Statistica Sinica **8**(1998), 827-839

CONSTRUCTING THE BIVARIATE TUKEY MEDIAN

Peter J. Rousseeuw and Ida Ruts

University of Antwerp

Abstract: The halfplane location depth of a point $\boldsymbol{\theta} \in I\!R^2$ relative to a bivariate data set $X = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ is the minimal number of observations in any closed halfplane that contains $\boldsymbol{\theta}$ (Tukey (1975)). The halfplane median or Tukey median is the $\boldsymbol{\theta}$ with maximal depth k^* (Donoho and Gasko (1992)). If this $\boldsymbol{\theta}$ is not unique, the Tukey median is defined as the center of gravity of the set of points with depth k^*. In this paper we construct two algorithms for computing the Tukey median. The first one is relatively straightforward but quite slow, whereas the second (called HALFMED) is much faster. A small simulation study is performed, and some examples are given.

Key words and phrases: Algorithm, halfplane depth, robustness, Tukey depth.

1. Introduction

Consider a bivariate data set $X = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$. The *halfplane location depth* of an arbitrary point $\boldsymbol{\theta} \in I\!R^2$ relative to X is defined as:

$$\mathrm{ldepth}(\boldsymbol{\theta}, X) = \min_H \#\{i;\, \mathbf{x_i} \in H\},$$

where H ranges over all closed halfplanes of which the boundary line passes through $\boldsymbol{\theta}$ (Tukey (1975)). Obviously, a point $\boldsymbol{\theta}$ outside the convex hull of X will have depth zero. A fast algorithm for the computation of the halfplane depth of $\boldsymbol{\theta}$ was constructed by Rousseeuw and Ruts (1996).

The *depth region* of depth k is defined as the set D_k of points $\boldsymbol{\theta}$ with $\mathrm{ldepth}(\boldsymbol{\theta}, X) \geq k$. Equivalently, D_k is the intersection of all closed halfplanes that contain (at least) $n - k + 1$ observations, hence D_k is a bounded convex polytope. (Note that $D_1 \supset D_2 \supset D_3 \supset \cdots$) The boundary of D_k is a convex polygon, which is called the *contour of depth k*. Therefore, each vertex of a depth contour is the intersection point of two lines, each passing through two observations.

The Tukey median is the center of gravity of the deepest depth region. We will construct two algorithms for the computation of the Tukey median. The first one is quite staightforward and has time complexity $O(n^5 \log n)$. The second one is called HALFMED and attains $O(n^2 \log^2 n)$. We will first explain the slower algorithm to illustrate some geometric aspects of the Tukey median (Section 3), but the focus of the paper will be on the algorithm HALFMED (Section 4).

By construction, the halfplane depth is *affine invariant*: if we consider an affine transformation $g(\mathbf{z}) = A\mathbf{z} + \mathbf{b}$ with any $\mathbf{b} \in \mathbb{R}^2$ and any nonsingular $A \in \mathbb{R}^{2\times 2}$, then $\text{ldepth}(g(\boldsymbol{\theta}), g(X)) = \text{ldepth}(\boldsymbol{\theta}, X)$. Consequently, the Tukey median is *affine equivariant*:

$$\text{halfmed}(g(X)) = g(\text{halfmed}(X)).$$

Other important affine equivariant bivariate medians are the Oja (1983) median and Liu's (1990) simplicial median. For surveys of bivariate medians see Small (1990), Niinimaa (1995) and Niinimaa and Oja (1997). The minimal computational complexity of most bivariate medians has not yet been established. A new (but not affine equivariant) median can be found in Grübel (1996). Other notions of location depth can be found in Liu (1992) and Dyckerhoff, Koshevoy and Mosler (1996). Recently, Rousseeuw and Hubert (1996) have introduced depth functions for regression which are analogous to halfplane depth and simplicial depth.

2. The Tukey Median

The function $\text{ldepth}(\boldsymbol{\theta}, X)$ takes on integer values between 0 and n, and thus attains a maximum value

$$k^*(X) = \max_{\boldsymbol{\theta}\in\mathbb{R}^2} \text{ldepth}(\boldsymbol{\theta}, X).$$

This maximum depends on the shape of X. For instance, $k^*(X)$ tends to be higher when X has certain symmetries, and $k^*(X) = n$ if all points of X coincide.

