Accepted Manuscript

Exactly computing bivariate projection depth contours and median

Xiaohui Liu, Yijun Zuo, Zhizhong Wang

PII: S0167-9473(12)00384-2
DOI: 10.1016/j.csda.2012.10.016
Reference: COMSTA 5429

To appear in: *Computational Statistics and Data Analysis*

Received date: 28 November 2011
Revised date: 21 October 2012
Accepted date: 22 October 2012

Please cite this article as: Liu, X., Zuo, Y., Wang, Z., Exactly computing bivariate projection depth contours and median. *Computational Statistics and Data Analysis* (2012), doi:10.1016/j.csda.2012.10.016

Exactly computing bivariate projection depth contours and median *

Xiaohui Liu[a,b], Yijun Zuo[b], Zhizhong Wang[c]

[a] School of Statistics, Jiangxi University of Finance and Economics, Nanchang, Jiangxi 330013, China

[b] Department of Statistics and Probability, Michigan State University, East Lansing, MI 48823, USA

[c] School of Mathematics Science and Computing Technology, Central South University, Hunan 410083, China

Abstract. Among their competitors, projection depth and its induced estimators are very favorable because they can enjoy very high breakdown point robustness without having to pay the price of low efficiency, meanwhile providing a promising center-outward ordering of multi-dimensional data. However, their further applications have been severely hindered due to their computational challenge in practice. In this paper, we derive a simple form of the projection depth function, when $(\mu, \sigma) = (\text{Med}, \text{MAD})$. This simple form enables us to extend the existing result of point-wise exact computation of projection depth (PD) of Zuo and Lai (2011) to depth contours and median for bivariate data.

Key words: Projection depth; Projection median; Projection depth contour; Exact computation algorithm; Linear fractional functionals programming;

2000 Mathematics Subject Classification Codes: 62F10; 62F40; 62F35

1 Introduction

To generalize order-related univariate statistical methods, depth functions have emerged as powerful tools for nonparametric multivariate analysis with the ability to provide a center-outward ordering of the multivariate observations. Points deep inside a data cloud obtain higher depth and those on the outskirts receive lower depth. Such depth-induced ordering enables one to develop favorable new robust estimators of multivariate location and scatter matrix. Since Tukey's introduction (Tukey, 1975), depth functions have gained much attention in the last two decades. Various depth notions have been introduced. To name a few, halfspace depth (Tukey, 1975), simplicial depth (Liu, 1990), regression depth (Rousseeuw and Hubert, 1999), and projection depth (Liu, 1992; Zuo and Serfling, 2000; Zuo, 2003).

Zuo and Serfling (2000) and Zuo (2003) found that among all the examined depth notions, projection depth is one of the favorite ones, enjoying very desirable properties. Furthermore, projection depth induced robust estimators, such as projection depth weighted means and median, can possess a very high breakdown point as well as high relative efficiency with appropriate choices of univariate location and scale estimators, serving as very favorable alternatives to the regular mean (Zuo, 2003; Zuo *et al.*, 2004). In fact, projection depth weighted means include as a special case the famous Stahel-Donoho estimator (Stahel, 1981; Donoho, 1982; Tyler, 1994; Maronna *et al.*, 1995; Zuo *et al.*, 2004), the latter is the first constructed location estimator in high dimensions enjoying high breakdown point robustness and affine equivariance, while the projection depth median has the highest breakdown point among all the existing affine equivariant multivariate location estimators (Zuo, 2003).

However, further prevalence of projection depth and its induced estimators is severely hindered by their computational intensity. The computation of projection depth seems intractable since it involves supremum over infinitely many direction vectors. There were only approximating algorithms in the last three decades until Zuo and Lai (2011), in which they proved that there is no need to calculate the supremum over infinitely many direction vectors in the bivariate data when the outlyingness function uses the very popular choice (Med, MAD) as the univariate location and scale pair. An exact algorithm

*Corresponding author's email: zuo@msu.edu, tel: 001-517-432-5413

for projection depth and its weighted mean, i.e. the Stahel-Donoho estimator, was also constructed in that paper.

In the current paper, we further generalize their idea to the higher dimensional cases by utilizing linear fractional functionals programming (Swarup, 1965). That is, we find that, with the choice of (Med, MAD), we only need to calculate the supremum over a finite number of direction vectors for $p \geq 2$, where p denotes the dimension of the data. Furthermore, these direction vectors are x-free, namely, independent of the point x for which the depth value is being computed, and depend only on the data cloud. Therefore, we derive a simple form of the projection depth function, and are able to compute the bivariate projection depth contours and median very conveniently through linear programming based on the procedure of Zuo and Lai (2011). It is found that sample projection depth contours are polyhedral under some mild conditions. Furthermore, it is noteworthy that the computational methods discussed in this paper have no limitation on the dimension p, and therefore may possibly be implemented to spaces with $p > 2$, as well as for the modified projection depth (Šiman, 2011) in a more general multidimensional regression context.

The rest of the paper is organized as follows. Section 2 provides the definitions of the projection depth contour and projection median. Section 3 presents the main idea of how to obtain a simple form of the projection depth function. Section 4 discusses the exact computational issue of the projection depth contour and projection median by linear programming. Some examples are given in Section 5. Both simulated and real data are considered in this section.

2 Definitions

For a given distribution F_1 on $\mathbb{R}^1$, let $\mu(F_1)$ be translation equivariant and scale equivariant, and $\sigma(F_1)$ be translation invariant and scale equivariant. Define the outlyingness of a point $x \in R^p$ ($p \geq 1$) with respect to the distribution F of the random variable $X \in R^p$ as (see (Zuo, 2003) and references therein)

$$O(x,\, F) = \sup_{\|u\|=1} |Q(u,\, x,\, F)|, \tag{1}$$

where $Q(u,\, x,\, F) = (u^\tau x - \mu(F_u))/\sigma(F_u)$. If $u^\tau x - \mu(F_u) = \sigma(F_u) = 0$, then define $Q(u,\, x,\, F) = 0$. F_u is the distribution of $u^\tau X$, which is the projection of X onto the unit vector u.

