Multivariate tools for compositional data analysis: the ToolsForCoDA package

Jan Graffelman
Universitat Politècnica de Catalunya
version 1.0.0
November 13, 2017

Abstract

Package **ToolsForCoDA** contains some functions for multivariate analysis with compositional data. It currently provides functions for doing compositional canonical correlation analysis. This analysis requires two data matrices of compositions, which can be adequately tranformed and used as entries in a specialized program for canonical correlation analysis, that is able to deal with singular covariance matrices. Some additional methods for the multivariate analysis of compositional data are planned to be included.

Keywords: log-ratio tranformation, canonical correlation analysis, generalized inverse.

---

1. Introduction

The **ToolsForCoDa** package provides some tools for the multivariate analysis of compositional data in the R environment (R Core Team 2014). The package is available from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=ToolsForCoDa.

This vignette describes the first version 1.0.0 of the package, which mainly provides functions for doing canonical correlation analysis with compositional data.

The remainder of this vignette shows an R example session showing how to analyze two sets of compositions with the functions of the package, using a small artificial data set included in the package.

2. An example session for a canonical analysis of compositions

The **ToolsForCoDa** package can be installed as usual via the command line or graphical user interfaces, e.g., the package can be installed and loaded by:

```
R> install.packages("ToolsForCoDa")
R> library("ToolsForCoDa")
```

The document describing the package (this document) can be consulted from inside R by typing:

```
R> vignette("ToolsForCoDa")
```

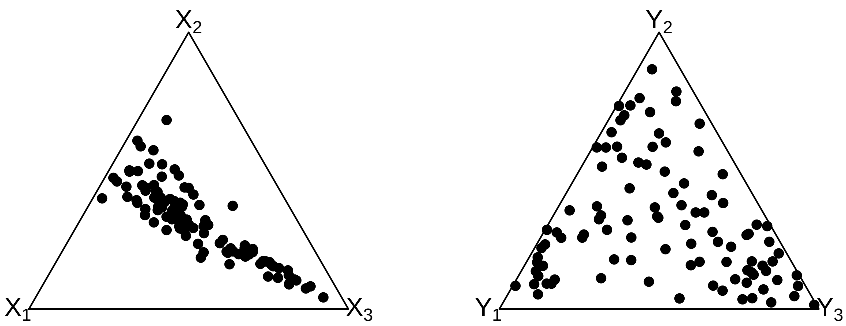
In the remainder we show how to perform the canonical analysis described in Section 3.1 of Graffelman et al. (2017).

We first load two artificial 3-part compositions.

```
R> library(HardyWeinberg) # needed for making some ternary diagrams
R> library(ToolsForCoDa)
R> data("Artificial")
R> Xsim.com <- Artificial$Xsim.com
R> Ysim.com <- Artificial$Ysim.com
```

We make the ternary diagrams of the two sets of compositions (Figure 1)

```
R> opar <- par(mfrow=c(1,2),mar=c(3,3,2,0)+0.5,mgp=c(2,1,0),pty="s")
R> par(mfg=c(1,1))
R> out <- HWTernaryPlot(Xsim.com,50,region=0,vbounds=FALSE,hwcurve=FALSE,
+                      vertexlab=c(expression(X[1]),
+                                  expression(X[2]),
+                                  expression(X[3])))
R> par(mfg=c(1,2))
R> out <- HWTernaryPlot(Ysim.com,50,region=0,vbounds=FALSE,
+                      hwcurve=FALSE,
+                      vertexlab=c(expression(Y[1]),
+                                  expression(Y[2]),
+                                  expression(Y[3])))
R> par(opar)
```



We do the centred log-ratio transformation

```
R> Xsub.clr <- clrmat(Xsim.com)
R> Ysub.clr <- clrmat(Ysim.com)
R> colnames(Xsub.clr) <- paste("X",1:3,sep="")
R> colnames(Ysub.clr) <- paste("Y",1:3,sep="")
```

We perform the canonical analysis:

```
R> res.cco <- canocov(Xsub.clr,Ysub.clr)
```

And we reproduce the results in Table 1. The canonical correlations are obtained as

```
R> round(diag(res.cco$ccor),digits=3)

[1] 0.944 0.129 0.000
```

The canonical weights of the X set and the Y set are obtained by:

```
R> res.cco$A

              [,1]      [,2]          [,3]
[1,]  0.0008130933  3.847198 -1.110223e-15
[2,] -0.7985815849 -3.446655  6.522560e-16
[3,]  0.7977684917 -0.400543  1.665335e-16

R> res.cco$B

           [,1]        [,2]          [,3]
[1,]  0.7624647 -0.05038131 -9.714451e-17
[2,] -0.7165761 -0.52116661  3.608225e-16
[3,] -0.0458886  0.57154792 -2.775558e-16
```

