1. The Distance Function Optimization for the Near Neighbors-based Classifiers

MARCEL JIŘINA*, Institute of Computer Science of the Czech Academy of Sciences, Czech Republic
SAID KRAYEM, Faculty of Applied Informatics, Tomas Bata University, Czech Republic

Based on the analysis of conditions for a good distance function we found four rules that should be fulfilled. Then we introduce two new distance functions, a metric and a pseudometric one. We have tested how they fit for distance-based classifiers, especially for the IINC classifier. We rank distance functions according to several criteria and tests. Rankings depend not only on criteria or nature of the statistical test, but also whether it takes into account different difficulties of tasks or whether it considers all tasks as equally difficult. We have found that the new distance functions introduced belong among the four or five best out of 23 distance functions. We have tested them on 24 different tasks, using the mean, the median, the Friedman aligned test and the Quade test. Our results show that a suitable distance function can improve behavior of distance-based classification rules.

CCS Concepts: • **Information systems** → **Nearest-neighbor search**; • **Mathematics of computing** → *Nonparametric statistics.*

Additional Key Words and Phrases: near neighbors, classification, distance function, metric

1 INTRODUCTION

In this work we deal with distances in a multidimensional space. Especially, we deal with the distance from the given point (the query point x) in multidimensional space R^d. In this way, the d-dimensional information is simplified to one-dimensional information, the distance. This is the cost we pay for this simplification. We show that this cost is not too high. The nearest neighbor method [5] remains popular and surprisingly effective and fairly often used up till now [1], [20], [22], [24], [30], [36], [38]. There is a lot of methods of classification based on the nearest neighbors [33]. The methods estimate the probability density at point x of the data space by ratio i/V_i of number i of points of a given class in a suitable ball of volume V_i with its center at point x [11]. These methods often optimize the best size of the neighborhood, i.e. the number i of points in the neighborhood of the point x or size of volume V_i. Nearest neighbors methods use various means to enhance the classification quality. One of them is setting up weights of individual features via learning, e.g. [9], [26], [32], [34], [35]. Li, Chen and Chen [21] search for local probability centers rather than for local class density. There are techniques for improving the speed of nearest neighbors search, see [37], [31], techniques for dealing with high dimensional data [25], techniques for dealing with missing feature values [3], techniques for dealing with uncertain data [2]. In [24] a guideline how to apply k-NN for classification tasks with the use of software package are given. Also, a proper selection of k is shortly discussed.

*Both authors contributed equally to this research.

Authors' addresses: Marcel Jiřina, marcel@cs.cas.cz, Institute of Computer Science of the Czech Academy of Sciences, Pod Vodárenskou věží 2, Prague, Czech Republic, 18207; Said Krayem, Faculty of Applied Informatics, Tomas Bata University, Nad Stranemi 4511, Zlin, Czech Republic, drsaid@seznam.cz.

1556-4681/2022/2-ART $15.00
https://doi.org/https://doi.org/10.1145/3434769

Here we use the Inverted Indexes of Neighbors Classifier (IINC) [17], see Chap. 2.4. It belongs among simple classifiers (data separators), eventually regressors, like the nearest neighbor rules 1-NN and k-NN or the naive Bayes classifier. The IINC method is distance-based as 1-NN and k-NN. Thus, distance plays a crucial role and is given by a distance function, mostly metric in the R^d. Another crucial role is played by the effective dimension of the data space. This is unknown for the 1-NN and k-NN methods. The IINC estimates probability of a class of a pattern or sample (a query point) according to a proper sum of reciprocals of indexes of neighbors of this sample; the nearest neighbor has index 1, the second nearest has index 2 etc. It can be shown that with the use of reciprocals of indexes we model scaling features of data and, at the same time, we correct errors caused by random placement of neighbors close to the query point. The ranking brings a similar advantage that has ranking statistics over standard statistics. Small variations are hidden in the same rank and eventually large changes influence the rank. Thus, there is a kind of robustness that brings also low classification error. The complexity of this method for large data set is given by $N \log N$, where N is the size of the learning set.