Throughout this paper, we select the very popular robust choice of μ and σ: the median (Med) and the median absolute deviation (MAD). Based on definition (1), the projection depth of any given point x with respect to F, $PD(x,\, F)$, can then be defined as (Liu, 1992; Zuo and Serfling, 2000; Zuo, 2003)

$$PD(x,\, F) = 1/(1 + O(x,\, F)).$$

With the outlyingness function and projection depth function defined above, we then define the projection depth median (PM) and contours (PC) as follows (Zuo, 2003)

$$PM(F) = \arg\sup_{x \in R^p} PD(x,\, F),$$

$$PC(\alpha,\, F) = \{x \in R^p : PD(x,\, F) = \alpha\},$$

where $0 < \alpha \leq \alpha^* = \sup_{x \in R^p} PD(x,\, F)$. Clearly, $PC(\alpha,\, F)$ is the boundary of the projection depth region (PR)

$$PR(\alpha,\, F) = \{x \in R^p : PD(x,\, F) \geq \alpha\}.$$

For a given sample $\mathcal{X}^n = \{X_1,\, X_2,\, \cdots,\, X_n\}$ from F, let F_n be the corresponding empirical distribution. By simply replacing F with F_n in $PM(F)$ and $PC(\alpha,\, F)$, we can obtain their sample version: $PM(F_n)$ and $PC(\alpha,\, F_n)$. Without confusion, we use $\mathcal{X}^n$ and F_n interchangeably in what follows. Furthermore, by noting the fact that for the choice of (Med, MAD), $Q(u,\, x,\, \mathcal{X}^n)$ in (1) is odd with respect to u, we drop the absolute value sign existing in definition (1), and consider

$$O(x,\, \mathcal{X}^n) = \sup_{\|u\|=1} Q(u,\, x,\, \mathcal{X}^n)$$

instead in what follows, where

$$Q(u,\, x,\, \mathcal{X}^n) = \frac{u^\tau x - \mathrm{Med}(u^\tau \mathcal{X}^n)}{\mathrm{MAD}(u^\tau \mathcal{X}^n)},$$

where $u^\tau x$ denotes the projection of x onto the unit vector u, and $u^\tau \mathcal{X}^n = \{u^\tau X_1,\, u^\tau X_2,\, \cdots,\, u^\tau X_n\}$. Let $Z_{(1)} \le Z_{(2)} \le \cdots \le Z_{(n)}$ be the order statistics based on the univariate random variables $\mathcal{Z}^n = \{Z_1,\, Z_2,\, \cdots,\, Z_n\}$, then

$$\mathrm{Med}(\mathcal{Z}^n) = \frac{Z_{(\lfloor (n+1)/2 \rfloor)} + Z_{(\lfloor (n+2)/2 \rfloor)}}{2},$$

$$\mathrm{MAD}(\mathcal{Z}^n) = \mathrm{Med}\{|Z_i - \mathrm{Med}(\mathcal{Z}^n)|,\, i = 1,\, 2,\, \cdots,\, n\},$$

where $\lfloor \cdot \rfloor$ is the floor function.

3 The main idea

Note that, for any given sample $\mathcal{X}^n$, the tasks of computing both $PM(\mathcal{X}^n)$ and $PC(\alpha,\, \mathcal{X}^n)$ mainly involve $O(x,\, \mathcal{X}^n)$, i.e.

$$PM(\mathcal{X}^n) = \arg \inf_{x \in R^p} O(x,\, \mathcal{X}^n),$$

$$PC(\alpha,\, \mathcal{X}^n) = \{x \in R^p : O(x,\, \mathcal{X}^n) = \beta\},$$

where $\beta = 1/\alpha - 1$. Thus, let's first focus on the computation of $O(x,\, \mathcal{X}^n)$. Without loss of generality, in what follows, we assume $\mathcal{X}^n$ to be in general position, which is commonly supposed in most existing literature; see for example Donoho and Gasko (1992).

The idea of a circular sequence (Edelsbrunner, 1987) (see also Dyckerhoff (2000); Cascos (2007)) implies that, for any given unit vector $v \in \mathcal{S} = \{u \in R^p : \|u\| = 1\}$, there must exist two permutations, say $(i_1,\, i_2,\, \cdots,\, i_n)$ and $(j_1,\, j_2,\, \cdots,\, j_n)$, of $(1,\, 2,\, \cdots,\, n)$ such that

$$v^\tau X_{i_1} \le v^\tau X_{i_2} \le \cdots \le v^\tau X_{i_n},$$

$$Y_{j_1} \le Y_{j_2} \le \cdots \le Y_{j_n},$$

where $Y_{j_l} = |v^\tau X_{j_l} - \mathrm{Med}(v^\tau \mathcal{X}^n)|$, $(1 \le l \le n)$. There is a small non-empty set $\mathcal{N}(v) \subset \mathcal{S}$ of v such that these hold true for any $v \in \mathcal{N}(v)$. This implies that the whole unit sphere $\mathcal{S}$ can be covered completely with at most N (of order $O(n^{4(p-1)})$) non-empty fragments $S_k = \{v \in \mathcal{S}$: v satisfies constraint conditions $\mathcal{Q}_k\}$ with $\mathcal{Q}_k$ being

$$\begin{cases} v^\tau(X_{i_{k,2}} - X_{i_{k,1}}) \ge 0, \\ v^\tau(X_{i_{k,3}} - X_{i_{k,2}}) \ge 0, \\ \quad\vdots \\ v^\tau(X_{i_{k,n}} - X_{i_{k,n-1}}) \ge 0, \\ Y_{j_{k,2}} - Y_{j_{k,1}} \ge 0, \\ Y_{j_{k,3}} - Y_{j_{k,2}} \ge 0, \\ \quad\vdots \\ Y_{j_{k,n}} - Y_{j_{k,n-1}} \ge 0, \\ \|v\| = 1, \end{cases}$$

for some fixed permutations $(i_{k,1},\, i_{k,2},\, \cdots,\, i_{k,n})$ and $(j_{k,1},\, j_{k,2},\, \cdots,\, j_{k,n})$ of $(1,\, 2,\, \cdots,\, n)$, where $1 \le k \le N$. We defer the discussion about the order of N until the Appendix; see *The order of N*.

Note that different fragments S_k $(k = 1,\, 2,\, \cdots,\, N)$ are connected and overlapped with each other only on the boundaries. Thus, to calculate $O(x,\, \mathcal{X}^n)$, it is sufficient to calculate

$$O(x,\, \mathcal{X}^n) = \max_{1 \le k \le N} O_k(x,\, \mathcal{X}^n)$$

with

$$O_k(x,\, \mathcal{X}^n) = \sup_{u \in S_k} Q(u,\, x,\, \mathcal{X}^n). \tag{2}$$

Furthermore, from the definition and properties of S_k, it is easy to see that, for any $u \in S_k$, the outlyingness function $Q(u, x, \mathcal{X}^n)$ can be simplified to

$$Q(u, x, \mathcal{X}^n) = \frac{u^\tau (x - X_{i_{k,m}})}{|u^\tau X_{j_{k,m}} - u^\tau X_{i_{k,m}}|}, \tag{3}$$

if n is odd with $m = (n+1)/2$, otherwise

$$Q(u, x, \mathcal{X}^n) = \frac{2u^\tau (x - X_{k,\iota})}{\left|u^\tau \left(X_{j_{k,m}} - X_{k,\iota}\right)\right| + \left|u^\tau \left(X_{j_{k,m+1}} - X_{k,\iota}\right)\right|}, \tag{4}$$

with $m = n/2$ and $X_{k,\iota} = (X_{i_{k,m}} + X_{i_{k,m+1}})/2$.