The canonical loadings of the X set and the Y set are obtained by

```
R> res.cco$Rxu

         [,1]       [,2]       [,3]
X1 -0.8857398  0.4641822 -0.9035794
X2 -0.9828511 -0.1844012 -0.4438392
X3  0.9940477 -0.1089461  0.6849272

R> res.cco$Ryv

         [,1]       [,2]       [,3]
Y1  0.8522677 -0.5231058  0.2439545
Y2 -0.6097840 -0.7925676  0.9387752
Y3 -0.3033098  0.9528920 -0.8183778
```

The adequacy coefficients of the X set and the Y set:

```
R> res.cco$fitXs

           [,1]       [,2]      [,3]
AdeX  0.9128873 0.08711271 0.4941914
cAdeX 0.9128873 1.00000000 1.4941914

R> res.cco$fitYs

           [,1]      [,2]      [,3]
AdeY  0.3967312 0.6032688 0.5368516
cAdeY 0.3967312 1.0000000 1.5368516
```

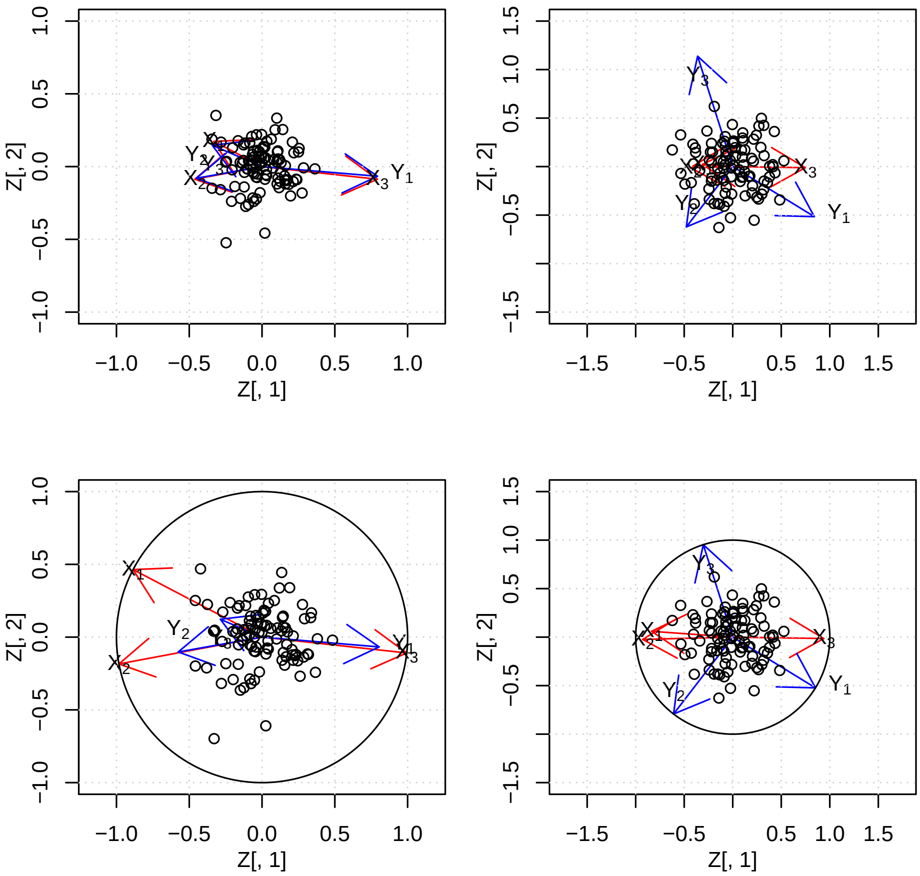
The redundancy coefficients of the X set and the Y set

```
R> res.cco$fitXp

           [,1]        [,2]       [,3]
RedX  0.8132984 0.001442577 0.06638809
cRedX 0.8132984 0.814740980 0.88112907

R> res.cco$fitYp

           [,1]        [,2]      [,3]
RedY  0.3534509 0.009990066 0.1440308
cRedY 0.3534509 0.363441013 0.5074718
```