We say that a bivariate data set X is in *regular position* if no more than two observations lie on a line. Under this assumption, Donoho and Gasko (1992), page 1806 show that

$$\left\lceil \frac{n}{3} \right\rceil \le k^*(X) \le \left\lceil \frac{n}{2} \right\rceil, \tag{1}$$

where the ceiling $\lceil \lambda \rceil$ is the smallest integer $\ge \lambda$. Therefore, $D_k \neq \varnothing$ whenever $k \le \lceil \frac{n}{3} \rceil$. Let us now prove that the upper bound $\lceil \frac{n}{2} \rceil$ in (1) can be lowered to $\lfloor \frac{n}{2} \rfloor$ for bivariate X. This is only a minor modification of (1), but our objective is to present a geometric proof through some actual constructions, which will then be used in Section 4 as basic tools for the algorithm HALFMED.

Proposition 1. *For any bivariate data set X in regular position*

$$k^*(X) \le \left\lfloor \frac{n}{2} \right\rfloor.$$

Proof. For n even we have $\lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil$ so there is nothing to prove. From here on we assume that n is odd. Suppose that there exists a depth region $D_k \neq \varnothing$ with $k = \lfloor \frac{n}{2} \rfloor + 1 = \lceil \frac{n}{2} \rceil = \frac{n+1}{2}$.

The depth region D_k is the intersection of all closed halfplanes which contain at least $n-k+1$ data points. Since $D_k \neq \varnothing$ and the data set is in regular position, there exists a directed line L containing two observations, and with $n-k-1$ observations strictly to the left of L. This is illustrated in Figure 1. Then D_k is a subset of the closed halfplane H to the left of L. The number of points strictly to the right of L is $k-1 = n-k$ (since $n = 2k-1$). This means that the closed halfplane to the right of L contains $n-k+2$ observations, hence D_k also must be a subset of this halfplane. This implies that $D_k \subset L$.

We now prove that no point of L has depth k. Let $\mathbf{x_1}$ be the first and $\mathbf{x_2}$ the second observation lying on the directed line L. We construct the directed line L_1 by slightly rotating L counterclockwise around $\mathbf{x_2}$ in such a way that it does not touch any of the other observations $\mathbf{x_3}, \ldots, \mathbf{x_n}$. Therefore, the observations strictly to the left of L_1 are $\mathbf{x_1}$ and the observations strictly to the left of L. Hence the closed halfplane to the right of L_1 (denoted by H_1) contains $k-1+1 = k = n-k+1$ observations.

Analogously, we construct the directed line L_2 by slightly rotating L clockwise around $\mathbf{x_1}$ in such a way that it does not touch any of the observations $\mathbf{x_3}, \ldots, \mathbf{x_n}$. Therefore, the closed halfplane to the right of L_2 (denoted by H_2) also contains $n-k+1$ observations. By definition of D_k it follows that $D_k \subset (H_1 \cap H_2)$. Therefore

$$D_k \subset (L \cap H_1 \cap H_2) = \varnothing$$

which ends the proof.

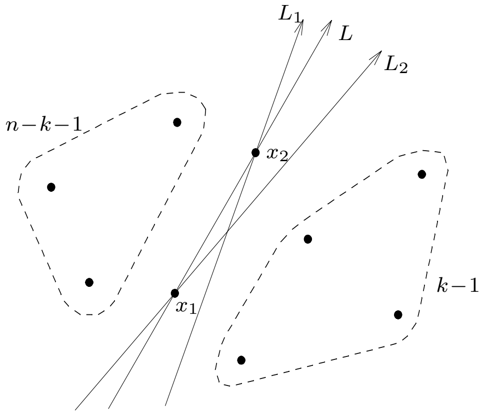

Figure 1. Illustration of the proof of Proposition 1 for $n = 9$. In Section 4, the directed line L is called a special k-divider.

The *Tukey median* or *halfplane median* is defined as the $\boldsymbol{\theta}$ with depth k^* (Donoho and Gasko (1992)). If D_{k^*} is not a singleton, they define the halfplane median as the center of gravity of D_{k^*}.

An important motivating property of the bivariate Tukey median is its *robustness.* Donoho and Gasko (1992), page 1811 prove that the *breakdown value* $\varepsilon^*(\text{halfmed}, X) \geq \frac{1}{3}$ for any sample X in regular position. This means that when replacing fewer than $n/3$ observations of X, the resulting halfmed(X') will still remain in a bounded region.

It is sometimes suggested to consider the *observation* $\mathbf{x_i}$ with largest ldepth$(\mathbf{x_i}, X)$ as a simple variant of the Tukey median. (If there are several observations with the maximal depth, one can again take their average.) This estimator is easier to compute than the Tukey median, and for 'well-behaved' data sets may be a fair approximation to it. However, the analog of (1) does not hold. For instance, if X consists of points on the half circle $\{(x, y);\ x^2+y^2 = 1 \text{ and } x < 0\}$, the maximal ldepth$(\mathbf{x_i}, X)$ is merely 1. Consequently, this estimator may have a breakdown value as low as $1/n$. For instance, consider the same X and replace one $\mathbf{x_j}$ by a point $(\lambda, 0)$ with $\lambda > 0$, and then let $\lambda \to \infty$. This shows that one outlier can cause breakdown. Therefore, we will not pursue this estimator further.

