ECONSTOR
Make Your Publications Visible.

A Service of
ZBW
Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Lange, Tatjana; Mosler, Karl; Mozharovskyi, Pavlo

Working Paper

Fast nonparametric classification based on data depth

Discussion Papers in Statistics and Econometrics, No. 1/12

Provided in Cooperation with:
University of Cologne, Institute of Econometrics and Statistics

Suggested Citation: Lange, Tatjana; Mosler, Karl; Mozharovskyi, Pavlo (2012) : Fast nonparametric classification based on data depth, Discussion Papers in Statistics and Econometrics, No. 1/12, University of Cologne, Seminar of Economic and Social Statistics, Cologne

This Version is available at:
https://hdl.handle.net/10419/67614

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

DISCUSSION PAPERS IN STATISTICS
AND ECONOMETRICS

SEMINAR OF ECONOMIC AND SOCIAL STATISTICS
UNIVERSITY OF COLOGNE

No. 1/12

Fast nonparametric classification based on data depth

by

Tatjana Lange
Karl Mosler
Pavlo Mozharovskyi

1st version
April 13, 2012

DISKUSSIONSBEITRÄGE ZUR
STATISTIK UND ÖKONOMETRIE

SEMINAR FÜR WIRTSCHAFTS- UND SOZIALSTATISTIK
UNIVERSITÄT ZU KÖLN

Albertus-Magnus-Platz, D-50923 Köln, Deutschland

DISCUSSION PAPERS IN STATISTICS
AND ECONOMETRICS

SEMINAR OF ECONOMIC AND SOCIAL STATISTICS
UNIVERSITY OF COLOGNE

No. 1/12

Fast nonparametric classification based on data depth

by

Tatjana Lange[1]
Karl Mosler[2]
Pavlo Mozharovskyi[3]

1st version
April 13, 2012

Abstract

A new procedure, called $DD\alpha$-procedure, is developed to solve the problem of classifying d-dimensional objects into $q \geq 2$ classes. The procedure is completely nonparametric; it uses q-dimensional depth plots and a very efficient algorithm for discrimination analysis in the depth space $[0,1]^q$. Specifically, the depth is the zonoid depth, and the algorithm is the α-procedure. In case of more than two classes several binary classifications are performed and a majority rule is applied. Special treatments are discussed for 'outsiders', that is, data having zero depth vector. The DDα-classifier is applied to simulated as well as real data, and the results are compared with those of similar procedures that have been recently proposed. In most cases the new procedure has comparable error rates, but is much faster than other classification approaches, including the SVM.

Keywords: Alpha-procedure, zonoid depth, DD-plot, pattern recognition, supervised learning, misclassification rate.

AMS Subject Classification: 62H30.

[1]Fachbereich Informatik und Kommunikationssysteme, Hochschule Merseburg, D-06217 Merseburg
[2]Seminar für Wirtschafts– und Sozialstatistik, Universität zu Köln, D-50923 Köln
[3]Seminar für Wirtschafts– und Sozialstatistik, Universität zu Köln, D-50923 Köln

1 Introduction

A steady interest in statistical learning theory has intensified recently since nonparametric tools have become available. A new impetus has been given to supervised classification by employing depth functions such as Tukey's ([22]) halfspace depth or Liu's ([17]) simplicial depth. In supervised learning a function is constructed from labeled training data that classifies an arbitrary data point by assigning it one of the labels [10]. Given two or more labeled clouds of training data in d-space, a data depth measures the centrality of a point with respect to these clouds. For any point in d-space it indicates the degree of closeness to each label. This can be employed in different ways for solving the classification task. Many authors have made use of data depth ideas in supervised classification. Liu et al. [18] were the first who stressed the usefulness and versatility of depth transformations in multivariate analysis. They introduced the notion of a DD-plot, that is the two-dimensional representation of multivariate objects by their data depths regarding two given distributions. In a straightforward way, an object can be classified to the class where it is deepest, that is, according to its maximum depth. Jornsten [13] and Ghosh and Chaudhuri [9] have followed this and similar approaches; see also Hoberg and Mosler [11]. Dutta and Ghosh [5, 4] employ a separator that is linear in a density based on kernel estimates of the projection depth, respectively L_p-depth. Recently, Li et al. [16] have used polynomial separators of the DD-plot to classify objects by their depth representation. These methods differ in the notion of depth used and allow for adaptive and other extensions.

The quoted literature has in common that a (possibly high-dimensional) space of objects is transformed into a lower-dimensional space of depth values of these objects and the classification task is performed in the depth space. In this context several questions arise:

1. Which particular notion of depth should be employed?
2. Which classification procedure should be applied to the depth-represented data?
3. How extends the procedure to $q > 2$ classes?

The above literature answers these questions in different ways. Ad (1), halfspace and simplicial depths, among others, have been employed in [8, 16, 18]. They depend only on the combinatorial structure of the data, being constant in the compartments spanned by them. Consequently, these depths are rather robust to outlying data, but calculating them in higher dimensions can be cumbersome if not impossible. On the other hand Mahalanobis depth [19], which has also been

used by these authors, is easily calculated but highly non-robust. Moreover, it depends on the first two moments only and does not reflect any asymmetries of the data. More robust forms of the Mahalanobis depth remain still insensitive to data asymmetries. L_1-depth as used in [13] has similar drawbacks. [4] employ L_p-depths, which are easily calculated if p is known, and choose p in an adaptive procedure; however the latter needs heavy computations. In [11] the maximum zonoid depth and a combination of it with the Mahalanobis depth are used; both can be efficiently calculated also in high dimensions but lack robustness. Ad (2), Li et al. [16] solve the classification problem of the DD-plot by designing a polynomial line that separates the unit square and provides a minimal average misclassification rate (AMR); the order (up to three) of the polynomial is selected by cross validation. Similarly, separators are determined in [5] and [4] by cross-validation.

Ad (3) with $q > 2$ classes a given point is usually classified in two steps according to majority rule: firstly $\binom{q}{2}$ classifications are performed that are restricted to pairs of classes in the object space, and secondly the point is assigned to that class where it was most often assigned in step 1.

In this paper, ad (1), we employ the zonoid depth [14, 20], as it can be efficiently calculated also in higher dimensions (up to $d = 20$ and more) and has excellent theoretical properties regarding continuity and statistical inference. However the zonoid depth has a low breakdown point. If, in a concrete application, robustness is an issue the data have to be preprocessed by some outlier detection procedure. Ad (2), for final classification in the depth space a variant of the α-procedure is employed. It operates simply and very efficiently on low-dimensional spaces like the depth spaces considered here. The α-procedure has been originally developed by Vasil'ev [24, 25] and Lange [26]. Ad (3) we employ DD-plots if there are two classes and q-dimensional depth plots if there are $q > 2$ classes. Assignment of a given point to a class is based on $\binom{q}{2}$ binary classifications in the q-dimensional depth space plus a majority rule. Note that in each binary classification the whole depth information regarding all q classes is used.

We call our approach the DDα-approach and apply it to simulated as well as real data. The results are contrasted with those obtained in [16], [5], and [4].

The contribution of this paper threefold. A classification procedure is proposed that

1. is efficiently computable for objects of higher dimensions,
2. employs a very fast classification procedure of the D-transformed data,
3. uses the full multivariate information when classifying into $q > 2$ classes,

The rest of the paper is organized as follows. Section 2 introduces the depth transform, which maps the data from d-dimensional object space to q-dimensional depth space, and provides a first discussion of the problem of 'outsiders', that are points having a vanishing depth vector. In Section 3 our modification of the α-procedure is presented in some detail. Section 4 provides a number of theoretical results regarding the behavior of the DDα-procedure on elliptical and mirror symmetric distributions. Section 5 contains extensive simulation results and comparisons. Calculations of real data benchmark examples are reported in Section 6 as well as a comparison of the DDα-procedure with the SVM approach. Section 7 concludes.