Remark 1. Based on the assumption of general position, the denominators in the above two formulas will not be 0 for any $u \in S_k$, since they are actually equal to $\mathrm{MAD}(u^\tau \mathcal{X}^n)$, and greater than 0 under such an assumption when $n \geq 2p$; see the proof of Theorem 3.4 in Zuo (2003) for a similar discussion.

By (3) and (4), we obtain the following proposition.

Proposition 1. Assume $\mathcal{X}^n$ are in general position. Then for any given k $(1 \leq k \leq N)$, the optimization problem (2) is equivalent to

$$O_k(x, \mathcal{X}^n) = \sup_z \frac{c_k^\tau z}{d_k^\tau z}, \tag{5}$$

subject to

$$\mathbb{A}_k z \geq 0 \tag{6}$$

where c_k, d_k and $\mathbb{A}_k$ will be specified in the Appendix. Here $\vartheta \geq 0$ means that ϑ is component-wise non-negative if ϑ is a vector, i.e. for any component ϑ_i, we have $\vartheta_i \geq 0$.

(5) with constraint conditions (6) is typically a linear fractional functionals programming problem. By Theorem 1 of Swarup (1965) (see also Šiman (2011) (p. 950) for a more general discussion), it is easy to show that the maximum of $c_k^\tau z / d_k^\tau z$ will only occur at a basic feasible solution of (6). Note that the number of fragments S_k is limited (at most N). Thus, we have

Theorem 1. Suppose that the choice of location and scale measures of projection depth function is the pair (Med, MAD). Then the number of direction vectors needed to compute the projection depth exactly is finite. Furthermore, these direction vectors only depend on the data cloud $\mathcal{X}^n$.

Remark 2. The idea of dividing the unit sphere $\mathcal{S}$ into fragments S_k by applying Med and MAD sequences was first used in Zuo and Lai (2011) for computing the bivariate projection depth; see also Paindaveine and Šiman (2011, 2012b) for other similar applications. Here we extend the result of Zuo and Lai (2011) to R^p $(p \geq 2)$. That is, one could compute PD in R^p exactly by only considering a finite number of direction vectors. Furthermore, the x-free property of these direction vectors can bring convenience to the computation of $\mathrm{PD}(x, \mathcal{X}^n)$ for any x, since we only need to search for the direction vectors once.

4 Exact computation of $PM(\mathcal{X}^n)$ and $PC(\alpha, \mathcal{X}^n)$

From the discussion above, we can obtain two observations as follows, namely, for any given x,

- the way to divide sphere $\mathcal{S}$ into fragments S_k $(k = 1, 2, \cdots, N)$ is fixed, i.e. x-free, as long as the data cloud $\mathcal{X}^n$ is fixed.
- there is no need to calculate $Q(u, x, \mathcal{X}^n)$ over an infinite number of direction vectors. It is sufficient to calculate it for $u_1, u_2, \cdots, u_M$, namely, the unit direction vectors corresponding to all the basic feasible solutions of N linear fractional functionals programming problems, where M denotes the number of such solutions. Clearly, $M \geq N$.

Based on the discussion and two observations above, we therefore can express the outlyingness function $O(x, \mathcal{X}^n)$ as follows

$$O(x, \mathcal{X}^n) = \max\left\{\frac{u_1^\tau x - \mathrm{Med}(u_1^\tau \mathcal{X}^n)}{\mathrm{MAD}(u_1^\tau \mathcal{X}^n)}, \frac{u_2^\tau x - \mathrm{Med}(u_2^\tau \mathcal{X}^n)}{\mathrm{MAD}(u_2^\tau \mathcal{X}^n)}, \cdots, \frac{u_M^\tau x - \mathrm{Med}(u_M^\tau \mathcal{X}^n)}{\mathrm{MAD}(u_M^\tau \mathcal{X}^n)}\right\},$$

where $\{u_i\}_{i=1}^M$ are some p-dimensional vectors depending only on the data cloud $\mathcal{X}^n$. For the sake of convenience, we write $g_i(x) = a_i^\tau x - b_i$ $(i = 1, 2, \cdots, M)$ hereafter, where $a_i = \frac{1}{\mathrm{MAD}(u_i^\tau \mathcal{X}^n)} u_i$ and $b_i = \frac{\mathrm{Med}(u_i^\tau \mathcal{X}^n)}{\mathrm{MAD}(u_i^\tau \mathcal{X}^n)}$.

Obviously, $O(x, \mathcal{X}^n) = \max_{1 \le i \le M}\{g_i(x)\}$ is in fact a piece-wise linear convex function with respect to x for the given data cloud $\mathcal{X}^n$. Therefore, its minimizers can be found by using common linear programming methods for solving the problem

$$s = \min_{\omega} \; t$$

subject to

$$t \ge g_i(x), \, i = 1, 2, \cdots, M,$$

where $\omega = (t, x^\tau)^\tau$. This kind of problem can be solved by some common solvers such as *linprog.m* in Matlab. Let $\omega_0 = (t_0, x_0^\tau)^\tau$ be a final solution to this problem. Then, it is easy to show that x_0 is one of the deepest points with depth value $PD(x_0, \mathcal{X}^n) = 1/(1 + t_0) = \alpha^*$.

Given the nature of the piecewise linear convex function $\max_{1 \le i \le M} g_i(x)$, there is either a single minimizer or a convex polyhedral set of minimizers. Then there naturally comes a question, namely, after obtaining the value α^*, how to get all of these vertices? Note that the projection median is a specific case of the projection depth contour. Therefore, let's focus now on the computation of projection depth contours.

By the definition, for any given $0 < \alpha \le \alpha^*$, $PC(\alpha, \mathcal{X}^n)$ is simply the boundary of

$$\begin{aligned} PR(\alpha, \mathcal{X}^n) &= \{x \in R^p : PD(x, \mathcal{X}^n) \ge \alpha\} \\ &= \{x \in R^p : O(x, \mathcal{X}^n) \le \beta\} \\ &= \{x \in R^p : g_i(x) \le \beta, \, i = 1, 2, \cdots, M\}. \end{aligned}$$

Typically, the regions constrained by linear inequalities such as

$$g_i(x) \le \beta, \, i = 1, 2, \cdots, M \tag{7}$$

are polytopes. Then all the vertices and facets of such a polytope can be found by means of the dual relationship between vertex and facet enumeration (Bremner *et al.*, 1998) and the program *qhull* (Barber *et al.*, 1996); see also Paindaveine and Šiman (2012a) for a more detailed discussion. In Matlab, similar tasks can be fulfilled by the function *con2vert.m*, which has been developed by Michael Kleder, and now can be downloaded from Matlab Central File Exchange.

However, in many practical applications, the number M of the direction vectors may be very large. When M is too large, it is difficult to obtain the boundary of the region formed by (7) by using some of the aforementioned procedures such as *con2vert.m*. Therefore, it is important to eliminate some redundant constraints before computing $PC(\alpha, \mathcal{X}^n)$ for a too large M.