The behavior of the method depends on the distance function, usually a metric in the R^d. It appears that rather uncommon metrics are the best. In study [38], the six distance metrics are considered. An interesting finding is that for $k \geq 6$ the differences was essentially negligible for particular task of mapping of species-level biomass. For smaller k the differences among distance functions are important. It can be found that an L_1 (taxicab or Manhattan) metric is very good. The Euclidean metric L_2 can be considered acceptable. On the other hand, for a particular task the IINC or the k-NN, with a distance function that generally (say, on average) is not good enough may work best. It was shown [1], [38] that an uncommon metric is much better than the L_1 and L_2 metrics. It applies to the 1-NN as well as the IINC rules. A distance function may be dependent on rotation. The only rule that does not depend on rotational position of the coordinate system is the Euclidean metric. Another uncommon thing is that in some metrics a ball (or circle in R^2) with a finite radius r need not have a finite volume at least for some values of r. Also, the distance function need not be a metric. A pseudo-metric that does not conform to the triangle inequality is sufficient.

1.1 Problem formulation

There is a crucial fact that behavior of k-NN classifiers, i.e. in the end the classification error, is influenced by the concept of the classifier and by a distance function, usually the metric used. Recently the Hassanat metric was published in the Journal of American Science (Marsland Press) [13]. It can be shown that it is one of the best distance functions for distance-based classifiers. The other factor is the design of the nearest neighbor scheme. This is often limited to proper selection of k in k-NN classifier, but more sophisticated schemes were published. One of them was published by Samworth in The Annals of Statistics (IMS) [32], the other is called the IINC and was published in the Journal of Classification (Springer) [17].

It was found that the use of the Hassanat metric with the IINC rule gives very good results compared with other distance functions and other nearest neighbor rules [13]. A question arises if there is another distance function giving generally better results. This question leads to another question, what are the conditions or rules for a "good" distance function.

It can be found (see [7] and [8]) that the various distance functions have been designed according to certain specific needs, but none of them in terms of classification accuracy.

We present here a systematic analysis conductive to the conditions that the distance function should meet in order to be a generally optimal distance function for distance-based classifiers. However, this requirement leads to a condition that cannot be fully met. It can only be partially accomplished. Therefore, this condition should include that the classification error must be minimal at least close to the threshold between classes. At the same time, this condition can be found too broad. Then the distance function should follow major features, which have

known distance functions that provide the best classification error. We are, therefore, looking for the conditions that the new distance function shoud have.

This is derived in Chaps. 2.5 and 2.6 and summarized in Chap. 2.7. Suggestions for two new distance functions are given in Section 3.

We concentrate here on distance functions and the IINC rule, as this rule was found to be the best among the other k-NN rules [1], [14].

At the same time, these references show that the distance functions have the same relative effect on the quality of classification for IINC and for k-NN.

Our experience with practical tasks also leads to this. We use it without considering any data preprocesing, features or class weighting often used to get better results with a given rule and a distance function, usually the Euclidean metric. Of course, the techniques mentioned can be applied with success.

Based on considerations above we present here two new distance functions. We prove that the first one is a metric one and the other is a pseudometric one. We show some of their features, especially we show the fact that a ball in R^d (a circle) has a very uncommon form and its volume can be infinite. Then we explain the IINC method for classification. Using the IINC method and 23 different distance functions, we analyze how the classification error depends on the distance function used. For this purpose we have tested the classifier on a set of 24 tasks. Having amassed enough data, we use simple comparisons like mean and median and the Friedman aligned test and the Quade test [6] for multiple comparisons. Finally we rank the distance functions according to these criteria and tests getting a set of four or five best distance functions. The set includes also new distance functions introduced here.

2 ANALYSIS

2.1 The data sets

Data set is usually represented as a real matrix with d+1 columns and N rows. Each of columns 1 till d corresponds to a feature, the column No. $d+1$ contains a class mark. Each row corresponds to one sample, point, pattern, eventually event that consists of values of d features and a class mark. Generally, there is no ordering of rows, i.e. of samples; they follow one after another randomly and are indexed $1, 2, \ldots N$. Samples are often called points as each sample can be viewed as a point in a d-dimensional space. The learning set U has total $N = N(U)$ samples. With classes the set U is decomposed into disjoint sets $U_c; U = \cup_{c=1}^{C} U_c, U_c \cap U_b = \emptyset; c, b \in 1, 2, \ldots, C; c \neq b$, C is the number of classes. Let the cardinality of set U_c be $N_c; \sum_{c=1}^{C} N_c = N$. Moreover, there is a testing set V such that $U \cap V = \emptyset$.