3. A Straightforward Algorithm

In this section we will construct a relatively straightforward algorithm for the Tukey median, several parts of which will also be used in the more sophisticated algorithm in Section 4. The first algorithm starts from the fact that each edge of the region D_{k^*} is part of a line passing through two observations. Therefore, each vertex of D_{k^*} is an intersection point of two such lines. The algorithm consists of the following steps:

Straightforward algorithm

1. Compute all $M = O(n^4)$ intersection points of lines passing through 2 observations.
2. Compute the halfplane depth of each of these M points by means of the subroutine LDEPTH of Rousseeuw and Ruts (1996), which runs in $O(n \log n)$ time. In short, LDEPTH uses the fact that ldepth$(\boldsymbol{\theta}, X) =$ ldepth$(\mathbf{0}, X - \boldsymbol{\theta})$. It then replaces all $\mathbf{x_i} - \boldsymbol{\theta}$ by $\mathbf{z_i} = (\mathbf{x_i} - \boldsymbol{\theta})/||\mathbf{x_i} - \boldsymbol{\theta}||$. Since the $\mathbf{z_i}$ lie on the unit circle, each $\mathbf{z_i}$ is characterized by its angle $\alpha_i \in [0, 2\pi[$. Also the angle β_i of $-\mathbf{z_i}$ is considered. Then LDEPTH sorts these $2n$ angles together in $O(n \log n)$ time. Finally, ldepth$(\mathbf{0}, \{\mathbf{z_1}, \ldots, \mathbf{z_n}\})$ is computed by a loop of length $2n$ over the sorted angles, where the number of angles in each halfplane is updated at

each step. Note that LDEPTH also yields the simplicial depth of Liu (1990). We now keep the highest halfplane depth k^* encountered, as well as the N points which attain it.

3. Compute the boundary of D_{k^*} as the convex hull of these N points with depth k^*. This convex hull can be obtained with an algorithm of Eddy (1977) or Preparata and Hong (1977). The latter algorithm runs in $O(N \log N) \leq O(n^4 \log n)$ time, and yields a list of points on the convex polygon in counterclockwise order.
4. From this list, delete all points lying inside an edge (this takes $\leq O(N)$ time), retaining only the actual vertices $\{\mathbf{y^1}, \ldots, \mathbf{y^m}\}$ of D_{k^*}. Since each edge of D_{k^*} is part of a line through two observations $\mathbf{x_i}$ and $\mathbf{x_j}$, and because each $\mathbf{x_i}$ can occur in at most two such lines (otherwise D_{k^*} is not convex), we have $m \leq n$.
5. Compute the central point
$$T^\circ = \frac{1}{m} \sum_{j=1}^{m} \mathbf{y^j}$$
which belongs to the convex set D_{k^*}.
6. If $m \leq 3$, the Tukey median T^* equals T°. For $m \geq 4$ the Tukey median is
$$T^* = \text{gravitycenter}(D_{k^*}) = \frac{\int x I(x \in D_{k^*}) d\lambda(x)}{\lambda(D_{k^*})},$$
where λ is the usual measure of area. Since T° is in the interior of D_{k^*} we can write D_{k^*} as the union of regular triangles
$$D_{k^*} = \triangle(T^\circ, \mathbf{y^1}, \mathbf{y^2}) \cup \triangle(T^\circ, \mathbf{y^2}, \mathbf{y^3}) \cup \cdots \cup \triangle(T^\circ, \mathbf{y^m}, \mathbf{y^{m+1}})$$
with $\mathbf{y^{m+1}} := \mathbf{y^1}$, hence
$$\begin{aligned} T^* &= \frac{\sum_{j=1}^{m} \text{area}(\triangle(T^\circ, \mathbf{y^j}, \mathbf{y^{j+1}}))\, \text{gravitycenter}(\triangle(T^\circ, \mathbf{y^j}, \mathbf{y^{j+1}}))}{\sum_{j=1}^{m} \text{area}(\triangle(T^\circ, \mathbf{y^j}, \mathbf{y^{j+1}}))} \qquad (2) \\ &= T^\circ + \frac{\sum_{j=1}^{m} |v_1^j v_2^{j+1} - v_1^{j+1} v_2^j| (\mathbf{v^j} + \mathbf{v^{j+1}})}{3 \sum_{j=1}^{m} |v_1^j v_2^{j+1} - v_1^{j+1} v_2^j|}, \end{aligned}$$
where $\mathbf{v^j} := \mathbf{y^j} - T^\circ$ and $\mathbf{v^{j+1}} := \mathbf{y^{j+1}} - T^\circ$. Clearly, (2) needs $O(m) \leq O(n)$ time.

