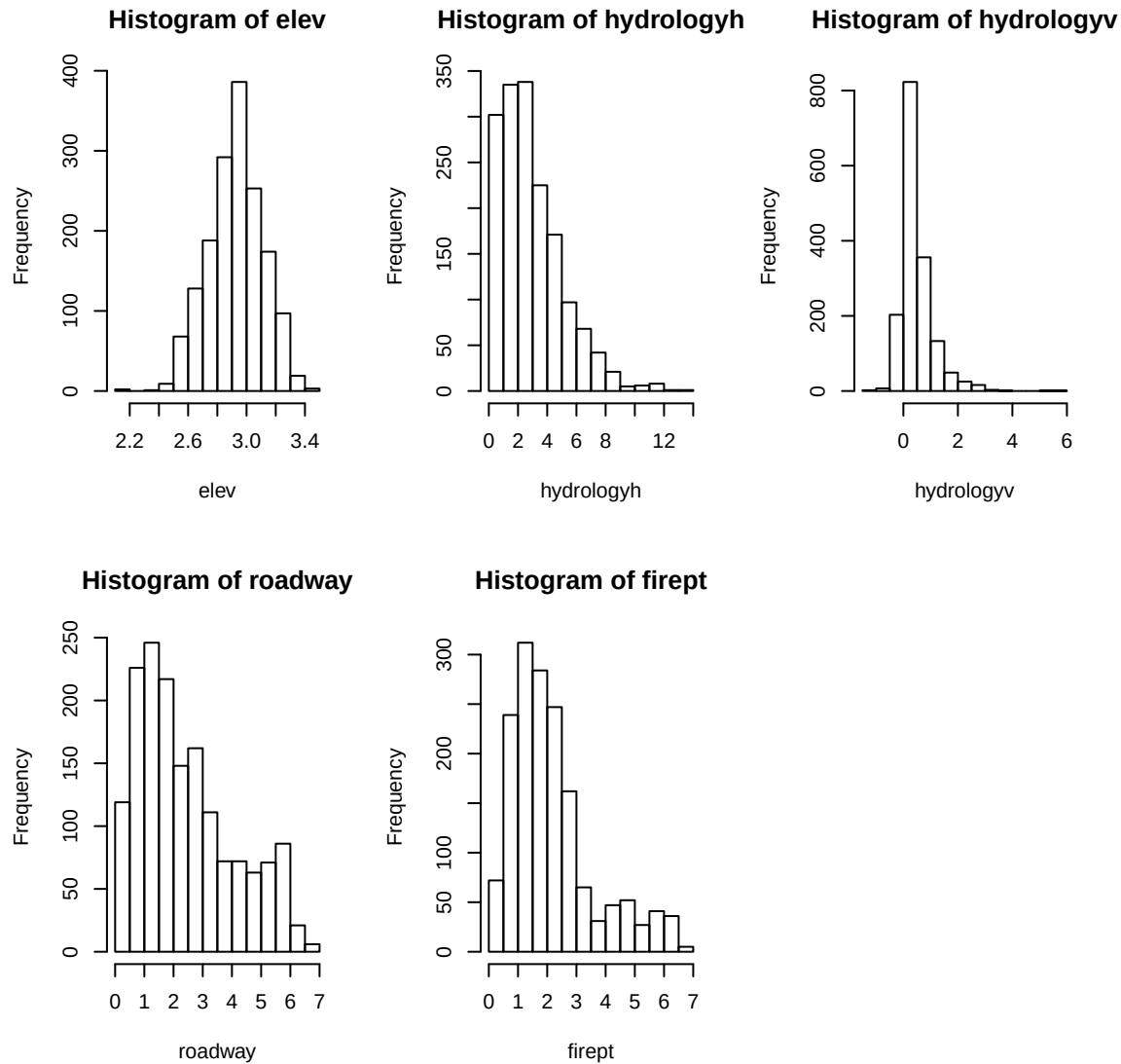
```
# Plots
par(mfrow=c(2,3))
attach(train1)
hist(elev)
hist(hydrologyh)
hist(hydrologyv)
hist(roadway)
hist(firept)
detach(train1)

par(mfrow=c(2,3))
```



```
attach(train2)
hist(elev)
hist(hydrologyh)
hist(hydrologyv)
```

```
hist(roadway)
hist(firept)
detach(train2)
```



Step 2. Fit univariate distributions for continuous variables for cover types 1 and 2

```
library(sn)
source("skewtnor-gammaWeibull.R") # skew-normal, Azzalini-Capitanio skew-t, gamma, Weibull
source("ggamma.R") # generalized gamma
source("jonesfaddy-skewt.R") # Jones-Faddy skew-t

udat1 = matrix(0,nrow(train1),5)
udat2 = matrix(0,nrow(train2),5)
```

```

## Cover type 1 training set
attach(train1)
n1=nrow(train1)
i1=((1:n1)-0.5)/n1

xbar=mean(elev); xsd=sd(elev)
MLElev1 = nlm(stJFnllk, c(xbar,xsd,15,20), hessian=T,print.level=1,dat=elev)
#> iteration = 0
#> Step:
#> [1] 0 0 0 0
#> Parameter:
#> [1] 3.1288802 0.1536791 15.0000000 20.0000000
#> Function Value
#> [1] -138.3218
#> Gradient:
#> [1] -8885.0939 580.5301 -258.1435 222.5632
#>
#> iteration = 58
#> Parameter:
#> [1] 3.2998258 0.1304073 5.9981880 10.6333112
#> Function Value
#> [1] -757.4648
#> Gradient:
#> [1] 1.524518e-07 -1.207877e-04 3.606857e-07 -2.370857e-07
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
par1=MLElev1$estimate
#q1=qstJF(i1,par1); plot(sort(elev),q1)
udat1[,1] = pstJF(elev,par1)

xbar=mean(hydrologyh); xsd=sd(hydrologyh)
MLhh1 = nlm(ggamllk, c(1,1,xbar), hessian=T,print.level=1,dat=hydrologyh)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 1.000000 1.000000 2.750284
#> Function Value
#> [1] 3258.961
#> Gradient:
#> [1] -7.129006e+01 -2.288652e+02 2.944805e-04
#>
#> iteration = 20
#> Parameter:
#> [1] 0.336467 2.363056 5.993264
#> Function Value
#> [1] 3197.771
#> Gradient:
#> [1] 1.897206e-03 2.126466e-04 -9.863933e-06
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.

