Computing zonoid trimmed regions of dimension $d > 2$

Karl Mosler, Köln*
Tatjana Lange, Merseburg
Pavel Bazovkin, Kiev

December 2, 2008

Abstract: A probability distribution on Euclidean d-space can be described by its zonoid regions. These regions form a nested family of convex sets around the expectation, each being closed and bounded. The zonoid regions of an empirical distribution introduce an ordering of the data that has many applications in multivariate statistical analysis, e.g. cluster analysis, tests for multivariate location and scale, and risk analysis. An exact algorithm is developed to constructing the zonoid regions of a d-variate empirical distribution by their facets when $d \geq 3$. The vertices of the region and their adjacency are characterized, and a procedure is suggested by which all vertices and facets can be determined. The algorithm is available as an R-package.

Keywords: Multivariate statistical analysis, central regions, zonoid data depth, convex polytope, algorithm.

1 Introduction

A probability distribution on Euclidean d-space can be described by its zonoid trimmed regions or, in brief, zonoid regions. These regions form a

*Corresponding author: Karl Mosler, Department of Social and Economic Statistics, University of Cologne, Albertus-Magnus-Platz, 50923 Köln, Germany. mosler@statistik.uni-koeln.de.

nested family of convex sets – so-called central regions – around the expectation, each being closed and bounded. The family is indexed by numbers that vary in the unit interval. Each zonoid region can be seen as a set-valued parameter that reflects the location, scale, and shape of the distribution.

Multivariate data are often asymmetrically distributed so that they cannot be modelled by normal or elliptical distributions. Zonoid regions offer a nonparametric and particularly visual approach to analyzing such data. A distribution - empirical as well as theoretical - is uniquely represented by a geometrical object; its family of zonoid regions. This object is visual and has attractive analytical properties. Moreover, the zonoid regions of an i.i.d. sample satisfy a law of large numbers, converging to the zonoid regions of the underlying probability distribution.

Zonoid regions were introduced by Koshevoy and Mosler (1997) and, since then, have found many applications in multivariate statistical analysis. They have been employed, e.g., in cluster analysis (Mosler and Hoberg, 2006), in the measurement of inequality (Koshevoy and Mosler, 2007) and polarization (Gigliarano and Mosler, 200x), and in tests for multivariate location and scale (Dyckerhoff, 2002); see also the monograph by Mosler (2002). Molchanov and Cascos (2007) propose a general geometric framework for measuring multivariate risks; in their approach zonoid regions serve as set-valued risk measures that generalize the usual univariate expected shortfall.

The boundary of a zonoid region forms a depth contour with respect to zonoid depth and can be regarded as a multivariate quantile. Therefore, given a d-variate empirical distribution, zonoid regions are used as trimmed regions that exclude "outlying" data and include "inlying", that is to say, central and relevant ones. Similar methodology has been based on alternative notions of data depth and trimmed regions, such as halfspace (= location) depth, simplicial depth, expected convex hull depth, among others. Zuo and Serfling (2000) provide some general theory of depth trimmed contour regions, while Liu, Parelius and Singh (1999) and Serfling (2006) broadly survey the theory and applications of various notions of depths in multivariate data. Lopez-Pintado and Romo (2007) investigate depth notions in functional data.

When applying such methods to given multivariate data, the crucial point is the availability of efficient numerical procedures to compute the data depths employed and the trimmed regions of an empirical distribution. To calculate the depth of a single point of an arbitrary dimension, algorithms have been

provided by Rousseeuw and Ruts (1996) and Rousseeuw and Struyf (1997) for the halfspace depth, and by Dyckerhoff et al. (1996) for the zonoid depth. Aloupis (2006) gives a survey of algorithms for calculating different notions of medians and depths. But, calculating a depth trimmed region appears to be a much more demanding task. So far, algorithms have been constructed for the halfspace trimmed regions by Ruts and Rousseeuw (1996) and Miller et al.(2003) in dimension 2, and by Fukuda and Rosta (2004) in arbitrary dimension. For bivariate zonoid trimmed regions, Dyckerhoff (2000) provides an algorithm that employs a circular sequence; Cascos (2006) uses the same approach for bivariate regions that are trimmed by the expected convex hull depth.

In this paper we present an exact algorithm that efficiently calculates zonoid regions of any dimension. In contrast to most classical statistical tools, zonoid regions are genuine geometric notions. Consequently, our algorithm makes ample use of tools from computational geometry.

Consider an empirical distribution that gives probability $\frac{1}{n}$ to each of the observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and let $n \geq d$. The zonoid regions of the empirical distribution are defined by

$$D_\alpha(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \left\{ \frac{1}{n\alpha} \sum_{i=1}^{n} \lambda_i \mathbf{x}_i : \sum_{i=1}^{n} \lambda_i = n\alpha, 0 \leq \lambda \leq 1 \ \ \forall i \right\}, \qquad (1)$$

$0 < \alpha < 1$. Immediately it can be seen from the definition that $D_{\frac{1}{n}}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the convex hull of the data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, while $D_1(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the set that contains the mean $\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ as a single point. At $0 < \alpha < 1$, $D_\alpha(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is a convex polytope that lies between the convex hull and the expectation point and decreases strictly with α. The border of such polytope consists of a finite number of facets. Each facet is part of a hyperplane in $\mathbb{R}^d$ and can be described by the direction of its normal vector and its distance from the origin, that is to say, by some element $\mathbf{p}$ of the unit sphere S^{d-1} and some $\lambda \in \mathbb{R}_+$. The main task is to identify, from all directions $\mathbf{p} \in S^{d-1}$, those directions that determine the facets and calculate them in an efficient way.

The algorithm constructs the zonoid regions of an empirical distribution by their facets. Thus, for any data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and any $\alpha \in [0, 1[$, the facets and vertices of $D_\alpha(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ are calculated and their coordinates are given. Our algorithm is efficient in that it computes the facets one after the other, proceeding from one facet to its neighbors.

In the dimension $d = 2$, Dyckerhoff (2000) has developed an algorithm for constructing zonoid regions. His procedure is based on the idea of a circular sequence (cf. Edelsbrunner, 1987): A ray starting at the center is turned like a clock's hand and the data points are projected onto this ray. However, the method of circular sequence works only with bivariate data. There is no obvious generalization of such a sequence to higher dimensions.

Our first task is to characterize the vertices and facets of a given zonoid region, given data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and α. For this, we introduce a global structure that partitions $\mathbb{R}^d$ into direction cones that correspond one-to-one to the vertices of the zonoid region. In this cone structure, the adjacency of vertices is investigated and characterized. A linear program is constructed to decide whether two vertices are neighbors. The resulting adjacency graph consists of elementary cycles that have either three or six nodes. Then we show that each facet of the zonoid region corresponds to exactly d data points and characterize the facet by a linear restriction on d data points.

Our second task is to put the facets of the zonoid region into an order according to which they can be efficiently calculated. For a given facet, a "jump-to-neighbor" procedure is introduced to transfer the calculation to the neighboring facets. Finally, a facet transversal graph is constructed, and a spanning tree order is realized to transverse this graph in an efficient way. This completes the algorithm.

Overview of the paper: Section 2 presents zonoid regions of general probability distributions and surveys their principal statistical properties. In Section 3 the set of supporting vectors that belong to a given vertex of the zonoid is investigated. In Section 4 a global structure of direction cones is set up, and the adjacency of vertices is described through conditions on these direction cones. Section 5 provides a linear program by which the adjacency of vertices can be checked. Section 6 presents the adjacency graph and a characterization of facets of the zonoid region. In Section 7 the "jump-to-neighbor" procedure and the spanning tree order are introduced, according to which all facets are transversed. Section 8 concludes with a discussion of the complexity of our algorithm and its use in calculating zonoid regions for different α. The last section also provides numerical experience and remarks on possible modifications of the algorithm.