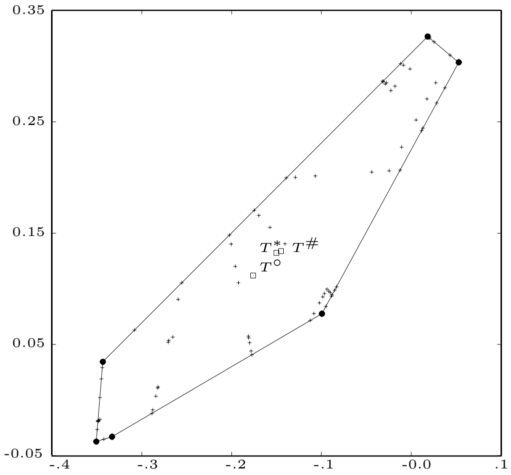

Figure 2. Depth region D_{k^*} of a data set with 15 observations. The maximal depth k^* is 6, and the total number of intersection points with depth 6 (indicated by + signs) is 80. Six of them (indicated by dots) are vertices of D_6. The Tukey median T^* as well as the estimates T° and $T^\#$ are indicated by squares.

The main principle of the algorithm is illustrated in Figure 2, which is based on an actual data set. The intersection points (step 1) with maximal depth k^* (step 2) are indicated by + signs. The boundary of D_{k^*} computed in step 3 is the convex polygon in the figure, and its vertices (from step 4) are shown as solid dots. The estimates T° (step 5) and T^* (step 6) are relatively close to each other, especially since the region D_{k^*} in Figure 2 is quite small relative to the original data cloud (not shown here).

The algorithm's computation time is determined by steps 1 and 2: applying the $O(n \log n)$ time algorithm LDEPTH to the $O(n^4)$ intersection points takes $O(n^5 \log n)$ operations. The other steps take less time, as indicated.

In the way the algorithm is described, its apparent storage is $O(n^4)$ due to step 1. However, this can be improved upon. By combining steps 1 and 2 we only need to store the intersection points with the currently highest depth (by overwriting the intersection points having a previous depth). We can even do with the minimal storage requirement $O(n)$, if we don't store the currently best intersection points but simply add them and count how many there are. This does not yield T^* but the related estimator

$$T^\# = \text{average}\{\text{all intersection points with depth } k^*\}. \qquad (3)$$

In this way we only need a modification of steps 1 and 2 while bypassing steps 3 to 6, yielding an algorithm with $O(n^5 \log n)$ time and only $O(n)$ storage. The estimate $T^{\#}$ is also indicated in Figure 2.

Note that both the set D_{k^*} and each of the estimators T°, T^* and $T^{\#}$ can be considered as valid formalizations of Tukey's notion of median. They all have maximal depth k^*, and when D_{k^*} is a singleton they all coincide. Moreover, they are all affine equivariant. Here we have emphasized T^* because this was the version suggested by Donoho and Gasko (1992), page 1809.

4. The Algorithm HALFMED

Making use of what we learned from the straightforward algorithm, we now construct a faster algorithm for the bivariate Tukey median. The basic idea is to construct several regions D_k for $\lceil \frac{n}{3} \rceil \leq k \leq \lfloor \frac{n}{2} \rfloor$ (using (1) and Proposition 1) in order to find the k^* for which $D_{k^*} \neq \varnothing$ and $D_{k^*+1} = \varnothing$. From D_{k^*} we can then compute the Tukey median T^* as in the straightforward algorithm.