```

```

par2=MLhh1$estimate
#q2=qggam(i1,par2); plot(sort(hydrologyh),q2)
udat1[,2] = pggam(hydrologyh,par2)

xbar=mean(hydrologyv); xsd=sd(hydrologyv)
MLhv1 = nlm(sknllk, c(xbar,xsd,4), hessian=T,print.level=1,dat=hydrologyv)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 0.4133519 0.5565337 4.0000000
#> Function Value
#> [1] 5698.179
#> Gradient:
#> [1] 20481.559 -13938.305 1939.541
#>
#> iteration = 27
#> Parameter:
#> [1] -0.1737265 0.8088254 4.8390302
#> Function Value
#> [1] 1062.351
#> Gradient:
#> [1] 9.413270e-05 9.345058e-05 -6.108368e-07
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
#q3=qskewn(i1,par3); plot(sort(hydrologyv),q3) # a little off in lower quarter
par3=MLhv1$estimate
udat1[,3] = pskewn(hydrologyv,par3)

xbar=mean(roadway); xsd=sd(roadway)
MLrw1 = nlm(ggamllk, c(1,1,xbar), hessian=T,print.level=1,dat=roadway)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 1.000000 1.000000 2.568743
#> Function Value
#> [1] 3148.335
#> Gradient:
#> [1] -5.783316e+02 -9.805200e+02 3.152923e-04
#>
#> iteration = 23
#> Parameter:
#> [1] 0.498800 2.825027 4.138517
#> Function Value
#> [1] 2819.71
#> Gradient:
#> [1] 3.128662e-04 1.141284e-04 -3.703014e-05
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
par4=MLrw1$estimate

```

```

#q4=qggam(i1,par4); plot(sort(roadway),q4) # upper tail a little off
udat1[,4] = pggam(roadway,par4)

xbar=mean(firept); xsd=sd(firept)
MLfp1 = nlm(ggamllk, c(1,1,xbar), hessian=T,print.level=1,dat=firept)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 1.000000 1.000000 2.005465
#> Function Value
#> [1] 2747.319
#> Gradient:
#> [1] -5.711827e+02 -9.527372e+02 4.038490e-04
#>
#> iteration = 15
#> Parameter:
#> [1] 1.458946 1.335924 1.605108
#> Function Value
#> [1] 2467.065
#> Gradient:
#> [1] 9.565943e-04 7.522819e-05 8.782693e-04
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
par5=MLfp1$estimate
#q5=qggam(i1,par5); plot(sort(firept),q5) # misses in 3rd quarter
udat1[,5] = pggam(firept,par5)

detach(train1)

## Cover type 2 training set
attach(train2)
n2=nrow(train2)
i2=((1:n2)-0.5)/n2

xbar=mean(elev); xsd=sd(elev)
MLElev2 = nlm(stJFnllk, c(xbar,xsd,15,20), hessian=T,print.level=1,dat=elev)
#> iteration = 0
#> Step:
#> [1] 0 0 0 0
#> Parameter:
#> [1] 2.9258938 0.1856381 15.0000000 20.0000000
#> Function Value
#> [1] 171.9028
#> Gradient:
#> [1] -7308.6209 394.1624 -257.9648 222.2283
#>
#> iteration = 100
#> Parameter:
#> [1] 3.9518997 0.1479843 54.2623766 140.1514850
#> Function Value

```

```

#> [1] -433.6
#> Gradient:
#> [1] 3.00180739 -15.96301939 0.05665160 -0.03410054
#>
#> Iteration limit exceeded. Algorithm failed.
par1=MLelev2$estimate
#q1=qstJF(i2,par1) ; plot(sort(elev),q1)
udat2[,1] = pstJF(elev,par1)

xbar=mean(hydrologyh); xsd=sd(hydrologyh)
MLhh2 = nlm(ggamllk, c(1,1,xbar), hessian=T,print.level=1,dat=hydrologyh)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 1.000000 1.000000 2.917298
#> Function Value
#> [1] 3354.466
#> Gradient:
#> [1] -2.813227e+02 -5.358400e+02 2.776216e-04
#>
#> iteration = 20
#> Parameter:
#> [1] 0.5324383 1.9378052 4.9632274
#> Function Value
#> [1] 3262.375
#> Gradient:
#> [1] 2.960405e-04 2.182443e-05 3.399225e-05
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
par2=MLhh2$estimate
#q2=qggam(i2,par2); plot(sort(hydrologyh),q2)
udat2[,2] = pggam(hydrologyh,par2)

xbar=mean(hydrologyv); xsd=sd(hydrologyv)
MLhv2 = nlm(sknllk, c(xbar,xsd,4), hessian=T,print.level=1,dat=hydrologyv)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 0.4822531 0.5954871 4.0000000
#> Function Value
#> [1] 5698.128
#> Gradient:
#> [1] 18682.107 -12761.439 1900.074
#>
#> iteration = 30
#> Parameter:
#> [1] -0.1400276 0.8611726 4.3892318
#> Function Value
#> [1] 1186.639
#> Gradient:

```

```

#> [1] 4.320100e-05 6.502887e-05 -2.590131e-07
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
par3=MLhv2$estimate
#q3=qskewn(i2,par3); plot(sort(hydrologyv),q3) # a little off in lower quarter
udat2[,3] = pskewn(hydrologyv,par3)

xbar=mean(roadway); xsd=sd(roadway)
MLrw2 = nlm(ggamllk, c(1,1,xbar), hessian=T,print.level=1,dat=roadway)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 1.000000 1.000000 2.428643
#> Function Value
#> [1] 3057.479
#> Gradient:
#> [1] -4.789701e+02 -7.950649e+02 3.334805e-04
#>
#> iteration = 18
#> Parameter:
#> [1] 0.9226264 1.5807906 2.8757966
#> Function Value
#> [1] 2879.357
#> Gradient:
#> [1] 0.0007898961 0.0001956162 0.0002996548
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
par4=MLrw2$estimate
#q4=qggam(i2,par4); plot(sort(roadway),q4) # upper tail a little off
udat2[,4] = pggam(roadway,par4)

xbar=mean(firept); xsd=sd(firept)
MLfp2 = nlm(ggamllk, c(1,1,xbar), hessian=T,print.level=1,dat=firept)
#> iteration = 0
#> Step:
#> [1] 0 0 0
#> Parameter:
#> [1] 1.000000 1.000000 2.152959
#> Function Value
#> [1] 2862.286
#> Gradient:
#> [1] -5.817685e+02 -9.480335e+02 3.761822e-04
#>
#> iteration = 41
#> Parameter:
#> [1] 4.0681448 0.7565003 0.3207762
#> Function Value
#> [1] 2569.238
#> Gradient:
#> [1] 3.352413e-05 -1.554158e-03 9.839346e-04

