KYBERNETIKA — VOLUME *57* (2021), NUMBER 1, PAGES 15–37

A DEPTH-BASED MODIFICATION OF THE K-NEAREST NEIGHBOUR METHOD

Ondřej Vencálek and Daniel Hlubinka

We propose a new nonparametric procedure to solve the problem of classifying objects represented by d-dimensional vectors into $K \geq 2$ groups. The newly proposed classifier was inspired by the k nearest neighbour (kNN) method. It is based on the idea of a depth-based distributional neighbourhood and is called k nearest depth neighbours (kNDN) classifier. The kNDN classifier has several desirable properties: in contrast to the classical kNN, it can utilize global properties of the considered distributions (symmetry). In contrast to the maximal depth classifier and related classifiers, it does not have problems with classification when the considered distributions differ in dispersion or have unequal priors. The kNDN classifier is compared to several depth-based classifiers as well as the classical kNN method in a simulation study. According to the average misclassification rates, it is comparable to the best current depth-based classifiers.

Keywords: Bayes classifier, data depth, k nearest depth neighbours, nonparametric

Classification: 62H30, 62G30

1. INTRODUCTION

The classification of multivariate observations to given groups is a classical statistical problem that has been studied many times and still is of high importance. Procedures based on data depth represent a modern nonparametric approach to the classification problem.

The notion of data depth provides a multivariate version of ranks. A depth function is any function which provides an ordering of points in multidimensional space with respect to a given distribution (or data cloud in the empirical case). Many depth functions have been introduced ad hoc. Their overview may be found in [30] where a general definition of a depth function is provided (the definition lists several desirable properties, suggested already by Liu in 1990, see [20]). Nowadays, the concept of data depth provides a basis for a wide range of nonparametric procedures. Its applications are reviewed in [21] or [26].

The idea of using data depth for classification was suggested by Liu in [20]. The classifier which assigns a new observation to the distribution where it has the most central position, i. e. maximal depth (referred as the max-depth classifier), was studied

DOI: 10.14736/kyb-2021-1-0015

in detail by Ghosh and Chaudhuri [10]. The development of new possible ways how to apply the data depth for classification has been intensified in recent years ([6, 14, 18], and [24]).

The current paper deals with a new classifier based on data depth. It is inspired by the k-nearest neighbour (kNN) method. This classical method is well-known for its versatility – it achieves low misclassification rates even if distributions of observations in individual groups are far from normality. On the other hand, the kNN could not utilise global properties of distributions like their symmetry which leads to its less satisfactory performance in higher dimensions. The newly suggested classifier retains the local character of the kNN procedure but is able to take advantage of the global properties of distributions.

The paper is organised as follows. The classification problem is briefly recalled in Section 2. The new kNDN classifier is introduced in Section 3. Its equivalence to the Bayes rule is proved for a class of general elliptically symmetric distributions. However, the method gives very good results for more general settings. In particular, this methods works better than the usual methods if the probability distributions of the respective groups belong to different families. Section 4 shows that the proposed method is at least comparable with the usual classifiers and in some settings it outperforms most of them. Two real data examples are also included in that section. The proof of the main result is provided in the Appendix.

2. THE DEPTH AND SUPERVISED CLASSIFICATION

Recall briefly the classification problem. Consider $2 \leq K < \infty$ groups of objects. Each object (in any group) is represented by $d \in \mathbb{N}$ numerical characteristics. Each group of objects is characterised by an (unknown) probability distribution on $\mathbb{R}^d$ of these numerical characteristics. Let us denote these distributions $P_1, \dots, P_K$. All distributions are assumed to be absolutely continuous with respect to the d-dimensional Lebesgue measure and, naturally, $P_i \neq P_j$ when $i \neq j$.

Consider further independent random samples $\boldsymbol{X}_{i,1}, \dots, \boldsymbol{X}_{i,n_i}$ from P_i, $i = 1, \dots, K$. These random samples (called the training set) provide the only available information on the unknown distributions. There is a need to find a rule assigning a new d-dimensional observation $\boldsymbol{X}$ to one of the groups. Such rule (called classifier) must have a form of some measurable function $c : \mathbb{R}^d \to \{1, \dots, K\}$.

The quality of the classification rule is usually measured by the average misclassification rate, i. e. by the proportion of incorrectly classified observations in the group of all observations to be classified. The average misclassification rate estimates the total probability of misclassification

$$\sum_{i=1}^{K} \pi_i \mathrm{P}\left(c(\boldsymbol{X}) \neq i | \boldsymbol{X} \sim P_i\right),$$

where $\pi_i = \mathrm{P}(\boldsymbol{X} \sim P_i)$ is the prior probability that $\boldsymbol{X}$ comes from the ith group, and $\mathrm{P}\left(c(\boldsymbol{X}) \neq i | \boldsymbol{X} \sim P_i\right)$ is the conditional probability of incorrect classification given that $\boldsymbol{X}$ comes from the ith group. A classifier minimising the average misclassification rate is called the *Bayes minimal error rule* or the *optimal Bayes rule.* It is well-known that the optimal classifier has the following form:

$$c(\boldsymbol{x}) = \arg \max_{i=1\dots,K} \pi_i f_i(\boldsymbol{x}), \tag{1}$$

where $f_i(\cdot)$ is the probability density function of the ith distribution. It is essential to realise that no classifier can have lower probability of misclassification than the Bayes classifier (1), see p. 307 in [22].

The Bayes rule (1) is based on probability density functions which in practice need to be estimated from the training set either by estimation of their parameters (parametric methods) or in some nonparametric way. Data depth can be successfully used for the density estimation in some special cases, e. g., if the distribution is elliptically symmetric and strictly decreasing from the centre. In that case, the halfspace depth becomes a strictly increasing function of f, see [7]. If the densities do not have global properties like symmetry, the use of data depth for classification purposes is not so straightforward. Zakai and Ritov [29] have shown that all consistent classifiers are necessarily localisable, i. e., they do not significantly change their response at a particular point when only the part of the training set that is close to that point is shown to them. However, data depth of a point is a measure of its centrality with respect to the whole distribution and therefore it is not of a local nature. There are two ways how to localise the depth-based procedures. The first possibility is to use some local depth, as suggested in, e. g., [1, 13, 16], or [23]. The second possibility is to plug-in some local classification procedure like k-nearest neighbours. Such classifiers were studied by Paindaveine and Van Bever [24] or by Vencalek [27].

3. CLASSIFICATION USING K-NEAREST DEPTH NEIGHBOURHOOD

In this section, we briefly recall the idea of k-nearest neighbour classifier and explain the new method of k-nearest depth neighbourhood (k-NDN).

3.1. Classical kNN classification

The classical k-nearest-neighbour method is based on a neighbourhood of a point with respect to the Euclidean distance, i. e. the neighbourhood $L_{E,\epsilon}(\boldsymbol{x})$ of the point $\boldsymbol{x}$ defined as

$$L_{E,\epsilon}(\boldsymbol{x}) = \left\{\boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{y}\| < \epsilon\right\}, \tag{2}$$

for some positive constant $\epsilon \in \mathbb{R}$. In what follows, we omit the subscript ϵ if it is possible without confusion. For a sufficiently small constant ϵ and a continuous density function f of a random vector $\boldsymbol{X}$, the approximation

$$\mathrm{P}(\boldsymbol{X} \in L_E(\boldsymbol{x})) = \int_{L_E(\boldsymbol{x})} f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} \cong f(\boldsymbol{x}) \cdot \lambda_d\left(L_E(\boldsymbol{x})\right), \tag{3}$$

where λ_d is the d-dimensional Lebesgue measure, may be used. The density f at the point $\boldsymbol{x}$ may then be estimated using the approximation

$$f(\boldsymbol{x}) \cong \frac{\mathrm{P}(\boldsymbol{X} \in L_E(\boldsymbol{x}))}{\lambda_d\left(L_E(\boldsymbol{x})\right)}. \tag{4}$$

The kNN classification is then based on (4) since the terms that include the group index i in the expression

$$\pi_i f_i(\boldsymbol{x}) \cong \pi_i \frac{P_i(L_E(\boldsymbol{x}))}{\lambda_d\left(L_E(\boldsymbol{x})\right)} \tag{5}$$

may be estimated easily. Let n be the total number of points in the training set ($n = n_1 + \ldots + n_K$), fix $k < n$ and find the smallest ball $L_E(\boldsymbol{x})$ centred at $\boldsymbol{x}$ that contains k points of the training set. Denote by K_i the number of points from the ith group of the training set lying in $L_E(\boldsymbol{x})$. A natural estimator of probability $P_i(L_E(\boldsymbol{x}))$ is then K_i/n_i, and the prior probabilities π_i may be estimated by n_i/n. Hence the empirical Bayes classifier is

$$\arg\max_i \widehat{\pi}_i \widehat{f}_i(\boldsymbol{x}) = \arg\max_i \frac{n_i}{n}\frac{K_i}{n_i}\frac{1}{\widehat{\lambda}_d(L_E(\boldsymbol{x}))} = \arg\max_i \frac{K_i}{n\widehat{\lambda}_d(L_E(\boldsymbol{x}))}, \tag{6}$$

which may be simplified to

$$c(\boldsymbol{x}) = \arg\max_{i=1,\ldots,K} K_i.$$

For more details see [8].