2 Zonoid regions

Given a d-variate probability distribution function F, a family $\{D_\alpha(F)\}$ of sets in d-space, called *zonoid regions*, is defined as follows: $D_0(F) = \mathbb{R}^d$ and for $\alpha \in]0, 1]$

$$D_\alpha(F) \;=\; \left\{ \int_{\mathbb{R}^d} \mathbf{x}\, g(\mathbf{x})\, dF(\mathbf{x}) : 0 \le g \le \frac{1}{\alpha}\,, \quad \int_{\mathbb{R}^d} g(\mathbf{x})\, dF(\mathbf{x}) = 1 \right\}. \quad (2)$$

For $\alpha \in]0, 1]$, these regions exist if and only if F has a finite expectation vector $\mu_F = \int_{\mathbb{R}^d} \mathbf{x} dF(\mathbf{x})$. It is obvious from the definition (2) that the zonoid regions are nested; the smallest region being the singleton set $D_1(F) = \{\mu_F\}$. Furthermore, each $D_\alpha(F)$ is bounded, closed, and convex. For an empirical distribution F, with equal mass on (not necessarily different) points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the definition specializes to the above definition (1).

In this section we list a few principal properties of zonoid regions, which make them useful for statistical description and inference. For details and many other theoretical results, the reader is referred to Mosler (2002), as well as for applications to the multivariate analysis of location, dispersion and dependency.

Firstly, for every $\alpha \in [0, 1]$, D_α is affine equivariant, i.e.,

$$D_\alpha(F_{\mathbf{XA}+\mathbf{c}}) = D_\alpha(F_X)\mathbf{A} + \mathbf{c} \quad \text{for any } d \times d \text{ matrix } \mathbf{A} \text{ having full rank,} \quad (3)$$

$\mathbf{c} \in \mathbb{R}^d$, and F having finite first moment. Hence, any statistical procedure based on zonoid regions is an **affine equivariant procedure**.

Secondly, the zonoid regions contain **full information** about the underlying distribution: For any two d-variate distribution functions F and G that have finite first moments, it holds

$$F = G \qquad \text{if and only if} \qquad D_\alpha(F) = D_\alpha(G) \quad \text{for all } \alpha \in]0, 1]\,. \quad (4)$$

The uniqueness property (4) implies that any claim about a distribution F can be equivalently formulated as a claim about the zonoid regions of F and, thus, can be analyzed by geometric means.

Thirdly, zonoid regions lend themselves easily to **projection methods**: By projecting the zonoid regions of a distribution F onto some lower-dimensional

subspace of $\mathbb{R}^d$, the zonoid regions of the projected distribution are obtained. In particular, any marginal of F has zonoid regions that are obtained by projection onto the respective coordinate space. See also Dyckerhoff (2004).

The fourth property is **continuity**: The zonoid region $D_\alpha(F)$ is continuous on α as well as on F. More precisely (see Mosler (2002, Th. 3.10)), given a distribution F and a sequence (α_n) in $]0,1]$ that converges to $\alpha > 0$, it holds

$$D_{\alpha_n}(F) \quad \xrightarrow{H} \quad D_\alpha(F)\,, \tag{5}$$

Also, given $\alpha \in]0,1]$ and a sequence $F^{(n)}$ of distributions that is uniformly integrable and weakly convergent to F, it holds

$$D_\alpha(F^{(n)}) \quad \xrightarrow{H} \quad D_\alpha(F)\,. \tag{6}$$

In (5) and (6), $\xrightarrow{H}$ means convergence with respect to Hausdorff distance. (The Hausdorff distance of two compacts C and D is the smallest ϵ for which C plus an ϵ-ball includes D and D plus an ϵ-ball includes C as well.)

Fifthly, zonoid regions satisfy a **Law of Large Numbers**, which serves as the basis of statistical inference: Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ denote an i.i.d. sample, with $\mathbf{X}_i \sim F$. For every α, it holds

$$D_\alpha(\mathbf{X}_1, \ldots \mathbf{X}_n) \quad \xrightarrow{H} \quad D_\alpha(F) \quad \text{almost surely}\,. \tag{7}$$

Thus, given an i.i.d. sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$, the zonoid region $D_\alpha(\mathbf{X}_1, \ldots \mathbf{X}_n)$ serves as a *set-valued statistic* that estimates $D_\alpha(F)$. So far, the practical application of the zonoid region statistic was confined to the dimension $d = 2$, since no algorithm existed to calculate it for higher dimensions. In the rest of this paper we develop an exact algorithm that works for any dimension $d \geq 3$.

3 Vertices and direction domains of a zonoid region

Let us first recall some standard notions and facts about convex sets and polytopes in $\mathbb{R}^d$. A *convex polytope* is the convex hull of a finite number of

points or, equivalently, the nonempty intersection of a finite number of closed halfspaces. A nonempty intersection of its boundary with a hyperplane is called a *facet* if it has an affine dimension $d-1$, and a *ridge* if it has an affine dimension $d-2$. It is called an *edge* if it is a line segment, and a *vertex* if it is a single point. The boundary of a convex polytope is the union of its facets. A convex polytope has a finite number of facets, ridges, edges, and vertices. An edge is the intersection of (at least) two facets, and a vertex is the intersection of (at least) two edges, $d-1$ ridges and d facets. A *hyperline* is an affine subspace of $\mathbb{R}^d$ that has a dimension 1.

A compact convex set is mentioned as a *convex body*. In particular, as a zonoid region is a bounded convex polytope, it forms a convex body in $\mathbb{R}^d$. The *support function* $h_C : S^{d-1} \to \mathbb{R}$ of a convex body $C \subset \mathbb{R}^d$ is defined by

$$h_C(\mathbf{p}) = \max\{\mathbf{p}'\mathbf{x} : \mathbf{x} \in C\}.$$

The support function of a convex body is closely related to its extreme points: A point $\mathbf{x}_0$ is extreme in C if and only if some $\mathbf{p} \in S^{d-1}$ exists so that

$$\mathbf{p}'\mathbf{x} = h_C(\mathbf{p}) \quad \text{implies} \quad \mathbf{x} = \mathbf{x}_0\,.$$

Now, for the given data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, denote

$$H = H(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \{\mathbf{p} \in S^{d-1} : \mathbf{p}'\mathbf{x}_i = \mathbf{p}'\mathbf{x}_j \quad \text{for some } \ i \neq j\}\,.$$

Given a direction $\mathbf{p} \in S^{d-1} \setminus H$, the inner product $\mathbf{p}'\mathbf{x}$ projects the data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ onto numbers $\mathbf{p}'\mathbf{x}_1, \ldots, \mathbf{p}'\mathbf{x}_n \in \mathbb{R}$. While the data as points in $\mathbb{R}^d$ have no natural total order, their projection does have. Thus, each $\mathbf{p} \in S^{d-1} \setminus H$ induces a total ordering of the data, i.e., a permutation $\pi_{\mathbf{p}}$ of the index set $1, \ldots, n$ given by

$$\mathbf{p}'\mathbf{x}_{\pi_{\mathbf{p}}(1)} < \mathbf{p}'\mathbf{x}_{\pi_{\mathbf{p}}(2)} < \cdots < \mathbf{p}'\mathbf{x}_{\pi_{\mathbf{p}}(n)}\,. \tag{8}$$

