R Code for Hausdorff and Simplex Dispersion Orderings in the 2D Case

Guillermo Ayala

Abstract. This paper proposes a software implementation using R of the Hausdorff and simplex dispersion orderings. A copy can be downloaded from `http://www.uv.es/~ayala/software/fun-disp.R`. The paper provides some examples using the functions *exactHausdorff* for the Hausdorff dispersion ordering and the function *simplex* for the simplex dispersion orderings. Some auxiliary functions are commented too.

1 The Introduction

The Hausdorff and simplex dispersion orderings have proposed in [6] and [1] respectively. Although the definitions are considered for d-dimensional random vectors we will assume in this paper 2-dimensional random vectors.

First, let us give basic notation used later. If $x \in \mathbb{R}^d$ and $r \in [0,\infty)$ then $B(x,r)$ is the ball centered at x with radius r. If $A \subset \mathbb{R}^d$ then $co(A)$ will denote the convex hull of the set A. For $A,B \subset \mathbb{R}^d$ then the Hausdorff distance between them will be denoted as $d_H(A,B)$ and $A+B$ the Minkowski addition. The usual stochastic ordering will be denoted as $\preceq_{st}$.

Let us begin by remembering those definitions. If X and Y are two random vectors and $r \in [0,\infty)$ then X is less dispersive than Y in the *Hausdorff dispersion ordering* for the index r, denoted as $X \preceq_H^r Y$, if

$$d_H\big(co\left(\{X\} \cup B_r(EX)\right), co\left(\{X'\} \cup B_r(EX)\right)\big) \tag{1}$$

$$\preceq_{st} d_H\big(co\left(\{Y\} \cup B_r(EY)\right), co\left(\{Y'\} \cup B_r(EY)\right)\big),$$

with X and X' i.i.d. (respectively Y and Y' i.i.d.).

Guillermo Ayala
Dpto. de Estadística e Investigación Operativa, Universidad de Valencia, 46100-Burjasot, Spain
e-mail: `Guillermo.Ayala@uv.es`

C. Borgelt et al. (Eds.): Combining Soft Computing & Stats. Methods, AISC 77, pp. 25–31.
springerlink.com

Let X and Y be random vectors then X is less dispersive than Y in the *simplex dispersion ordering*, denoted as $X \preceq_{sx} Y$, if

$$d_H(\mathscr{S}_{\boldsymbol{X}}, \mathscr{S}_{\boldsymbol{X}'}) \preceq_{st} d_H(\mathscr{S}_{\boldsymbol{Y}}, \mathscr{S}_{\boldsymbol{Y}'}) \tag{2}$$

where $\boldsymbol{X} = (X_1, X_2, X_3)$ and $\boldsymbol{Y} = (Y_1, Y_2, Y_3)$ are two random samples of X and Y and $\mathscr{S}_{\boldsymbol{X}}$ is the convex hull of $\boldsymbol{X}$.

I have developed a collection of R functions in order to evaluate both dispersion orderings. This collection of R functions can be download from `http://www.uv.es/~ayala/software/fun-disp.R`. In this paper, we explain how they can be used. Note that the analyses included in this paper can be reproduced by a simple copy-paste of the code.

2 Data

First, we declare our functions and load the packages needed later. In particular, we will need the R packages *geometry, mvtnorm* and *Hmics* [4],[3], [5]. We will give later details about their use.

```
> source("fun-disp.R"); library(geometry); library(mvtnorm)
```

We will use multivariate normal distribution data generated using the package *mvtnorm* [3]. Let us consider the $\mathbb{R}^d$-valued random vectors X and Y with normal distributions, $X \sim_{st} N(\mu_X, \Sigma_X)$ and $Y \sim_{st} N(\mu_Y, \Sigma_Y)$, where $\Sigma_X = AA^t$, $A \in M_{d\times d}$ being a matrix whose values are randomly chosen with uniform distribution in the interval $(0,1)$, the super index ^{t} denoting the transpose matrix, and $\Sigma_Y = \Sigma_X + \lambda I_d$, with $\lambda \geq 0$. It is well-known that the eigenvalues of Σ_Y are those of Σ_X plus the value λ. It holds that $X \preceq_{sx} Y$ [1]. Roughly speaking, larger values of λ will produce larger dispersion for the random vector Y.