Note that, for any given α $(0 < \alpha \le \alpha^*)$, the number of the non-redundant constraints in (7) is much smaller compared to M, which implies that numerous inequalities in (7) could be eliminated during the computation of the α-contour. In fact, it is not difficult to show that

$$PR(\alpha, \mathcal{X}^n) = \mathcal{C}_1 \cap \mathcal{C}_2 \cap \cdots \cap \mathcal{C}_s,$$

where $\mathcal{C}_k = \{x : g_j(x) \le \beta, \, j = i_k, i_k + 1, \cdots, i_{k+1}\}$ with $1 \le k \le s - 1$, and $1 = i_1 < i_2 < \cdots < i_s = M$ with $\min_{1 \le l \le s-1}\{i_{l+1} - i_l\} \gg p$. Then, when M is large, a procedure for exactly computing $PC(\alpha, \mathcal{X}^n)$ is: (1) to find the non-redundant constraints in each $\mathcal{C}_i$ at first, then (2) to use all of these non-redundant constraints together to compute $PC(\alpha, \mathcal{X}^n)$.

With the vertices at hand, some common graphical softwares can be utilized to visualize these contours very easily in spaces of $p = 2$ or 3. It is noteworthy that, although all the methods discussed above can possibly be implemented to spaces with $p \geq 2$ theoretically, a feasible exact algorithm for computing the projection depth exists now only for bivariate data (Zuo and Lai, 2011). Therefore, we can only provide some exact results about the bivariate projection depth contours and projection median in the current paper. All the direction vectors are found by using a Matlab implementation based on the algorithm of Zuo and Lai (2011). *The corresponding codes can be obtained from the authors through email* (zuo@msu.edu or csuliuxh912@gmail.com).

It is worth mentioning that, although M aforementioned seems to be very large even when $p = 2$ at first sight, there is a possibility of filtering a great number of direction vectors in the computation; see Figure 1 for an illustration. The reason lies in the fact that, when u passes from one fragment S_k to some of its adjacent fragments, $Q(u, x, \mathcal{X}^n)$ may still equal $c_k^\tau u / d_k^\tau u$, with both c_k and d_k unchanged. Therefore, we can merge some of such fragments into a large one to filter some direction vectors in practice. For the bivariate case, this tactic has already been used in the algorithm of Zuo and Lai (2011), and therefore in the corresponding Matlab implementation of the current paper.

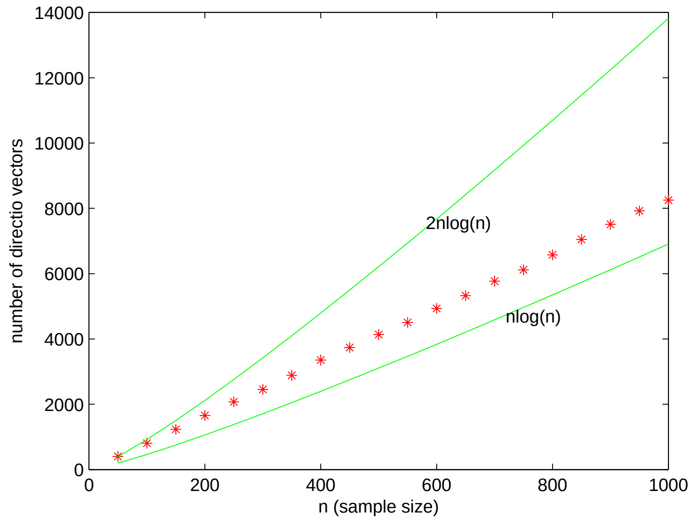


Figure 1: The average number (the star points) of the final direction vectors after being filtered. For the sake of comparison, we also provide two lines that correspond to $n \log n$ and $2n \log n$, respectively. Here the data are generated from the bivariate standard normal distribution. The sample sizes are taken to be $n = 50, 100, \cdots, 1000$. For each sample size n, we compute the average number of the direction vectors based on 20 repeated simulations. The figure reveals that the number of the remaining direction vectors can be quite small relative to M, which is greater than $n(n-1)$ when $p = 2$ according to the discussion above.

5 Examples

In order to gain more insight into the sample version of projection depth contours and projection median, we provide some data examples in this section. Both simulated and real data are used here.

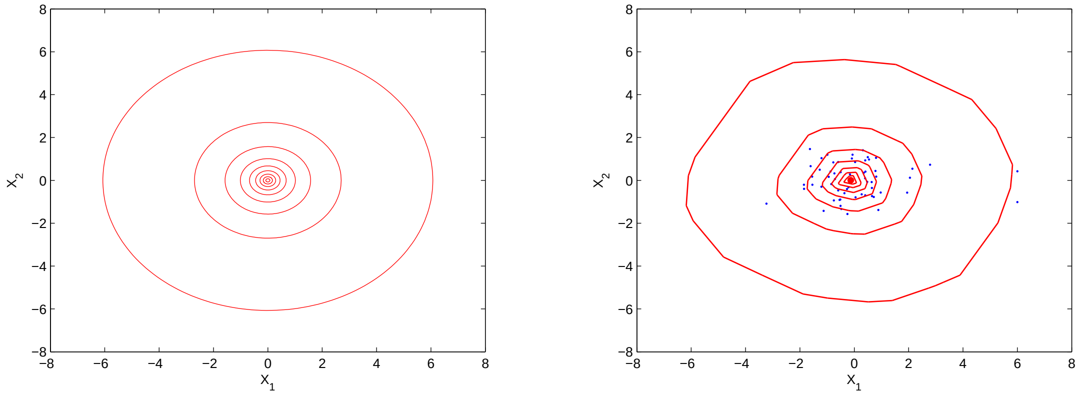
5.1 Simulation results

To illustrate the robustness and the shape of the bivariate projection depth contours and median, we present two examples as follows. The data are mainly generated from the normal distribution, but contain a few outliers.

Example 1. We generate 60 points $X = (X_1, X_2)^\tau$ from the normal distribution $N(0, \mathbb{I}_2)$, where $\mathbb{I}_2$ denotes the 2×2 identity matrix, and then modify these points randomly by replacing their first components by 6 with probability 0.05.