In the sequel we notate the $\mathbf{p}$-ordered data by

$$\mathbf{X}^{\mathbf{p}} = (\mathbf{x}_1^{\mathbf{p}}, \ldots, \mathbf{x}_n^{\mathbf{p}}) \quad \text{with } \ \mathbf{x}_i^{\mathbf{p}} = \mathbf{x}_{\pi_{\mathbf{p}}(i)}\,, \quad i = 1, \ldots, n\,.$$

Proposition 1. (Dyckerhoff, 2000) *Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ be pairwise distinct, $d \in \mathbb{N}$. For any $\mathbf{p} \in S^{d-1} \setminus H$ define*

$$\mathbf{x}_{\mathbf{p},\alpha} = \frac{1}{n\alpha} \sum_{i=1}^{n} \lambda_i^{\mathbf{p}} \mathbf{x}_i\,, \tag{9}$$

where

$$\lambda_i^{\mathbf{p}} = \begin{cases} 1 & \textit{if } \pi_{\mathbf{p}}(i) > n - [n\alpha] \ , \\ n\alpha - [n\alpha] & \textit{if } \pi_{\mathbf{p}}(i) = n - [n\alpha] \ , \\ 0 & \textit{if } \pi_{\mathbf{p}}(i) < n - [n\alpha] \ . \end{cases}$$

Then the set of vertices of the zonoid region D_α is given by

$$\mathcal{V}(D_\alpha) = \{\mathbf{x}_{\mathbf{p},\alpha} \in \mathbb{R}^d : \mathbf{p} \in S^{d-1} \setminus H\} \, .$$

The set of all directions that yield vertices (= extreme points) of D_α is $S^{d-1} \setminus H$. Let $S(\mathbf{v}) \subset S^{d-1} \setminus H$ denote the subset of those directions that belong to a given vertex $\mathbf{v} \in S^{d-1} \setminus H$. $S(\mathbf{v})$ is named a *direction domain*. According to Proposition 1, all directions that provide the same permutation of the data belong to the same direction domain, i.e. to some common vertex $\mathbf{v}$. The family of direction domains $S(\mathbf{v}), \mathbf{v} \in \mathcal{V}(D_\alpha)$, forms a finite partition of $S^{d-1} \setminus H$.

Thus, the proposition yields a discretization of the continuum of possible directions of the vector $\mathbf{p}$, where the cardinality of the set of direction domains equals the number of vertices of the zonoid region. Thus, a one-to-one relation between domains of directions and vertices has been established.

In the sequel we assume that the data are *in general position*, i.e., every subset of $k+1$ data points generates an affine space of the dimension k, $k = 1, \ldots, d-1$. (If the data are not in general position, the subsequent discussion and the algorithm need to be modified, e.g., by slightly perturbing the data.) Also, without loss of generality, we shall assume that the mean of the data is at the origin, $\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i = 0$.

4 Adjacent vertices

In this section we investigate the transition of one vertex to another, that is to say, of one direction domain to another. In our procedure we let a support vector $\mathbf{p}$ – that represents direction – continuously rotate on the unit sphere S^{d-1}. We start with an arbitrary $\mathbf{p} \in S^{d-1}$, which provides an initial permutation of the data points. As was mentioned, all $\mathbf{p}$ that produce the same permutation of points form a direction domain that belongs to a common vertex. When searching for all vertices, it obviously suffices to

traverse each direction domain once. To do this, we shall characterize and identify the possible transitions of one domain to a neighboring one. Our identification procedure is based on the following observations:

The vector $\mathbf{p}$ *hits the boundary of a direction domain only if, for some* $i \neq j$, $\mathbf{p}'\mathbf{x}_i = \mathbf{p}'\mathbf{x}_j$ *holds, i.e.,* $\mathbf{p}$ *is orthogonal to* $\mathbf{x}_i - \mathbf{x}_j$.

Note that the pair $(\mathbf{p}'\mathbf{x}_i, \mathbf{p}'\mathbf{x}_j)$ is not unique. However, at most $d-1$ such pairs can arise, as the space of all vectors that are orthogonal to $\mathbf{p}$ has an affine dimension $d-1$, and the data are in general position.

The vector $\mathbf{p}$ *crosses the boundary of a direction domain only if, for some* $i \neq j$, $\mathbf{p}'x_i$ *and* $\mathbf{p}'x_j$ *change their order. That is to say, any transition from one permutation to another is done by swapping one of the pairs of data points.*

With Proposition 1 follows:

Theorem 1. (Identification of vertices) *The vector* $\mathbf{p}$ *passes from one direction domain (and one vertex) to a neighboring one if and only if* i *and* j *exist,* $i \neq j$, *so that* $\mathbf{p}$ *is orthogonal to* $\mathbf{x}_i - \mathbf{x}_j$, $\pi_{\mathbf{p}}(i) = n - [n \cdot \alpha]$, *and* $|\pi_{\mathbf{p}}(i) - \pi_{\mathbf{p}}(j)| = 1$.

Theorem 1 provides the basis for an algorithm that calculates all vertices of the zonoid region.

So far, we have considered parts of the unit sphere; the direction domains. They correspond to the vertices of the zonoid region. In the following discussion we will use the corresponding closed cones: For $\mathbf{v} \in \mathcal{V}(D_\alpha)$ define the *direction cone* $C(\mathbf{v})$,

$$C(\mathbf{v}) = \mathrm{cl}\{\lambda\mathbf{y} : \lambda \in \mathbb{R}_+, \mathbf{y} \in S(\mathbf{v})\}\,,$$

where $\mathrm{cl}(U)$ means closure of a set $U \in \mathbb{R}^d$. Then each $C(\mathbf{v})$ is a closed convex cone in $\mathbb{R}^d$ with an apex at the origin. It is finitely generated (i.e. it consists of all non-negative linear combinations of a finite number of vectors), and has a maximum of $n-1$ facets. The normals of its facets are described in Theorem 1.

The family of direction cones provides a global structure that divides the space $\mathbb{R}^d$ into sets corresponding to the vertices of D_α.

Now, consider three data points $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ and the hyperplanes through the origin that are orthogonal to $\mathbf{x}_i - \mathbf{x}_j$, $\mathbf{x}_i - \mathbf{x}_k$, and $\mathbf{x}_j - \mathbf{x}_k$, respectively. These hyperplanes intersect at a common hyperline that possibly contains a ridge of a direction cone. On the other hand, every ridge of a direction cone is contained in such a hyperline for some i, j, and k.

We can conclude: *In the global structure a ridge of a direction cone belongs to another direction cone only if it is a ridge of the latter, too.*

In the sequel we shall say that two direction cones are *adjacent cones* if they have a full facet in common. Turning the vector $\mathbf{p}$ through a boundary facet implies a transition from a direction cone to an adjacent one. It means that, to leave a current cone and enter an adjacent one, we have to move the vector $\mathbf{p}$ in an arbitrary way beyond one of the hyperplanes that carry the facets of the current cone.