2 Depth transform

A data depth is a function that measures, in a certain sense, how close a given point $\mathbf{x}$ is located to the "center" of a finite set X in $\mathbb{R}^d$, that is, how "deep" it is in the set. More precisely, a data depth is a function

$$(\mathbf{x}, X) \mapsto D_X(\mathbf{x}) \in [0,1]\,, \quad \mathbf{x} \in \mathbb{R}^d\,, \quad X \subset \mathbb{R}^d\,,$$

that satisfies the following restrictions: affine invariant; upper semicontinuous in $\mathbf{x}$; quasiconcave in $\mathbf{x}$ (that is, having convex upper level sets), vanishing if $||\mathbf{x}|| \to \infty$. Sometimes two weaker restrictions are imposed: orthogonal invariant; decreasing on rays from a point of maximal depth (that is, starshapedness of the upper level sets). For surveys of these restrictions and many special notions of data depth, see e.g. [27, 20, 6, 21, 2].

Now, assume that data in $\mathbb{R}^d$ are to be classified into $q \geq 2$ classes and that $X_1, \ldots, X_q \subset \mathbb{R}^d$ are training sets for these classes each having finite size $n_j = |X_j|$. Let D be a data depth. The function $\mathbb{R}^d \to [0,1]^q$ mapping

$$\mathbf{x} \mapsto \mathbf{d} := (D_{X_1}(\mathbf{x}), \ldots, D_{X_q}(\mathbf{x})) \tag{1}$$

will be mentioned as a *depth representation*. Each object is represented by a vector whose q components indicate its depth or closeness regarding the q classes. In particular, the training sets $X_j \subset \mathbb{R}^d$ are transformed to sets in $[0,1]^q$ that represent the classes in the depth space. It should be noted that 'closeness' of points in the original space translates to 'closeness' of their representations. The classification problem then becomes one of partitioning the depth space $[0,1]^q$ into q parts.

A simple rule, e.g., is to classify a point to that class where it has the largest depth value; see [9, 13]. This means that the depth space decomposes into q compartments which are separated by (parts of) q bisecting hyperplanes. Maximum

depth classification is a linear rule. A nonlinear classification rule is used in Li et al. [16], who treat the case $q = 2$ by constructing a polynomial line up to degree 3 that separates the depth space $[0, 1]^2$; see also [5, 4].

With several important notions of data depth, $D_X(x)$ vanishes outside the convex hull of X. This is, e.g., the case with the halfspace, simplicial, and zonoid depths, but not with the Mahalanobis and L_p-depths. A point that is not within the convex hull of at least one training set then is mapped to the origin in the depth space. Such a point will be mentioned as an *outsider*. Of course, it can be neither regarded as correctly classified nor ignored. To classify this point we may consider three principal approaches, each allowing for several variants.

- Classify randomly, with probabilities equal to the expected proportions of origin of points to be classified.
- Use the k-nearest neighbors method with a properly chosen distance: Euclidean distance, L_p-distance, Mahalanobis distance with moment estimates, Mahalanobis distance with robust estimates (MCD, cf. e.g. [12]).
- Classify with maximum Mahalanobis depth (using moment estimates or MCD) or a with the maximum of another depth that is properly extended beyond the convex hull as e.g. in [11].

In the sequel we will use either random classification, k-nearest neighbors (with different distances), or maximum Mahalanobis depth (with moment and robust estimates).

3 The α-procedure

To separate the q classes in the multi-depth space we use the *α-procedure*, which has been developed by Vasil'ev [24, 25] and Lange [26], see also [15].

Let us first present the procedure in the case of $q = 2$ classes. As above consider two clouds of training data in $\mathbb{R}^d$, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_1}\}$ and $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_{n_2}\}$ and notate $\mathbf{x}_{n_1+m} = \mathbf{y}_m$, $m = 1, ..., n_2$. By calculating the depth of all $\mathbf{x}_i$ with respect to each of the two clouds, their depth representation, $(D_X(\mathbf{x}_i), D_Y(\mathbf{x}_i))$, is obtained, $i = 1, 2, \ldots, n_1 + n_2$. The set

$$\mathcal{D} = \{\mathbf{d}_i \in [0, 1]^2 | \mathbf{d}_i = (D_X(\mathbf{x}_i), D_Y(\mathbf{x}_i)), i = 1, \ldots, n_1 + n_2\}$$

is the DD-plot of the data ([18]).

We use a modified version of the α-procedure to construct a nonlinear separator in $[0, 1]^2$ that classifies the D-represented data points. The construction is based

on depth values and the products of depth values up to some degree p. For this, a linearized representation of the two classes in a depth feature space is produced as follows: If $p = 2$, say, we include the squared depth values and the product of the two depths as additional components of the representation, yielding an *extended D-representation* in $[0,1]^5$,

$$\mathbf{Z} = \{\mathbf{z}_i \quad | \quad \mathbf{z}_i = \big(D_X(\mathbf{x}_i), D_Y(\mathbf{x}_i), D_X(\mathbf{x}_i) \cdot D_Y(\mathbf{x}_i), D_X^2(\mathbf{x}_i), D_Y^2(\mathbf{x}_i)\big)\,, \\ i = 1, ..., n_1 + n_2\}\,.$$

Each element of the extended D-representation is mentioned as a *basic D-feature* and the space $[0,1]^r$ as the **feature space**. When the maximum exponent is $p \geq 1$, $\mathbf{z}_i$ is a vector in $\mathbb{R}^r$ having components

$$D_X(\mathbf{x}_i)^{k_\nu} \cdot D_Y(\mathbf{x}_i)^{\ell_\nu}, \quad \text{where } 1 \leq k_\nu + \ell_\nu \leq p\,, \quad \nu = 1, \ldots, r\,. \tag{2}$$

The number of basic D-features, that is the dimension of the feature space, equals $r = \binom{p+2}{2} - 1$, which is easily seen by induction. We index the basic D-features by ν and notate $\mathbf{z}_i = (z_{i\nu})_{\nu=1,\ldots,r}$.

The α-procedure now, in a stepwise way, performs linear discrimination in subspaces of the feature space. It is a bottom-up approach that successively builds new features from the basic D-features. In each step certain two-dimensional subspaces of $\mathbf{Z}$ are considered, and the projection of $\mathbf{Z}$ to each of these subspaces is separated by a straight discrimination line. Out of these subspaces the α-procedure selects a subspace whose discrimination line provides the least classification error. Clearly any discrimination line that separates the DD-plot must pass through the origin since $D_X(\mathbf{x}_i) = D_Y(\mathbf{x}_i) = 0$ implies that the point $\mathbf{x}_i$ cannot be classified to either of the two classes. The same must hold for all discrimination lines in subspaces of the extended depth space.

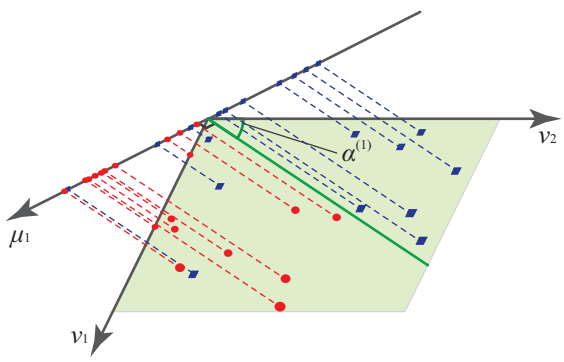

Figure 1: α-procedure; step 1.