3.2. Construction of the k nearest depth neighbours classifier

The basic idea of the kNDN classifier is to use a different notion of the neighbourhood in approximation (4). Approximation (4) is appropriate if the density is almost constant on the used neighbourhood $L_E(\boldsymbol{x})$. In that case, it makes sense to use the density-based neighbourhood – a set of points where the density is similar to the value $f(\boldsymbol{x})$. Consider now a depth function $D(\cdot, P)$ which assigns measures of centrality to points in $\mathbb{R}^d$ with respect to the distribution P characterised by the density f. Assuming that small difference in density between two (arbitrary) points small difference in depths of these points, it is possible to use the depth-based neighbourhood instead of the density-based neighbourhood.

The depth-based neighbourhood of a point $\boldsymbol{x}$ can be defined as a set of points which have the depth similar (w.r.t. a given probability distribution) to the point $\boldsymbol{x}$. The depth-based neighbourhood was defined in [9].

Definition 3.1. For a depth function D, positive real constant ϵ, and probability measure P the depth ϵ-neighbourhood of $\boldsymbol{x} \in \mathbb{R}^d$ with respect to P is defined as

$$L_{D,\epsilon}(\boldsymbol{x}; P) = \left\{\boldsymbol{y} \in \mathbb{R}^d : |D(\boldsymbol{x}; P) - D(\boldsymbol{y}; P)| < \epsilon\right\}. \tag{7}$$

The difference between the distance-based neighbourhood and the depth-based neighbourhood of a point is illustrated in Figure 1. In the considered example, contours of the pdf of the bivariate normal distribution $N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 & 1 \\ 1 & 1 \end{pmatrix}\right)$ are plotted. The classical and the depth neighbourhood of a point $\boldsymbol{x} = (2.5, 0.5)^T$ are shown as the grey regions.

Note that the depth-based neighbourhood involves the whole distribution, hence some intrinsic geometric properties of the distribution may be captured in the shape of the neighbourhood.

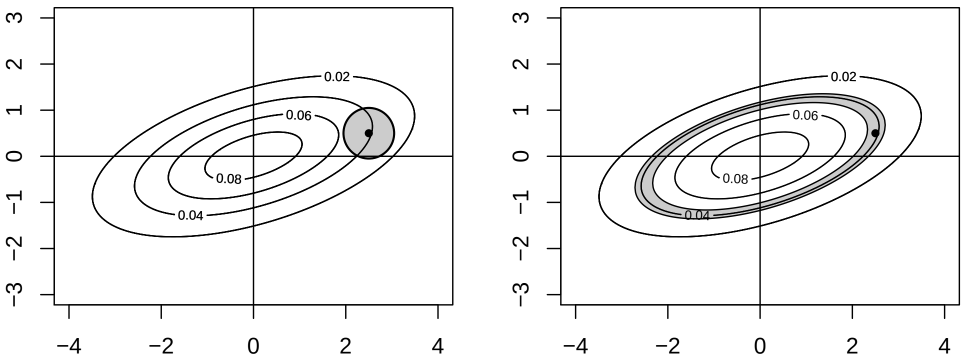

Fig. 1. A classical neighbourhood of a point $\boldsymbol{x} = (2.5, 0.5)^T$ (left) and a distributional neighbourhood of the same point (right), when the considered distribution is centred bivariate normal.

Using the depth neighbourhood instead of the distance-based neighbourhood brings the following advantage. Let us consider constants ϵ_1 and ϵ_2 such that it holds $P(L_{D,\epsilon_1}(\boldsymbol{x}; P)) = P(L_{E,\epsilon_2}(\boldsymbol{x}))$. Figure 1 indicates that the density f is less varying in the region $L_{D,\epsilon_1}(\boldsymbol{x}; P)$ than in the ball $L_{E,\epsilon_2}(\boldsymbol{x})$. Hence, the approximation

$$f(\boldsymbol{x}) \cong \frac{\mathrm{P}\big(\boldsymbol{X} \in L_{D,\epsilon_1}(\boldsymbol{x})\big)}{\lambda_d\big(L_{D,\epsilon_1}(\boldsymbol{x})\big)}. \tag{8}$$

may be even more precise than the approximation (4).

The depth neighbourhood depends on the underlying distribution while the classical neighbourhood is distribution free. This is the principal difference and the classification rule must be modified properly. Our classifier is based on the approximation

$$\pi_i f_i(\boldsymbol{x}) \cong \pi_i \frac{P_i(L_{D,\epsilon_i}(\boldsymbol{x}, P_i))}{\lambda_d\big(L_{D,\epsilon_i}(\boldsymbol{x}, P_i)\big)}, \tag{9}$$

which is only a slight modification of (5). In practice, empirical distributions based on the training set $\widehat{P}_1, \ldots, \widehat{P}_K$ are used instead of their unknown theoretical counterparts.

We proceed similarly as in the case of the classical kNN method. Fix $k < \min_i n_i$ and find the smallest depth neighbourhoods $L_{D,\epsilon_1}(\boldsymbol{x}, \widehat{P}_1), \ldots, L_{D,\epsilon_K}(\boldsymbol{x}, \widehat{P}_K)$ each containing k points from the corresponding part of the training set. Note that the number of points in each neighbourhood is now fixed and equal for each group but the volumes of the neighbourhoods are different while for the classical kNN method it is the opposite. The volumes need to be estimated, what might be a nontrivial task, as discussed in Section 3.5. The classifier based on the empirical version of the approximation (9) has the following form:

$$c(\boldsymbol{x}) = \arg \max_{i=1,\ldots,K} \frac{n_i}{n} \frac{k}{n_i} \frac{1}{\widehat{\lambda}_d\big(L_{D,\epsilon_i}(\boldsymbol{x}, \widehat{P}_i)\big)},$$

which may be simplified to

$$c(\boldsymbol{x}) = \arg \min_{i=1,\dots,K} \widehat{\lambda}_d\big(L_{D,\epsilon_i}(\boldsymbol{x}, \widehat{P}_i)\big). \tag{10}$$

An observation is therefore classified into the group with the smallest volume of its depth neighbourhood containing k points from the corresponding training group.

The choice of the number of neighbours k should follow the rules well known in the classical kNN method: with an increasing size of the training set, k should also increase whereas the proportion of points in the neighbourhood should decrease, see Section 3.4. In practice, k is usually chosen by cross-validation.

3.3. Depth functions

The above-presented method of the depth neighbourhood may be applied with an arbitrary depth, or localised depth function. In this paper, we mainly consider the *halfspace depth*, the *projection depth* or the Mahalanobis depth. These depth functions are widely used for classification purposes.

Let us recall that the halfspace (Tukey) depth of a point $\boldsymbol{x}$ w.r.t. a probability distribution P of a random variable $\boldsymbol{X}$ is defined as

$$D(\boldsymbol{x}, P) = \inf_{\|\boldsymbol{u}\|=1} \mathrm{P}[\boldsymbol{u}^\top(\boldsymbol{X} - \boldsymbol{x}) \geq 0], \tag{11}$$

i. e., the halfspace depth is the infimum of probabilities of closed halfspaces containing the point $\boldsymbol{x}$.

The projection depth of a point $\boldsymbol{x}$ w.r.t. a probability distribution P is defined as

$$D(\boldsymbol{x}, P) = \frac{1}{1 + O(\boldsymbol{x}, P)}, \ O(\boldsymbol{x}, P) = \sup_{\|u\|=1} \frac{|(\boldsymbol{u}^T\boldsymbol{x} - \mu_{P_{\boldsymbol{u}}})|}{\sigma_{P_{\boldsymbol{u}}}}, \tag{12}$$

where $\mu_{P_{\boldsymbol{u}}}$ is some location and $\sigma_{P_{\boldsymbol{u}}}$ is some scale characteristic of the distribution of the random variable $\boldsymbol{u}^T\boldsymbol{X}$, usually the median and the median absolute deviation (MAD), respectively. These characteristics are preferred to the mean and the standard deviation because of their robustness.

The Mahalanobis depth of a point $\boldsymbol{x}$ w.r.t. a probability distribution P is defined as

$$D(\boldsymbol{x}, P) = \frac{1}{1 + M(\boldsymbol{x}, P)}, \tag{13}$$

where $M(\boldsymbol{x}, P) = (\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$ is squared Mahalanobis distance of the point $\boldsymbol{x}$ from the centre of P denoted as $\boldsymbol{\mu}$.

The k depth nearest neighbours classifier with the above-mentioned depth functions is quite good if the level sets of density f are convex, or even elliptically symmetric. Then the approximation (9) may be even better than the approximation (5) (see Figure 1). Other kinds of symmetry (l_p-symmetry) can be utilized with specific depth functions, see [5]. For asymmetric distributions, the correspondence of level sets of density and depth can be approached by the use of local depth, as discussed in [13].

On the other hand, if the level sets of the probability density function are not convex, then the approximation (9) needs not be sufficiently good. This problem may be solved by using some version of a *localised depth*, see [13, 16] or [1], rather than a usual global depth function such as the halfspace depth or the projection depth.

3.4. Optimality of the k nearest depth neighbours classifier

In this section we show that the classifier (10) with the halfspace, projection or Mahalanobis depth is optimal when considering elliptically symmetric distributions P_i. However, the distributions P_i need not be of the same nature, they may differ not only in location and scatter matrix but quite generally in the family of distributions. Also, the priors π_i need not be equal for the optimality. These assumptions are then less restrictive than is usual for basic depth-based methods like max-depth classifier (see [10]). Although optimality is guaranteed only for elliptically symmetric distributions, the classifier (10) is applicable in a more general situation, as shown in the simulation study 4.