Example 2. We generate 400 points $X = (X_1, X_2)^\tau$ from the normal distribution $N(0, \Sigma_0)$, and then modify these points randomly by replacing their first components by 6 with probability 0.10, where

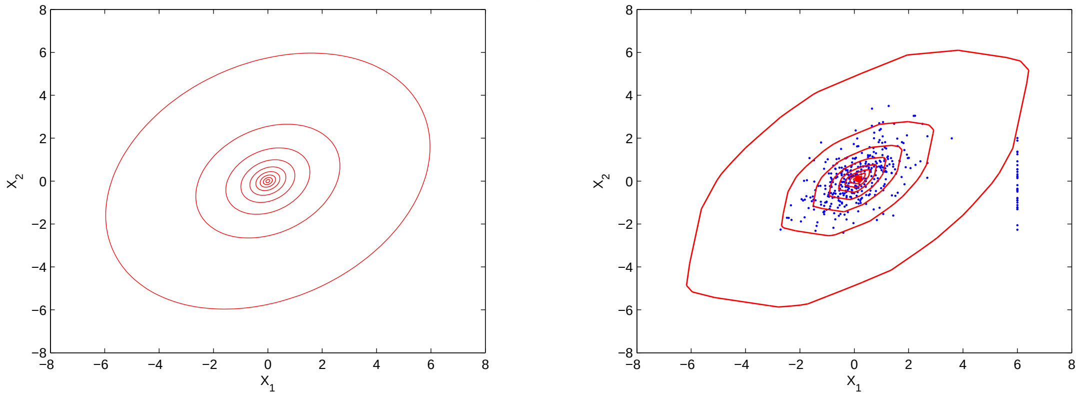
$$\Sigma_0 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$



(a) Population versions $PC(\alpha, X)$.

(b) Sample versions $PC(\alpha, \mathcal{X}^n)$ with $n = 60$. The small points denote the observations, and the big point in the interior of all the contours denotes the projection median.

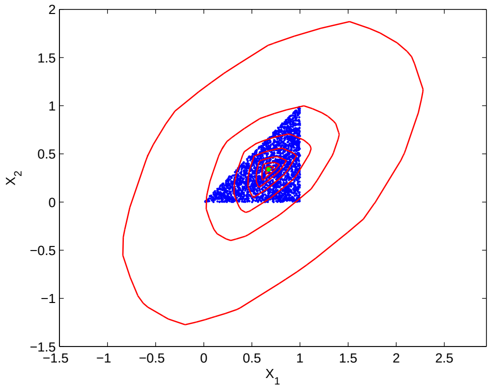
Figure 2: Shown are the population and sample versions of contours for Example 1, where $\alpha = 0.1, 0.2, \cdots, 0.9$ in Figure 2(a), and $\alpha = 0.1, 0.2, \cdots, 0.7$ in Figure 2(b) from the periphery inwards, respectively.



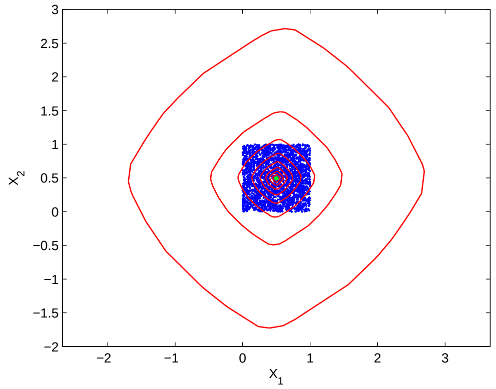
(a) Population versions $PC(\alpha, X)$.

(b) Sample versions $PC(\alpha, \mathcal{X}^n)$ with $n = 400$. The small points denote the observations, and the big point in the interior of all the contours denotes the projection median.

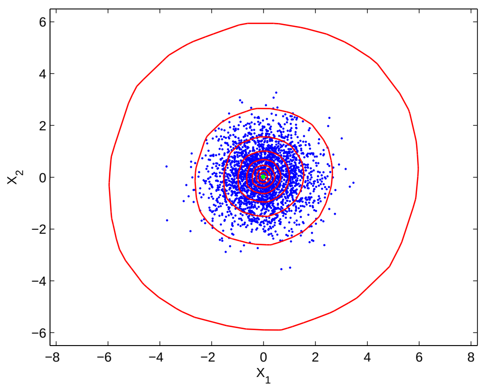
Figure 3: Shown are the population and sample versions of contours for Example 2, where $\alpha = 0.1, 0.2, \cdots, 0.9$ in Figure 3(a), and $\alpha = 0.1, 0.2, \cdots, 0.8$ in Figure 3(b) from the periphery inwards, respectively.



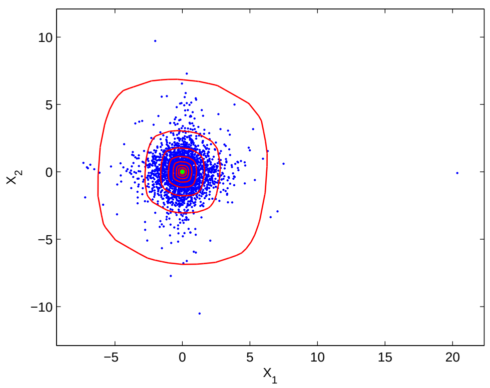
(a) $PC(\alpha, \mathcal{X}^n)$ of the uniform distribution over the triangle formed by vertices: (0, 0), (0, 1) and (1, 1).



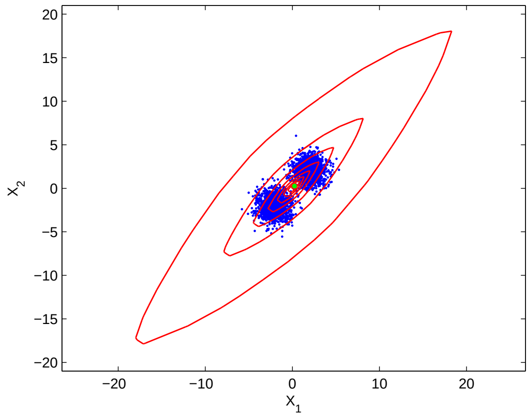
(b) $PC(\alpha, \mathcal{X}^n)$ of the uniform distribution over $(0, 1) \times (0, 1)$.



(c) $PC(\alpha, \mathcal{X}^n)$ of the bivariate standard normal distribution $N(0, \mathbb{I}_2)$.



(d) $PC(\alpha, \mathcal{X}^n)$ of the bivariate t-distribution with 4 degrees of freedom. The components X_1 and X_2 are independently distributed.



(e) $PC(\alpha, \mathcal{X}^n)$ of the random vector $\delta X + (1-\delta)Y$, where $X \sim N(\mu_1, \mathbb{I}_2)$, $Y \sim N(\mu_2, \mathbb{I}_2)$, $\mu_1 = (-2, -2)$, $\mu_2 = (2, 2)$, and δ satisfies $P(\delta = 0) = P(\delta = 1) = 0.5$. X, Y and δ are all independently distributed.

Figure 4: Shown are the sample contours $PC(\alpha, \mathcal{X}^n)$ of different distributions with $n = 2500$, where $\alpha = 0.1, 0.2, \cdots, 0.7$ in Figure 4(a), and $\alpha = 0.1, 0.2, \cdots, 0.9$ in Figure 4(b), 4(c), 4(d), and 4(e) from the periphery inwards, respectively. The small points denote the observations, and the big point in the interior of all the contours denotes the projection median.