In a *first step* a pair (ν_1, ν_2) of D-features (2) is chosen with $(k_1+k_2)(\ell_1+\ell_2) > 0$. The latter restriction implies that the two D-features do not solely relate to one of the classes. A straight discrimination line is calculated in the two-dimensional coordinate subspace defined by the pair (ν_1, ν_2). As the line passes through the origin it is characterized by an angle $\alpha \in [0, 2\pi[$. The best discriminating angle α_{ν_1,ν_2} is determined by minimizing the *average misclassification rate (AMR)*,

$$\begin{aligned}\Delta(\alpha; \nu_1, \nu_2) \;=\; & \frac{1}{n_1+n_2}\Big[\sum_{i=1}^{n_1} I(z_{i,\nu_1}\cos\alpha - z_{i,\nu_2}\sin\alpha < 0) \\ & + \sum_{i=n_1+1}^{n_1+n_2} I(z_{i,\nu_1}\cos\alpha - z_{i,\nu_2}\sin\alpha > 0)\Big]\,. \end{aligned} \tag{3}$$

Here $I(A)$ denotes the indicator function of A. If the minimum is attained in an interval, its middle value is selected for α_{ν_1,ν_2}; see Figure 1. The same is done for all pairs of D-features satisfying the above restriction, and the pair (ν_1^*, ν_2^*) is selected that minimizes (3). If the minimum is not unique the pair with the smallest k and ℓ is chosen. Let $\alpha^{(1)} = \alpha_{\nu_1^*,\nu_2^*}$ and denote the respective AMR by $\Delta^{(1)}$. Next the D-features ν_1^* and ν_2^* are replaced by a new D-feature which is indexed by μ_1 and gives value

$$z_{i,\mu_1} = z_{i,\nu_1}\cos\alpha^{(1)} - z_{i,\nu_2}\sin\alpha^{(1)}\,, \quad i = 1, \ldots n_1+n_2\,, \tag{4}$$

to each $\mathbf{x}_i$. Geometrically the values are obtained by projecting $(z_{i,\nu_1}, z_{i,\nu_2})$ to a straight line in the (ν_1, ν_2)-plane that is perpendicular to the discrimination line; see Figure 1. The first step results in the new D-feature μ_1 and the AMR $\Delta^{(1)}$ produced by classifying according to this feature.

The *second step* couples the new D-feature μ_1 with each of the basic D-features ν that have not been replaced so far. For each of these pairs of D-features a best discriminating angle $\alpha_{\mu_1,\nu}$ is determined, and among these the pair of D-features is selected that provides the minimum AMR. The minimum error is denoted by $\Delta^{(2)}$ and the angle at which it is attained by $\alpha^{(2)}$. This is visualized in Figure 2. The best pair of D-features is replaced by a new D-feature μ_2, where the values z_{i,μ_2} are calculated as in (4).

The last step is repeated with μ_2 in place of μ_1, etc. The procedure stops after step t if either the additional discriminating power $\Delta^{(t)} - \Delta^{(t+1)} = 0$ or $t = r$, that is, all basic D-features have been replaced. Then the angle $\alpha^{(t)}$ defines a linear rule for discriminating between two (up to) p-th order polynomials in $D_X(\mathbf{z})$ and $D_Y(\mathbf{z})$, which correspond to the two finally constructed D-features, according to their sign. This yields a polynomial separation of the classes in the depth space.

For example, let in step 1 the basic features D_X and D_Y^2 be selected and, consequently, $D_X \cdot D_Y$ and D_X^2 be included in steps 2 and 3. If the procedure terminates

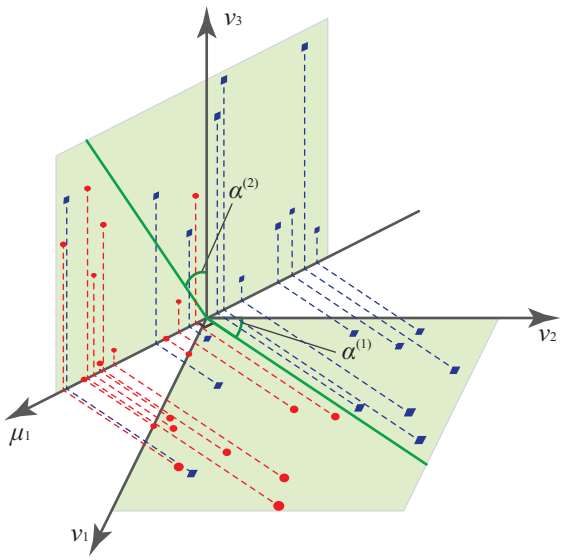

Figure 2: α-procedure; step 2.

after step 3, the result is a polynomial in the two depths $D_X(\mathbf{x})$ and $D_Y(\mathbf{x})$ that has form

$$aD_X(\mathbf{x}) + bD_X^2(\mathbf{x}) + cD_Y^2(\mathbf{x}) + dD_X(\mathbf{x})D_Y(\mathbf{x})$$

A given point $\mathbf{x}$ of the object space then is classified according to the sign of the polynomial.

If there are more than two classes, say $X_1, \ldots, X_q$, each data point $\mathbf{x}_i$ is represented by the vector of depth values $\mathbf{d} = (D_{X_1}(\mathbf{x_i}), \ldots, D_{X_q}(\mathbf{x}_i))$ in $[0,1]^q$. Again a depth feature space is considered of some order p; it has dimension $r = \binom{p+q}{q} - 1$. With $q > 2$ classes every two training classes $X_j, X_k, j \neq k$, are separated by the α-procedure in the same way as above: In each step a pair of D-features is replaced by a new D-feature as long as the AMR decreases and basic D-features are left to be replaced. For each pair of classes the procedure results in a hypersurface that separates the q-dimensional depth space into two sets of attraction. A given point $\mathbf{x}$ is finally assigned to that class to which it has been most often attracted.

4 Some theoretical aspects

In order to investigate some properties of the DDα-approach we transfer it to a more general probabilistic setting and define a depth function as the population version of a data depth. Let $\mathcal{P}$ be a properly chosen set of probability distributions on $\mathbb{R}^d$ that includes the empirical distributions. A *depth function* D is a function

that assigns a value $D_P(\mathbf{x}) \in [\mathbf{0}, \mathbf{1}]$ to every $\mathbf{x} \in \mathbb{R}^d$ and $P \in \mathcal{P}$ in an affine invariant way and has convex compact upper level sets. Obviously, the restriction of a depth function D to the class of empirical distributions is an affine invariant quasiconvex data depth. For details on general depth functions, see e.g. the above cited surveys [2, 20, 21, 27].

While data depth is an intrinsically nonparametric notion, the behavior of depth functions and depth based procedures on parametric classes is of special interest as it indicates how the nonparametric approach relates to the more classical parametric one. As a generalization of multivariate Gaussian distributions, spherical and elliptical distributions play an important role in parametric multivariate analysis. A random vector $\mathbf{X}$ in $\mathbb{R}^d$ has a *spherical distribution* if $\mathbf{X} = R \cdot \mathbf{U}$, where $\mathbf{U}$ is a random vector uniformly distributed on the sphere S^{d-1} and R is a random variable having support $[0, \infty[$ and being independent of $\mathbf{U}$. A random vector $\mathbf{Y}$ has an *elliptical distribution* if it is an affine transform of a spherically distributed $\mathbf{X}$, $\mathbf{Y} = \mu + B\mathbf{X}$. If R has a density r we notate $\mathbf{Y} \sim \mathrm{Ell}(\mu, BB', r)$. As, by definition, a depth function is affine invariant, it operates on elliptical distributions in a rather simple way. The following propositions give some insight into the the behavior of depth functions and the DDα-procedure if the data generating processes are elliptical.

Proposition 4.1 *If D is an affine invariant depth function and P an elliptical distribution, then for every $\alpha \in]0, 1]$ the upper level set*

$$D_\alpha(P) = \{\mathbf{x} \in \mathbb{R}^d | D_P(\mathbf{x}) \geq \alpha\}$$

is an ellipsoid.

