Applications Flexible asymmetric multivariate distributions based on two-piece univariate distributions

Jonas Baillien, Department of Mathematics and Leuven statistics Research Center (LStat), Celestijnenlaan 200 B, Heverlee, 3001, Belgium
Irène Gijbels, Department of Mathematics and Leuven statistics Research Center (LStat), Celestijnenlaan 200 B, Heverlee, 3001, Belgium
Anneleen Verhasselt, Center for Statistics, Data Science Institute, Agoralaan Building D, Diepenbeek, 3590, Belgium

02-03-2022

```
# change to the correct files downloadable at:
# https://github.com/Anonymous162222/LCQBA
source("C:/Users/public/Code part 1 Functions.R")
```

The first data example is often encountered in papers on the topic of multivariate asymmetric distributions, the AIS-dataset. The data, plotted below,

concerns the body mass index (bmi) calculated as height (in cm) divided by squared mass (in kg) and lean body mass (lbm, expressed in kg), which is the body mass without fat mass, of 202 Australian athletes. The data is freely available in the `DAAG`-package in `R` and originates from (Cook and Weisberg 1994).

This document will show the process how we obtained the results concerning the linear combination of QBA-distributions shown in the paper.

The first thing we do, is create all possible combinations. This is up to a permutation. We consider four options for the independent components: the QBA-normal, -Laplace, -logistic and -Student's t. Since we have a bivariate dataset, we need to combine two of these. This leaves us with a total of ten different models. Each of these ten models is then fitted to the data using the following code

```
dat=as.matrix(ais[,c(6,9)])

### matrix of all possible combinations of base functions ###
#############################################################

d=length(dat[1,])
bf=c("normal","t","laplace","logistic")
expandlist=list()
for(i in 1:d){
  expandlist[[i]]=bf
}
basefuncs=as.matrix(expand.grid(expandlist))
basefuncs=apply(basefuncs,1,sort)
basefuncs=t(basefuncs)
basefuncs=unique(basefuncs,margin=1)

nmods=length(basefuncs[,1])
```