Let us start with two elliptically symmetric distributions P_1 and P_2 on $\mathbb{R}^d$ with densities $f_1(\cdot)$ and $f_2(\cdot)$. Assume:

(Q1) $f_i(\boldsymbol{x}) = \frac{\gamma}{|\boldsymbol{\Sigma}_i|^{1/2}} g_i\left(M_i(\boldsymbol{x})\right)$, where $M_i(\boldsymbol{x}) = \left[(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right]^{1/2}$ denote the Mahalanobis distances of $\boldsymbol{x}$ from the $\boldsymbol{\mu}_i$, $i=1,2$, and γ is a normalising constant,

(Q2) g_i are continuous functions, such that $g_i(cx) < g_i(x)$ for arbitrary $x \in \mathbb{R}^+$ such that $g_i(x) > 0$ and $c > 1$, and $g_i(x) = 0 \Rightarrow g_i(cx) = 0\ \forall c > 1$, i. e. g_i are continuous and monotone decreasing and strictly decreasing on sets $\{\boldsymbol{x}\colon g_i(\boldsymbol{x}) > 0\}$.

Assumption (Q2) means that there are no central areas with high depth but zero probability.

Let us denote $n = n_1 + n_2$ the (total) size of the training set, n_1 and n_2 denoting the number of observations from group 1 and group 2 in the training set. Further, denote k_n the number of points (from each group) included in the depth neighbourhood. Assume

$$k_n \overset{n\to\infty}{\longrightarrow} \infty, \quad \frac{k_n}{n} \overset{n\to\infty}{\longrightarrow} 0.$$

We consider a sequence of independent d-dimensional random vectors $\boldsymbol{X}_1, \boldsymbol{X}_2, \dots$ with the same distribution $P = \pi_1 P_1 + \pi_2 P_2$. For any fixed $n \in \mathbb{N}$ and any fixed $\boldsymbol{x} \in \mathbb{R}^d$ denote $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ the training set and $\boldsymbol{X}_{1,1}, \dots, \boldsymbol{X}_{1,n_1}$ and $\boldsymbol{X}_{2,1}, \dots, \boldsymbol{X}_{2,n_2}$ the observations belonging to group 1 and 2, respectively. Denote an ordering relation $\prec_{M_i}$ by

$$\boldsymbol{x} \prec_{M_i} \boldsymbol{y} \text{ if } M_i(\boldsymbol{x}) \leq M_i(\boldsymbol{y}).$$

Observations $\boldsymbol{X}_{i,1}, \dots, \boldsymbol{X}_{i,n_i}$ can be ordered according to their Mahalanobis distances from $\boldsymbol{\mu}_i$ in an increasing order: $\boldsymbol{X}_{i:1} \prec_{M_i} \boldsymbol{X}_{i:2} \prec_{M_i} \dots \prec_{M_i} \boldsymbol{X}_{i:m_i(n)} \prec_{M_i} \boldsymbol{x} \prec_{M_i} \boldsymbol{X}_{i:m_i(n)+1} \prec_{M_i} \dots \prec_{M_i} \boldsymbol{X}_{i:m_i(n)+k_n} \prec_{M_i} \dots \prec_{M_i} \boldsymbol{X}_{i:n_i}$

In what follows, we use Mahalanobis distance based neighbourhood of a point $\boldsymbol{x} \in \mathbb{R}^d$ defined as

$$O_i^M(\boldsymbol{x}, h) := \left\{\boldsymbol{y} \in \mathbb{R}^d : M_i(\boldsymbol{y}) \in [M_i(\boldsymbol{x}), M_i(\boldsymbol{x}) + h]\right\}, i = 1, 2,$$

where $M_i(\cdot)$ denotes the Mahalanobis distance from $\boldsymbol{\mu}_i$. Recall that $\lambda_d\big(O_i^M(\boldsymbol{x}, h)\big) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} |\boldsymbol{\Sigma}_i|^{1/2} \left[\big(M_i(\boldsymbol{x}) + h\big)^d - \big(M_i(\boldsymbol{x})\big)^d\right]$.

Theorem 3.2. Consider the mixture of two distributions $P = \pi_1 P_1 + \pi_2 P_2$, where the distributions P_i satisfy (Q1–Q2) above. Let $\boldsymbol{x}$ be any fixed point in $\mathbb{R}^d$ such that $f_i(\boldsymbol{x}) > 0$ for both $i = 1, 2$. For any $n \in \mathbb{N}$ define a random variable $C_1(n) := M_1(\boldsymbol{X}_{1:m_1(n)+k_n}) - M_1(\boldsymbol{x})$ and $C_2(n) := M_2(\boldsymbol{X}_{2:m_2(n)+k_n}) - M_2(\boldsymbol{x})$. Then it holds:

$$\frac{\lambda_d\big(O_1^M(\boldsymbol{x}, C_1(n))\big)}{\lambda_d\big(O_2^M(\boldsymbol{x}, C_2(n))\big)} \to \frac{\pi_2 f_2(\boldsymbol{x})}{\pi_1 f_1(\boldsymbol{x})} \quad \text{in probability as } n \to \infty.$$

Under the assumptions (Q1–Q2), the halfspace depth or projection depth is a decreasing function of the Mahalanobis distance, as shown in [10]. Mahalanobis distance based neighbourhood therefore coincides with depth-based neighbourhood defined as

$$O_i^D(\boldsymbol{x}, \ell) := \left\{\boldsymbol{y} \in \mathbb{R}^d : D_i(\boldsymbol{y}) \in [D_i(\boldsymbol{x}) - \ell, D_i(\boldsymbol{x})]\right\},$$

where $D_i(\cdot)$ denotes the depth with respect to the distribution P_i and ℓ is a constant which can be chosen to make the neighbourhoods coincide:

$$O_i^M\big(M_i(\boldsymbol{x}), C_i(n)\big) = O_i^D\big(D_i(\boldsymbol{x}), \ell(C_i(n))\big). \tag{14}$$

As a corollary, the theorem implies the asymptotic equivalence of the classifier (10) and the Bayes classifier (1) under the assumption that the volumes of O_i^M, $i = 1, 2$ (and equivalently of O_i^D) may be consistently estimated. This is possible due to the assumption on n and k and due to the fact that the depth central regions are convex and the relation (14) holds.

3.5. Practical issues

Several practical issues need to be addressed when implementing the newly suggested classifier.

- Choice of the depth function: The classifier is suggested in the way that enables the use of any depth-function. However, it is advisable to use a depth function which orders points unambiguously (with no ties – points of the same empirical depth). Therefore, projection depth or Mahalanobis depth functions are convenient. On the other hand, use of the halfspace depth or the convex hull peeling depth implies the occurrence of the groups of points with equal depth. In this case, it is sometimes necessary to choose a subset of points from a group of points with equal empirical depth that should be included in the neighbourhood. The choice may affect the procedure considerably.

- Estimation of the volume of the neighbourhood: The classifier is based on estimates of volumes of neighbourhoods. However, the estimation may be a nontrivial task. There are several ways how to deal with this problem. First, a certain shape of the depth central regions may be assumed, e. g. it may be assumed they are of an elliptic shape. In that case, it is sufficient to estimate the parameters of the considered areas and subsequently compute their volumes. A more general approach relies on the assumption of quasi-concavity of the depth-central regions, see [26]. In this case, the central regions may be estimated using convex hulls of points with given or higher depth. This approach was used in the current paper – we used function convhulln() from the R package geometry [12], for details see [2]. In the most general case, one needs to deal with estimation of volumes of non-convex sets (when using, e. g., some localised depth whose central areas need not be convex). In that case, one can plug-in, e. g., Voronoi diagram to solve the estimation problem. However, this idea is still being explored and has not been used in practice so far.

- Inclusion of the new point into its neighbourhood: It may be a bit surprising that a new point (which needs to be classified) may not be included in its estimated distributional neighbourhood. Nevertheless, it happens from time to time if the estimation is based only on the points from the training set. For illustration, consider theoretical neighbourhood in the case of bivariate normal distribution which is an area between two ellipses. Empirical counterparts of these ellipses are polygons. Some points of the theoretical neighbourhood of a given point (including the point itself) may therefore be outside of the empirical neighbourhood (lie outside the area between the two polygons). We recommend to include the new point in its empirical neighbourhood.
- The problem of so-called outsiders: An outsider is a point that is not within the convex hull of at least one training set, see [18]. The outsiders may be classified in the same way as other points by the newly suggested classifier since they are included in their estimated neighbourhoods, as suggested above. However, in many situations, lower AMR may be obtained by the classification of outsiders based on the maximum Mahalanobis depth.

4. SIMULATION STUDY AND ANALYSIS OF BENCHMARK DATA SETS

We explored the properties of the newly proposed classifier in a simulation study. We compared its performance to several depth-based classifiers as well as the traditional k nearest neighbour method. The distributional settings used in the simulation are based on the settings used in [19] and [18]. This enables a straightforward comparison with two efficient depth-based classifiers – the DD classifier and the DD-alpha procedure. Various two-dimensional distributions are considered in this main part of the simulation. Moreover, we added a short simulation study dealing with applicability of the procedure in higher dimensions. The practical applicability of the new method is illustrated by an analysis of two benchmark data sets.