Proof. Let $P = \mathrm{Ell}(\mu, BB', r)$ and $\alpha \in]0, 1]$. Consider $P_0 = \mathrm{Ell}(\vec{0}, I_d, r)$. Then, for all $\beta \geq \alpha$, $\{\mathbf{x} \in \mathbb{R}^d | D_P(\mathbf{x}) = \beta\}$ is a sphere since D is, in particular, orthogonal invariant. Hence, $D_\alpha(P_0) = \{\mathbf{x} \in \mathbb{R}^d | D_{P_0}(\mathbf{x}) \geq \alpha\}$ is a ball and, by affine transformation with μ and B, $D_\alpha(P)$ is an ellipsoid. $\square$

Proposition 4.2 **(i)** *Let D be the zonoid depth and P a unimodal elliptical distribution, that is $P = \mathrm{Ell}(\mu, BB', r)$. Then for every non-empty density level set $\{\mathbf{x} \in \mathbb{R}^d | f(\mathbf{x}) \geq \beta\}$ exists some $\alpha = \phi(\beta)$ such that*

$$\{\mathbf{x} \in \mathbb{R}^d | f(\mathbf{x}) \geq \beta\} = D_\alpha(P)\,.$$

(ii) *If, in addition, r has an interval support then ϕ is a continuous, strictly increasing function. It holds $D_P(\mathbf{x}) = \phi(f(\mathbf{x}))$ and therefore*

$$f(\mathbf{x}) \geq f(\mathbf{y}) \quad \Longleftrightarrow \quad D_P(\mathbf{x}) \geq D_P(\mathbf{y})\,. \tag{5}$$

Proof. (i): Note that $D_0 = \mathbb{R}^d$. Thus, if $\beta \leq 0$, the claim holds with $\alpha = 0$. Now let $\beta > 0$ and assume w.l.o.g. that P is spherical. Then $\{\mathbf{x} \in \mathbb{R}^d | f(\mathbf{x}) \geq \beta\}$ is a ball with center at the origin. Let $\mathbf{x}^*$ be a point on its surface. Also the central regions D_α are balls around the origin. By Theorems 3.9 and 3.14 in [20], the D_α are continuous and strictly decreasing on the convex hull of the support of P and it holds $\alpha^* := D_P(\mathbf{x}^*) > 0$. We conclude $D_{\alpha^*} = \{\mathbf{x} \in \mathbb{R}^d | f(\mathbf{x}) \geq \beta\}$.
(ii): Under the additional premise, the density level sets are continuously and strictly decreasing in $\beta > 0$, which yields the result. □

Corollary 4.1 *Consider a mixture of unimodal elliptical distributions* $P_j = \mathrm{Ell}(\mu_j, B_j B_j', r_j)$, $j = 1, \ldots, q$, *with mixing probabilities* π_j *and assume that all* r_j *have an interval support. Let* D *be the zonoid depth.*

Then, for each j *and* k *exists a strictly increasing function* ψ_{jk} *so that*

$$\pi_j \cdot f_j(\mathbf{x}) < \pi_k \cdot f_k(\mathbf{x}) \quad \Longleftrightarrow \quad D_{P_j}(\mathbf{x}) < \psi_{jk}(D_{P_k}(\mathbf{x}))\,.$$

Proof. From Proposition 4.2 continuous and strictly increasing functions ϕ_j and ϕ_k are obtained with $D_{P_j}(\mathbf{x}) = \phi_j(f_j(\mathbf{x}))$ and $D_{P_k}(\mathbf{x}) = \phi_k(f_k(\mathbf{x}))$. Consequently,

$$\pi_j \cdot f_j(\mathbf{x}) < \pi_k \cdot f_k(\mathbf{x}) \quad \Leftrightarrow \quad D_{X_j}(\mathbf{x}) < \phi_j\left(\frac{\pi_k}{\pi_j}\,\phi_k^{-1}(D_{X_k}(\mathbf{x}))\right)\,,$$

which proves the claim by use of the function $\psi_{jk}(\cdot) = \phi_j\left(\frac{\pi_k}{\pi_j}\,\phi_k^{-1}(\cdot)\right)$. □

A similar result holds for other data depths including the halfspace, simplicial, projection and Mahalanobis depths; see Prop. 1 in [16]. In the rest of section we consider the limit behavior of the DDα-procedure under independent sampling. For this, we assume that the empirical depth is a consistent estimator of its population version. This is particularly true for the zonoid, halfspace, simplicial, projection and Mahalanobis depths.

Theorem 4.1 (Bayes rule) *Let* F *and* G *be probability distributions in* $\mathbb{R}^d$ *and* H *be a hyperplane* H *such that* G *is the mirror image of* F *with respect to* H*. Then based on a 50:50 independent sample from* F *and* G *the DDα-procedure will asymptotically yield the linear separator that corresponds to the bisecting line of the DD-plot.*

Note that the rule given in the theorem corresponds the Bayes rule, see [10].

Proof. Due to the mirror symmetry of the distributions in $\mathbb{R}^d$ the DD-plot is symmetric as well. Symmetry axis is the bisector, which is obviously the result of the α-procedure when the sample is large enough. □

Theorem 4.2 *Let F, G be elliptical, $F = \mathrm{Ell}(\mu_F, BB', r)$, $G = \mathrm{Ell}(\mu_G, BB', r)$. Then based on a 50:50 independent sample from F and G the DDα-procedure will asymptotically yield the linear separator that corresponds to the bisecting line of the DD-plot.*

Proof. If F and G are spherically symmetric, they satisfy the premise of the previous theorem. A common affine transformation of F and G does not change the DD-plot. □

5 Simulation study

The DDα-procedure has been implemented on a standard PC in an R-environment. To explore its specific potencies we apply it to simulated as well as to real data. The same data have been analyzed with several classifiers in the literature. In this section results on simulated data are presented regarding the average misclassification rate of nine procedures besides the DDα-classifier (Section 5.1). Then the speed of the DDα-procedure is quantified (Section 5.2). The following Section 6 covers the relative performance of the the DDα- and other classifiers on several benchmark data sets.

5.1 Comparison of performance

To simplify the comparison with known classifiers, we use the same simulation settings as in [16]. These are supervised classification tasks with two equally sized training classes. Data are generated by ten pairs of distributions according to Table 1. Here N and Exp denote the Gaussian and exponentional distributions, respectively, and

$$\mathrm{MixN}(\mu, \sigma_1, \sigma_2) = \begin{cases} -\sigma_1 * |\mathrm{N}(0,1)| + \mu & \text{with probability } 1/2, \\ \sigma_2 * |\mathrm{N}(0,1)| + \mu & \text{with probability } 1/2. \end{cases}$$

The DDα-classifier is contrasted with the following nine classifiers: linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-nearest neighbors classification (k-NN), maximum depth classification based on Mahalanobis (MM), simplicial (MS), and halfspace (MH) depth, and DD-classification with the same depths (DM, DS and DH, correspondingly). For more details about the data and the procedures as well as for some motivation the reader is referred to [16].

All simulations of [16] are recalculated following their paper as close as possible. The LDA, QDA and k-NN classifiers are computed with the R-packages "MASS"

Table 1: Distributional settings used in the simulation study.

Alternative	1-st class	2-nd class
Normal location	$\mathrm{N}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{N}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$
Normal location-scale	$\mathrm{N}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{N}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}4&4\\4&16\end{smallmatrix}\right])$
Cauchy location	$\mathrm{Cauchy}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{Cauchy}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$
Cauchy location-scale	$\mathrm{Cauchy}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{Cauchy}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}4&4\\4&16\end{smallmatrix}\right])$
Normal contaminated location	Learning sample: 90% as s1, 10% from $\mathrm{N}(\left[\begin{smallmatrix}10\\10\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$. Testing sample: as s1	as s1
Normal contaminated location-scale	Learning sample: 90% as s2, 10% from $\mathrm{N}(\left[\begin{smallmatrix}10\\10\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$. Testing sample: as s2	as s2
Exponential location	$(\mathrm{Exp}(1), \mathrm{Exp}(1))$	$(\mathrm{Exp}(1)+1, \mathrm{Exp}(1)+1)$
Exponential location-scale	$(\mathrm{Exp}(1), \mathrm{Exp}(1/2))$	$(\mathrm{Exp}(1/2)+1, \mathrm{Exp}(1)+1)$
Asymmetric location	$(\mathrm{MixN}(0;1,2), \mathrm{MixN}(0;1,4))$	$(\mathrm{MixN}(1;1,2), \mathrm{MixN}(1;1,4))$
Normal-exponential	$\mathrm{N}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right])$	$(\mathrm{Exp}(1), \mathrm{Exp}(1))$

and "class", where the parameter k of the k-NN-classifier is selected by leave-one-out cross-validation over a relatively wide range. The simplicial, and halfspace depths have been determined by exact calculations with the R-package "depth". The zonoid depth has been exactly computed by the algorithm in [7]. Recall that in dimension two calculations of all these depths can be efficiently done by a circular sequence and note that the problem of prior probabilities is avoided by choosing test samples of equal size from both classes.