5 A linear program for constructing adjacent vertices

Our next task is to find an explicit way of constructing all vertices that are neighbors of a given vertex $\mathbf{v}$. In other words, we will explicitly determine all direction cones that are adjacent to a given direction cone $C(\mathbf{v})$. These neighbors correspond to the facets of $C(\mathbf{v})$. Every facet of the cone is defined by a hyperplane. The set of hyperplanes that determine the cone's boundary is a subset of the hyperplanes described in Section 3. Each of these $n-1$ hyperplanes is represented by its "inner" normal $\mathbf{z}_j$, i.e., the normal pointing inside the cone:

$$\mathbf{z}_j = \begin{cases} \mathbf{x}^{\mathbf{p}}_{n-[n\alpha]} - \mathbf{x}^{\mathbf{p}}_j & \text{if } \; j = 1, \ldots, n - [n\alpha] - 1\,, \\ \mathbf{x}^{\mathbf{p}}_j - \mathbf{x}^{\mathbf{p}}_{n-[n\alpha]} & \text{if } \; j = n - [n\alpha] + 1, \ldots, n\,. \end{cases}$$

The $n-1$ normals are codirected with all directions $\mathbf{p}$ in the cone, i.e., they have non-negative inner products. In fact, $C(\mathbf{v})$ is the intersection of all corresponding halfspaces. In other words, $C(\mathbf{v})$ is the set of all vectors $\mathbf{p}$ that are codirected with the normals (5),

$$C(\mathbf{v}) = \{\mathbf{p} \in \mathbb{R}^d : \mathbf{z}_j'\mathbf{p} \geq 0 \quad \text{for all } \; j = 1, \ldots, n-1\}\,.$$

Our task is to identify those hyperplanes (or, equivalently, their normals) that belong to the boundary of the cone $C(\mathbf{v})$. To determine whether $\mathbf{z}_j$ is a boundary normal, we shall solve the following minimization problem:

$$\begin{aligned} & \mathbf{z}_j'\mathbf{p} \to \min \\ \text{s.t.} \quad & \mathbf{p} \in C(\mathbf{v})\,, \\ & \sum_{i=1}^{d} |\mathbf{p}_i| = 1\,. \end{aligned} \tag{10}$$

If (10) has a positive minimal value, no support vector, $\mathbf{p} \in C(\mathbf{v})$, exists that is orthogonal to $\mathbf{z}_j$. Hence, $\mathbf{z}_j$ is not a boundary normal of $C(\mathbf{v})$. If (10) is minimized with value $\mathbf{z}_j'\mathbf{p}^* = 0$, we can conclude that $\mathbf{p}^*$ is a support vector that belongs to the boundary of $C(\mathbf{v})$ and is an element of the hyperplane that has normal $\mathbf{z}_j$. Consequently, $\mathbf{p}^*$ also belongs to the boundary of the direction cone $C(\tilde{\mathbf{v}})$ of some vertex $\tilde{\mathbf{v}}$ that borders on the current vertex $\mathbf{v}$.

(10) can be rewritten as a linear program (LP_j),

$$\begin{aligned} & \mathbf{z}_j'(\mathbf{p}^+ - \mathbf{p}^-) \to \min \\ \text{s.t.} \quad & \mathbf{z}_j'(\mathbf{p}^+ - \mathbf{p}^-) \geq 0 \quad j = 1, \ldots, n-1\,, \\ & \sum_{i=1}^{d} (p_i^+ + p_i^-) = 1\,, \\ & p_i^+ \geq 0\,, \quad p_i^- \geq 0\,, \quad i = 1, \ldots, d\,. \end{aligned} \tag{11}$$

Here we have inserted $\mathbf{p} = \mathbf{p}^+ - \mathbf{p}^-$, where $\mathbf{p}^+ = (p_1^+, \ldots, p_d^+)'$ and $\mathbf{p}^- = (p_1^-, \ldots, p_d^-)'$ are the positive and negative parts of $\mathbf{p}$, respectively. The linear program (11) is solved by the simplex method.

To find all vertices bordering on $\mathbf{v}$, we may solve the linear programs (11) for $j = 1, \ldots, n-1$. As all programs have the same set of feasible solutions, the calculations can be shortened by solving them simultaneously. In fact, the number of neighbors is small compared to $n-1$. Therefore, the number of basic feasible solutions in the simplex method will be relatively small, too, which leads to a high average efficiency of the simplex method. If n is large, the dual simplex method may outperform the primal approach.

Each ridge of a direction cone is an intersection of three hyperplanes. On the ridge either three or six direction cones touch each other. Let us consider these two cases in more detail:

Figure 1: Neighboring cones near the common ridge.

Remember that, given a vertex $\mathbf{v}$, for all $\mathbf{p} \in C(\mathbf{v})$ the point $\mathbf{x}^{\mathbf{p}}_{n-[n\alpha]}$ does not depend on $\mathbf{p}$. We call this point the *main point* of $C(\mathbf{v})$.

Consider three direction cones $C(\mathbf{v}), C(\mathbf{w})$, and $C(\mathbf{u})$ that have a common ridge and denote their main points by $\mathbf{a}, \mathbf{b}, \mathbf{c}$, respectively. For all $\mathbf{p} \in C(\mathbf{v})$ these three main points are $\mathbf{p}$-ordered in the same way, either with $\mathbf{a}$ in the middle or not. If $\mathbf{a}$ is in the middle, $\mathbf{b}$ and $\mathbf{c}$ cannot switch their positions. Hence, $C(\mathbf{w})$ and $C(\mathbf{u})$ have no boundary hyperplane in common, and $\mathbf{w}$ and $\mathbf{u}$ are not adjacent vertices. In this case, a total of six direction cones meet at the common ridge. If, on the other hand, $\mathbf{b}$ and $\mathbf{c}$ are on the same $\mathbf{p}$-side of $\mathbf{a}$, their positions can switch and $\mathbf{w}$ and $\mathbf{u}$ are neighboring vertices. In this case, only three direction cones unite at the common ridge. The two cases are illustrated in Figure 1.

6 Edges and facets

Having obtained an efficient procedure for finding extreme points we now must create an efficient one for constructing all facets of the zonoid region. Next we characterize the edges of the zonoid region.

Lemma 1 (Vertices and edges). *Let $C(\mathbf{v})$ and $C(\mathbf{w})$ be direction cones. The line connecting $\mathbf{v}$ and $\mathbf{w}$ is an edge if and only if $C(\mathbf{v})$ and $C(\mathbf{w})$ are adjacent cones.*

Proof. Recall that the zonoid region D_α is a convex polytope, and its extreme points form the vertices of this polytope. As $C(\mathbf{v})$ and $C(\mathbf{w})$ are direction cones of vertices $\mathbf{v}$ and $\mathbf{w}$, for all $\mathbf{x} \in D_\alpha$ it holds that:

$$\mathbf{p}'\mathbf{x} \leq \mathbf{p}'\mathbf{v} \quad \text{if } \mathbf{p} \in C(\mathbf{v}), \quad \text{and} \quad \mathbf{p}'\mathbf{x} \leq \mathbf{p}'\mathbf{w} \quad \text{if } \mathbf{p} \in C(\mathbf{w}). \tag{12}$$

Now assume that $C(\mathbf{v})$ and $C(\mathbf{w})$ are adjacent cones. Then for each $\mathbf{p}$ in their common boundary $C(\mathbf{v}) \cap C(\mathbf{w})$ it holds that $\mathbf{p}'\mathbf{v} = \mathbf{p}'\mathbf{w}$. Hence, for all $\mathbf{x} \in D_\alpha$

$$\mathbf{p}'\mathbf{x} \leq \mathbf{p}'(\lambda\mathbf{v} + (1-\lambda)\mathbf{w}) \quad \text{if } \lambda \in [0,1]. \tag{13}$$

It follows that the line connecting $\mathbf{v}$ and $\mathbf{w}$ is an edge of the polytope. On the other hand, assume that this line $\overline{\mathbf{vw}}$ is an edge. Then some $\mathbf{p}$ exists that, for all $\mathbf{x} \in D_\alpha$, satisfies (13). For this $\mathbf{p}$, it must hold that $\mathbf{p}'\mathbf{v} = \mathbf{p}'\mathbf{w}$. In view of (12), we can conclude that (13) is true for all $\mathbf{x} \in D_\alpha$ if and only if

$$\mathbf{p}'(\mathbf{v}-\mathbf{w}) = 0, \ \mathbf{p}'\mathbf{v} \geq 0, \text{ and } \mathbf{p}'\mathbf{w} \geq 0,$$

that means, $\mathbf{p}$ is in a $d-1$-dimensional cone which is a subset of $C(\mathbf{v})$ and of $C(\mathbf{w})$. Consequently, $C(\mathbf{v})$ and $C(\mathbf{w})$ have a full facet in common, and are, thus, adjacent cones. □

Lemma 1 provides a unique correspondence between the adjacency of direction cones in the global cone structure and the existence of edges of the zonoid region.