4.1. Simulation in $\mathbb{R}^2$: settings

Let us first list the compared classifiers. The newly suggested procedure – k nearest depth neighbour method (kNDN) – is compared to three depth-based classifiers and one classical classifier. The considered depth-based procedures are the DD-plot classifier (DD) suggested in [19], the DD-alpha procedure (DDalpha) suggested in [18], and the classifier based on the idea of symmetrisation (Sym) suggested in [24]. We used the implementation of the first two procedures that is available in the R package *ddalpha* [25]. The last procedure (Sym) was implemented by ourselves. The classical methods are represented by the k-nearest neighbour method (kNN) which outperforms the LDA and QDA in eight of the ten considered examples (except for the first two examples). The parameter k in the kNN was selected by the leave-one-out cross-validation. Finally, the performance of the optimal Bayes classifier (which is based on theoretical density and is not available in practice) is also studied and visualised. This serves as a benchmark of the best attainable classification.

There are many factors influencing performance of the depth based classifiers. We have gained a basic insight into influence of size of the training set (see the following paragraphs), and choice of the depth function. Projection depth and Mahalanobis depth were used for all classifiers (see section 3.5 where reasons for this choice are explained). Projection depth was computed by a procedure using $1,000$ randomly selected projections, for more details see documentation of the R package ddalpha [25]. In examples including asymmetric distributions (7–10), the newly suggested classifier was also used with local depth introduced in [23] implemented in the R package DepthProc [17].

Let us now describe distributional settings (10 different examples) used in the simulation. All the examples deal with a two class problem in two-dimensional real space. The first six examples deal with symmetric distributions, the remaining four examples include asymmetric distributions. The examples are summarised in Table 1. Let us briefly explain the shortcuts used in Table 1. The covariance matrix $\mathbf{\Sigma}_0$ used in examples 1–6 has the following form:

$$\mathbf{\Sigma}_0 = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}.$$

In the fifth example, we consider the same distributions as in the first example, but now there are 10 % of the points in the training set of the group 1 which come from $N(10 \cdot \mathbf{1}, \mathbf{\Sigma}_0)$. Similarly, this contamination is considered in the sixth example where we consider the same distributions as in the second example. In the case of bivariate exponential distributions, we consider the distribution with independent marginal exponential distributions. The bivariate mix-normal distribution used in example 9 comes up as a product of two independent distributions of the following form:

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

One hundred repetitions of the simulations were performed for each distributional setting. Each run was performed in the following way: the training data set containing N points from each of the considered distributions was generated. Two different training sample sizes were considered: $N = 50$ and $N = 250$. Subsequently, another 100 points from each group constituting the test set were generated and classified. The average misclassification rate was then computed and recorded.

The depth-based procedures that are inspired by the kNN method – kNDN and Sym – are dependent on number of considered neighbours (parameter k). For Sym procedure, we considered k corresponding to 1%, 5%, 10%, 15%, and 20% of N (number of points from individual groups in the training set). For kNDN procedure, only 10 and 20 percent were considered.

	Group 1		Group 2	
Ex.	Distribution	Parameters	Distribution	Parameters
1	Normal	$\mathbf{0}, \mathbf{\Sigma}_0$	Normal	$\mathbf{1}, \mathbf{\Sigma}_0$
2	Normal	$\mathbf{0}, \mathbf{\Sigma}_0$	Normal	$\mathbf{1}, 4\mathbf{\Sigma}_0$
3	Cauchy	$\mathbf{0}, \mathbf{\Sigma}_0$	Cauchy	$\mathbf{1}, \mathbf{\Sigma}_0$
4	Cauchy	$\mathbf{0}, \mathbf{\Sigma}_0$	Cauchy	$\mathbf{1}, 4\mathbf{\Sigma}_0$
5	Contamin. normal	$\mathbf{0}, \mathbf{\Sigma}_0$	Normal	$\mathbf{1}, \mathbf{\Sigma}_0$
6	Contamin. normal	$\mathbf{0}, \mathbf{\Sigma}_0$	Normal	$\mathbf{1}, 4\mathbf{\Sigma}_0$
7	Bivar. exponential	1, 1	Shifted bivar. expon. (+**1**)	1, 1
8	Bivar. exponential	1, 1/2	Shifted bivar. expon. (+**1**)	1/2, 1
9	Bivar. mix-normal	(0,1,2), (0,1,4)	Bivar. mix-normal	(1,1,2), (1,1,4)
10	Normal	$\mathbf{0}, \boldsymbol{I}$	Bivar. exponential	1, 1

Tab. 1. Examples used in the simulation study.

4.2. Simulation in $\mathbb{R}^2$: results

The main results of the simulation study are presented in Figures 2 and 3 (for examples 1–6), Figures 4 and 5 (for examples 7–10), respectively. Boxplots of average misclassification rates of the considered classifiers are plotted there.

For each of the four depth-based methods (kNDN, Sym, DDalpha, and DD), several possible versions were examined from which the best one was always highlighted by light grey colour. Shortcuts of the classifiers is amended by one of the letters P, M, and L denoting used depth function – projection, Mahalanobis or local (parameter of lacality beta = 0.7 turned out to be appropriate choice in all considered situations). For the kNDN classifier, percentage determining k (10p or 20p) is indicated. For the Sym classifier, values of k are added to the shortcut Sym.

The only classical classifier (kNN) can be distinguished by a darker shade of grey and the lowest achievable misclassification rate obtained by the Bayes classifier (not available in practice since it assumes known densities) is plotted in dark grey.

Main observations based on the simulation:

- For the kNDN classifier, the projection depth provides better results than the Mahalanobis depth in examples 3-10. Considering asymmetric cases, the local depth is appropriate in all four examples (although in example 10 it is slightly outperformed by the projection depth). Improvement gained by the use of the local depth is more visible for the larger training set (N=250).

- Let us evaluate examples 1, 3, and 5, in which the considered distributions are elliptically symmetric with equal characteristics of dispersion. In these situations, the compared classifiers perform similarly well. Slightly worse performance can be recorded for kNN in examples 1 and 3 irrespective of N, the Sym classifier in example 1 for $N = 50$, and the kNDN procedure in example 5 for $N = 50$.

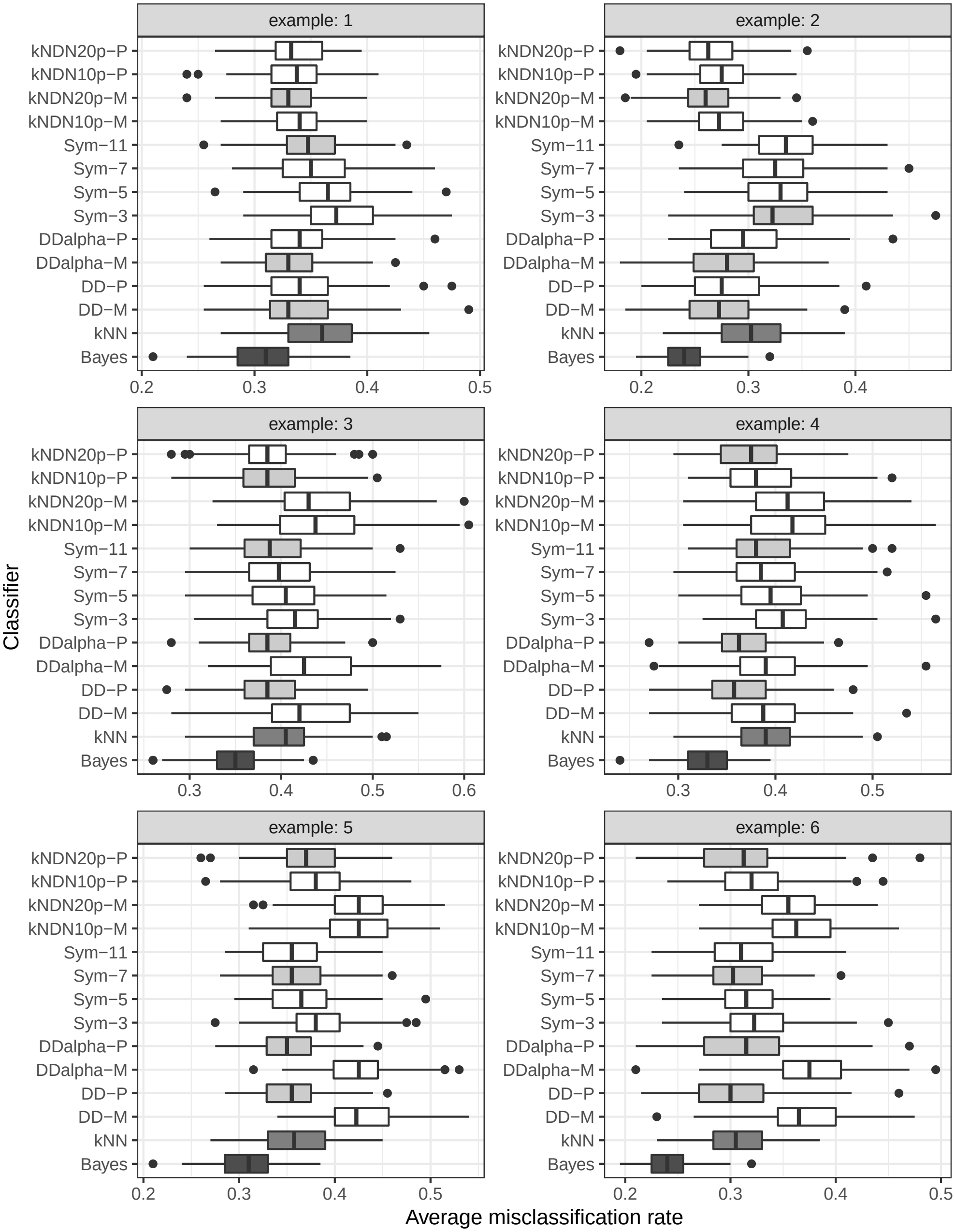

Fig. 2. Average misclassification errors, examples 1–6, $N = 50$.