For the DD-classifiers a polynomial line (up to degree three) is determined to discriminate in the two-dimensional DD-Plot, a tenfold cross-validation is employed to choose the optimal degree of the polynomial, a smoothing constant t=100 is selected in the logistic function, and the DD-Plot is never rotated. Each experiment includes a training phase and an evaluation phase: From the given pair of distributions 400 observations (200 of each class) are generated to train the classifier, and 1000 (500 of each) observations to evaluate its AMR. For each distribution pair and each classifier 100 experiments are performed, and the resulting sample of AMRs is visualized as a box-plot; see Figures 3 to 7.

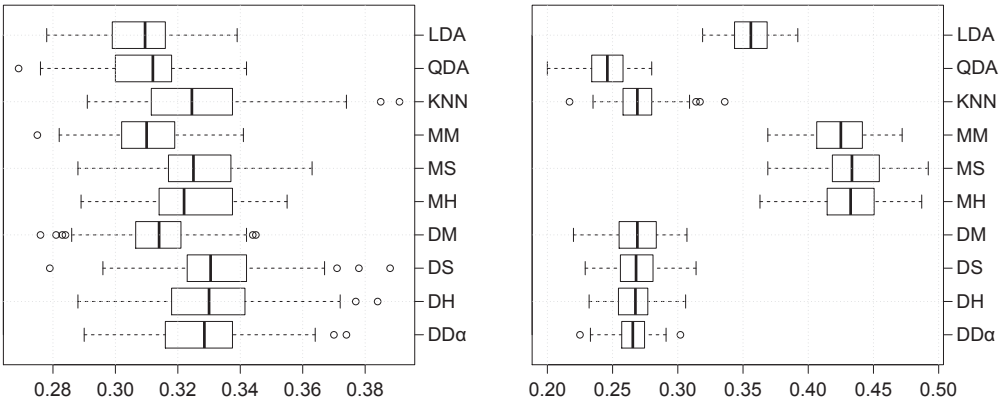

Figure 3: Normal location (left) and location-scale (right) alternatives.

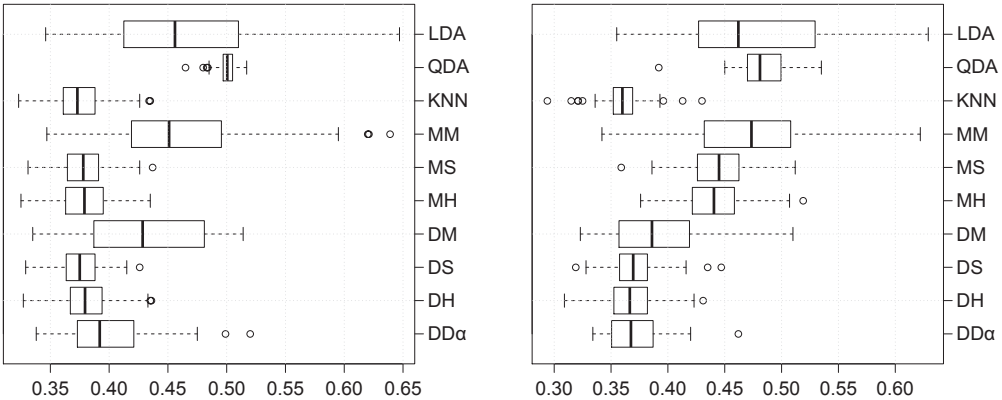

Figure 4: Cauchy location (left) and location-scale (right) alternatives.

As we have discussed at the end of Section 2, with depths like the simplicial, halfspace and zonoid depth the problem of outsiders arises. An outsider is, in the DD-plot, represented by the origin. A simple approach is to assign the outsiders randomly to the two classes. Throughout our simulation study we have chosen the random assignment rule, which results in kind of worst case AMR. Observe that this choice of assignment rule discriminates against the procedures that yield outsiders and advantages those that do not, in particular LDA, QDA, MM, DM and k-NN for all distribution settings.

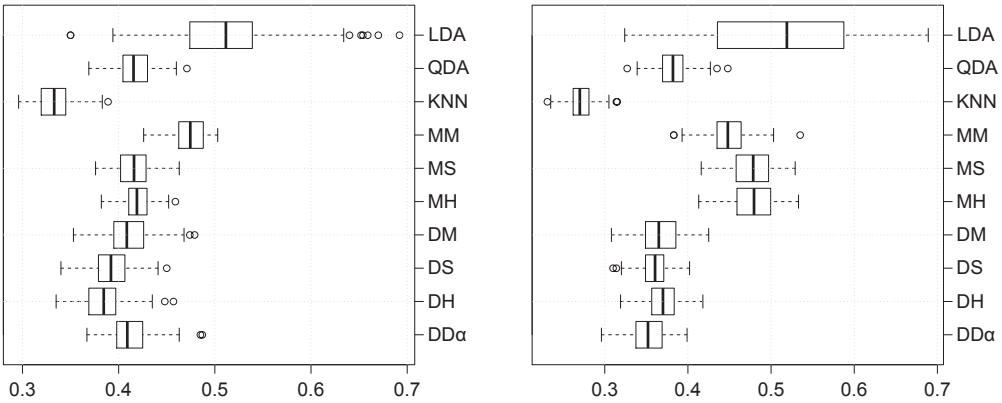

Figure 5: Normal contaminated location (left) and location-scale (right) alternatives.

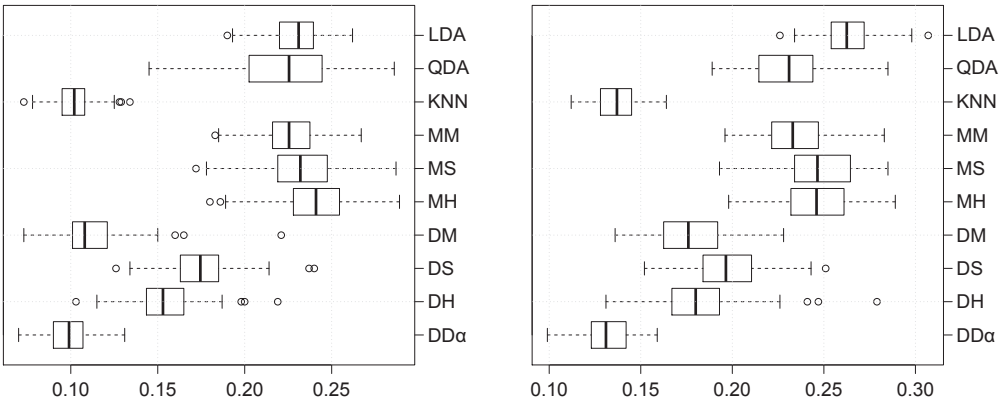

Figure 6: Exponential location (left) and location-scale (right) alternatives.

The principal results of the simulation study are collected in Figures 3 to 7. Under the normal location-shift model (Figure 3, left) all classifiers behave satisfactorily, and the DDα-classifier performs well among them. However LDA, QDA, MM and DM show slightly better results since they do not have to cope with outsiders like the other depth-based procedures.

Also under the normal location-scale alternative (Figure 3, right) the DDα-classifier performs rather well, like all DD-classifiers. A slightly worse perfor-

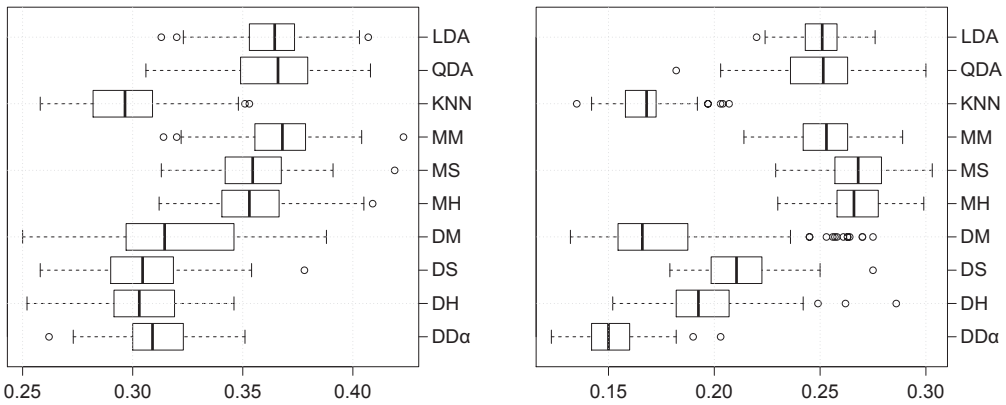

Figure 7: Asymmetric location (left) and normal-exponential (right) alternatives.

mance of the DDα-classifier is observed when discriminating the Cauchy location alternative (Figure 4, left), but it is still close to the DD-classifiers. This can be attributed to the lower robustness of the zonoid depth. However, when scaling enters the game (Cauchy location-scale alternative, Figure 4, right), the DDα-classifier again performs quite satisfactorily. The same picture arises when considering contaminated normal settings (Figure 5, left and right). Under a location alternative, the DDα-classifier is a bit worse than the DD-classifiers, while it slightly outperforms them in a location-scale setting.