Algorithm HALFMED

1. We first test whether any two data points $\mathbf{x_i}$ and $\mathbf{x_j}$ coincide, which can be done in $O(n \log n)$ time if we sort the data according to their horizontal coordinate (and then sort the vertical coordinates among the points with identical horizontal coordinate). Next we consider all $\binom{n}{2}$ lines L_{ij} through any two data points $\mathbf{x_i}$ and $\mathbf{x_j}$. Each L_{ij} makes an angle $0 \leq \alpha_{ij} < \pi$ with the horizontal axis. We then sort these $O(n^2)$ lines according to their angles in $O(n^2 \log n)$ time, and whenever two or more angles are equal we check whether their lines have a point in common. If that doesn't happen, the data are in regular position. Overall, step 1 takes $O(n^2 \log n)$ time and $O(n^2)$ storage. We now initialize $k^{\text{lower}} \leftarrow \lceil \frac{n}{3} \rceil$ so we know that $D_{k^{\text{lower}}} \neq \varnothing$ by (1), and set $k^{\text{upper}} \leftarrow \lfloor \frac{n}{2} \rfloor + 1$ so we know that $D_{k^{\text{upper}}} = \varnothing$ by Proposition 1 above. Then put $k \leftarrow \lfloor \frac{1}{2}(k^{\text{lower}} + k^{\text{upper}}) \rfloor$.
2. Let us construct D_k. If $D_k \neq \varnothing$ we want to obtain its vertices; if $D_k = \varnothing$ the algorithm should tell us so. This step is a slight extension of steps 2 and 3 in Ruts and Rousseeuw (1996), where a more detailed description is given. The main concept is that of a *special k-divider*, which is a directed (oriented) line passing through two observations such that exactly $n - k - 1$ observations lie strictly to its left and exactly $k - 1$ observations lie strictly to its right. For instance, the line L in Figure 1 is a special k-divider. We can find all the special k-dividers by running through the $O(n^2)$ lines L_{ij} sorted according to their angles α_{ij} (available from step 1) while making use of the *circular sequence* technique of Goodman and Pollack (1980). This means that we start from the projection of the data points on the horizontal direction, and we then let the directed line on which we project rotate counterclockwise. At discrete steps, we update the ranks of the projected points on the current directed line.

This takes $O(n^2)$ operations. Now note that D_k (which may be empty) is the intersection of the closed halfplanes to the left of the special k-dividers. Based on this fact, Ruts and Rousseeuw (1996) devised an $O(n^2 \log n)$ time algorithm (described in step 3 of their algorithm ISODEPTH) which constructs D_k by 'spiraling down' to it by marching only on the special k-dividers and taking left turns where appropriate. Checking when they are appropriate is done by calling LDEPTH. The spiral stops when a point is encountered twice. If its depth is less than k we know that $D_k = \varnothing$; otherwise we have found the vertices of D_k.

3. If $D_k = \varnothing$ we put $k_{\text{new}}^{\text{lower}} \leftarrow k_{\text{old}}^{\text{lower}}$ and $k_{\text{new}}^{\text{upper}} \leftarrow k$. If $D_k \neq \varnothing$ we put $k_{\text{new}}^{\text{lower}} \leftarrow k$ and $k_{\text{new}}^{\text{upper}} \leftarrow k_{\text{old}}^{\text{upper}}$. If $k_{\text{new}}^{\text{upper}} - k_{\text{new}}^{\text{lower}} \geq 2$ we put $k \leftarrow \lfloor \frac{1}{2}(k_{\text{new}}^{\text{lower}} + k_{\text{new}}^{\text{upper}}) \rfloor$ and return to step 2. On the other hand, if $k_{\text{new}}^{\text{upper}} - k_{\text{new}}^{\text{lower}} = 1$ we have found $k^* \leftarrow k_{\text{new}}^{\text{lower}}$ as well as the corresponding D_{k^*}.
4. Apply steps 5 and 6 of the straightforward algorithm in Section 3 above, yielding the Tukey median T^*.

The algorithm HALFMED takes $O(n^2 \log^2 n)$ time and $O(n^2)$ storage. Clearly, step 1 needs $O(n^2 \log n)$ time and $O(n^2)$ storage. Step 2 also takes $O(n^2 \log n)$ time, but has to be repeated $O(\log_2 n)$ times in view of the bisection strategy in step 3, yielding $O(n^2 \log^2 n)$ overall. The time spent on step 4 is negligible by comparison.

Table 1. Execution times (in seconds on a Pentium PC) of the straightforward algorithm and the proposed algorithm HALFMED, for various sample sizes n

n	straightforward	HALFMED
10	0.15	0.02
20	6.47	0.12
30	52.22	0.32
40	236.58	0.59
50	768.52	0.94
100		5.75
200		26.67
300		61.61
400		117.43
500		187.86

Naturally, we have verified that the straightforward algorithm and HALFMED produce exactly the same Tukey median T^*. In Table 1 we compare the execution times of the straightforward algorithm and HALFMED for various sample sizes n, using the same generated data sets for both. The times are in seconds on a Pentium PC. We see that the straightforward algorithm takes a lot of time even for small n, which is not surprising in view of its $O(n^5 \log n)$ complexity.

In contrast, HALFMED computes the Tukey median of 100 points in under 6 seconds.