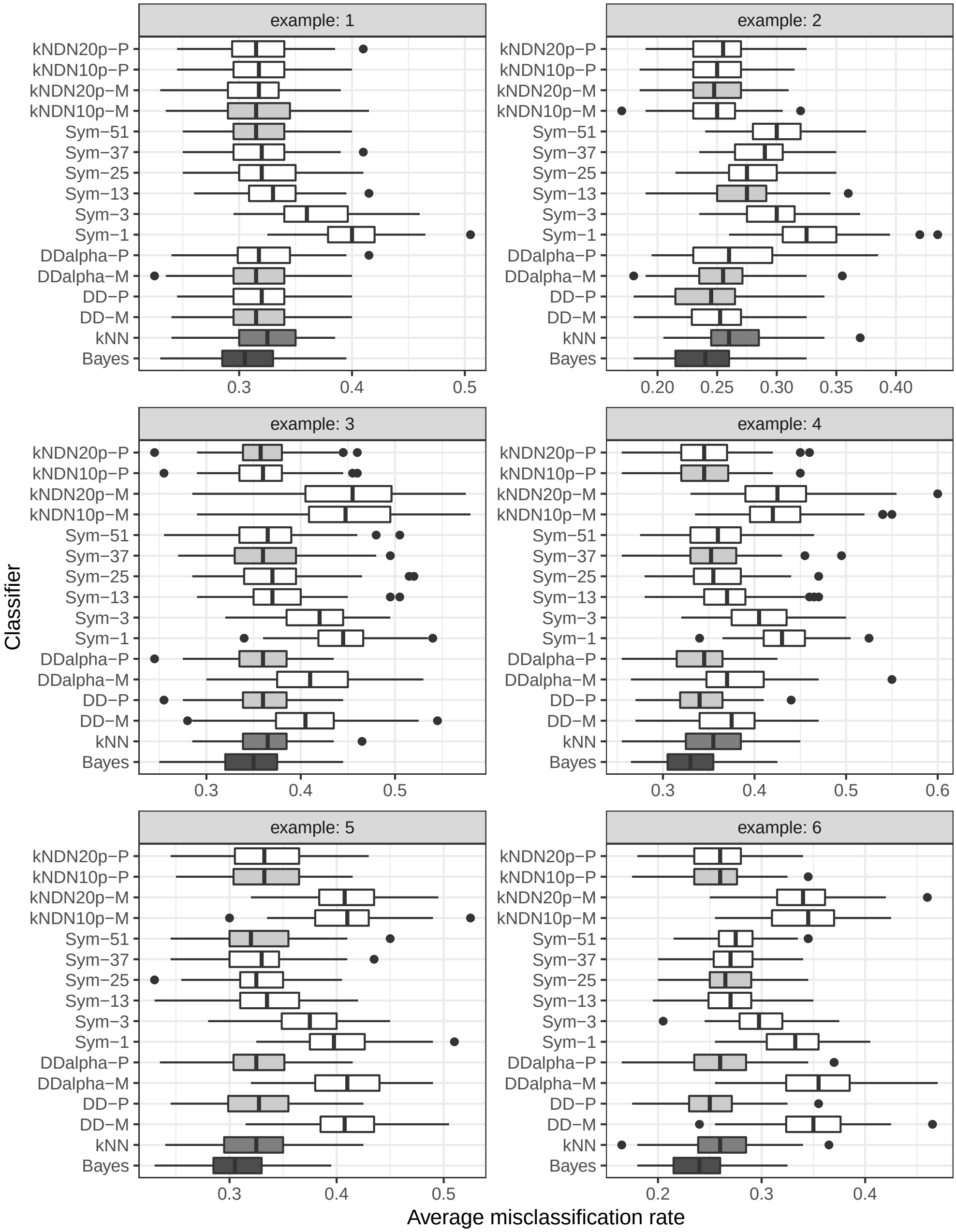

Fig. 3. Average misclassification errors, examples 1–6, $N = 250$.

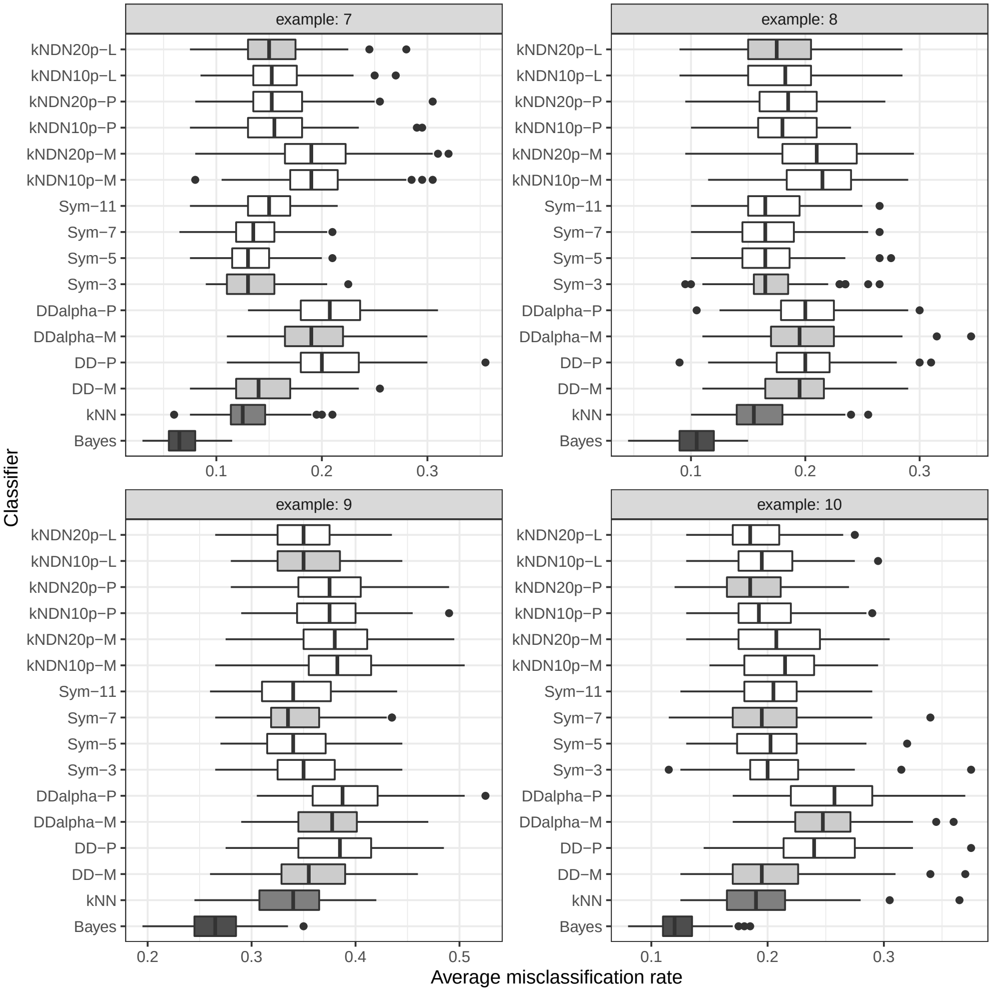

Fig. 4. Average misclassification errors, examples 7–10, $N = 50$.

- Consider now examples 2, 4, and 6, in which the considered distributions are elliptically symmetric but differ also in dispersion. The most significant differences are noticeable in example 2 in which kNN and Sym procedures perform worse than the other three classifiers. The kNDN classifier outperformed all its competitors in this example in the case $N = 50$. In examples 4 and 6, all classifiers perform similarly well. The kNDN classifier is slightly worse than the DD and the DDalpha and slightly better than the Sym classifier and the kNN in example 4, but the differences get smaller with increasing N.