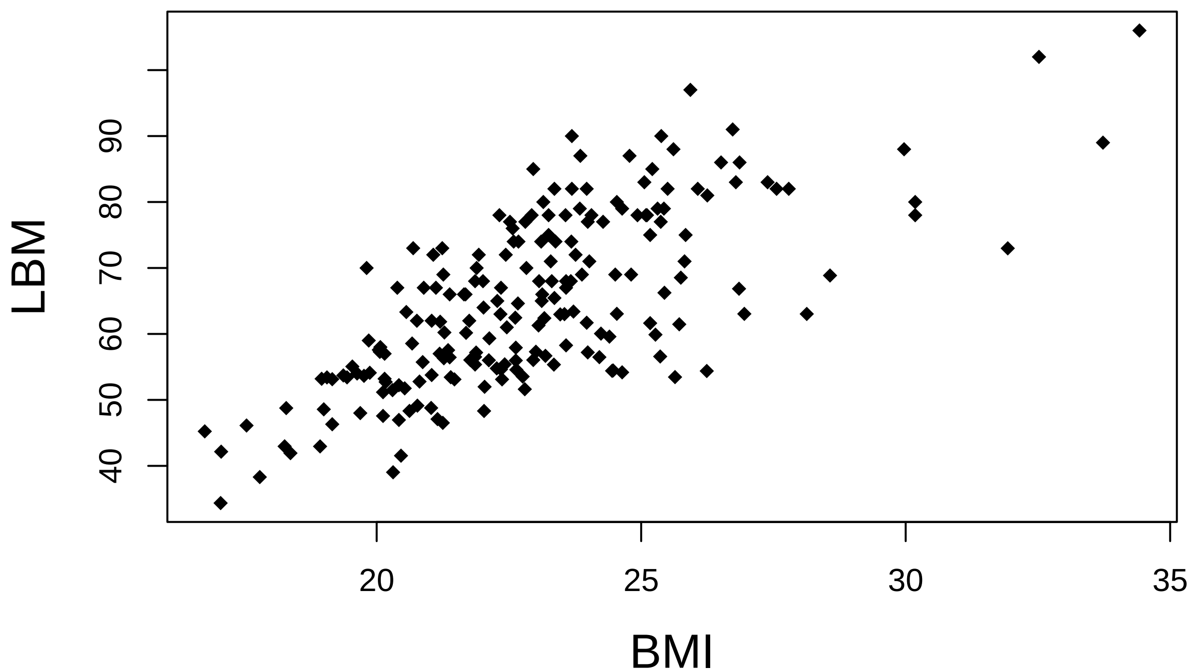

Figure 1: AIS data.

```r
### fitting the models to the data ###
######################################

# for the fitter
seed=126
maxiter=50000
tol=10^-5
numstarts=20
result=list("models"=basefuncs)

for(i in 1:nmods){

  # optional to track progress
  # print(paste0("Fitting model ", i, " of ",nmods))

  # basefunctions of the model fitted to the data
  basefunc=basefuncs[i,]

  # fitting the model to the data
  fit=LCQBAFD_fit(data=dat,basefunc=basefunc,seed=seed,maxiter=maxiter,tol=tol,
                      numstarts=numstarts)
  output=c(as.numeric(fit$`fitted parameters`),fit$`log likelihood fit`)
  result[[paste0("model",i)]]=output

}
```

After tidying up and combining the output of the different fits, the following table of parameter estimates per model is obtained.

Table 1: AIS data: Fitted models using linear combinations of the four mentioned QBA-distributions.

BF1	BF2	alpha1	alpha2	mu1	mu2	A11	A21	A12	A22	df1	df2	loglikelihood
normal	normal	0.78	0.7	20.05	54.61	-0.75	-0.65	-0.91	-5.22	NA	NA	-1208.06
normal	t	0.3	0.77	20.14	54.57	0.63	-0.68	5.18	-0.86	NA	7.42	-1205.51
laplace	normal	0.8	0.7	19.94	54.64	-0.54	-0.67	-0.61	-5.26	NA	NA	-1210.81
logistic	normal	0.22	0.36	20.39	57.34	0.4	0.72	0.41	5.77	NA	NA	-1206.37
t	t	0.3	0.78	20.02	54.43	0.63	-0.71	5.09	-0.86	53.69	32.15	-1207.4
laplace	t	0.66	0.29	21.08	54.5	-0.84	0.39	-1.62	4.66	NA	465.88	-1212.02
logistic	t	0.75	0.3	20.31	54.68	-0.46	0.6	-0.62	5.11	NA	251.66	-1205.66
laplace	laplace	0.25	0.74	20.13	53.11	0.44	-0.66	3.77	-0.82	NA	NA	-1222.78
laplace	logistic	0.75	0.71	20.28	54.43	-0.64	-0.34	-0.81	-2.93	NA	NA	-1215.32
logistic	logistic	0.72	0.78	20.05	54.36	-0.37	-0.41	-2.93	-0.45	NA	NA	-1209.48

Using these parameter estimates and the asymptotic theory, standard errors for the parameter estimates can be calculated. This is done for the "normal-normal" model to illustrate the process. Note that the parameter estimates are not yet in their final formulation, as they still need to be transformed to fulfill the constraints mentioned in the paper. The operation in order to put them in the desired formulation are: (1) Change the sign of the first row of A to make the diagonal element positive. When doing this, alpha1 needs to be transformed to 1-alpha1. (2) Do the same as in (1) for the second row of A.

```r
basefunc=c("normal","normal")
alpha=c(0.22,0.3)
```

```r
mu=c(20.05,54.61)
A=matrix(c(0.75,0.65,0.91,5.22),nrow=2,ncol=2)
tpars=c(NA,NA)

IM=FIM(alpha = alpha,mu = mu,A = A,basefunc = basefunc,tpars = NULL)

VarCov=1/202*solve(IM)
rownames(VarCov)=c("alpha1","alpha2","mu1","mu2","A11","A21","A12","A22")
colnames(VarCov)=c("alpha1","alpha2","mu1","mu2","A11","A21","A12","A22")
kable(round(VarCov,5),caption = "Variance-Covariance matrix of the parameter estimates of
      QBA-normal - QBA-normal model.")
```

Table 2: Variance-Covariance matrix of the parameter estimates of QBA-normal - QBA-normal model.