The source code of the new program HALFMED can be obtained at our website http://win-www.uia.ac.be/u/statis/. Since step 2 of HALFMED actually constructs D_k, the program has been set up so that (apart from T^*) it will also yield the vertices of D_k for any value(s) of k specified by the user. (Therefore, the program HALFMED encompasses and supersedes our older program ISODEPTH.) The extra time for each additional D_k is $O(n^2 \log n)$. Afterwards, these contours can easily be plotted, as in Figures 4 and 5. For data analytic purposes we don't want to draw all depth contours for k between 1 and k^*, but just a few representative ones.

Note that the algorithm HALFMED desribed here is for bivariate data only. The ideas behind the algorithm cannot be extended directly to higher dimensional problems. We actually use a special feature of 2-dimensional space, namely the possibility of identifying all angles with points on a circle, where they can be ordered. This feature first occurs in the algorithm LDEPTH described in Step 2 of Section 3. In Step 2 of the HALFMED algorithm we apply the ISODEPTH algorithm which is based on the idea of circular sequences, which again use the ordering of angles on a circle. Moreover, the ISODEPTH algorithm often calls the LDEPTH subroutine.

We have recently worked on ways to compute the halfspace location depth in higher dimensions (Rousseeuw and Struyf (1998)). In that paper we compute the exact depth in 3 dimensions by a more complicated algorithm, which needs an extra factor n of computer time. In addition to this, the paper also constructs approximate algorithms which are faster and can deal with more than 3 dimensions. However, algorithms for the Tukey median in 3 or more dimensions are not yet available.

5. Simulation: Finite-Sample Efficiency

For several sample sizes n we generated $m = 1000$ samples from the bivariate standard normal distribution, and applied the algorithm HALFMED to each sample. From the m Tukey medians $T_1^*, \ldots, T_m^*$ we computed the bias, the empirical covariance matrix and the empirical efficiency. The results are shown in Table 2. The first column contains the sample size n. The next two columns show the coordinates of the bias:

$$(bias_1, bias_2) = \bar{T}^* = \underset{j=1,\ldots,m}{\text{average}}\, T_j^*.$$

Next we consider the empirical variance-covariance matrix $\hat{C}$:

$$\hat{C} = \frac{1}{m-1}\sum_{j=1}^{m}(T_j^* - \bar{T}^*)(T_j^* - \bar{T}^*)^t.$$

Columns 4 and 5 contain the n-fold variances $n\hat{C}_{11}$ and $n\hat{C}_{22}$, whereas column 6 contains the n-fold covariance $n\hat{C}_{12}$. The final column contains the empirical (Monte Carlo) efficiency, computed as

$$\mathit{eff} = \frac{1}{n\sqrt{det(\hat{C})}}.$$

In Table 2 we see that the finite-sample efficiency of the Tukey median is around 78%. Let us compare this with the analogous results for the coordinatewise median, given by

$$\left(\operatorname*{med}_{i=1}^{n} x_i, \operatorname*{med}_{i=1}^{n} y_i\right)$$

which is easy to compute but not affine equivariant. In Table 3 (using $m = 10,000$) we see that its efficiency is that of the univariate median, and lower than that of the Tukey median in Table 2.

Table 2. Empirical efficiency of the Tukey median for various sample sizes n

n	$bias_1$	$bias_2$	$n\hat{C}_{11}$	$n\hat{C}_{22}$	$n\hat{C}_{12}$	*eff*
10	−0.0014	−0.0018	1.3153	1.2981	−0.0150	0.7654
20	0.0025	−0.0004	1.3140	1.2902	0.0109	0.7681
30	−0.0069	0.0049	1.2930	1.3089	−0.0001	0.7687
40	0.0040	0.0001	1.2564	1.3256	−0.0609	0.7757
50	−0.0055	0.0044	1.2303	1.2437	−0.0318	0.8087
60	0.0017	−0.0066	1.2298	1.3007	−0.0206	0.7908
70	−0.0036	−0.0048	1.3271	1.2364	0.0003	0.7807
80	−0.0015	0.0011	1.2207	1.2947	0.0085	0.7955
90	−0.0026	−0.0024	1.2632	1.2724	−0.0085	0.7888
100	0.0043	−0.0117	1.2848	1.3387	−0.0127	0.7625