For the sake of comparison, the population versions of $PC(\alpha,\, X)$ corresponding to these two examples are also provided here, and plotted according to the formula

$$(x_1,\, x_2) \times \Sigma^{-1} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{C_N^2(1-\alpha)^2}{\alpha^2},$$

which was developed in Zuo (2003). Here $C_N = \Phi^{-1}(\frac{3}{4}) \approx 0.6744898$, Σ denotes the covariance matrix of normally distributed X, namely, $\mathbb{I}_2$ in Example 1 and Σ_0 in Example 2. The population versions $PC(\alpha,\, X)$ of Example 1 and 2 are given in Figure 2(a) and 3(a), respectively, while the sample versions $PC(\alpha,\, \mathcal{X}^n)$ are plotted in Figure 2(b) and 3(b), respectively.

The comparison of sample and population contours indicates that the sample contours are really similar to their population counterparts, resistant to the outliers (Zuo, 2004) and polyhedral like the halfspace depth contours (Paindaveine and Šiman, 2012a).

Furthermore, to gain more information about the shape of the projection depth contours, we also provide some other examples in the following. The sample size we used is 2500. Here Figure 4(a) reports the projection depth contours $PC(\alpha,\, \mathcal{X}^n)$ corresponding to the uniform distribution over the triangle with its vertices being (0, 0), (0, 1) and (1, 1) (D1). Figure 4(b) corresponds to the uniform distribution over the region $[0,\, 1] \times [0,\, 1]$ (D2). Figure 4(c) gives the contours of the normal distribution $N(0,\, \mathbb{I}_2)$ (D3). Figure 4(d) corresponds to the bivariate t-distribution with 4 degrees of freedom and independently distributed components (D4). Figure 4(e) provides the contours of $\delta X + (1-\delta)Y$ (D5), where $X \sim N(\mu_1,\, \mathbb{I}_2)$, $Y \sim N(\mu_2,\, \mathbb{I}_2)$, $\mu_1 = (-2,\, -2)$, $\mu_2 = (2,\, 2)$, and δ is a discrete random variable with $P(\delta = 0) = P(\delta = 1) = 0.5$. Here X, Y and δ are all independently distributed. These figures show that the projection contours are also polyhedral and convex.

Note that somebody may be interested in the execution time of the proposed algorithm in practice. Therefore, this section also presents some empirical results about the execution time based on some simulated bivariate data. The data are generated from five distributions D1-D5 investigated in Figure 4. For any sample size $n \in \{50,\, 100,\, 200,\, 300,\, 400,\, 500,\, 800\}$, we run the computation ten times for each distribution. Table 1 reports the average execution times (in seconds), namely t_C, of computing the projection depth median and contours $PC(\alpha, \mathcal{X}^n)$ for $\alpha = 0.1,\, 0.2,\, \cdots,\, \lfloor 10 \times \alpha^*(\mathcal{X}^n) \rfloor/10$, where $\alpha^*(\mathcal{X}^n)$ denotes the depth value corresponding to the projection depth median of $\mathcal{X}^n$. Furthermore, in order to illustrate the benefits of the x-free property, we also list the average execution times (in seconds), write t_V, of calculating all the final direction vectors for each combination of n and D $\in$ {D1, D2, $\cdots$, D5} in the same table. Table 1 shows that a great proportion ($\geq 99\%$) of the time is spent on finding the final direction vectors on average. This confirms that there would be a considerable improvement in time when several contours have to be calculated for the same data set.

Table 1: Average execution times (in seconds) of the proposed algorithm for the data from five different distributions D1-D5.

D \ n		50	100	200	300	400	500	800
D1	t_V	0.2768	1.7661	13.4223	44.1912	107.7082	208.9788	851.6945
	t_C	0.2841	1.7738	13.4336	44.2053	107.7258	208.9983	851.7224
D2	t_V	0.2536	1.8236	13.4294	44.1836	104.5274	210.2240	834.7482
	t_C	0.2608	1.8317	13.4410	44.1989	104.5461	210.2472	834.7816
D3	t_V	0.2791	1.9218	14.4595	47.0722	113.2368	215.3458	869.2130
	t_C	0.2865	1.9303	14.4714	47.0879	113.2554	215.3688	869.2457
D4	t_V	0.2657	1.9464	15.0405	48.4866	117.1220	224.3819	911.1383
	t_C	0.2727	1.9544	15.0527	48.5031	117.1418	224.4050	911.1711
D5	t_V	0.2403	1.6682	12.4515	39.7955	94.3297	188.6513	756.2787
	t_C	0.2470	1.6757	12.4629	39.8105	94.3479	188.6730	756.3091

5.2 Real data example

Here a real data example is presented to illustrate the performance of projection depth contours.

Table 2 here is taken from Table 7 of Rousseeuw and Leroy (1987) (p.57). Total 28 animals' brain weight (in grams) and body weight (in kilograms) are presented in this table. Before the analysis, logarithmic transformation was taken for the sake of convenience. According to the results of Rousseeuw and Leroy (1987), there are five cases considered as outlying, i.e. diplodocus, human, triceratops, rhesus monkey and brachiosaurus. Among them, the most severe cases are diplodocus, triceratops and brachiosaurus. In fact, these three cases are referred to as dinosaurs because they possess a small brain as compared with a heavy body (see Table 2) and their highly negative residuals can lead to a low slope for the least squares fit. For the remaining two cases, although their actual brain weights are higher than those predicted by the linear model, they are not worse than the three previous cases since they do not obey the same trend as that one followed by the majority of the data.

We plot the projection depth contours in Figure 5, where the big point in the interior of all the contours denotes the projection median with depth value 0.73257, five labeled points denote the outliers mentioned above with the points 1-3 corresponding to the case of diplodocus, triceratops and brachiosaurus and 4-5 corresponding to those of human and rhesus monkey. 8 contours are plotted there. From Figure 5, we can see that all of these three dinosaurs lie outside the contour for $\alpha = 0.1$, while points 4-5 lie between the contours for $\alpha = 0.1$ and 0.15. These results are consistent with those of Rousseeuw and Leroy (1987), implying that projection depth contours can capture the structures of the objective data and identify outliers.

Furthermore, it is worth mentioning that the shape of these plotted contours is not affected by a few atypical points at the margin/border of the data cloud, namely, both the inner and outer depth contours are roughly elliptical, unlike those of halfspace depth contours (see Figure 6) (Ruts and Rousseeuw, 1996). This is the most outstanding difference between projection and halfspace depth contours, confirming the higher robustness of projection depth and its contours (see Zuo (2004)).

Table 2: Body and Brain Weight for 28 Animals (Rousseeuw and Leroy, 1987).