Let us generate two point sets from the model just considered.

```
> n = 100; mu1 = rep(0,2); mu2 = mu1; lambda = 0.5; n1 = n2 = n = 100
> sigma1 = matrix(runif(4), nrow = 2, ncol = 2)
> sigma1 = t(sigma1) %*% sigma1
> sigma2 = sigma1 + lambda * diag(1, 2)
> A = rmvnorm(n1, mean = mu1, sigma = sigma1)
> B = rmvnorm(n2, mean = mu2, sigma = sigma2)
```

Figure 1 shows the original point set A and the corresponding convex hull obtained by using the package *geometry* [4] that contains an interface to *qhull* (`http://www.qhull.org/`).

```
> plot(A); chA = convhulln(A); lines(A[chA[1, ], ])
> for (i in 2:nrow(chA)) lines(A[chA[i, ], ])
```

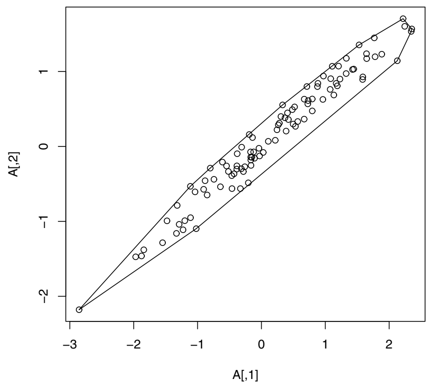

Fig. 1 A data set and the corresponding convex hull

3 Hausdorff Dispersion Ordering

The basic reference is [6]. We need to calculate the Hausdorff distance between $x_1 + B(y,r)$ and $x_2 + B(y,r)$ where $x_1, x_2, y \in \mathbb{R}^2$ and $B(m,r)$ is the disc centered at m with radius r. Our first approach will be to discretize $co(x + \partial B(y,r))$. The number of points used in the discretization is *NOP*. Then the continuous set $x + B(y,r)$ is replaced by the corresponding discrete set. Finally we calculate the Hausdorff distance between the corresponding convex hulls of the discrete sets composed by 100 points. The function *cone2* calculates this distance. Let us see how to calculate this Hausdorff distance.

```
> x1 = c(3, 5); x2 = c(7, 9); center = c(4, 4); radius = 1;
> NOP = 100
> cone2(x1, x2, center, radius, NOP)
```

```
[1] 5.656854
```

Given a random vector X and a random sample $\{x_1, \ldots, x_n\}$, The function *bootcone* provides us a bootstrap sample of $d_H(co(X_1^* + B(y,r)), co(X_2^* + B(y,r)))$ where X_1^*, X_2^* is a random sample without replacement from $\{x_1, \ldots, x_n\}$. Let us generate a sample and see the empirical distribution function in Figure 2.

```
> A.bc = bootcone(A, radius = 0.2, NOP = 100, nresamples = 10)
> Ecdf(A.bc)
```

If we have two samples x and y from X and Y then it can be tested if X is less dispersive than Y in the Hausdorff dispersion ordering using the following code. Note that the function *uso.test* tests the usual stochastic ordering using the Wilcoxon and Kolmogorov-Smirnov tests.

```
> AB.bc = bootcone2(A, B, radius, NOP = 100, nresamples = 10)
> uso.test(AB.bc$dhx, AB.bc$dhy)
```

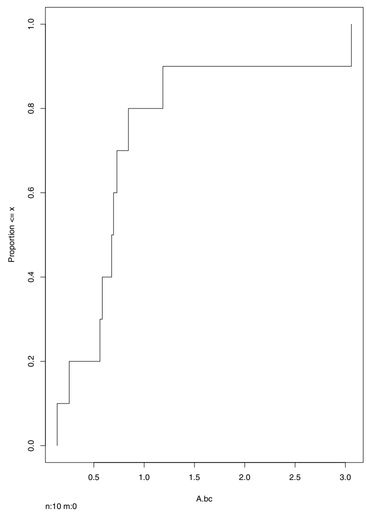

Fig. 2 Empirical distribution function of the Hausdorff distances $d_H(co(X_1^* + B(y,r)), co(X_2^* + B(y,r)))$.