Table 3. Empirical efficiency of the coordinatewise median for various sample sizes n

n	$bias_1$	$bias_2$	$n\hat{C}_{11}$	$n\hat{C}_{22}$	$n\hat{C}_{12}$	*eff*
10	−0.0047	0.0020	1.3815	1.3790	−0.0062	0.7245
20	0.0021	0.0012	1.4699	1.4510	−0.0129	0.6848
30	−0.0006	0.0021	1.4674	1.5062	0.0020	0.6726
40	−0.0001	0.0030	1.4986	1.5222	−0.0133	0.6621
50	0.0002	0.0005	1.5208	1.5086	−0.0104	0.6602
60	0.0035	−0.0018	1.5076	1.5013	−0.0108	0.6647
70	−0.0007	0.0001	1.5265	1.5123	0.0177	0.6582
80	0.0006	−0.0016	1.5184	1.4963	−0.0117	0.6634
90	−0.0007	−0.0002	1.5148	1.5429	−0.0187	0.6542
100	−0.0020	0.0016	1.5314	1.5423	−0.0203	0.6507

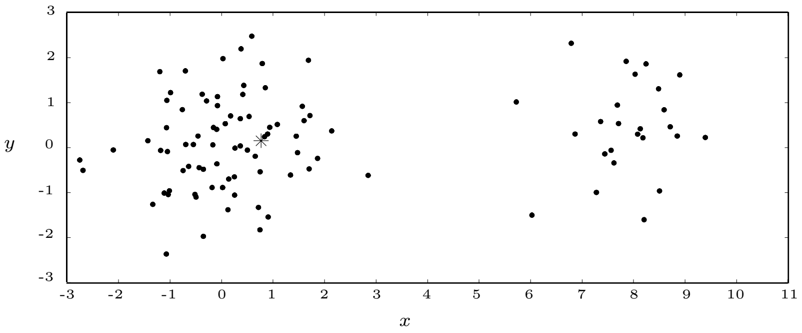

Figure 3. Tukey median $(*)$ of 75 points generated from the bivariate standard normal distribution, and 25 points generated from the normal distribution with center $(8,0)$ and unit covariance matrix.

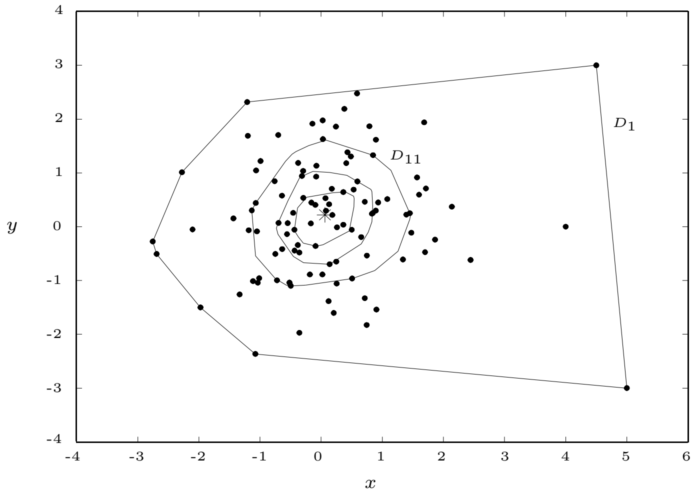

Figure 4. The Tukey median $(*)$ and the contours of depths 1, 11, 21 and 31 of 100 points generated from the standard bivariate normal distribution plus 3 outliers. The maximal depth k^* equals 44.

6. Examples

In the first example we illustrate the robustness property of the Tukey median. We consider 75 points generated from the bivariate standard normal distribution, and 25 points generated from the bivariate normal distribution with center $(8,0)$ and unit covariance matrix. Applying the algorithm HALFMED to these $n = 100$ observations yields the Tukey median T^*, indicated by a $*$ in Figure 3. If we move the 25 points further away to the right, the Tukey median essentially does not change any more.

In a second example we consider 100 data points generated from the bivariate standard normal distribution, and 3 outliers with coordinates $(4, 0)$, $(4.5, 3)$ and $(5, -3)$. The maximal depth k^* was equal to 44. In Figure 4 we have plotted the data and the Tukey median, as well as the contours of depths 1, 11, 21 and 31. The region D_1 is simply the convex hull of the whole data cloud. The outliers influence only the outer contour line, whereas the inner contours are roughly spherical around the median.