```

```

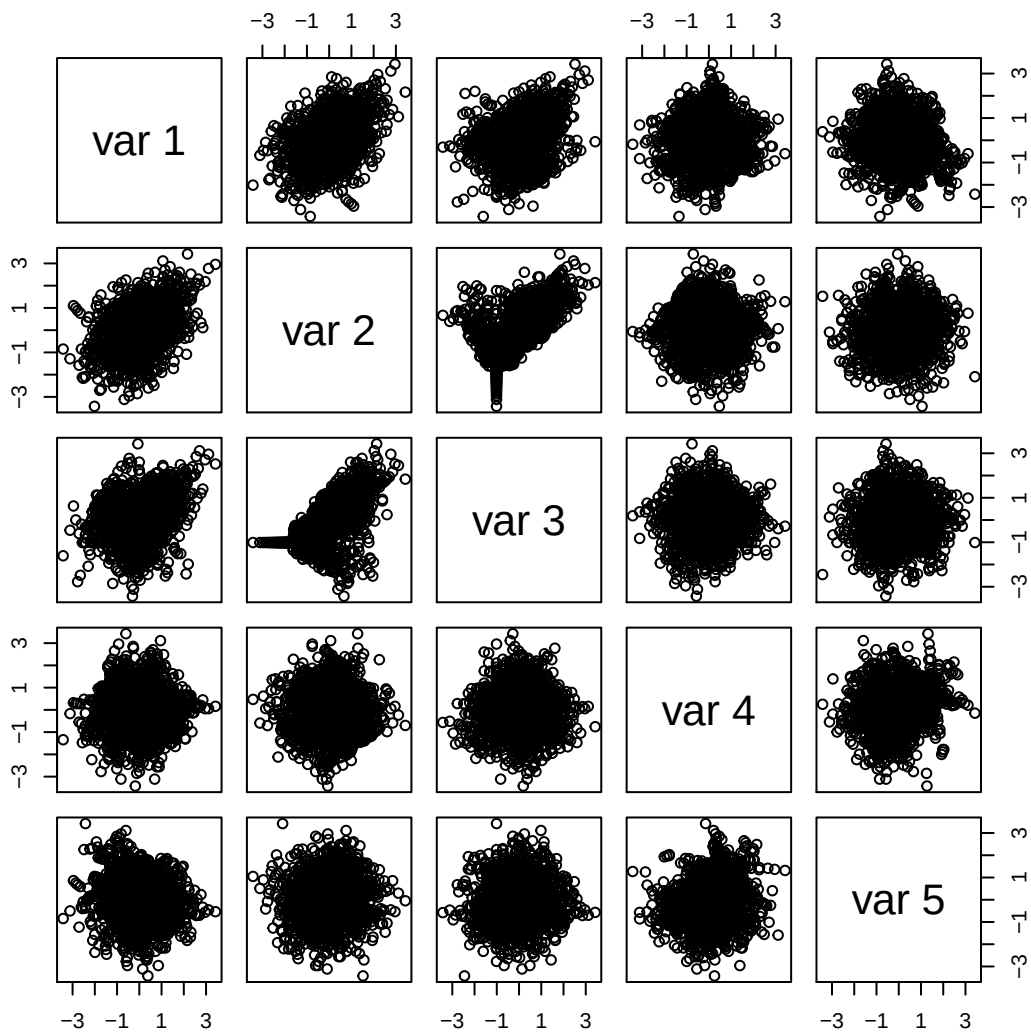
#>
#> Relative gradient close to zero.
#> Current iterate is probably solution.
par5=MLfp2$estimate
#q5=qggam(i2,par5); plot(sort(firept),q5) # misses in 3rd quarter
udat2[,5] = pggam(firept,par5)

detach(train2)

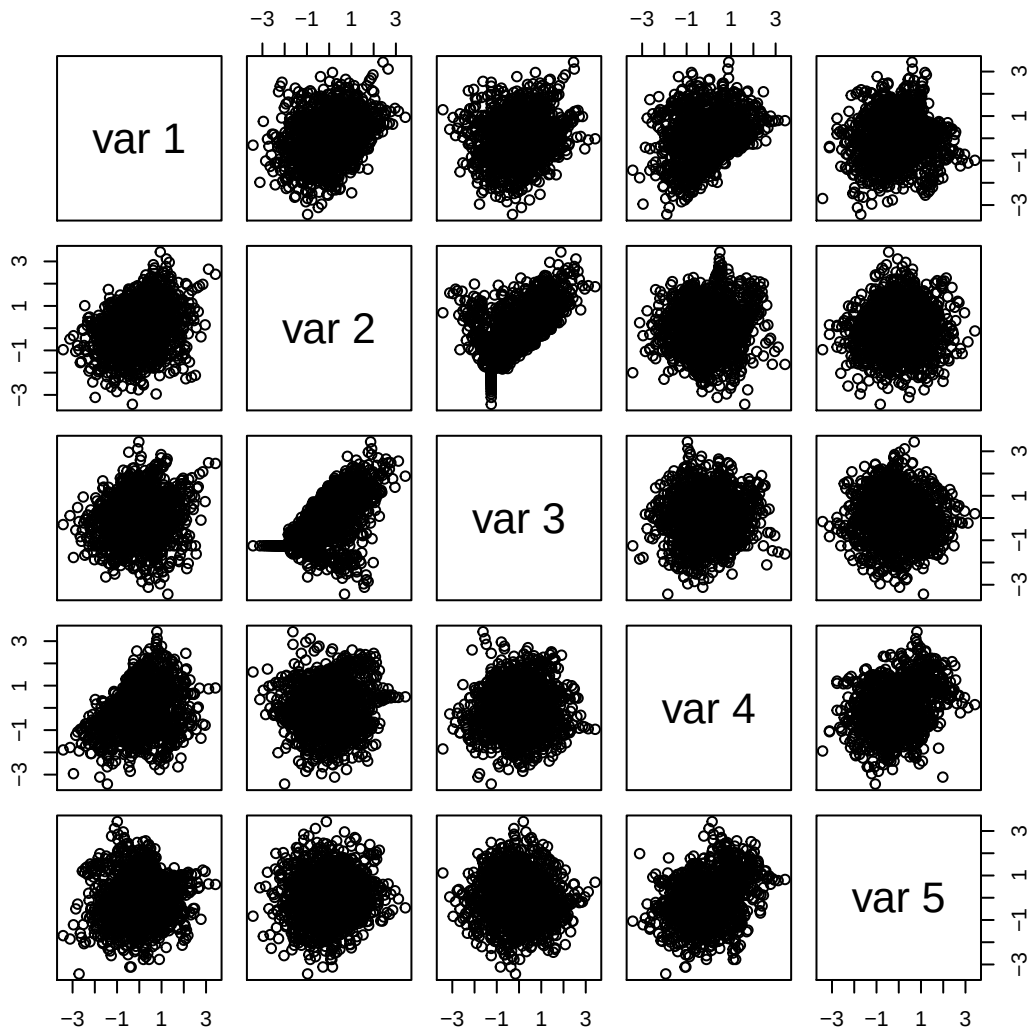
colnames(udat1)=labels[1:5]
colnames(udat2)=labels[1:5]
print(summary(udat1),3)
#>      elev      hydrologyh      hydrologyv      roadway
#> Min.      :0.001469   Min.      :0.002148   Min.      :0.0000011   Min.      :0.006626
#> 1st Qu.:0.255112   1st Qu.:0.247403   1st Qu.:0.2157757   1st Qu.:0.236546
#> Median :0.514938   Median :0.516345   Median :0.3913119   Median :0.476356
#> Mean    :0.501011   Mean    :0.502538   Mean    :0.4538984   Mean    :0.494172
#> 3rd Qu.:0.750650   3rd Qu.:0.743762   3rd Qu.:0.6674392   3rd Qu.:0.747454
#> Max.    :0.999911   Max.    :0.998575   Max.    :0.9999974   Max.    :0.991396
#>      firept
#> Min.      :0.0003302
#> 1st Qu.:0.2595045
#> Median :0.5213862
#> Mean    :0.5014389
#> 3rd Qu.:0.7376762
#> Max.    :0.9972083
print(summary(udat2),3)
#>      elev      hydrologyh      hydrologyv      roadway
#> Min.      :0.0001414   Min.      :0.0000048   Min.      :0.0000   Min.      :0.001327
#> 1st Qu.:0.2492040   1st Qu.:0.2693467   1st Qu.:0.2286   1st Qu.:0.231742
#> Median :0.5065997   Median :0.4937920   Median :0.4073   Median :0.473179
#> Mean    :0.4996181   Mean    :0.5049750   Mean    :0.4575   Mean    :0.495615
#> 3rd Qu.:0.7369467   3rd Qu.:0.7429208   3rd Qu.:0.6649   3rd Qu.:0.758275
#> Max.    :0.9974669   Max.    :0.9997646   Max.    :1.0000   Max.    :0.984271
#>      firept
#> Min.      :0.0000607
#> 1st Qu.:0.2660336
#> Median :0.5022863
#> Mean    :0.4977682
#> 3rd Qu.:0.7079980
#> Max.    :0.9883990

# Normal score plots. some deviations can be seen.
library(CopReg)
pairs(nscore(udat1)) # nscore is function in above library