Recall that adjacent direction cones are cones that have a common facet. The adjacency information of the zonoid region, which is a polytope, is represented by its *adjacency graph*, which consists of the polytope's vertices and edges. Above, we have demonstrated that either three or six direction cones

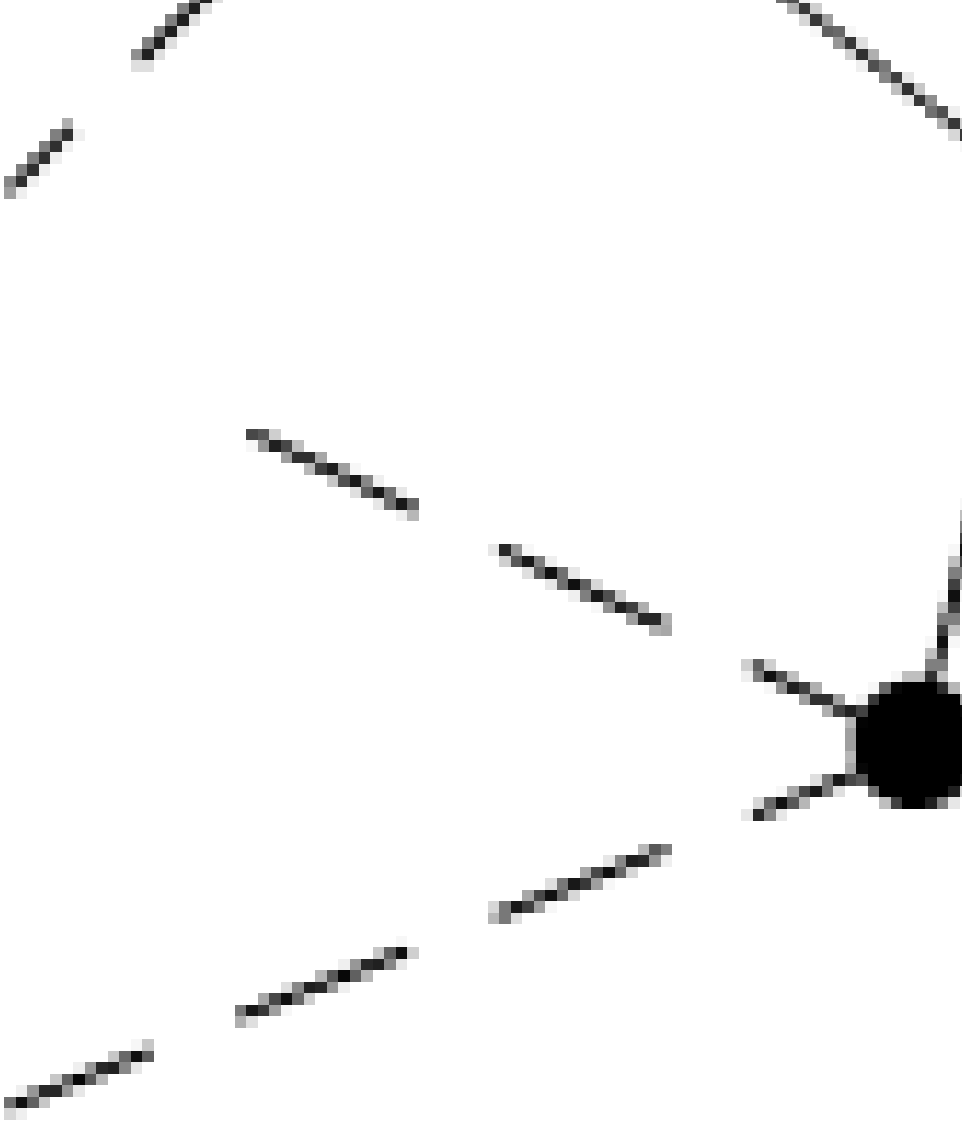

Figure 2: The structure of the adjacency graph.

touch each other on a ridge. Hence the adjacency graph is a concatenation of elementary cycles, each connecting either three or six vertices. This is illustrated in Figure 2:

Now, consider a facet of the zonoid region. Let the vector $\mathbf{p}$ be directed orthogonally to the facet. Then, if points from the data cloud are in general position, exactly d points $\mathbf{x}_{\pi_{\mathbf{p}}(k)}, \mathbf{x}_{\pi_{\mathbf{p}}(k+1)}, \ldots, \mathbf{x}_{\pi_{\mathbf{p}}(k+d-1)}$ exist so that:

$$\left.\begin{array}{l} \mathbf{p}'\mathbf{x}_{\pi_{\mathbf{p}}(k)} = \mathbf{p}'\mathbf{x}_{\pi_{\mathbf{p}}(k+1)} = \cdots = \mathbf{p}'\mathbf{x}_{\pi_{\mathbf{p}}(k+d-1)} \\ \text{with some } k\,, \quad k \leq n - [n\alpha] \leq k + d - 1\,. \end{array}\right\} \tag{14}$$

Obviously, the indices $k, \ldots, k+d-1$ in (14) are not unique. However, any permutation of these d points yields the same facet. We may conclude the

following theorem.

Theorem 2. (Identification of facets) *Each facet can be identified by a set of exactly d points from the data cloud and one of its vertices. Moreover, if there is a support vector that defines a permutation satisfying (14), then this permutation and these d data points define a facet of the zonoid region.*

It can easily be seen from Theorem 2 that, if $k < n-[n\alpha] < k+d-1$, the facet can have more than d vertices . A facet will be mentioned as *redundant* if it has more than d vertices, and as *non-redundant* if it has exactly d vertices. In any case, there are only d different main points that belong to a facet. Let $\ell = n - [n\alpha] - k$. We obtain:

Corollary 1. *The number of vertices of the facet equals* $d \cdot \binom{d-1}{\ell}$.

Proof. As stated above, the total number of possible relative positions of d points is $d!$. But, according to (9), the relative position of points in

$$[k, (n - [n\alpha])[\quad \text{and} \quad](n - [n\alpha]), k + d - 1]$$

is not significant, i.e. $\binom{d-1}{\ell}$ different cases remain. This number is multiplied by d, which is the number of possible main points. □

Corollary 2. *Each set of d points from the data cloud defines, at most, one facet of a zonoid region unless:*

$$k \leq n - [n\alpha] \leq k + d - 1 \quad \textit{and} \quad k \leq n - [n - n\alpha] \leq k + d - 1. \tag{15}$$

Otherwise it defines exactly two parallel facets.