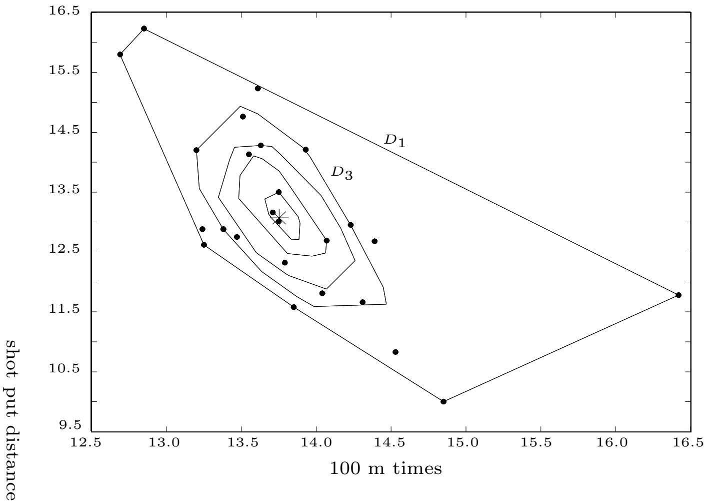

Figure 5. The Tukey median ($*$) and the contours of depth 1, 3, 5, and 7 for the heptathlon data.

Our third example considers the 100 metres time (in seconds) and the shot put distance (in metres) of $n = 25$ heptathletes in the women's heptathlon of the 1988 Olympics (Hand, Daly, Lunn, McConway and Ostrowski (1994)). We applied HALFMED to this data set, yielding the Tukey median T^* indicated by a $*$ in Figure 5. The maximal depth k^* was 10. In Figure 5 we also plotted the regions with depth 1, 3, 5 and 7. There is an outlier on the right of the figure which only affects D_1, whereas the inner contours and the Tukey median are robust against the outlier. Making use of the Tukey median and certain depth contours, we have recently developed a bivariate generalization of the boxplot (Rousseeuw and Ruts (1997)).

References

Donoho, D. L. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Ann. Statist.* **20**, 1803-1827.

Dyckerhoff, R., Koshevoy, G. and Mosler, K. (1996). Zonoid data depth: theory and computation. In *COMPSTAT* 1996 *Proceedings in Computational Statistics* (Edited by A. Prat), 235-240, Physica-Verlag, Heidelberg.

Eddy, W. F. (1977). A new convex hull algorithm for planar sets. *ACM Trans. on Math. Software* **3**, 398-403.

Goodman, J. E. and Pollack, R. (1980). On the combinatorial classification of nondegenerate configurations in the plane. *J. Comb. Theory A* **29**, 220-235.

Grübel, R. (1996). Orthogonalization of multivariate location estimators: the orthomedian. *Ann. Statist.* **24**, 1457-1473.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J. and Ostrowski, E. (1994). *A Handbook of Small Data Sets.* Chapman and Hall, London.

Liu, R. Y. (1990). On a notion of data depth based on random simplices. *Ann. Statist.* **18**, 405-414.

Liu, R. Y. (1992). Data depth and multivariate rank tests. In L_1*-Statistical Analysis and Related Methods* (Edited by Y. Dodge), 279-294, North-Holland, Amsterdam.

Niinimaa, A. (1995). Bivariate generalizations of the median. In *New Trends in Probability and Statistics, Vol* 3 : *Multivariate Statistics and Matrices in Statistics. Proceedings of the* 5*th Tartu Conference*, TEV, Vilnius, pages 163-180.

Niinimaa, A. and Oja, H. (1997). Multivariate median. In *Encyclopedia of Statistical Sciences, Update Volume* 2 (Edited by S. Kotz, C. Read and D. Banks). John Wiley, New York, to appear.

Oja, H. (1983). Descriptive statistics for multivariate distributions. *Statist. Probab. Lett.* **1**, 327-332.

Preparata, F. P. and Hong, S. J. (1977). Convex hulls of finite sets of points in two and three dimensions. *Comm. ACM* **20**, 87-93.

Rousseeuw, P. J. and Hubert, M. (1996). Regression depth. Technical Report, submitted.

Rousseeuw, P. J. and Ruts, I. (1996). AS 307: Bivariate location depth. *Appl. Statist.* (*JRSS-C*) **45**, 516-526.

Rousseeuw, P. J. and Ruts, I. (1997). The bagplot: a bivariate box-and-whiskers plot. Technical Report, submitted.

Rousseeuw, P. J. and Struyf, A. (1998). Computing location depth and regression depth in higher dimensions. *Statist. Comput.*, to appear.

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

Small, C. G. (1990). A survey of multidimensional medians. *Internat. Statist. Rev.* **58**, 263-277.

Tukey, J. W. (1975). Mathematics and the picturing of data. *Proc. Int. Congress Math., Vancouver* **2**, 523-531.

Department of Mathematics and Computer Science, UIA, Universiteitsplein 1, B-2610 Antwerp, Belgium.

E-mail: rousse@uia.ua.ac.be

E-mail: ruts@uia.ua.ac.be

(Received August 1996; accepted May 1997)