```



```
pairs(nscore(udat2))
```



```
## Holdout set transforms

#  $Z \sim \text{Gamma}(th, 1)$ ,  $W = Z^{(1/zeta)}$ ,  $Y = \text{sigma} * W$ ,
# then  $Y$  is  $g\text{gamma}(\text{shape}=\text{theta}, \text{power}=\text{zeta}, \text{scale}=\text{sigma})$ 

# var 1 elev      stJF  dstJF(x,param)  loc scale a b
# var 2 hydrologyh ggam  dggam=function(y,param,log=F) theta zeta scale
# var 3 hydrologyv skewm dskewn=function(y,param) xi=loc omega=scale alpha=skew
# var 4 roadway   ggam
# var 5 firept    ggam

par1a=MLelev1$estimate; par2a=MLhh1$estimate; par3a=MLhv1$estimate;
par4a=MLrw1$estimate; par5a=MLfp1$estimate;
par1b=MLelev2$estimate; par2b=MLhh2$estimate; par3b=MLhv2$estimate;
par4b=MLrw2$estimate; par5b=MLfp2$estimate;

# Product of univariate densities for Models A, B from training for classes 1,2
```

```

apdf = dstJF(holdo[,1],par1a) * dggam(holdo[,2],par2a) * dskewn(holdo[,3],par3a) *
      dggam(holdo[,4],par4a) * dggam(holdo[,5],par5a);
bpdf = dstJF(holdo[,1],par1b) * dggam(holdo[,2],par2b) * dskewn(holdo[,3],par3b) *
      dggam(holdo[,4],par4b) * dggam(holdo[,5],par5b);

# Set-up cdf transforms of holdout set for models A and B
nh = nrow(holdo)
uholda = matrix(0,nh,5)
uholdb = matrix(0,nh,5)
uholda[,1] = pstJF(holdo[,1],par1a)
uholda[,2] = pggam(holdo[,2],par2a)
uholda[,3] = pskewn(holdo[,3],par3a)
uholda[,4] = pggam(holdo[,4],par4a)
uholda[,5] = pggam(holdo[,5],par5a)
uholdb[,1] = pstJF(holdo[,1],par1b)
uholdb[,2] = pggam(holdo[,2],par2b)
uholdb[,3] = pskewn(holdo[,3],par3b)
uholdb[,4] = pggam(holdo[,4],par4b)
uholdb[,5] = pggam(holdo[,5],par5b)

# Add the binary variable soil (0 or 1) as 6th variable for discrimination
print(table(holdo$soil, holdo$cover))
#>
#>      1    2
#>  0    2  42
#>  1 538 498

soilecdf1 = ecdf(train1$soil)
soilecdf2 = ecdf(train2$soil)
usoil1 = soilecdf1(train1$soil)
usoil1[usoil1==1] = 0.999999
usoil2 = soilecdf2(train2$soil)
usoil2[usoil2==1] = 0.999999

# ECDF transform for holdout set based on models A,B for classes 1,2
pr0a = min(usoil1) # 0.008641975
pr0b = min(usoil2) # 0.05432099
uplusa = soilecdf1(holdo$soil)
uplusb = soilecdf2(holdo$soil)
uplusa[uplusa==1] = 0.999999
uplusb[uplusb==1] = 0.999999

# Holdout set: discrete binary: CDF+, CDF- and PMF for models A and B
uminua = uplusa
uminua[uplusa==pr0a] = 0.000001; uminua[uplusa>=0.9999] = pr0a
upmfa = uplusa-uminua
uminub = uplusb
uminub[uplusb==pr0b] = 0.000001; uminub[uplusb>=0.9999] = pr0b
upmfb = uplusb-uminub

# Univariate densities including for soil
# Naive Bayes assuming feature variables are independent

```

```

apdf6 = apdf*upmfa
bpdf6 = bpdf*upmfb
pra6 = apdf6/(apdf6+bpdf6) # probability for model A based on uniform prior
naiveBayes6 = table(holdo$cover,pra6<0.5)
colnames(naiveBayes6)=c("pred1","pred2")
rownames(naiveBayes6)=c("actual1","actual2")
print(naiveBayes6)
#>
#>           pred1 pred2
#>  actual1    409   131
#>  actual2    145   395

#save(file="step2.RData", train1,train2,holdo, udat1,udat2,
#  MLelev1,MLhh1,MLhv1,MLrw1,MLfp1,
#  MLelev2,MLhh2,MLhv2,MLrw2,MLfp2,
#  usoil1, usoil2, uholda, uholdb, upmfa, upmfb,
#  apdf, bpdf, apdf6, bpdf6, pra6)

```

Step 3. Fit vine densities for classes 1 and 2

```

library(VineCopula)

# BB1 fam7, BB8 fam10, Gum fam4, Frank fam5
famset = c(1,2,4,5,7,10,14,17,20)
# 6 variables with discrete variable soil
udat61 = cbind(udat1,usoil1)
udat62 = cbind(udat2,usoil2)

# Class 1: fit joint vine density
n1 = nrow(udat61)
rr1 = vWcor(udat61, is_disc=c(F,F,F,F,F,T))
print(round(rr1,3))
#>           [,1] [,2] [,3] [,4] [,5] [,6]
#> [1,]  1.000 0.393 0.310 0.061 -0.122  0.197
#> [2,]  0.393 1.000 0.586 0.011  0.041  0.000
#> [3,]  0.310 0.586 1.000 0.005  0.051  0.060
#> [4,]  0.061 0.011 0.005 1.000  0.156  0.026
#> [5,] -0.122 0.041 0.051 0.156  1.000 -0.033
#> [6,]  0.197 0.000 0.060 0.026 -0.033  1.000

vine61 = GaussVineMST(rr1,n1,1.01);
#print(vine61)
joint61 = RVineCopSelectMix(udat61,is_disc=c(F,F,F,F,F,T),
  familyset=famset, vine_array=vine61$VineA, iprint=T)
#>           [,1] [,2] [,3] [,4] [,5] [,6]
#> [1,]    1    1    2    1    5    1
#> [2,]    0    2    1    2    1    2
#> [3,]    0    0    3    3    2    3
#> [4,]    0    0    0    5    3    5
#> [5,]    0    0    0    0    4    4
#> [6,]    0    0    0    0    0    6
#>           [,1] [,2] [,3] [,4] [,5] [,6]