Proof. The first statement is clear from Theorem 2. The second is based on the fact that, if the condition (15) is met, then the inverted support vector also defines a permutation satisfying (14). □

Corollary 3. *For each set of d data points some α exists, so that the set defines a facet of the zonoid α-region.*

Proof. In fact, for a set of exactly d data points it is always possible to find α so that the condition (14) is met. Then the statement follows from Theorem 2. □

In the case of a *non-redundant facet* we have $\ell = 0$. Then, according to Corollary 1 the facet is a $(d-1)$-dimensional simplex having d vertices. The vertices identify the facet; they are pairwise adjacent and correspond to the pairwise adjacent direction cones. In turn, every set of d vertices that correspond to pairwise neighboring cones defines either a facet or a cut of the zonoid region. Thus, also in this case, the identification of the facet is based on the adjacency graph. Based on Corollary 2 we can generate an arbitrary facet as follows:

1. Choose an arbitrary set of d points.
2. Check whether this set defines a facet. If not, go back.
3. Create the corresponding facet.

We will also use this procedure to initialize our algorithm by creating a first facet. Thus, Theorem 2 and its corollaries provide a procedure for identifying each facet of the zonoid region.

7 Sequencing the facets

To complete the algorithm, we have to create a procedure that generates all facets in a sequential way. For this, we specify a total ordering of the set of facets. In the case $d = 2$ such an order is easily created by a circular sequence; see Dyckerhoff (2000). In the dimension $d \geq 3$ we can solve this problem by introducing a *spanning tree order (STO)*. Consider a given facet. Each ridge of it corresponds to exactly one neighboring facet and all neighboring facets can be found by passing through the ridges. The following theorem and its proof tell us the number of adjacent faces, that is the number of ridges, and how the ridges are obtained.

Theorem 3. (Neighboring facets)

(i) A facet has either $2d$ *or* d *neighboring facets.*

(ii) It has d *neighbors if and only if it is non-redundant.*

Proof. Let the support vector $\mathbf{p}$ be orthogonal to the facet and condition (14) be met. A minimal violation of orthogonality is achieved by an infinitesimal move of $\mathbf{p}$ in a direction that is perpendicular to a ridge of the facet and non-perpendicular to its other ridges. This corresponds to the following change in the first equation of (14): Either $\mathbf{p}'\mathbf{x}_{\pi_\mathbf{p}(k)} < \mathbf{p}'\mathbf{x}_{\pi_\mathbf{p}(k+1)} = \cdots = \mathbf{p}'\mathbf{x}_{\pi_\mathbf{p}(k+d-1)}$ or $\mathbf{p}'\mathbf{x}_{\pi_\mathbf{p}(k)} = \cdots = \mathbf{p}'\mathbf{x}_{\pi_\mathbf{p}(k+d-2)} < \mathbf{p}'\mathbf{x}_{\pi_\mathbf{p}(k+d-1)}$. That is, a ridge is obtained by removing one of the points that define the facet according to Theorem 2.

By removing one point to the higher and one to the lower part of the permutation, we obtain two ridges. Note that in the non-redundant case we can generate only one ridge in order not to violate the second equation of (14). The number of ridges is $2d$ (d in a non-redundant case), as there are d points which have to be removed. Obviously, the number of facets that neighbor the current one equals the number of its ridges. □

Thus, by removing one of the points that characterize the current facet according to the Theorem 2, we are directed to either one or two neighboring facets. If the current facet is redundant, removing a point yields two parallel ridges. Thus, we get two neighbors for a redundant, and one for a non-redundant facet.

Next, we search for a new adjacent facet that shares a given ridge. For this, rotate the support vector $\mathbf{p}$ in the plane orthogonal to the ridge defined by the $d-1$ points. Obviously, if $\mathbf{x}_i$ has been removed from the higher (resp. lower) part of the permutation, $\mathbf{p}$ has to be rotated in the direction of increasing (resp. decreasing) $\mathbf{p}'\mathbf{x}_i$. The rotation stops when the (14) first equation is met, that means, $\mathbf{p}$ has reached a normal of a new facet.

As this procedure produces a "jump" from the current facet to one of its neighbors, we shall mention it as the *"jump-to-neighbor" procedure*. According to this procedure, sequentially generated facets are identified in a similar way, which allows for an efficient implementation. Moreover, as the traversal through all neighbors guarantees the absence of "gaps", no facet will be lost.

Based on Theorem 3 we shall construct a special graph; the *facet traversal graph (FTG)*. The vertices of the FTG correspond to the set of all facets of the zonoid region, and the edges of the FTG indicate the neighborhood of facets. Each vertex of this graph joins either d or $2d$ edges.

A sequential procedure for determining the zonoid region consists in transvers-

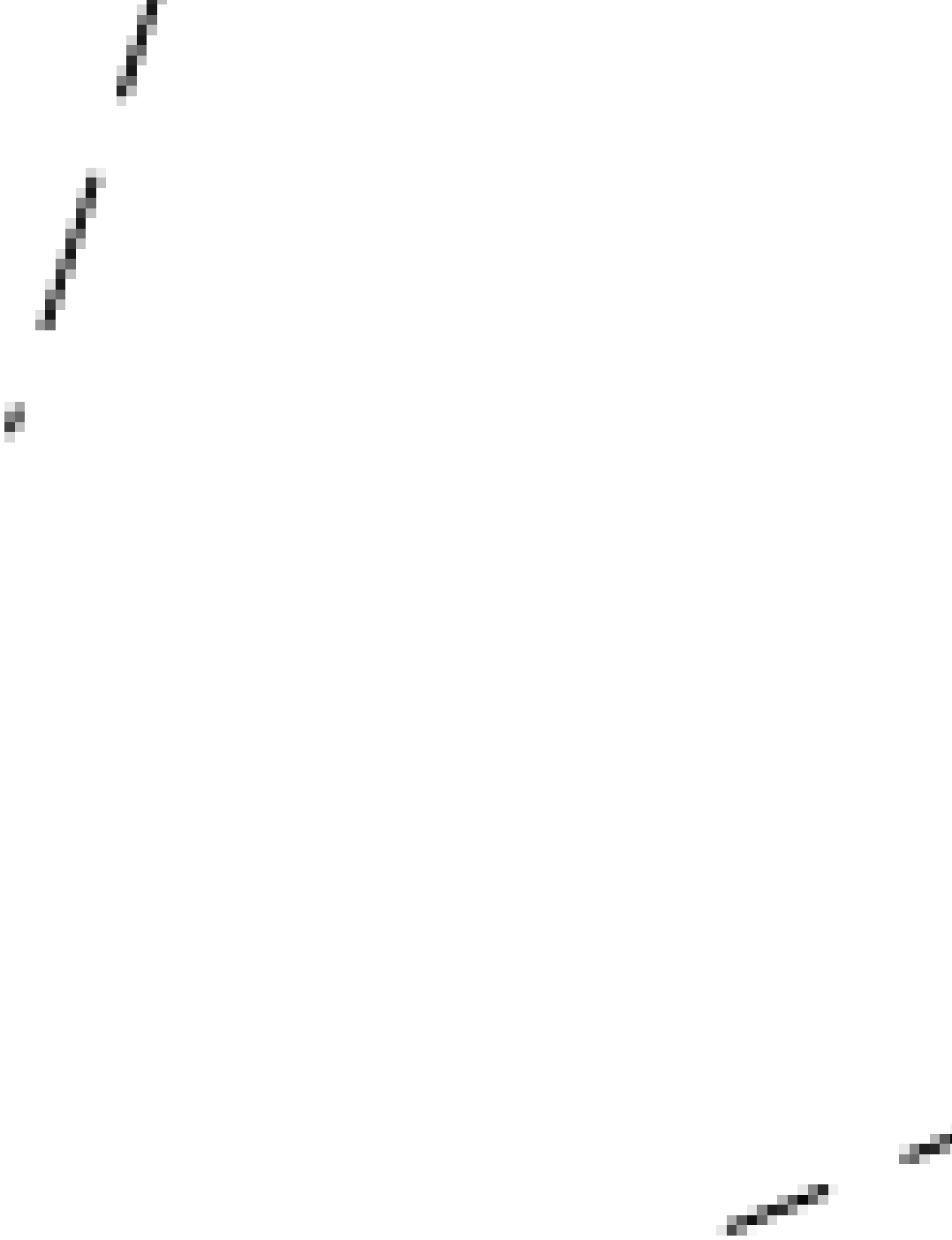

Figure 3: Examples of the facet traversal graph.