Index i	Species	Body Weight X_i	Brain Weight Y_i
1	Mountain beaver	1.350	8.100
2	Cow	465.000	423.000
3	Gray wolf	36.330	119.500
4	Goat	27.660	115.000
5	Guinea pig	1.040	5.500
6	Diplodocus	11700.000	50.000
7	Asian elephant	2547.000	4603.000
8	Donkey	187.100	419.000
9	Horse	521.000	655.000
10	Potar monkey	10.000	115.000
11	Cat	3.300	25.600
12	Giraffe	529.000	680.000
13	Gorilla	207.000	406.000
14	Human	62.000	1320.000
15	African elephant	6654.000	5712.000
16	Triceratops	9400.000	70.000
17	Rhesus monkey	6.800	179.000
18	Kangaroo	35.000	56.000
19	Hamster	0.120	1.000
20	Mouse	0.023	0.400
21	Rabbit	2.500	12.100
22	Sheep	55.500	175.000
23	Jaguar	100.000	157.000
24	Chimpanzee	52.160	440.000
25	Brachiosaurus	87000.000	154.500
26	Rat	0.280	1.900
27	Mole	0.122	3.000
28	Pig	192.000	180.000

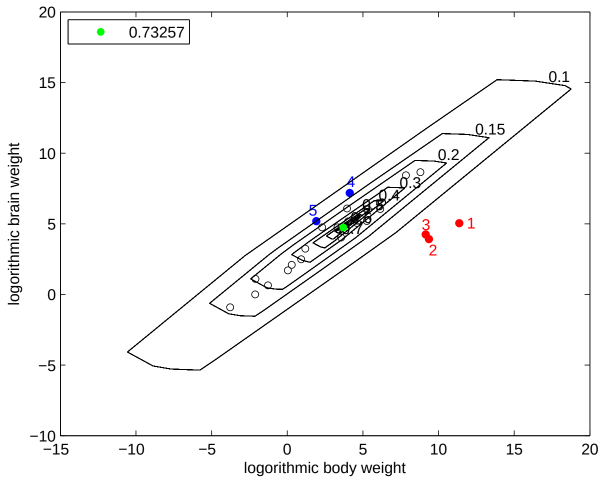


Figure 5: Projection depth contours with $\alpha = 0.1, 0.15, 0.2, 0.3, \cdots, 0.7$ from the periphery inwards. The big point in the interior of all the contours is the computed projection median with depth value 0.73257. The other points denote the observations, where the five labeled points are those considered as outlying, and points 1-3 are the so-called dinosaurs mentioned in Rousseeuw and Leroy (1987).

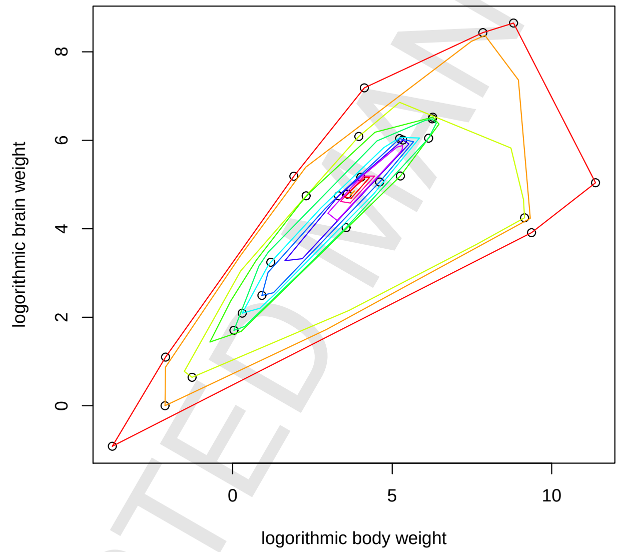


Figure 6: Halfspace depth contours; see Ruts and Rousseeuw (1996) for details.

Acknowledgments

The authors greatly appreciate the insightful and constructive remarks of two anonymous referees, which led to distinct improvements in this paper. They are also very grateful to Professor James Stapleton, an associate editor and the chief editor Stanley Azen for their great assistance. This work was done during Xiaohui Liu's visit to the Department of Statistics and Probability at Michigan State University as a joint PhD student. He thanks his co-advisor Professor Yijun Zuo for stimulating discussions and insightful comments and suggestions and the department for providing excellent studying and working condition. This article was partially supported by the National Natural Science Foundation of China (No. 61070236).

Appendix Proofs of main results

The order of N**.** Note that there is a bijection between the fragments S_k and the cones $\mathcal{C}_k = \{z \in R^p : \mathbb{A}_k^\tau z \leq 0\}$, where $\mathbb{A}_k = (\mathbb{A}_{k1}^\tau, \mathbb{A}_{k2}^\tau)^\tau$. $\mathbb{A}_{k1}$ and $\mathbb{A}_{k2}$ will be specified in the end of this Appendix. Therefore, N is the same as the number of $\mathcal{C}_k$.

Let $\mathcal{D}_{k'} = \{z \in R^p : \mathbb{A}_{k1}^\tau z \leq 0\}$. Obviously, (1) $\mathcal{C}_k \subset \mathcal{D}_{k'}$, (2) all of such $\mathcal{D}_{k'}$ together can span the whole space R^p, (3) for any $u (u \neq 0) \in \mathcal{D}_{k'}$, the permutation $(i_{k,1}, i_{k,2}, \cdots, i_{k,n})$ is fixed. Note that $\mathcal{D}_{k'}$ is formed by some hyperplanes, such as $(X_{i_{k,2}} - X_{i_{k,1}})^\tau z = 0$, that are intersect at the origin. In the literature, it has been shown by Winder (1966) (p. 816) that the number of such cones equals $2\sum_{i=1}^{p-1}\binom{k-1}{i}$ for k hyperplanes, which is of order $O(k^{p-1})$. Based on $\mathcal{X}^n$, we can obtain at most $n(n-1)/2$ hyperplanes. Therefore, the upper bound for the number of $\mathcal{D}_{k'}$'s is about $O(n^{2(p-1)})$. That is, we can divide R^p into up to $O(n^{2(p-1)})$ cones, which are determined by some permutations such as $(i_{k,1}, i_{k,2}, \cdots, i_{k,n})$. Similarly, for each given $\mathcal{D}_{k'}$, we can further divide it into a finite number, with an upper bound $O(n^{2(p-1)})$, of $\mathcal{C}_k$'s based on some permutations such as $(j_{k,1}, j_{k,2}, \cdots, j_{k,n})$. Therefore, the order of N is about $O(n^{4(p-1)})$. See also Bazovkin and Mosler (2012) for a similar discussion.