```

```

#> [1,] 1 1 2 1 4 1
#> [2,] 0 2 1 2 1 2
#> [3,] 0 0 3 3 2 3
#> [4,] 0 0 0 4 3 4
#> [5,] 0 0 0 0 5 5
#> [6,] 0 0 0 0 0 6
#> elev hydrologyh hydrologyv firept roadway usoil1
#> [1,] 0.009009565 0.009252263 0.1821840 0.9972083 0.4124801 0.999999
#> [2,] 0.013774511 0.236496281 0.5445930 0.9938265 0.6576982 0.999999
#> [3,] 0.008727421 0.109131157 0.1821840 0.9926046 0.6903713 0.999999
#> [4,] 0.011500357 0.115496525 0.3478247 0.9931336 0.6787981 0.999999
#> [5,] 0.088097356 0.247403211 0.2330456 0.9295551 0.9564531 0.999999
#> [6,] 0.028542169 0.721289745 0.8352680 0.9946876 0.6552929 0.999999
#> elev hydrologyh hydrologyv firept roadway usoil1
#> [1,] 0.009009565 0.009252263 0.1821840 0.9972083 0.4124801 0.008641975
#> [2,] 0.013774511 0.236496281 0.5445930 0.9938265 0.6576982 0.008641975
#> [3,] 0.008727421 0.109131157 0.1821840 0.9926046 0.6903713 0.008641975
#> [4,] 0.011500357 0.115496525 0.3478247 0.9931336 0.6787981 0.008641975
#> [5,] 0.088097356 0.247403211 0.2330456 0.9295551 0.9564531 0.008641975
#> [6,] 0.028542169 0.721289745 0.8352680 0.9946876 0.6552929 0.008641975
#> [1] "family" "par" "par2"
#> [4] "npars" "familyname" "tau"
#> [7] "beta" "taildep" "call"
#> [10] "nobs" "logLik" "AIC"
#> [13] "BIC" "emptau" "p.value.indeptest"
#> Bivariate copula: Survival BB1 (par = 0.42, par2 = 1.08, tau = 0.24)
#> [1] 142.2714
#> [1] "family" "par" "par2"
#> [4] "npars" "familyname" "tau"
#> [7] "beta" "taildep" "call"
#> [10] "nobs" "logLik" "AIC"
#> [13] "BIC" "emptau" "p.value.indeptest"
#> Bivariate copula: BB8 (par = 6, par2 = 0.64, tau = 0.48)
#> [1] 455.243
#> [1] "family" "par" "par2"
#> [4] "npars" "familyname" "tau"
#> [7] "beta" "taildep" "call"
#> [10] "nobs" "logLik" "AIC"
#> [13] "BIC" "emptau" "p.value.indeptest"
#> Bivariate copula: Gaussian (par = -0.12, tau = -0.08)
#> [1] 12.58269
#> [1] "family" "par" "par2"
#> [4] "npars" "familyname" "tau"
#> [7] "beta" "taildep" "call"
#> [10] "nobs" "logLik" "AIC"
#> [13] "BIC" "emptau" "p.value.indeptest"
#> Bivariate copula: BB8 (par = 1.45, par2 = 0.82, tau = 0.1)
#> [1] 22.52869
#> [1] "family" "par" "par2" "npars" "familyname"
#> [6] "tau" "beta" "taildep" "call" "logLik"
#> [11] "AIC" "BIC" "AIClist"
#> Bivariate copula: Frank (par = 100, tau = 0.96)
#> [1] 49.95172

```

```

#> Bivariate copula: BB8 (par = 1.13, par2 = 0.99, tau = 0.07)
#> [1] 18.15483
#> Bivariate copula: BB8 (par = 1.34, par2 = 0.8, tau = 0.07)
#> [1] 10.46027
#> Bivariate copula: BB1 (par = 0.13, par2 = 1, tau = 0.06)
#> [1] 9.663027
#> Bivariate copula: Gaussian (par = -0.52, tau = -0.35)
#> [1] 4.701402
#> Bivariate copula: Survival Gumbel (par = 1.01, tau = 0.01)
#> [1] 2.255718
#> Bivariate copula: Frank (par = -0.23, tau = -0.03)
#> [1] 1.297864
#> Bivariate copula: Gaussian (par = 0.16, tau = 0.1)
#> [1] 0.3783073
#> Bivariate copula: Frank (par = -0.32, tau = -0.04)
#> [1] 2.175075
#> Bivariate copula: Frank (par = -0.59, tau = -0.07)
#> [1] 0.1257136
#> Bivariate copula: Frank (par = -1.14, tau = -0.12)
#> [1] 0.2357644

# Class 2: fit joint vine density
n2 = nrow(udat62)
rr2 = vWcor(udat62, is_disc=c(F,F,F,F,F,T))
print(round(rr2,3))
#>      [,1] [,2] [,3] [,4] [,5] [,6]
#> [1,] 1.000 0.320 0.165 0.286 0.045 0.282
#> [2,] 0.320 1.000 0.581 0.052 0.061 0.084
#> [3,] 0.165 0.581 1.000 0.041 -0.034 -0.023
#> [4,] 0.286 0.052 0.041 1.000 0.324 0.148
#> [5,] 0.045 0.061 -0.034 0.324 1.000 0.186
#> [6,] 0.282 0.084 -0.023 0.148 0.186 1.000

vine62 = GaussVineMST(rr2,n2,1.01);
#print(vine62)
joint62 = RVineCopSelectMix(udat62,is_disc=c(F,F,F,F,F,T),
  familyset=famset, vine_array=vine62$VineA, iprint=T)
#>      [,1] [,2] [,3] [,4] [,5] [,6]
#> [1,] 1 1 2 1 4 1
#> [2,] 0 2 1 2 1 4
#> [3,] 0 0 3 3 2 5
#> [4,] 0 0 0 4 3 2
#> [5,] 0 0 0 0 5 3
#> [6,] 0 0 0 0 0 6
#>      [,1] [,2] [,3] [,4] [,5] [,6]
#> [1,] 1 1 2 1 4 1
#> [2,] 0 2 1 2 1 4
#> [3,] 0 0 3 3 2 5
#> [4,] 0 0 0 4 3 2
#> [5,] 0 0 0 0 5 3
#> [6,] 0 0 0 0 0 6
#>      elev hydrologyh hydrologyv roadway firept usoil2
#> [1,] 0.24920400 0.5390599 0.64106030 0.720647507 0.9816077 0.999999