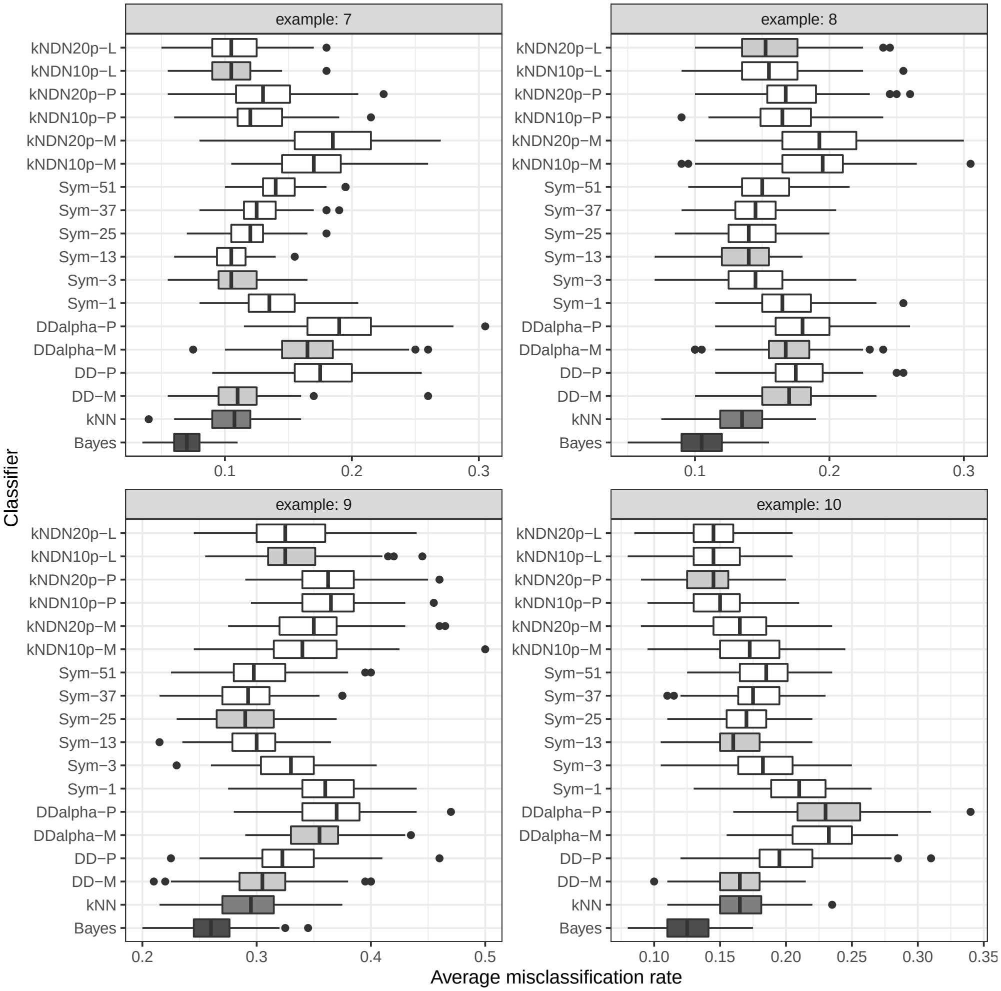

Fig. 5. Average misclassification errors, examples 7–10, $N = 250$.

- Regarding examples including asymmetric distributions (7–10), the kNDN overcomes the DDalpha in all cases. In example 7, kNDN is comparable to the DD, the Sym, and the kNN when $N = 250$, but is slightly worse than these classifiers when $N = 50$. In example 8, the ordering of classifiers (from the best one) is kNN, Sym, kNDN, DDalpha, and DD. The kNN and Sym classifiers outperform the other classifiers also in example 9. In example 10, the kNDN overcomes all its competitors when $N = 250$ and performs similarly well as DD, Sym, and kNN when $N = 50$.

4.3. Simulation in $\mathbb{R}^5$

We explored possibility of using the kNDN classifier also in higher-dimensional space. We considered two simple cases, both with equal priors:

- Example 1: two d-dimensional normal distributions with covariance matrix equal to identity matrix differing only in the mean value of the first coordinate (difference equal to one), i. e.

$$P_1 = N_d\left((0,0,\ldots,0)', \boldsymbol{I}_d\right) \quad P_2 = N_d\left((1,0,\ldots,0)', \boldsymbol{I}_d\right).$$

- Example 2: two d-dimensional normal distributions differing in the mean value of the first coordinate (difference equal to one), as well as in scale. More precisely:

$$P_1 = N_d\left((0,0,\ldots,0)', \boldsymbol{I}_d\right) \quad P_2 = N_d\left((1,0,\ldots,0)', 4\boldsymbol{I}_d\right).$$

First, we examined proportion of so called outsiders – points from the test set located out of the convex hulls of both groups of points in the training set. Table 2 shows increasing proportion of outsiders (see the last column of the table) implied by increasing dimension. This proportion also closely relates to the size of the training set – the larger the training set, the less outsiders. Therefore, we decided to work with training sets containing $N = 1000$ or $N = 4000$ points from each group.

We used function inhulln() from the R package geometry [12] for testing whether a given point lies within a convex hull of some other points. We found that the current implementation of this function works properly only to dimension 8 when 4000 points are considered. This determines the current limit of applicability of the newly suggested method.

Table 2 also includes information about time (in seconds) needed for computation of the values in the corresponding line. These values are based on 100 repetitions in which N points from each group (example 1) are generated and another 100 points from each group are tested (whether they belong to both convex hulls). Considering time demands, we decided to perform the simulation for dimension $d = 5$.

Results of the simulation study in five-dimensional space are shown in Figure 6. From this figure, we can conclude that all the compared classifiers perform similarly well in example 1, but different performance is recorded in example 2. In the second example, the Sym method as well as the kNN method lead to higher error rates, while the kNDN, DD and DDalpha remain close to the lowest achievable error rates (represented by the Bayes classifier).

4.4. Real-data examples

In this section, we show the performance of the newly proposed classifier on two well known datasets. Both of them were studied in [18] and [19], the later one was also studied in [24].

- The biomedical data first discussed by Cox, Johnson and Kafadar [4] are four-dimensional data divided into two groups. The first group consists of 127 subjects, the second one of 67 subjects. From these observations, we repeatedly generated training sets of 100 + 50 subjects and used the rest of the data as test sets.

dim	N	time [s]	in both	only one – correct	only one – false	outsiders
2	200	1	85.93	8.71	2.18	3.18
	1000	1	95.18	3.75	0.40	0.67
	4000	1	98.09	1.63	0.13	0.15
3	200	1	72.36	14.90	3.65	9.09
	1000	1	90.32	6.20	1.17	2.31
	4000	2	96.03	2.86	0.40	0.71
4	200	1	57.57	17.19	6.24	19.00
	1000	2	82.97	9.03	2.42	5.58
	4000	7	92.48	4.73	0.85	1.94
5	200	2	41.42	18.65	6.41	33.52
	1000	11	70.48	12.74	3.75	13.03
	4000	46	86.43	7.20	1.57	4.80
6	200	10	26.54	16.66	7.40	49.40
	1000	94	57.80	15.30	4.57	22.33
	4000	428	78.69	9.65	2.39	9.27
7	200	69	16.53	12.72	6.42	64.33
	1000	692	45.62	15.35	5.54	33.49
	4000	3392	68.63	11.68	3.46	16.23
8	200	445	9.06	9.52	4.90	76.52
	1000	5304	34.77	15.01	5.75	44.47
	4000	30335	58.83	13.00	4.27	23.90

Tab. 2. Example 1, section 4.3: Percentage of points lying inside of both convex hulls of points in training set (column: in both), only in the convex hull corresponding to the group from which the point was generated (only one – correct), only in the convex hull corresponding to the other group (only one – false), and to none of the convex hulls (outsiders).

- The blood transfusion data was first used by Yeh, Yang and Ting [28]. These data are three-dimensional and they are divided into two groups. The first group consists of 570 subjects, the second one of 178 subjects. We repeatedly generated training sets of 400 + 100 subjects and used the rest of the data as test sets.

We compared the same classifiers as in Section 4.1. For the depth-based classifier, we tried the projection and the Mahalanobis depth. For the kNDN method, we choose number of neighbours (k) equal to 5, 10, and 20, respectively. For the Sym method, we choose number of neighbours (k) equal to 1, 5, and 11, respectively. We recomputed the misclassification rates presented in the literature.

The results are presented in Table 3 which includes average misclassification rates (in %), standard errors (in brackets) and information which depth and which number of neighbours turned out to be the best choice in the considered situation.

The kNDN classifier achieved the 2nd highest average misclassification rate (it was outperformed by 3 competitors)in both cases. For the biomedical data, failure of the

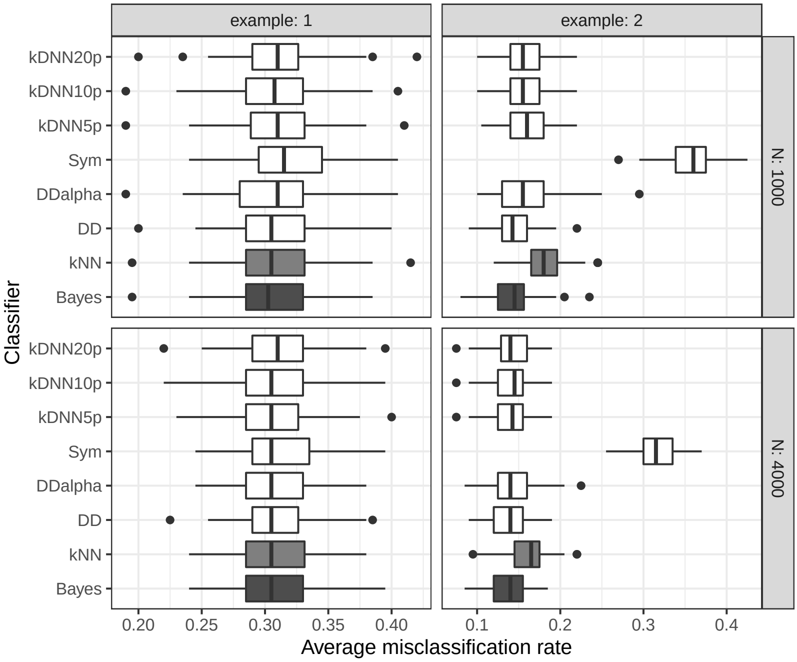

Fig. 6. Average misclassification errors, examples in Section 4.3.

Sym classifier is evident while the other four classifiers do not differ much – the difference between the best one (DDalpha) and the 4th one (kNDN) is less than 1%. For the blood transfusion data, classical kNN method was surprisingly the worst of the considered methods while the best results were achieved by the DDalpha and the Sym classifier.