ing all vertices of the corresponding FTG. Note, that in the dimension $d = 2$, the FTG is an elementary cycle and its traversal is trivial and unique. In fact $d = 2$ is a degenerated case. For $d \geq 3$ we construct a spanning tree order (STO) of the FTG, which orders the set of facets. By "jumps-to-neighbor" the STO is created in a dynamic way:

1. Organize a queue.
2. In each step, pop from the queue a current facet, which corresponds to a vertex of the FTG. Add to the queue all adjacent vertices of the current vertex that have not been processed so far. Mark the current vertex as processed.
3. Marking of the vertices is done through a hash table, where hash codes of all the processed vertices (i.e. facets of the zonoid region) are stored.

According to Corollary 2 the record for each facet in the hash table is fully described by d integer numbers. These numbers are the labels of main elements of d points that define the facet. If these points define two parallel facets, these facets can easily be generated in one step, thus making it possible to have one record for them in the hash table.

The linear order of the vertices (facets) is provided by its final positioning in the queue. The realization of the STO is illustrated by *Figure 4*.

So far, we have constructed an algorithm to compute $D\alpha(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ for a single given $\alpha \in]0, 1[$. Finally, this procedure is modified to efficiently calculate $D^*_\alpha(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ for all α^* in an interval around α.

Given $\mathbf{p}_0$, consider the intervall A_k of those α whose main element has the same index k, that is $n - [n\alpha] = \pi_{\mathbf{p}_0}(k)$ or, equivalently,

$$\frac{n - \pi_{\mathbf{p}_0}(k)}{n} \leq \alpha < \frac{n - \pi_{\mathbf{p}_0}(k) + 1}{n}\,. \tag{16}$$

For all $\alpha \in A_k$, the global cone structure is the same. When we calculate the zonoid region for some $\alpha \in A_k$, we can simultaneously determine the zonoid regions for all $\alpha \in A_k$ by using the same global cone structure and only recalculating the distances of facets from the origin.

Note that that there are only n possible different global cone structures. Hence the global cone structure is fully determined by the initial position $\mathbf{p}_0$ of the support vector and the corresponding main point, which can be any of the n data points. Consequently, for determining the *whole family of zonoid regions*, $D\alpha(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ for all α, it suffices to run the modified algorithm n times only.

8 Discussion

An exact algorithm has been constructed to compute the zonoid regions of an empirical distribution in d-space. It calculates all of the vertices, edges, and facets of a zonoid region at any given depth $\alpha \in]0, 1[$. (Recall that $\alpha = 0$ and $\alpha = 1$ are trivial cases.) This approach requires that the dimension $d-2$ of ridges is not lower than 1, which is the dimension of edges. It works for any dimension $d \geq 3$ and any number n of data points.

A hash table plays a significant role, as it stores the vertices, once generated, in a special structure and facilitates a fast check of whether the vertex has already been processed. Each facet is generated only once. Thus the algorithm has as many loops as the zonoid region has facets. Obviously, this is the minimum number of facet generating loops in this sort of algorithm.

In a single facet generating loop, the most costly operations are as follows: Calculate the hyperplane equation of the current facet, calculate its distance from the origin and obtain the neighboring facets. This is done by solving linear equations and finding inner products only. The complexity of the first operation is $O(d^3)$. Up to d such operations are performed in each loop. The complexity of getting $n-1$ inner products is assessed $O(nd)$. Hence, the complexity of one facet generating loop is described by $O(d^2(d^2+n))$.

The number of computational loops of the algorithm is equal to the number of facets of the zonoid region. If the average number of facets is denoted $N(n, d)$, the average computational complexity of the algorithm amounts to $O(d^2(d^2+n) \cdot N(n, d))$.

Note that the complexity increases only moderately with d. For example, consider two data clouds of dimensions d_1 and d_2 ($d_1 < d_2$) that contain the same number of points n. Then the second data cloud will form a polytope

that has a more trivial structure in $\mathbb{R}^{d_2}$ than the first has in $\mathbb{R}^{d_1}$. It is also easy to see that there is no operation in the algorithm whose complexity grows exponentially with the growth of the dimension. Altogether the complexity is polynomial in n and d. This confirms the efficiency of our algorithm.

General memory resources are used, in the first place, for storing a hash table and created facets. Each facet occupies $O(d)$ storage size, while a hash table in almost any case has a constant size C, not depending on n and d. Therefore, the use of general memory is of the order $O(N(n,d) \cdot d + C)$. Facets, once they have been created, are put into a secondary store, thus considerably reducing the storage cost.

Figure 5 illustrates its application by exhibiting zonoid regions for a small data set of five points of the dimension three and for several values of α. Each zonoid region is depicted in three directions (by revolving it on a vertical axis). The data points are shown as little pyramids. (Note that these 3d-pictures employ a perspective view.)

The algorithm can be downloaded as an R-package from the first author's webpage (`www.wisostat.uni-koeln.de/Forschung/ZonoidRegions`). It has been implemented on a standard PC. Table 1 exhibits, for different choices of d and n, average values of total time (in seconds), number of facets, and time per facet. Note that for all calculations the same $\alpha = 0.317$ was taken. As the number of facets depends on the data, the efficiency of the algorithm may be judged by its computation time per facet. Table 1 gives an idea of how the time for computing one facet grows with d and n. It appears that the "time per facet" increases polynomially with d; moreover, the increase is close to being linear. Concerning n, a tendency towards saturation at some constant value is indicated. The given results also suggest that the growth in the "number of facets" is polynomial as well. Hence we can suspect that the aggregate complexity is polynomial.

Much computational load can be spared when we simultaneously calculate zonoid regions for several α that are sufficiently close to each other. If $[n \cdot \alpha_1] = \ldots = [n \cdot \alpha_k]$ holds, complete facets have to be computed for α_1 only, while for $\alpha_2, \ldots, \alpha_k$ all facets are parallel to them; so, only their distances from the origin have to be calculated.

The algorithm as it stands generates the facets one after the other in a deterministic way. It may be modified in order to gradually cover certain specified

d	n	time per facet	number of facets	total time [sec]
3	10	0.002437	106	0.259
3	15	0.003022	241	0.722
3	20	0.003474	476	1.653
4	10	0.003142	257	0.809
4	15	0.004444	948	4.216
4	20	0.005163	2840	14.667
5	10	0.003714	377	1.403
5	15	0.006250	2556	15.980
5	20	0.007257	13082	95.133
6	10	0.004083	377	1.542
6	15	0.008112	5005	40.600
6	20	0.009385	38177	356.648

Table 1: First computational results.

parts of the zonoid region that are of special interest. For instance, all facets belonging to the lower boundary may be constructed without generating the remaining facets of the zonoid region.

Moreover, the structure of the algorithm lends itself well to being parallelized on a high-performance cluster system.