```

```

#> [2,] 0.21902913 0.4937920 0.87468125 0.705082321 0.9829086 0.999999
#> [3,] 0.03573825 0.5915061 0.06677756 0.004281274 0.9802083 0.999999
#> [4,] 0.40334684 0.6943002 0.35891993 0.935299439 0.9023925 0.999999
#> [5,] 0.15962463 0.3168770 0.66487125 0.726537689 0.9811526 0.999999
#> [6,] 0.39099531 0.4462833 0.39128534 0.920179323 0.9213436 0.999999
#>      elev hydrologyh hydrologyv roadway firept usoil2
#> [1,] 0.24920400 0.5390599 0.64106030 0.720647507 0.9816077 0.05432099
#> [2,] 0.21902913 0.4937920 0.87468125 0.705082321 0.9829086 0.05432099
#> [3,] 0.03573825 0.5915061 0.06677756 0.004281274 0.9802083 0.05432099
#> [4,] 0.40334684 0.6943002 0.35891993 0.935299439 0.9023925 0.05432099
#> [5,] 0.15962463 0.3168770 0.66487125 0.726537689 0.9811526 0.05432099
#> [6,] 0.39099531 0.4462833 0.39128534 0.920179323 0.9213436 0.05432099
#> [1] "family"      "par"      "par2"
#> [4] "npars"      "familyname" "tau"
#> [7] "beta"      "taildep"    "call"
#> [10] "nobs"      "logLik"     "AIC"
#> [13] "BIC"      "emptau"     "p.value.indeptest"
#> Bivariate copula: Frank (par = 2.07, tau = 0.22)
#> [1] 87.41163
#> [1] "family"      "par"      "par2"
#> [4] "npars"      "familyname" "tau"
#> [7] "beta"      "taildep"    "call"
#> [10] "nobs"      "logLik"     "AIC"
#> [13] "BIC"      "emptau"     "p.value.indeptest"
#> Bivariate copula: Frank (par = 5.37, tau = 0.48)
#> [1] 405.9774
#> [1] "family"      "par"      "par2"
#> [4] "npars"      "familyname" "tau"
#> [7] "beta"      "taildep"    "call"
#> [10] "nobs"      "logLik"     "AIC"
#> [13] "BIC"      "emptau"     "p.value.indeptest"
#> Bivariate copula: BB1 (par = 0.4, par2 = 1.02, tau = 0.18)
#> [1] 81.7438
#> [1] "family"      "par"      "par2"
#> [4] "npars"      "familyname" "tau"
#> [7] "beta"      "taildep"    "call"
#> [10] "nobs"      "logLik"     "AIC"
#> [13] "BIC"      "emptau"     "p.value.indeptest"
#> Bivariate copula: BB8 (par = 1.83, par2 = 0.89, tau = 0.22)
#> [1] 119.2709
#> [1] "family"      "par"      "par2"      "npars"      "familyname"
#> [6] "tau"      "beta"      "taildep"    "call"      "logLik"
#> [11] "AIC"      "BIC"      "AIClist"
#> Bivariate copula: Survival BB1 (par = 2.75, par2 = 1.13, tau = 0.63)
#> [1] 76.33037
#> Bivariate copula: Frank (par = -0.26, tau = -0.03)
#> [1] 1.445934
#> Bivariate copula: Frank (par = -0.4, tau = -0.04)
#> [1] 3.493444
#> Bivariate copula: t (par = -0.1, par2 = 8.64, tau = -0.06)
#> [1] 14.37245
#> Bivariate copula: Survival Gumbel (par = 1.05, tau = 0.04)
#> [1] 1.443597

```

```

#> Bivariate copula: Frank (par = -0.24, tau = -0.03)
#> [1] 1.149773
#> Bivariate copula: Gaussian (par = 0.07, tau = 0.04)
#> [1] 3.850758
#> Bivariate copula: BB8 (par = 2.19, par2 = 1, tau = 0.39)
#> [1] 22.70315
#> Bivariate copula: Frank (par = -0.86, tau = -0.09)
#> [1] 14.24807
#> Bivariate copula: t (par = -0.06, par2 = 8.58, tau = -0.04)
#> [1] 2.111629
#> Bivariate copula: Frank (par = -1.86, tau = -0.19)
#> [1] 7.963544

#save(file="step3.RData", udat61, udat62, rr1, rr2,
# vine61,vine62, joint61,joint62)

```

Step 4. Holdout set comparisons

```

nh = nrow(holdo)
Amat6 = matrix(0,nh,2)
Bmat6 = matrix(0,nh,2)

# Output of RVineLogPDFMix has columns "cllk", "ullk"
# for copula density log-likelihood and
# univariate contribution to log-likelihood from discrete variables.
Amat6 = RVineLogPDFMix(cbind(uholda,uplusa),cbind(uholda,uminua),
  idiscv=c(F,F,F,F,F,T),
  A=vine61$VineA, ntrunc=5, fammat=joint61$family,
  param1=joint61$par, param2=joint61$par2, iprint=F)

Bmat6 = RVineLogPDFMix(cbind(uholdb,uplusb),cbind(uholdb,uminub),
  idiscv=c(F,F,F,F,F,T),
  A=vine62$VineA, ntrunc=5, fammat=joint62$family,
  param1=joint62$par, param2=joint62$par2)

#print(summary(exp(Amat6[,2])-upmfa)) # 0s
#print(summary(exp(Bmat6[,2])-upmfb)) # 0s

# Classification for holdout set based on fitted vine densities
# Joint densties from copula and univariate contributions
ajpdf6 = exp(Amat6[,1])*apdf6
bjpdf6 = exp(Bmat6[,1])*bpdf6
jpra6 = ajpdf6/(ajpdf6+bjpdf6)

print(summary(jpra6))
#>      Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
#> 0.0000001 0.2121904 0.5373379 0.5009504 0.7941061 0.9996187

jointBayes6 = table(holdo$cover,jpra6<0.5)
colnames(jointBayes6)=c("pred1","pred2")
rownames(jointBayes6)=c("actual1","actual2")