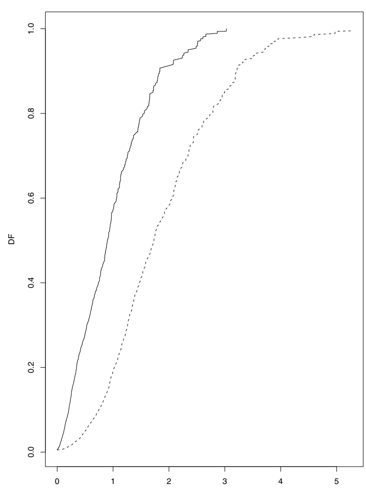

Fig. 3 Empirical distribution functions of the Hausdorff distances $d_H(co(X_1^* + B(y,r)), co(X_2^* + B(y,r)))$.

In the 2-dimensional case, an exact algorithm to calculate the distance $d_H(co(X_1^* + B(y,r)), co(X_2^* + B(y,r)))$ has been proposed [2]. It has been implemented in the function *exactHausdorff*. The following code calculates these distance from the two data sets and displays the empirical distribution functions. See Figure 3.

```
> r = 0.1; n= 100; prob = rep(1/n, n)
> HA = exactHausdorff(A, prob, r); HB = exactHausdorff(B, prob, r)
> plot(HA$distance, cumsum(HA$probability), type = "l", xlab = "",
+     ylab = "DF", xlim = range(c(HA,HB)))
> lines(HB$distance, cumsum(HB$probability), lty = 2)
```

Finally, the test if performed using the following code. The Kolmogorov-Smirnov test is used to test the usual stochastic ordering.

```
> y = rbind(A, B); x = c(rep(1, nrow(A)), rep(2, nrow(B)))
> z = testHDO(y, x, r = 0.2)
```

4 Simplex Dispersion Ordering

A detailed explanation of this algorithm can be found in [2].

If $A = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{n_1}\}$ and $B = \{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{n_2}\}$, let $\{i_1,\ldots,i_{d+1},i_{d+2},\ldots,i_{2(d+1)}\}$ be a sample without replacement from $\{1,\ldots,n_1\}$, and $U = d_H(\mathrm{co}(\boldsymbol{x}_{i_1},\ldots,\boldsymbol{x}_{i_{d+1}}),$ $\mathrm{co}(\boldsymbol{x}_{i_{d+2}},\ldots,\boldsymbol{x}_{i_{2(d+1)}}))$. Therefore, s_1 independent extractions from the set $\{1,\ldots,$ $n_1\}$ will produce a random sample of the corresponding bootstrap distribution $u_1,\ldots,u_{s_1}$. Replacing $\boldsymbol{x}$ by $\boldsymbol{y}$, we obtain $v_1,\ldots,v_{s_2}$, a random sample of the bootstrap distribution associated to the vector $\boldsymbol{y}$. Now, these values can be used for the proposed tests.