To speed up the procedure, our exact algorithm can also be modified by imposing heuristic rules on the choice of adjacent facets. For instance, as a heuristic rule we may prefer redundant facets to non-redundant ones, since a redundant facet borders on more other facets and, thus, can be regarded as a facet that determines the zonoid region "more strictly" than others do. In any case, the exact algorithm serves as a benchmark procedure to be compared with any heuristic procedure, regarding precision as well as speed.

9 References

Aloupis, G. (2006). Geometric measures of data depth. In R. Y. Liu, R. Serfling, D. L. Souvaine, eds., *Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications, DIMACS Series in Discrete Mathematics and Theoretical Computer Science,* 147-158, American Mathematical Society.

Cascos, I. (2006). The expected convex hull trimmed regions of a sample. *Working Paper 06-69, Statistic and Econometric Series 19,* Departamento de Estadística, Universidad Carlos III de Madrid.

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.

Dyckerhoff, R. (2002). Inference based on data depth, Chapter 5 in: Mosler, K., Multivariate Dispersion, Central Regions and Depth. Springer, New York.

Dyckerhoff, R. (2004). Data depths satisfying the projection property. *AStA: Advances in Statistical Analysis,* 99, 163-190.

Dyckerhoff, R., Koshevoy, G. and Mosler, K. (1996). Zonoid data depth: Theory and computation. In A. Pratt, ed., COMPSTAT 1996, *Proceedings in Computational Statistics,* 235-240, Physica, Heidelberg.

Edelsbrunner, H. (1987). *Algorithms in Combinatorial Geometry.* Springer, Berlin.

Fukuda, K. and Rosta, V. (2004). Exact parallel algorithms for the location depth and the maximum feasible subsystem problems. In C.A. Floudas and P. M. Pardalos, eds., *Frontiers in Global Optimization,* 123-133, Kluwer Acad. Publ., Boston, MA.

Gigliarano, C. and Mosler, K. (200x). Constructing indices of multivariate polarization. To appear in: Journal of Economic Inequality.

Mosler, K. and Hoberg, R. (2006). Data analysis and classification with the zonoid depth. In R. Y. Liu, R. Serfling, D. L. Souvaine, eds., *Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications, DIMACS Series in Discrete Mathematics and Theoretical Computer Science,* 49-59, American Mathematical Society.

Koshevoy, G. and Mosler, K. (1997). Zonoid trimming for multivariate distributions. *Annals of Statistics,* 25, 1998-2017.

Koshevoy, G. and Mosler, K. (2007). Multivariate Lorenz dominance based on zonoids. *AStA: Advances in Statistical Analysis,* 91, 57-76.

Liu, R.Y., Parelius, J.M. and Singh, K. (1999). Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *Annals of Statistics,* 27, 783-858.

Lopez-Pintado, S. and Romo, J. (2007). Depth-based inference for functional data. *Computational Statistics and Data analysis* 51, 4957-4968.

Mosler, K. (2002). *Multivariate Dispersion, Central Regions and Depth.* Springer, New York.

Rousseeuw, P.J. and Ruts, I. (1996). Algorithm AS 307. Bivariate location depth. *Applied Statistics,* 45, 516-526.

Rousseeuw, P.J. and Struyf, A. (1998). Computing location depth and regression depth in higher dimensions. *Statistics and Computing,* 8, 193–203,

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

Serfling, R. (2006). Depth functions in nonparametric multivariate inference. In R. Y. Liu, R. Serfling, D. L. Souvaine, eds., *Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications,* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 2006, 1-16.

Serfling, R. and Zuo, Y. (2000). Structural properties and convergence results for contours of sample statistical depth functions. *Annals of Statistics,* 28, 483-499.

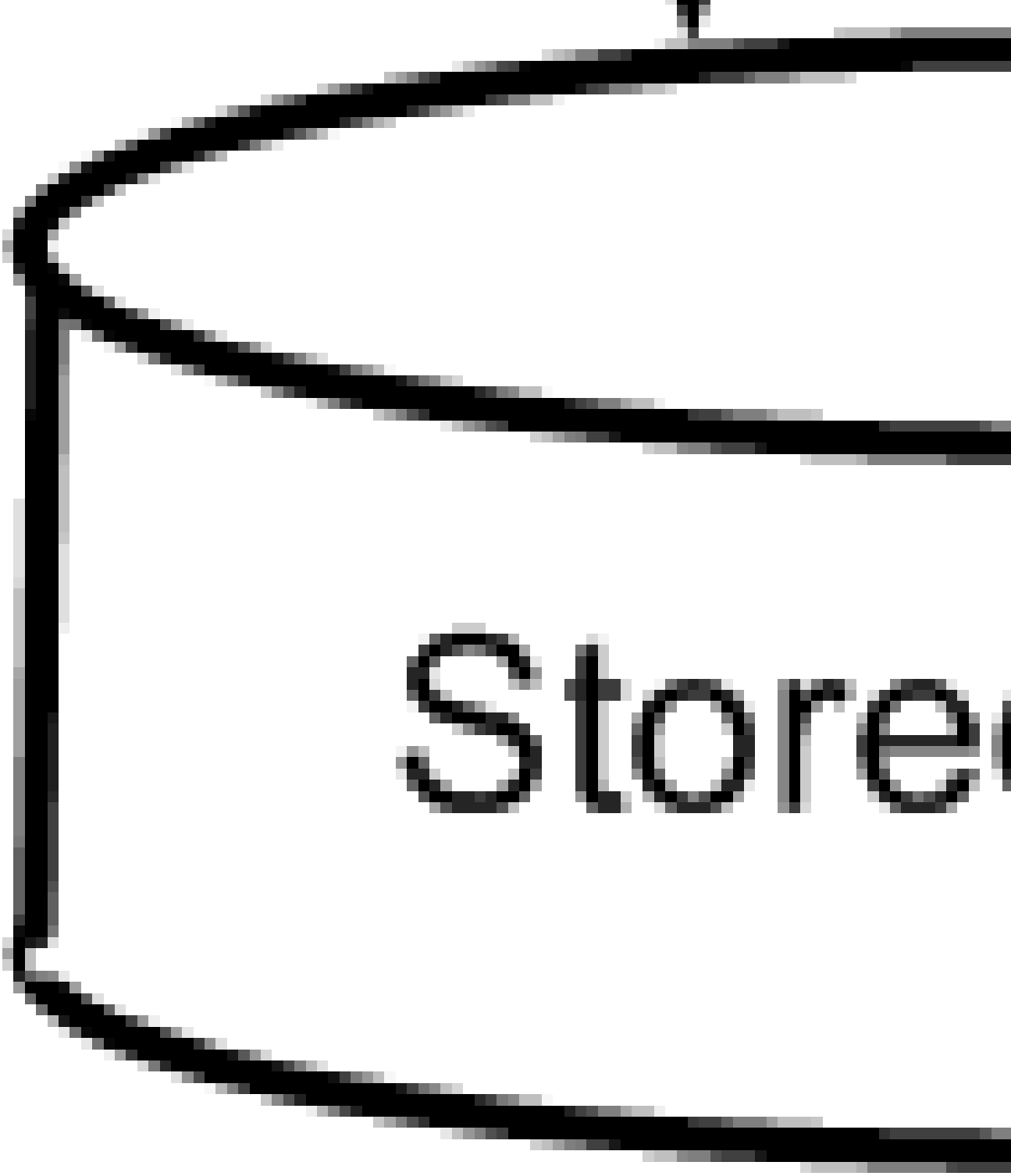

Figure 4: Realization of the STO.

Figure 5: Zonoid regions of five points of the dimension three.