```

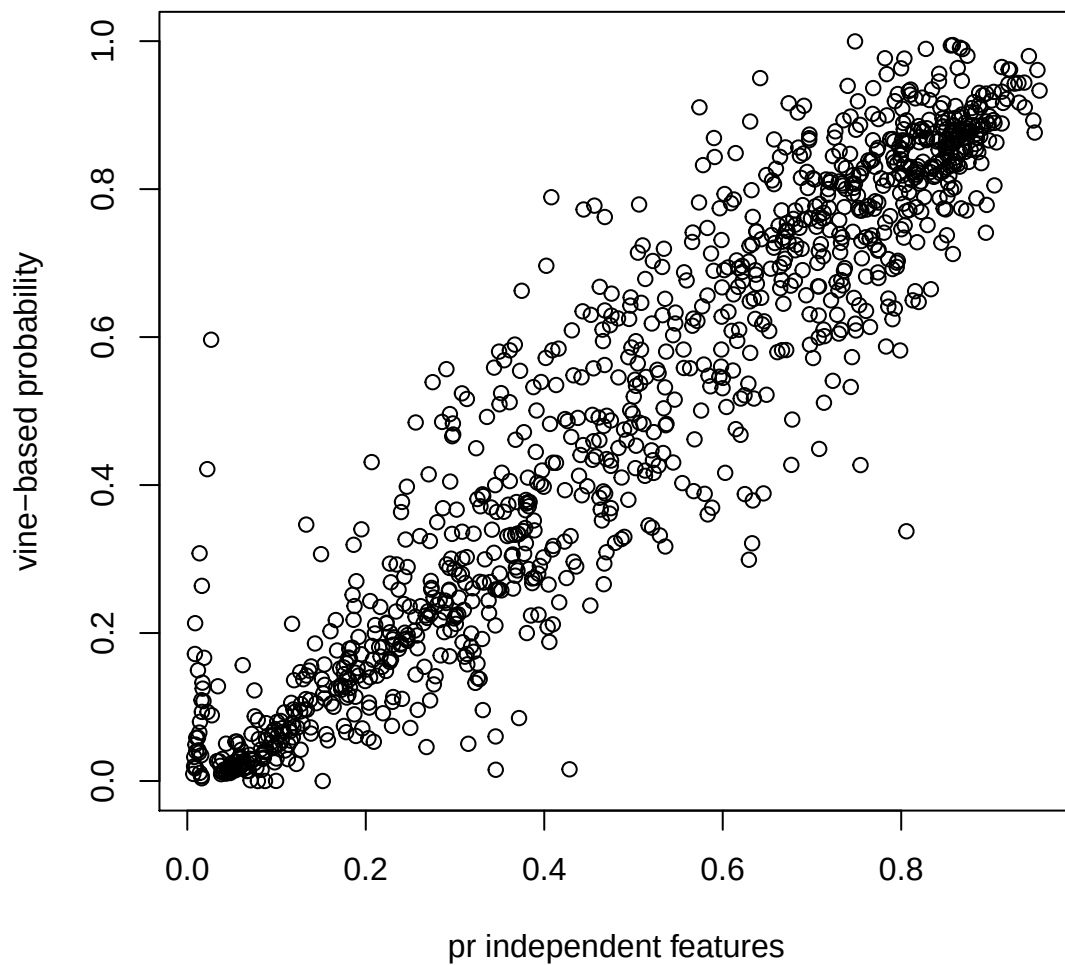
```

print(jointBayes6)
#>
#>      pred1 pred2
#> actual1  420  120
#> actual2  145  395

print(naiveBayes6)
#>
#>      pred1 pred2
#> actual1  409  131
#> actual2  145  395

par(mfrow=c(1,1))
plot(pra6,jpra6,xlab="pr independent features", ylab="vine-based probability")

```



```

# Visible scatter, despite similar tables based on cutpoint of 0.5

print(cor(pra6,jpra6))
#> [1] 0.9409787

# Compare predicted probability values for class A to see effect of joint densities
pra6categ = cut(pra6, breaks=c(0,0.1,0.2,0.5,0.8,0.9,1.0))
jpra6categ = cut(jpra6, breaks=c(0,0.1,0.2,0.5,0.8,0.9,1.0))

# classification by predprob in (0,.1),(.1,.2),(.2,.5),(.5,.8),(.8,.9),(.9,1)
print(table(holdo$cover,pra6categ)) # naive Bayes
#>      pra6categ
#>      (0,0.1] (0.1,0.2] (0.2,0.5] (0.5,0.8] (0.8,0.9] (0.9,1]
#>  1         6        25        100        227        160        22
#>  2       112        68       215       118        25         2

print(table(holdo$cover,jpra6categ)) # joint densities for each class
#>      jpra6categ
#>      (0,0.1] (0.1,0.2] (0.2,0.5] (0.5,0.8] (0.8,0.9] (0.9,1]
#>  1        13        24         83        194        173        53
#>  2       141        82       172       105        29        11

# More confident correct cases with jpra6categ.
# Also more confident incorrect cases.

```