Proof of Proposition 1. Here, without loss of generality, we prove only the odd n case. Note that, for any $u \in \mathcal{Q}_k$, we have

$$u^T X_{i_{k,1}} \leq u^T X_{i_{k,2}} \leq \cdots \leq u^T X_{i_{k,m}} \leq \cdots \leq u^T X_{i_{k,n}}$$

according to the definition of $\mathcal{Q}_k$. This implies that

$$|u^T(X_i - X_{i_{k,m}})| = \begin{cases} -u^T(X_i - X_{i_{k,m}}), & \text{if } i \in \{i_{k,1}, i_{k,2}, \cdots, i_{k,m-1}\}, \\ u^T(X_i - X_{i_{k,m}}), & \text{if } i \in \{i_{k,m}, i_{k,m+1}, \cdots, i_{k,n}\}. \end{cases}$$

That is, we can remove the absolute value signs of Y_{j_l} based on the order information existing in the permutation $(i_{k,1}, i_{k,2}, \cdots, i_{k,n})$. Therefore, for any $u \in \mathcal{Q}_k$, (3) can be further simplified to

$$Q(u, x, \mathcal{X}^n) = \frac{c_k^T u}{d_k^T u},$$

where $c_k = x - X_{i_k,m}$ and $d_k = s_k(j_{k,m}) \cdot (X_{j_k,m} - X_{i_k,m})$ $(1 \leq k \leq N)$, with

$$s_k(i) = \begin{cases} -1, & \text{if } i \in \{i_{k,1}, i_{k,2}, \cdots, i_{k,m-1}\}, \\ 1, & \text{if } i \in \{i_{k,m}, i_{k,m+1}, \cdots, i_{k,n}\}. \end{cases}$$

Next, note that, for any positive λ and $z = \lambda u$, it holds that $c_k^\tau z / d_k^\tau z = c_k^\tau u / d_k^\tau u$ and $\eta^\tau z \geq 0$ if $\eta^\tau u \geq 0$. Then, (3) and constraint conditions $\mathcal{Q}_k$ lead to

$$O_k(x, \mathcal{X}^n) = \sup_z \frac{c_k^\tau z}{d_k^\tau z} \tag{8}$$

subject to

$$\mathbb{A}_k z \geq 0,$$

$1 \leq k \leq N$, with $\mathbb{A}_k = \binom{\mathbb{A}_{k1}}{\mathbb{A}_{k2}}$, where

$$\mathbb{A}_{k1} = (X_{i_{k,2}} - X_{i_{k,1}}, X_{i_{k,3}} - X_{i_{k,2}}, \cdots, X_{i_{k,n}} - X_{i_{k,n-1}})^\tau$$

and

$$\mathbb{A}_{k2} = \begin{pmatrix} s_k(j_{k,2}) \cdot (X_{j_{k,2}}^\tau - X_{i_{k,m}}^\tau) - s_k(j_{k,1}) \cdot (X_{j_{k,1}}^\tau - X_{i_{k,m}}^\tau) \\ s_k(j_{k,3}) \cdot (X_{j_{k,3}}^\tau - X_{i_{k,m}}^\tau) - s_k(j_{k,2}) \cdot (X_{j_{k,2}}^\tau - X_{i_{k,m}}^\tau) \\ \vdots \\ s_k(j_{k,n}) \cdot (X_{j_{k,n}}^\tau - X_{i_{k,m}}^\tau) - s_k(j_{k,n-1}) \cdot (X_{j_{k,n-1}}^\tau - X_{i_{k,m}}^\tau) \end{pmatrix}.$$

This completes the proof of Proposition 1.

References

Barber, C.B., Dobkin, D.P., Huhdanpaa, H., 1996. The quickhull algorithm for convex hulls. ACM Transactions Math. Software 22, 469-483.

Bazovkin, P., Mosler, K., 2012. An exact algorithm for weighted-mean trimmed regions in any dimension. J. Statist. Software 47.

Bremner, D., Fukuda, K., Marzetta, A., 1998. Primal-dual methods for vertex and facet enumeration. Discrete Comput. Geometry 20, 333-357.

Cascos, I., 2007. The expected convex hull trimmed regions of a sample. Comput. Statist. 22, 557-569.

Donoho, D.L., 1982. *Breakdown properties of multivariate location estimators.* Ph.D. Qualifying Paper. Dept. Statistics, Harvard University.

Donoho, D.L., Gasko, M., 1992. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. Ann. Statist. 20, 1808-1827.

Dyckerhoff, R., 2000. Computing zonoid trimmed regions of bivariate data sets. In J. Bethlehem and P. van der Heijden, eds., COMPSTAT 2000. Proceedings in Computational Statistics, 295-300. Physica, Heidelberg.

Edelsbrunner, H., 1987. *Algorithms in Combinatorial Geometry.* Springer, Heidelberg.

Liu, R. Y., 1990. On a notion of data depth based on random simplices. Ann. Statist. 18, 191-219.

Liu, R. Y., 1992. Data depth and multivariate rank tests. In L1-Statistical Analysis and Related Methods (Y. Dodge, ed.), 279-294. North-Holland, Amsterdam.

Maronna, R.A., Yohai, V.J., 1995. The behavior of the Stahel-Donoho robust multivariate estimator. J. Amer. Statist. Assoc. 90, 330-341.

Paindaveine, D., Šiman, M., 2011. On directional multiple-output quantile regression. J. Multivariate Anal. 102, 193-392.

Paindaveine, D., Šiman, M., 2012a. Computing multiple-output regression quantile regions. Comput. Statist. Data Anal. 56, 840-853.

Paindaveine, D., Šiman, M., 2012b. Computing multiple-output regression quantile regions from projection quantiles. Comput. Statist. 27, 29-49.

Rousseeuw, P.J., Leroy. A., 1987. *Robust regression and outlier detection.* Wiley New York.

Rousseeuw, P.J., Hubert, M., 1999. Regression depth (with discussion). J. Amer. Statist. Assoc. 94, 388-433.

Ruts, I., Rousseeuw, P.J., 1996. Computing depth contours of bivariate point clouds. Comput. Statist. Data Anal. 23, 153-168.

Šiman, M., 2011. On exact computation of some statistics based on projection pursuit in a general regression context. Comm. Statist. Simulat. Comput. 40, 948-956.

Stahel, W.A., 1981. Breakdown of covariance estimators. Research Report 31. Fachgruppe für Statistik. ETH, Zürich.

Swarup, K., 1965. Linear fractional functionals programming. Operations Res. 13, 1029-1036.

Tukey, J.W., 1975. Mathematics and the picturing of data. In Proceedings of the International Congress of Mathematicians, 523-531. Cana. Math. Congress, Montreal.

Tyler, D.E., 1994. Finite sample breakdown points of projection based multivariate location and scatter statistics. Ann. Statist. 22, 1024-1044.

Winder, R., 1966. Partitions of N-Space by Hyperplanes. SIAM J. Applied Math. 14, 811-818.

Zuo, Y.J., 2003. Projection based depth functions and associated medians. Ann. Statist. 31, 1460-1490.

Zuo, Y.J., 2004. Robustness of weighted L_p-depth and L_p-median. Allgemeines Statistisches Archiv 88, 1-20.

Zuo, Y.J., Cui, H.J., He, X.M., 2004. On the Stahel-Donoho estimators and depth-weighted means for multivariate data. Ann. Statist. 32, 189-218.

Zuo, Y.J., Lai, S.Y., 2011. Exact computation of bivariate projection depth and the Stahel-Donoho estimator. Comput. Statist. Data Anal. 55, 1173-1179.

Zuo, Y.J., Serfling, R., 2000. General notions of statistical depth function. Ann. Statist. 28, 461-482.