5. CONCLUDING REMARKS

The newly proposed k nearest depth neighbour (kNDN) method is an alternative to the methods based on density estimation. We have shown that, in contrast to the classical kNN, it can utilize global properties of the considered distributions like their symmetry. In contrast to the maximal depth classifier and related classifiers, the kNDN method does not have problems with classification when the considered distributions differ in the dispersion.

In the simulation study, it was shown that the newly suggested classifier perform competitively and is able to overcome the other depth-based classifiers in some situations.

There are already two different classifiers that combine depth-based classification and the k-nearest neighbour procedure, see [24] and [27]. It is useful to realise that these approaches are based on different notions of the neighbourhood. Vencalek [27] uses depth transformation followed by the classical kNN in the DD-space and, therefore, the neighbouring points are those with similar depths w.r.t. both (all) distributions.

Dataset	kNN	DD	DDalpha	symkNN	kNDN
Biomedical	14.32	13.89	13.84	19.11	14.82
	(0.45)	(0.47)	(0.48)	(0.53)	(0.45)
		Mahal.	Mahal.	$k = 1$	proj., $k = 20$
Blood transfusion	29.74	28.10	26.80	26.83	28.94
	(0.13)	(0.23)	(0.22)	(0.15)	(0.24)
		proj.	Mahal.	$k = 11$	proj., $k = 20$

Tab. 3. Average misclassification rates (in %) with standard errors (in brackets); comparison of classifiers on real datasets.

Paindaveine and Van Bever [24] use the idea of symmetrisation and, therefore, the neighbouring points are those with the highest depth w.r.t. the symmetrical distribution with a centre at a point which has to be classified. In the current paper, we use a distributional neighbourhood defined as an area of points with similar depth, w.r.t. a given distribution, as the studied point.

APPENDIX

Proof of Theorem 3.2: Let us consider a fixed point $\boldsymbol{x} \in \mathbb{R}^d$. Throughout the proof, we denote Mahalanobis distance based neighbourhoods $O_i^M(\boldsymbol{x}, C_i(j))$ by $O_i(C_i(j))$ for simplicity. We want to show that

$$\forall \epsilon > 0 \; \exists j_0 \in \mathbb{N} : \; j \geq j_0 \Rightarrow \mathrm{P}\left(\left| \frac{\lambda_d(O_1(C_1(j)))}{\lambda_d(O_2(C_2(j)))} / \frac{\pi_2 f_2(\boldsymbol{x})}{\pi_1 f_1(\boldsymbol{x})} - 1 \right| > \epsilon \right) < \epsilon. \tag{15}$$

For any given $\epsilon > 0$ we can find constant $c_0(\epsilon) > 0$ such that $\frac{g_i(M_i + c_0(\epsilon))}{g_i(M_i)} > 1 - \epsilon$ for both $i = 1, 2$. Notice that this inequality implies $\frac{g_i(M_i + c)}{g_i(M_i)} > 1 - \epsilon$ for both $i = 1, 2$ for all $c \in [0, c_0(\epsilon)]$. Denote $p(\epsilon) := \min\{\pi_1 P_1(O_1(c_0(\epsilon))), \pi_2 P_2(O_2(c_0(\epsilon)))\}$.

Assume $j \in \mathbb{N}$ to be large enough to ensure

(A1) $k_j / n_j < p(\epsilon)/2$ and

(A2) $k_j^{-1/4} < \epsilon$.

In the three following steps, we show that for any $j \in \mathbb{N}$ satisfying these two assumptions the inequality in (15) holds. Since now assume j to be fixed (satisfying conditions above) and we write k, n and $C_i, i = 1, 2$ instead of k_j, n_j and $C_i(j), i = 1, 2$ for simplicity.

Step 1:
We can find positive (uniquely determined) constants c_1 and c_2 such that

$$\pi_1 P_1(O_1(c_1)) = k/n = \pi_2 P_2(O_2(c_2)). \tag{16}$$

Obviously $0 < c_i < c_0$ for both $i = 1, 2$.

Now

$$\frac{\frac{\lambda_d(O_1(c_1))}{\lambda_d(O_2(c_2))}}{\frac{\pi_2 f_2(\boldsymbol{x})}{\pi_1 f_1(\boldsymbol{x})}} = \frac{\frac{|\boldsymbol{\Sigma}_1|^{1/2}\left[(M_1+c_1)^d-M_1^d\right]}{|\boldsymbol{\Sigma}_2|^{1/2}\left[(M_2+c_2)^d-M_2^d\right]}}{\frac{\pi_2|\boldsymbol{\Sigma}_2|^{-1/2}g_2(M_2)}{\pi_1|\boldsymbol{\Sigma}_1|^{-1/2}g_1(M_1)}} = \frac{\pi_1 g_1(M_1)\left[(M_1+c_1)^d-M_1^d\right]}{\pi_2 g_2(M_2)\left[(M_2+c_2)^d-M_2^d\right]}. \tag{17}$$

Using the equation (16) the ratio (17) can be written as

$$\frac{\pi_2 P_2(O_2(c_2))}{\pi_2 g_2(M_2)\left[(M_2+c_2)^d-M_2^d\right]} \cdot \frac{\pi_1 g_1(M_1)\left[(M_1+c_1)^d-M_1^d\right]}{\pi_1 P_1(O_1(c_1))} =$$

$$= \frac{\int_{M_2}^{M_2+c_2} g_2(r)r^{d-1}\mathrm{d}r}{g_2(M_2)\left[(M_2+c_2)^d-M_2^d\right]} \cdot \frac{g_1(M_1)\left[(M_1+c_1)^d-M_1^d\right]}{\int_{M_1}^{M_1+c_1} g_1(r)r^{d-1}\mathrm{d}r}. \tag{18}$$

Now we can find upper bound for this ratio (and analogous lower bound). Since $g_i(\cdot)$ are decreasing functions, it holds $g_2(r) > g_2(M_2+c_2)$ for all $r \in [M_2, M_2+c_2)$ and $g_1(r) < g_1(M_1)$ for all $r \in (M_1, M_1+c_1)$. Hence (18) is bounded from below by

$$\frac{\int_{M_2}^{M_2+c_2} g_2(M_2+c_2)r^{d-1}\mathrm{d}r}{g_2(M_2)\left[(M_2+c_2)^d-M_2^d\right]} \cdot \frac{g_1(M_1)\left[(M_1+c_1)^d-M_1^d\right]}{\int_{M_1}^{M_1+c_1} g_1(M_1)r^{d-1}\mathrm{d}r} = \frac{g_2(M_2+c_2)}{g_2(M_2)} \cdot \frac{g_1(M_1)}{g_1(M_1)} > 1-\epsilon.$$

Similarly, the upper bound for the ratio can be computed.

Step 2:
We find positive constants c_1^L, c_1^U and c_2^L, c_2^U such that

$$\begin{aligned} \pi_1 P_1\left(O_1(c_1^L)\right) &= \tfrac{k-k^{3/4}}{n} &&= \pi_2 P_2\left(O_2(c_2^L)\right), \\ \pi_1 P_1\left(O_1(c_1^U)\right) &= \tfrac{k+k^{3/4}}{n} &&= \pi_2 P_2\left(O_2(c_2^U)\right). \end{aligned}$$

These constants are again unique and less than c_0.

Now consider any constant $c_1^S \in [c_1^L, c_1^U]$ and $c_2^S \in [c_2^L, c_2^U]$. We do not assume $\pi_1 P_1\left(O_1(c_1^S)\right) = \pi_2 P_2\left(O_2(c_2^S)\right)$. Nevertheless, it can proved that the ratio $\frac{\lambda_d\left(O_1\left(c_1^S\right)\right)}{\lambda_d\left(O_2\left(c_2^S\right)\right)} / \frac{\pi_2 f_2(\boldsymbol{x})}{\pi_1 f_1(\boldsymbol{x})}$ is close to one.

We can proceed similarly as in the first step:

$$\frac{\lambda_d\left(O_1\left(c_1^S\right)\right)}{\lambda_d\left(O_2\left(c_2^S\right)\right)} / \frac{\pi_2 f_2(\boldsymbol{x})}{\pi_1 f_1(\boldsymbol{x})} = \frac{\pi_1 g_1(M_1)\left[(M_1+c_1^S)^d-M_1^d\right]}{\pi_2 g_2(M_2)\left[(M_2+c_2^S)^d-M_2^d\right]}. \tag{19}$$

The fraction can be extended by

$$\frac{\pi_1 P_1(O_1(c_1^S))}{\pi_1 P_1(O_1(c_1^S))} \frac{\pi_2 P_2(O_2(c_2^S))}{\pi_2 P_2(O_2(c_2^S))} \frac{\pi_2 P_2(O_2(c_2))}{\pi_1 P_1(O_1(c_1))},$$

where the last term is equal to one from (16). After a convenient arrangement we get (19) equals to

$$\frac{g_1(M_1)\left[(M_1+c_1^S)^d-M_1^d\right]}{P_1(O_1(c_1^S))} \cdot \frac{P_1(O_1(c_1^S))}{P_1(O_1(c_1))} \cdot \frac{P_2(O_2(c_2^S))}{g_2(M_2)\left[(M_2+c_2^S)^d-M_2^d\right]} \cdot \frac{P_2(O_2(c_2))}{P_2(O_2(c_2^S))}.$$

Ratios $\frac{\mathrm{P}(O_i(c_i^S))}{\mathrm{P}(O_i(c_i))}$ are not greater than $\frac{(k+k^{3/4})/n}{k/n} = 1 + k^{-1/4}$ and not smaller than $\frac{(k-k^{3/4})/n}{k/n} = 1 - k^{-1/4}$. Recall that k is so big that $k^{-1/4} < \epsilon$. The first and the third term are both bounded similarly as the ratio in the step 1.