The relative robustness of the DDα-classifier may be explained by two of its features: First it maps the original data points to a compact set, the q-dimensional unit hypercube. Second, for classification in the unit hypercube it employs the α-procedure, which, by choosing a median angle in each step, is rather insensitive to outliers.

Under exponential alternatives (Figure 6, left and right) the DDα-classifier shows excellent performance, which is even similar to that of the the k-NN for both location and location-scale alternatives. Its results for the asymmetric location alternative (Figure 7, left) are somewhat ambiguous, though still close to those of the DD-classifiers. Concerning the normal-exponential alternative (Figure 7, right) the DDα-classifier performs distinctly better than the others considered here.

On the basis of the simulation study we conclude: The DDα-classifier (1) performs quite well under various settings of elliptically distributed alternatives, it (2) is rather robust to outlier prone data, and (3) shows a distinctly good behavior under the asymmetrically distributed alternatives considered and when the two

classes originate from different families of distributions.

5.2 Speed of the DDα-procedure

To estimate the speed of the DDα-classification we have quantified the total time of training and classification times under two simulation settings, a shift and and a location-shift alternative concerning d-variate normals (see Table 2, header), with various values of dimension d and of total size of training classes n. An experiment consists of a training phase based on two samples (each of size $n/2$) and an evaluation phase, where 2500 points (1250 from each distribution) are classified. Each experiment is performed 100 times, then the average computation time is determined. All these computations have been conducted on a single kernel of the processor Core i7-2600 (3.4 GHz) having enough physical memory.

Table 2 exhibits the average computation times (in seconds, with the standard deviations in parentheses) under the two distributional settings and for different d and n. As it is seen from the table, the DDα-classifier is a very fast, in the learning phase as well as in classifying high amounts of data. However, computation times increase considerably with the number of training points, which is due to the many calculations of zonoid depth needed. With dimension d computation time grows slower, which may be explained as follows. With increasing dimension of the data space, more points come to lie on the convex hull (thus having depth $= 2/n$) or outside it (thus having depth $= 0$). The algorithm from [7] computes the depth of such points much faster than that of points having larger depths.

6 Benchmark studies

Concerning real data, we take benchmark examples from [16, 5, 4] to compare the performance of the DDα-classifier with respect to AMR (Section 6.1). In addition we use four real data sets from from the UCI machine learning repository [1] to contrast the DDα-classifier with the support vector machine (SVM) of [23] regarding both performance and time (Section 6.2).

6.1 Benchmark comparisons with nonparametric classifiers

As our benchmark examples are well known, we refer to the literature for their detailed description and restrict ourselves to mentioning the dimension d, the number of classes q, the total number of points in the training classes (# train), and the number of points classified (# class); see Table 3.

Table 2: Computing times of DDα-classification, in seconds.

	N($\mathbf{0}_d, \mathbf{I}_d$) N($0.25 \cdot \mathbf{1}_d, \mathbf{I}_d$)			
	$d = 5$	$d = 10$	$d = 15$	$d = 20$
$n = 200$	0.14 (0.00014)	1.55 (0.00014)	1.89 (-)	2.24 (-)
$n = 500$	1.04 (0.00046)	10.37 (0.00052)	12.58 (0.00062)	14.14 (-)
$n = 1000$	5.33 (0.0012)	42.54 (0.0014)	53.66 (0.0017)	59.18 (-)
	N($\mathbf{0}_d, \mathbf{I}_d$) N($(0.25\ \mathbf{0}'_{d-1})', 5 \cdot \mathbf{I}_d$)			
	$d = 5$	$d = 10$	$d = 15$	$d = 20$
$n = 200$	0.15 (0.00014)	1.62 (0.00016)	1.94 (0.00021)	2.2 (0.00027)
$n = 500$	1.09 (0.00044)	11.33 (0.00059)	14.44 (0.00079)	15.18 (0.0010)
$n = 1000$	5.24 (0.0011)	47.63 (0.0016)	67.22 (0.0022)	74.15 (0.0026)

Table 3: Overview of benchmark examples; dimension (d), classes (q), training points (# train), classified points (# class), total data (# data).

N	Dataset	Results	q	d	# train	# class	# data
1	Biomedical	Tables 5, 4	2	4	150	44	194
		Table 6	2	4	100	94	194
2	Blood Transfusion	Table 6	2	3	374	374	748
		Table 4	2	3	500	248	748
3	Diabetes (1)	Table 6	3	5	100	45	145
4	Diabetes (2)	Table 7	2	8	767	1	768
5	Ecoli	Table 7	3	7	271	1	272
6	Glass	Tables 5, 6	2	5	100	46	146
		Table 7	2	9	145	1	146
7	Hemophilia	Table 6	2	2	50	25	75
8	Image Segmentation	Table 4	2	10	500	160	660
9	Iris	Table 7	3	4	149	1	150
10	Synthetic	Tables 5, 6	2	2	250	1000	1250

Tables 4, 5 and 6 exhibit the performance (in terms of AMR, with standard errors in parentheses) of the DDα-classifier together with the performance of the different classifiers investigated in [16], [5] and [4] and based on the respective

benchmark data. When applying the DDα-classifier an auxiliary procedure has to be chosen by which outsiders are treated. In our benchmark study we employ several such procedures.

Table 4: Benchmark performance with DD- and other classifiers.

Dataset	LDA	QDA	k-NN	MM	MH	DM	DH	DDα
Biomedical	17.05 (0.49)	13.05 (0.38)	14.32 (0.45)	27.14 (0.6)	18.00 (0.49)	12.25 (0.4)	17.48 (0.51)	24.59 (0.63)
Blood Transfusion	29.49 (0.08)	29.11 (0.13)	29.74 (0.13)	32.56 (0.29)	30.47 (0.3)	26.82 (0.19)	28.26 (0.19)	32.27 (0.25)
Image Segmentation	8.17 (0.2)	9.44 (0.19)	5.59 (0.19)	9.12 (0.23)	11.87 (0.25)	9.54 (0.2)	13.98 (0.29)	43.58 (0.34)

In Table 4 the DDα-procedure is contrasted with the real data results in [16]. Here we use the same settings as in Section 5.1 and classify the outsiders on a random basis. All results in Table 4 have been recalculated.

As we see from the Table, the performance of our new classifier is mostly worse than the classifiers considered in [16]. Only in the Blood Transfusion case the AMR has comparable size. However, in this comparison the eventual presence and treatment of outsiders plays a decisive role. Observe that [16] in their procedures MH and DH use the random Tukey depth [3] to approximate the halfspace depth of a data point in dimension three and more. But the random Tukey depth generally overestimates the halfspace depth so that some of the outsiders remain undetected. This implies that, in the procedures MH and DH, considerably fewer points (we observed around 16%, 4% and 11% correspondingly) are treated as outsiders and assigned on a random basis.

In fact, as exactly determined by calculating the zonoid depth, the rate of outsiders in the Biomedical Data (with $d = 4$) totals some 35%, in the Blood Transfusion Data ($d = 3$) about 11%, and in the Image Segmentation Data with $d = 10$ about 86%. This is in line with our expectation: the higher the dimension of the data the higher is the outsider rate. In contrast to the MH and DH procedures, the DDα-procedure detects all outsiders and, in the comparison of Table 4, assigns them randomly. Obviously the performance of the latter can be improved with a proper non-random procedure of outsider assignment. In the subsequent benchmark comparisons several such procedures of non-random outsider assignment are included.