First, we describe some auxiliary functions. The function *rotatePHA* provides the angle to rotate a point w to the positive x-axis.

```
> w = c(-1, 5)
```

The angle is given by

```
> (tau = rotatePHA(w))
```

```
[1] 1.768192
```

and the rotated point can be found using

```
> AA = rbind(c(cos(tau), sin(tau)), c(-sin(tau), cos(tau)))
> w.rotated = t(AA %*% t(t(w)))
```

Given three points (corresponding with the rows of pp), we need to know if the convex hull of these points is a triangle, a segment or just they are the same point. This is given by the function *whichShape*.

```
> pp = matrix(data = c(0, -1, 0, -3, -3, 0), ncol = 2, byrow = T)
> whichShape(pp)
 [1] "triangle"
```

In order to calculate the Hausdorff distance between the point z and the convex hull of the points corresponding to the rows of pp, we have to move all points.

```
> z = c(-9, 1)
> pp = matrix(data = c(0, -1, 0, -3, -3, 0), ncol = 2, byrow = T)
> zpp = moveShapeAndPoint(z, pp, dib = T)
```

The different steps are illustrated in figure 4.

The function *distShape* calculates the Hausdorff distance by taking into account if the convex hull is a triangle, a segment or just a point. A detailed explanation of the calculations can be found in [2].

```
> z = c(-9, 1)
> pp = matrix(data = c(0, -1, 0, -3, -3, 0), ncol = 2, byrow = T)
> distShape(z, pp)
         1
2 6.082763
```

The function *simplex* provides us a sample of u's.

```
> d1 = simplex(A, withBootstrap = TRUE, nresamples = 100)
```

If we consider two different samples we can test the simplex dispersion ordering using

```
> d1 = simplex2(A, B, withBootstrap = TRUE, nresamples = 10)
> uso.test(d1$dhx, d1$dhy)
```

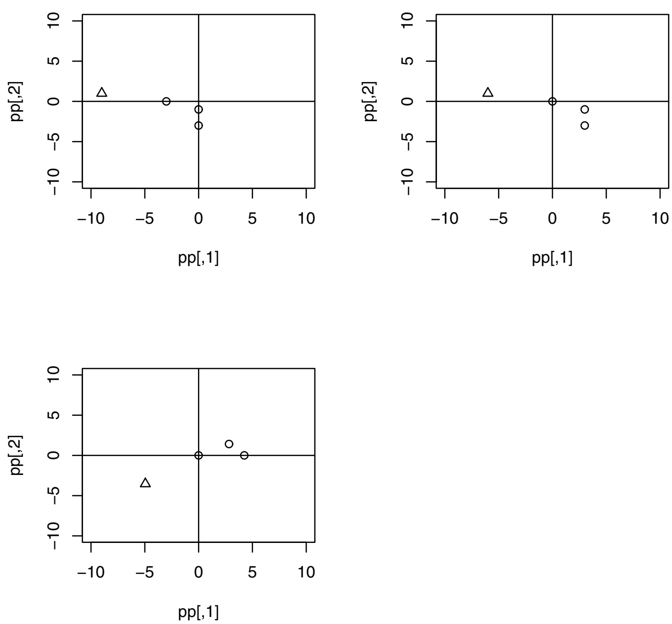

Fig. 4 Moving a triangle.

Acknowledgements. This paper has been funded by grant TIN2009-14392-C02-01. The software is a result of the collaboration with Miguel López-Díaz from the University of Oviedo.

References

1. Ayala, G., López-Díaz, M.: The simplex dispersion ordering and its application to the evaluation of human corneal endothelia. J. Multivariate Anal. 100, 1447–1464 (2009)
2. Ayala, G., López-Díaz, M.C., López-Díaz, M., Martínez-Costa, L.: Methods and algorithms to test the simplex and hausdorff dispersion orders with a simulation study and an ophthalmological application. Technical report, Universidad de Oviedo (2010)
3. Genz, A., Bretz, F., Hothorn, T.: mvtnorm: Multivariate Normal and t Distributions. R package version 0.9-2 (2008)
4. Grasman, R., Gramacy, R.B.: geometry: Mesh generation and surface tesselation. R package version 0.1-3 (2008)
5. Harrell, F.E.: Hmisc: Harrell Miscellaneous. R package version 3.7-0 (2009)
6. López-Díaz, M.: An indexed multivariate dispersion ordering based on the Hausdorff distance. J. Multivariate Anal. 97(7), 1623–1637 (2006)