The considered ratio is thus not greater than $\frac{(1+\epsilon)^2}{1-\epsilon}$ and not less than $\frac{1-\epsilon}{(1+\epsilon)^2}$.

Step 3:
We show that $C_i \in [c_i^L, c_i^U]$ with probability greater than $1-2\epsilon$ both for $i = 1, 2$. Consider a random sample of n points from the mixture P (some of the randomly sampled points are from P_1 and some are from P_2).

Let $Z_i^L, i = 1, 2$, denote numbers of points from P_i lying in $O_i(c_i^L)$. $Z_i^L, i = 1, 2$, are binomial random variables: $Z_i^L \sim \mathrm{Bi}\left(\frac{k-k^{3/4}}{n}, n\right)$. Let $Z_i^U, i = 1, 2$, denote numbers of points from P_i lying in $O_i(c_i^U)$. $Z_i^U, i = 1, 2$, are binomial random variables: $Z_i^U \sim \mathrm{Bi}\left(\frac{k+k^{3/4}}{n}, n\right)$.

Obviously $C_i \notin [c_i^L, c_i^U]$ iff either $Z_i^L > k$ (in that case $C_i < c_i^L$) or $Z_i^U < k$ (in that case $C_i > c_i^U$). Now

$$\mathrm{P}\left(Z_i^L > k\right) = \mathrm{P}\left(\frac{Z_i^L - EZ_i^L}{SD(Z_i^L)} > \frac{k - (k - k^{3/4})}{\sqrt{(k - k^{3/4})(1 - \frac{k-k^{3/4}}{n})}}\right).$$

The standard deviation of the considered binomial distribution is smaller than $k^{1/2}$, hence

$$\mathrm{P}\left(Z_i^L > k\right) < \mathrm{P}\left(\frac{Z_i^L - EZ_i^L}{SD(Z_i^L)} > k^{1/4}\right) \le k^{-1/2} < \epsilon,$$

where the second inequality follows from the Chebyshev's inequality and the last inequality follows from the assumption (A2).

Similarly it can be shown that $\mathrm{P}\left(Z_i^U < k\right) < \epsilon$. Hence $P\left(C_i \in [c_i^L, c_i^U]\right) > 1 - 2\epsilon$. □

ACKNOWLEDGEMENT

This work was partially supported the grant PRIMUS/17/SCI/03 Advanced geometric methods in statistics and by the grant of The Ministry of Education, Youth and Sports CZ.02.1.01/0.0/0.0/17_049/0008408.

(Received March 15, 2019)

REFERENCES

[1] C. Agostinelli and M. Romanazzi: Local depth. J. Statist. Plann. Inference *141* (2011), 817–830. DOI:10.1016/j.jspi.2010.08.001

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa: The quickhull algorithm for convex hulls. ACM Trans. Math. Software (TOMS) *22* (1996), 4, 469–483. DOI:10.1145/235815.235821

[3] A. Christmann and P. J. Rousseeuw: Measuring overlap in binary regression. Comput. Statist. Data Analysis *37* (2001), 65–75. DOI:10.1016/S0167-9473(00)00063-3

[4] L. H. Cox, M. M. Johnson, and K. Kafadar: Exposition of statistical graphics technology. In: ASA Proc Stat. Comp Section 1982, pp. 55–56. DOI:10.1016/S0167-9473(00)00063-3

[5] S. Dutta and A. K. Ghosh: On classification based on Lp depth with an adaptive choice of p. Preprint, 2011.

[6] S. Dutta and A. K. Ghosh: On robust classification using projection depth. Ann. Inst.Statist. Math. *64* (2012), 3, 657–676. DOI:10.1007/s10463-011-0324-y

[7] S. Dutta, A. K. Ghosh, and P. Chaudhuri: Some intriguing properties of Tukey's halfspace depth. Bernoulli *17* (2011), 4, 1420–1434. DOI:10.3150/10-BEJ322

[8] E. Fix and J. L. Hodges: Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties. Technical Report 4, Randolph Field, Texas: USAF School of Aviation Medicine, 1951.

[9] R. Fraiman, R. Y. Liu, and J. Meloche: Multivariate density estimation by probing depth. Lecture Notes-Monograph Series 1997, pp. 415–430.

[10] A. K. Ghosh and P. Chaudhuri: On maximum depth and related classifiers. Scand. J. Statist. *32* (2005), 327–350. DOI:/10.1111/j.1467-9469.2005.00423.x

[11] A. K. Ghosh and P. Chaudhuri: On data depth and distribution-free discriminant analysis using separating surfaces. Bernoulli *11* (2005), 1, 1–27. DOI:10.3150/bj/1110228239

[12] K. Habel, R. Grasman, R. B. Gramacy, P. Mozharovskyi, and D. C. Sterratt: Geometry: Mesh Generation and Surface Tessellation. R package version 0.4.5. `https://CRAN.R-project.org/package=geometry`

[13] D. Hlubinka, L. Kotík, and O. Vencálek: Weighted data depth. Kybernetika *46* (2010), 1, 125–148.

[14] M. Hubert and S. van der Veeken: Fast and robust classifiers adjusted for skewness. In: COMPSTAT 2010: Proceedings in Computational Statistics: 19th Symposium held in Paris 2010 (Y. Lechevallier and G. Saporta, eds.), Springer, Heidelberg 2010, pp. 1135–1142.

[15] R. Jörnsten: Clustering and classification based on the L_1 data depth. J. Multivar. Anal. *90* (2004), 67–89.

[16] L. Kotík and D. Hlubinka: A weighted localization of halfspace depth and its properties. J. Multivar. Anal. *157* (2017), 53–69. DOI:10.1016/j.jmva.2017.02.008

[17] D. Kosiorowski and Z. Zawadzki: DepthProc An R Package for Robust. Exploration of Multidimensional Economic Phenomena, 2020.

[18] T. Lange, K. Mosler, and P. Mozharovskyi: Fast nonparametric classification based on data depth. Statist. Papers *55* (2014), 1, 49–69.

[19] J. Li, J. A. Cuesta-Albertos, and R. Y. Liu: DD-classifier: Nonparametric classification procedure based on DD-plot. J. Amer. Statist. Assoc. *107* (2012), 498, 737–753. DOI:10.1080/01621459.2012.688462

[20] R. Y. Liu: On a notion of data depth based on random simplices. Ann. Statist. *18* (1990), 1, 405–414. DOI:10.1214/aos/1176347507

[21] R. Y. Liu, J. M. Parelius, and K. Singh: Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion). Ann. Statist. *27* (1999), 783–858. DOI:10.1214/aos/1018031260

[22] K. Mardia, J. Kent, and J. Bibby: Multivariate Analysis. Academic Press, 1979.

[23] D. Paindaveine and G. Van Bever: From depth to local depth: a focus on centrality. J. Amer. Statist. Assoc. *105* (2013), 1105–1119.

[24] D. Paindaveine and G. Van Bever: Nonparametrically consistent depth-based classifiers. Bernoulli *21* (2015), 1, 62–82. DOI:10.3150/13-BEJ561

[25] O. Pokotylo, P. Mozharovskyi, and R. Dyckerhoff: Depth and depth-based classification with R package ddalpha. J. Statist. Software *91* (2019), 5, 1–46.

[26] R. Serfling: Depth functions in nonparametric multivariate inference. In: Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications (R. Y. Liu, R. Serfling, and D. L. Souvaine, eds.), American Mathematical Society, DIMACS Series in Discrete Mathematics and Theoretical Computer Science *7*, New York 2006, pp. 1–16.

[27] O. Vencalek: k-Depth-nearest neighbour method and its performance on skew-normal distributons. Acta Univ. Palacki Olomouc., Fac. Rer. Nat., Mathematica *52* (2013), 2, pp. 121–129.

[28] I. C. Yeh, K. J. Yang, and T. M. Ting: Knowledge discovery on RFM model using Bernoulli sequence. Expert Systems Appl. *36* (2009), 5866–5871. DOI:10.1016/j.eswa.2008.07.018

[29] A. Zakai and Y. Ritov: Consistency and localizability. J. Machine Learning Res. *10* (2009), 827–856.

[30] Y. Zuo and R. Serfling: General notion of statistical depth function. Ann. Statist. *28* (2000), 461–482. DOI:10.1214/aos/1016218226

[31] Y. Zuo and R. Serfling: Structural properties and convergence results for contours of sample statistical depth functions. Ann. Statist. *28* (2000) 2, 483–499. DOI:10.1214/aos/1016218227

Ondřej Vencálek, Palacky University, Faculty of Science, Department of Mathematical Analysis and Applications of Mathematics, 17. listopadu 12, 771 46 Olomouc. Czech Republic.

e-mail: ondrej.vencalek@upol.cz

Daniel Hlubinka, Charles University, Faculty of Mathematics and Physics, Department of Probability and Mathematical Statistics, Sokolovská 83, 186 75 Praha 8. Czech Republic.

e-mail: hlubinka@karlin.mff.cuni.cz