Dutta and Ghosh [5] introduce classification based on projection depth and compare it with several variants of the maximum-Mahalanobis-depth (MD)classifier. The same authors [4] propose an L_p-depth classifier (with optimized p) and contrast it with two types of MD. To compare the DDα-classifier on a par with [5, 4] we implement the following rules for handling outsiders: First, k-nearest-neighbor

rules are used with various k and either Euclidean or Mahalanobis distance, the latter with moment or, alternatively, MCD estimates. Second, maximum Mahalanobis depth is employed, again based on moment or MCD estimation. As the k-NN results of the benchmark examples do not vary much with k, we restrict to $k = 1$. (However, the performance of the classifiers can be improved by an additional cross-validation over k.) Consequently, five different rules for treating outsiders remain for comparison. Tables 5 and 6 exhibit the performance of the DDα-classifier *vs.* the projection-depth classifiers of [5] and the L_p-depth classifiers of [4], respectively, regarding the benchmark examples investigated in these papers. The last five columns of Table 5 and the bottom part of Table 6 report the AMR (standard deviations in parentheses) of the DDα-classifier when one of the five outsiders treatments is chosen. The remaining columns are adopted as they stand in [5] and [4].

Table 5: Benchmark comparison with projection depth classifiers.

Dataset	MD (SS)	MD (MS)	$MD_{\frac{3}{4}}$ (SS)	$MD_{\frac{3}{4}}$ (MS)	PD (SS)	PD (MS)
Synthetic	13.00	11.60	10.30	10.40	10.00	10.50
Glass	26.59 (0.25)	26.14 (0.25)	24.92 (0.25)	24.43 (0.25)	25.70 (0.34)	25.24 (0.33)
Biomedical	12.44 (0.13)	12.04 (0.12)	14.25 (0.13)	14.03 (0.14)	12.37 (0.14)	12.18 (0.13)

	DDα-classifier				
	1-NN			Mahalanobis depth	
	Eucl.	Mah. dist.			
Dataset	dist.	Mom.	MCD	Mom.	MCD
Synthetic	12.10	11.90	12.00	11.90	12.00
Glass	29.45 (0.20)	25.79 (0.17)	24.73 (0.18)	30.09 (0.18)	35.06 (0.22)
Biomedical	13.51 (0.14)	19.59 (0.18)	17.90 (0.17)	12.91 (0.14)	15.23 (0.16)

Regarding the Biomedical Data, [5] do not specify the sample sizes they use in training and testing. For the DDα-classifier, we select 100 observations of the larger class and 50 of the smaller class to form the training sample; the remaining observations constitute the testing sample. As it is seen from Table 5 the DDα-classifier shows results similar to the projection-depth classifier (except with the Synthetic Data), while the performance of outsider-handling methods varies depending on the type of the data. Specifically, with the Glass Data 1-NN based on the Mahalanobis distance (both with the moment and the robust estimate) performs best in handling outsiders. On the other hand, with the Biomedical Data the same approach performs quite poorly, while treating outsiders with

moment-estimated Mahalanobis depth or Euclidean 1-NN yields best results.

Table 6: Benchmark comparison with L_p-depth classifiers.

Data-set	MD		L_pD		**DDα-classifier**				
					1-NN			Mahalanobis depth	
					Eucl. dist.	Mah. dist.			
	Mom.	MCD	Mom.	MCD		Mom.	MCD	Mom.	MCD
Syn.	10.20	10.60	9.60	10.70	12.10	11.90	12.00	11.90	12.00
Hem.	15.84 (0.30)	17.13 (0.32)	15.39 (0.32)	16.43 (0.32)	16.63 (0.20)	17.98 (0.20)	18.36 (0.19)	18.65 (0.22)	19.39 (0.22)
Gla.	26.80 (0.26)	24.80 (0.29)	27.64 (0.29)	24.75 (0.26)	30.13 (0.19)	28.37 (0.22)	26.63 (0.20)	32.88 (0.22)	36.82 (0.23)
Biom.	12.35 (0.14)	14.48 (0.15)	12.68 (0.15)	15.11 (0.15)	13.74 (0.09)	22.09 (0.16)	20.89 (0.14)	14.34 (0.12)	17.28 (0.14)
Diab.	8.22 (0.18)	11.49 (0.22)	9.39 (0.21)	11.92 (0.27)	10.77 (0.12)	18.36 (0.18)	18.33 (0.20)	12.70 (0.18)	15.90 (0.19)
B.Tr.	22.75 (0.07)	22.17 (0.08)	22.30 (0.07)	22.06 (0.07)	23.11 (0.06)	22.73 (0.06)	22.92 (0.06)	22.59 (0.06)	22.17 (0.06)

Table 6 presents a similar comparison of the DDα-classifier with the L_p-classifier of [4]. The same approaches are included to treat outsiders. In all six benchmark examples the DDα-classifier generally performs worse than the best L_p-depth classifier. However, its performance substantially depends on the chosen treatment of outsiders. In all examples the AMR of the DDα-classifier comes close to that of the L_p-depth classifier, provided the outsider treatment is properly selected. On the Hemophilia Data, e.g., Euclidean 1-NN should be chosen. On the Glass Data a 1-NN outsider treatment with robust Mahalanobis distance performs relatively best, etc. On the Blood Transfusion Data all outsider-handling approaches show equally good performance, which appears to be typical when n is relatively large compared to d.

6.2 Benchmark comparisons with SVM

The support vector machine (SVM) is a powerful solver of the classification problem and has been widely used in applications. However, different from the DDα-classifier, the SVM is a parametric approach, as in applying it certain parameters have to be adjusted: the box-constraint and the kernel parameters. The AMR performance of the SVM depends heavily on the choice of these parameters. In applications, optimal parameters are selected by some cross-validation, which affords extensive calculations. Once these parameter have been optimized, SVM-classification is usually very fast and precise.

In comparing the SVM with the DDα-procedure, this step of parameter optimization has to be somehow accounted for. Here we introduce a two-fold view on the comparison problem: Two values of the AMR are calculated and reported, first the *best AMR* when the parameters have been optimally selected, second the *expected AMR* when the parameters are randomly selected over specified ranges. As ranges we choose the intervals between the smallest and the largest number that arise as an optimal value in one of our benchmark data examples. This seems us a fair and, regarding the parameter ranges, rather conservative approach.

As benchmark four well-known data sets are employed in the sequel, Diabetes, Ecoli, Glass, and Iris Data being taken [1]. Following [5] the two biggest classes of the Glass Data have been selected, and similarly to [4] we have chosen three of the bigger classes from the Ecoli Data. The DDα-classifier is calculated with the same outsider treatments as above. For the SVM-classifier we use radial basis function kernels as implemented in LIBSVM with the R-Package "e1071" as an R-interface. Leave-one-out cross validation is employed for performance estimation of the all classifiers. The computation has been done on the same PC as in Section 5.2.

The results on the best AMR and the expected AMR are collected in the Table 7, together with time quantities and portions of outsiders. The Iris Data appears twice in the Table. First the original are used, and second the same data after a preprocessing step. The preprocessing consists in the exclusion of an obvious outlier in the DD-plot that was identified by visual inspection of the plot.

The overall analysis of the Table 7 shows that, even if using an arbitrary technique for handling outsiders, the DDα-classifier mostly performs not much worse than an SVM where the parameters have been optimally chosen. In contrast, if the SVM is employed with some non-optimized parameters, its AMR can be considerably larger than that of the DDα-classifier. Average errors over the relevant intervals are given in parentheses.

The times needed to classify a new object (also given in Table 7) are quite comparable. But as the parameters of the SVM have to be adjusted first by running it many times for cross-validation, the computational burden of its training phase is much higher than that of the DDα-classifier, which has to be run only once. Recall that the latter is fully nonparametric. For example, in our implementation it took 875 seconds to determine approximate optimal values of SVM parameters for the Diabetes Data and similar times for the others.

7 Discussion and conclusions

A new classification procedure has been proposed that is completely nonparametric. The DDα-classifier transforms the d-variate data to a q-variate depth

Table 7: Benchmark comparison with the support vector machine; γ - kernel parameter, C - box constraint.