Finally, we make the biplots given in Figure 2 of

```
R> opar <- par(mfrow=c(2,2),mar=c(3,3,2,0)+0.5,mgp=c(2,1,0))
R> par(mfg=c(1,1))
R> #
R> # Figure A
R> #
R> Z <- rbind(res.cco$Fs,res.cco$Gp)
R> plot(Z[,1],Z[,2],type="n",xlim=c(-1,1),ylim=c(-1,1),asp=1)
R> arrows(0,0,Z[1:3,1],Z[1:3,2],col="red")
R> arrows(0,0,Z[4:6,1],Z[4:6,2],col="blue")
R> text(res.cco$Fs[,1],res.cco$Fs[,2],
+      c(expression(X[1]),expression(X[2]),expression(X[3])))
R> text(res.cco$Gp[,1],res.cco$Gp[,2],
+      c(expression(Y[1]),expression(Y[2]),expression(Y[3])),pos=c(4,3,1))
R> grid()
R> fa <- 0.15
R> points(fa*res.cco$U[,1],fa*res.cco$U[,2])
R> par(mfg=c(1,2))
R> #
R> # Figure B
R> #
R>
R> Z <- rbind(res.cco$Fp,res.cco$Gs)
R> plot(Z[,1],Z[,2],type="n",xlim=c(-1.5,1.5),ylim=c(-1.5,1.5),asp=1)
R> arrows(0,0,Z[1:3,1],Z[1:3,2],col="red")
R> arrows(0,0,Z[4:6,1],Z[4:6,2],col="blue")
R> text(res.cco$Fp[,1],res.cco$Fp[,2],
+      c(expression(X[1]),expression(X[2]),expression(X[3])))
R> text(res.cco$Gs[,1],res.cco$Gs[,2],
+      c(expression(Y[1]),expression(Y[2]),expression(Y[3])),pos=c(4,3,1))
R> grid()
R> fa <- 0.25
R> points(fa*res.cco$V[,1],fa*res.cco$V[,2])
R> par(mfg=c(2,1))
R> #
R> # Standardizing the transformed data
R> #
R>
R> Xstan.clr <- scale(Xsub.clr)
R> Ystan.clr <- scale(Ysub.clr)
R> res.stan.cco <- canocov(Xstan.clr,Ystan.clr)
R> #
R> # Figure C
R> #
R>
R> Z <- rbind(res.stan.cco$Fs,res.stan.cco$Gp)
R> plot(Z[,1],Z[,2],type="n",xlim=c(-1,1),ylim=c(-1,1),asp=1)
R> arrows(0,0,Z[1:3,1],Z[1:3,2],col="red")
```

```
R> arrows(0,0,Z[4:6,1],Z[4:6,2],col="blue")
R> text(res.stan.cco$Fs[,1],res.stan.cco$Fs[,2],
+       c(expression(X[1]),expression(X[2]),expression(X[3])))
R> text(res.stan.cco$Gp[,1],res.stan.cco$Gp[,2],
+       c(expression(Y[1]),expression(Y[2]),expression(Y[3])),pos=c(4,3,1))
R> grid()
R> fa <- 0.2
R> points(fa*res.stan.cco$U[,1],fa*res.stan.cco$U[,2])
R> circle()
R> par(mfg=c(2,2))
R> #
R> # Figure D
R> #
R>
R> Z <- rbind(res.stan.cco$Fp,res.stan.cco$Gs)
R> plot(Z[,1],Z[,2],type="n",xlim=c(-1.5,1.5),ylim=c(-1.5,1.5),asp=1)
R> arrows(0,0,Z[1:3,1],Z[1:3,2],col="red")
R> arrows(0,0,Z[4:6,1],Z[4:6,2],col="blue")
R> text(res.stan.cco$Fp[,1],res.stan.cco$Fp[,2],
+       c(expression(X[1]),expression(X[2]),expression(X[3])))
R> text(res.stan.cco$Gs[,1],res.stan.cco$Gs[,2],
+       c(expression(Y[1]),expression(Y[2]),expression(Y[3])),pos=c(4,3,1))
R> grid()
R> fa <- 0.25
R> points(fa*res.stan.cco$V[,1],fa*res.stan.cco$V[,2])
R> circle()
R> par(opar)
```



Z[, 1]
Z[, 2]
Y1
Y2
Y3
X1
X2
X3

Acknowledgments

This work was partially supported by grant 2014SGR551 from the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) of the Generalitat de Catalunya, by grant MTM2015-65016-C2-2-R (MINECO/FEDER) of the Spanish Ministry of Economy and Competitiveness and European Regional Development Fund.

References

Graffelman J, Pawlowsky-Glahn V, Egozcue J, Buccianti A (2017). "Compositional Canonical Correlation Analysis." doi:10.1101/144584. URL http://dx.doi.org/10.1101/144584.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Affiliation:

Jan Graffelman
Department of Statistics and Operations Research
Universitat Politècnica de Catalunya
Barcelona, Spain
E-mail: jan.graffelman@upc.edu
URL: http://www-eio.upc.es/~jan/