	alpha1	alpha2	mu1	mu2	A11	A21	A12	A22
alpha1	0.00187	0.00000	0.01306	0.01585	0.00458	0.00000	0.00556	0.00000
alpha2	0.00000	0.00229	0.01132	0.09093	0.00000	0.00284	0.00000	0.02279
mu1	0.01306	0.01132	0.18105	0.70062	0.03018	0.01194	0.02441	0.11006
mu2	0.01585	0.09093	0.70062	5.10714	0.00218	0.14815	-0.24697	0.94723
A11	0.00458	0.00000	0.03018	0.00218	0.01455	-0.00215	0.03089	-0.00260
A21	0.00000	0.00284	0.01194	0.14815	-0.00215	0.00960	-0.01723	0.04273
A12	0.00556	0.00000	0.02441	-0.24697	0.03089	-0.01723	0.14370	-0.02091
A22	0.00000	0.02279	0.11006	0.94723	-0.00260	0.04273	-0.02091	0.30149

In order to obtain the standard deviations mentioned in Table 4, the square root of the diagonal of this matrix needs to be taken. Finally, the DD plot can be created. To calculate the depth in the sample, we calculate the halfspace depth of each of the datapoints in the AIS dataset itself. The theoretical depth is calculated as the depth of the datapoints in a sample of size 10000 from the fitted distribution.

```r
sampsize=10000
seed=12457

sample=rLCQBA(A=A,mu=mu,basefunc=basefunc,alpha=alpha,sampsize=sampsize,d=d,
              tpars=tpars,seed=seed+23)
X=sample[[2]]

DinDist=hdepth(x=X,z=dat)$depthZ # theoretical depth
DinSample=hdepth(x=dat,z=dat)$depthZ # sample depth

plot(DinDist,DinSample,xlab="Theoretical depth",ylab="Depth in sample",cex.lab=1.5,cex.axis=1.5)
abline(a=0,b=1,lwd=2)
```

references

Cook, R. D., and S. Weisberg. 1994. *An Introduction to Regression Graphics*. Wiley Series in Probability and Mathematical Statistics. New York (N.Y.): Wiley.

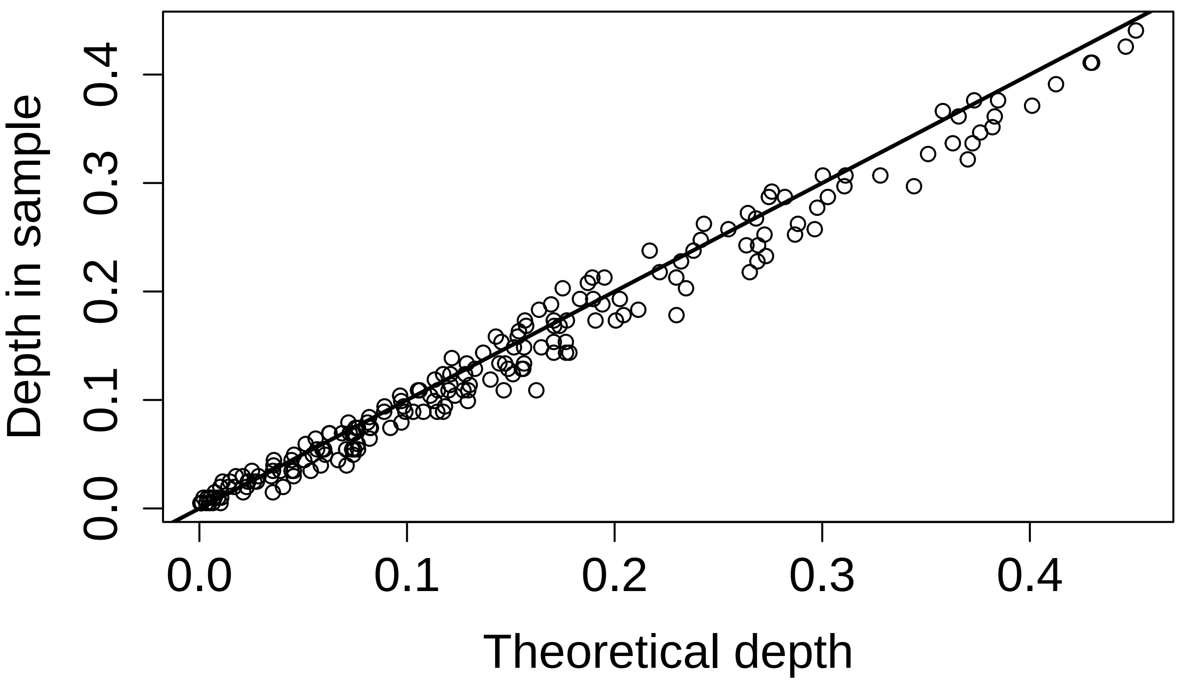

Figure 2: DD-plot of the QBA-normal - QBA-normal model.