Data-set	Legend	**DDα-classifier**					**SVM**
		1-NN			Mahalanobis		
		Eucl.	Mah. dist.		depth		
		dist.	Mom.	MCD	Mom.	MCD	Opt. (Avr.)
Diab.	Error	28.26	30.6	34.51	24.35	31.77	23.18 (44.99)
	Time:train	16.63	16.62	16.59	16.58	17.39	0.05
	Time:test	0.033	0.009	0.0092	0.0035	0.0037	0.0023
	γ/C						0.056/1
	% outsiders	62.24	62.24	62.24	62.24	63.54	
Ecoli	Error	10.29	11.4	12.13	12.13	16.18	3.68 (47.43)
	Time:train	0.26	0.26	0.26	0.26	0.26	0.0077
	Time:test	0.014	0.0026	0.0032	0.001	0.00044	0.0019
	γ/C						5.62/1.78
	% outsiders	75	75	75	75	75	
Glass	Error	18.49	26.03	31.51	34.93	34.93	21.23 (51.96)
	Time:train	0.31	0.32	0.31	0.32	0.32	0.0082
	Time:test	0.0083	0.0019	0.0016	0.00014	0.00055	0.0024
	γ/C						0.56/1
	% outsiders	95.89	95.89	95.89	95.89	95.89	
Iris	Error	37.33	37.33	37.33	36	46.67	4.67 (66.67)
	Time:train	0.07	0.07	0.07	0.07	0.07	0.0051
	Time:test	0.0046	0.0018	0.0013	0.00033	0.00047	0.0017
	γ/C						0.056/10
	% outsiders	50	50	50	50	50	
Iris (Pre.)	Error	3.36	3.36	4.03	2.68	13.42	2.68 (66.44)
	Time:train	0.07	0.07	0.07	0.07	0.07	0.0052
	Time:test	0.0046	0.0011	0.0013	0.0006	0.00027	0.0017
	γ/C						0.1/3.16
	% outsiders	51.68	51.68	51.68	51.68	51.68	

plot and performs linear classification in an extended depth space. The depth transformation is done by the zonoid depth, and the final classification by the α-procedure. The procedure has attractive properties: First, it proves to be very fast and efficient in the training as well as in the testing phase; in this it highly outperforms existing alternative nonparametric classifiers, and also - regarding the training phase - the support vector machine. Second, in many settings of elliptically distributed alternatives, its AMR is of similar size than that of the competing classifiers. Moreover, it is rather robust to outlier prone data. As a nonparametric approach, the new procedure shows a particularly good behavior under asymmetrically distributed alternatives and when the two classes originate from different families of distributions. Other than many competitors, it considers

all classes in the multi-class classification problem even when performing binary classification. Also several theoretical properties of the DDα-procedure have been derived: It operates in a rather simple way if the data generating processes are elliptical, and a Bayes rule holds if $q = 2$ and the two classes are mirror symmetric.

The zonoid depth has many theoretical and computational advantages: Most important here, it is efficiently computed also in higher dimensions. However, as it takes its maximum at the mean of the data, the zonoid depth lacks robustness. Nevertheless, the DDα-classifier shows a rather robust behavior. Its relative robustness can be explained as follows: The original data points are mapped to a compact set, the q-dimensional unit hypercube, and then classified by the α-procedure. The latter, by choosing a median angle in each step, is rather insensitive to outliers.

Points that are not within the convex hull of at least one training set must be specially treated as their depth representation is zero. To classify those so called outsiders several approaches have been used and compared. Instead of assigning them randomly, which disadvantages the DDα-procedure like other procedures based on halfspace or simplicial depth, one should classify outsiders by 1-NN and some distance or by a properly chosen maximum depth rule.

To contrast the DDα-procedure with an SVM approach, a novel way of comparison has been taken: An optimal performance of an SVM has been evaluated, that arises under an optimal choice of the parameters, as well as an average performance, where the parameters vary over specified conservative intervals. It came out that, even with an arbitrary handling of outsiders, the DDα-classifier mostly performs not much worse than an SVM whose parameters have been optimally chosen. If the SVM is employed with some non-optimized parameters, the AMR can be considerably larger than that of the DDα-classifier.

More investigations are needed on the behavior of the DDα-classifier on skewed or fat-tailed data, the - possibly adaptive - choice of outsider treatments, and the use of alternative notions of data depth. These are intended for future research.

Acknowledgements 1 *Thanks are to Rainer Dyckerhoff for his constructive remarks on the paper as well as to the other participants of the Witten Workshop on "Robust methods for dependent data" for discussions.*

References

[1] A. Asuncion and D. Newman. UCI machine learning repository. URL http://archive.ics.uci.edu/ml/ (2007).

[2] I. Cascos. Data depth: Multivariate statistics and geometry. *New Perspectives in Stochastic Geometry*, (W. Kendall and I. Molchanov, eds.) (2009).

[3] J.A. Cuesta-Albertos and A. Nieto-Reyes. The random Tukey depth. *Computational Statistics and Data Analysis*, 52, 4979-4988 (2008).

[4] S. Dutta and A.K. Ghosh. On classification based on L_p depth with an adaptive choice of p. Preprint 2011.

[5] S. Dutta and A.K. Ghosh. On robust classification using projection depth. *Annals of the Institute of Statistical Mathematics*, 64, 657–676 (2012).

[6] R. Dyckerhoff. Data depths satisfying the projection property. *AStA - Advances in Statistical Analysis*, 88, 163-190 (2004).

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

[8] A.K. Ghosh. and P. Chaudhuri. On data depth and distribution free discriminant analysis using separating surfaces. *Bernoulli*, 11, 1-27 (2005).

[9] A.K. Ghosh. and P. Chaudhuri. On maximum depth and related classifiers. *Scandinavian Journal of Statistics*, 32, 327-350 (2005).

[10] T. Hastie, R. Tibshirani and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd Edition. Springer Verlag. New York (2009).

[11] R. Hoberg and K. Mosler. Data analysis and classification with the zonoid depth. *DIMACS Ser. Math. Theo. Comput. Sci.*, (R. Liu, R. Serfling and D. Souvaine, eds.), 72, 49-59 (2006).

[12] Mia Hubert and Katrien van Driessen. Fast and robust discriminant analysis. *Computational Statistics and Data Analysis*, 45, 301-320 (2004).

[13] R. Jornsten. Clustering and classification based on the L1 data depth. *J. Multivariate Anal.* 90, 67-89 (2004).

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

[15] T. Lange, P. Mozharovskyi and G. Barath. Two approaches for solving tasks of pattern recognition and reconstruction of functional dependencies. *ASMDA Conference*, Rome, 7-10 June 2011.

[16] J. Li, J.A. Cuesta-Albertos, and R.Y. Liu. DD-classifier: Nonparametric classification procedure based on DD-plot. Preprint 2011.

[17] R.Y. Liu. On a notion of data depth based on random simplices. *Annals of Statistics*, 18, 405-414 (1990).

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

[19] P. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Academy India* 12, 49-55 (1936).

[20] K. Mosler. *Multivariate Dispersion, Central Regions and Depth: The Lift Zonoid Approach.* Springer Verlag. New York (2002).

[21] R. Serfling. Depth functions in nonparametric multivariate inference. *Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications*, (R. Liu, R. Serfling and D. Souvaine, eds.), 1-16 (2006).

[22] J.W. Tukey. Mathematics and the picturing of data. *Proc. Inter. Cong. Math.*, Vancouver, 523-531 (1975).

[23] V.N. Vapnik. *Dtatistical learning theory.* Wiley. New York (1998).

[24] V.I. Vasil'ev. The reduction principle in pattern recognition learning (PRL) problem. *Pattern Recognition and Image Analysis* 1, 1 (1991).

[25] V.I. Vasil'ev. The reduction principle in problems of revealing regularities I. *Cybernetics and Systems Analysis* 39, 686-694 (2003).

[26] V.I. Vasil'ev and T. Lange. The duality principle in learning for pattern recognition (in Russian). *Kibernetika i Vytschislit'elnaya Technika* 121, 7-16 (1998).

[27] Y.J. Zuo and R. Serfling. General notions of statistical depth function. *Annals of Statistics* 28, 461-482 (2000).