2.1.1 Learning and testing set. There is a different terminology used in literature. Here we speak about the learning set used for setting up the classifier, and the testing set (checking set) "never seen before" for evaluation of true classification capability. The learning set can be divided into the training set really used in setting up the classifier and validating set for validation of the previous setting cycle. We do not use this option here.

2.1.2 Indexing data and neighbors' distances. As we need to express which sample is closer or further from some given sample x, we can rank samples of the learning set according to distance r_i of sample x_i from sample x. Therefore, let samples of U be indexed (ranked) so that for any two samples $x_i, x_j \in U$ there is $i < j$ if $r_i < r_j; i, j = 1, 2, \ldots N$, and class $U_c = \{x_i \in U | T(x_i) = c\}$. Of course, ranking depends on sample x and eventually metrics in R^d.

From now on we use numbering of samples according to their order as neighbors of sample x; x_i being the i-th nearest neighbor of sample x.

2.2 Metrics in R^d

A natural metric in R^d is the Euclidean metric and (R^d, ρ_E) is a metric space. At the same time one uses another metric ρ_x. This metric we use for stating distances between points in R^d, for stating ranks of neighbors, and for defining geometrical objects. Their areas and volumes are computed in a standard way. For example, in (R^2, L_1), i.e. in a two-dimensional space with Manhattan or taxicab metric, a "circle" with unit radius has the form of a diamond with edge $\sqrt{2}$. Its area computed in a standard way is equal to 2, whereas a circle in (R^2, L_2) with unit radius has an area equal to π.

2.3 Measure-dependency

Metric as well as semimetric can be measure-dependent and then values of coordinates should be normalized. Normalization (sometimes called standardization) means transforming each feature, taken as a random variable, into a new variable with zero mean and unit standard deviation. Thus, individual features are comparable. Normalization is a practical issue; in analysis it is usually supposed that all features are of the same nature.

2.4 IINC rules

A basic notion of the IINC rules is the distribution mapping exponent q. Suppose a fixed point x and its distance to its nearest neighbor, the second nearest neighbor etc. The distribution of individual points is given by the probability distribution of points in space R^d. Imagine a graph where the order number of a neighbor as a function of distance from point x is depicted. We call this dependence the distribution mapping function. If logarithmic scales on both coordinates are used, points are arranged approximately along a straight line. The slope of this straight line is called the distribution mapping exponent q. It is, in fact, the scaling exponent in the sense of the theory of fractals [23]. It can be proved that the average of all distribution mapping functions formed for all points of a set, is the correlation integral [12] and mean of distribution mapping exponents is close to the correlation dimension. It is clear that the distributon mapping exponent q describes behavior of data locally around point x and the correlation dimension ν globally for the whole set considered. This is the basis for the IINC rules.

The local method (the L-rule) [15] is given by formula for probability that point x is of class c for $x \neq x_1$ (x_1 is the first nearest neighbor of x)

$$\hat{p}(c|x) = \frac{\sum_{i:x_i \in U_c} r_i^{-q}}{\sum_{i:x_i \in U} r_i^{-q}} \tag{1}$$

and for $x = x_1$ there is

$$c(x) = c(x_1). \tag{2}$$

Here the two-class problem with the same a priori probabilities of both classes is considered. In this formula the sum in the numerator goes over points of class c (set U_c), and the sum in the denominator goes over all points of the learning set U.

The global method (G-rule) [16] is given by a similar formula with correlation dimension ν instead of the distribution mapping exponent q.

The IINC rule uses the order numbers of neighbors (ranks) instead of r^q. It holds [17] for $x \neq x_1$

$$\hat{p}(c|x) = \frac{\sum_{i:x_i \in U_c} 1/i}{\sum_{i:x_i \in U} 1/i}, \tag{3}$$

and for $x = x_1$ there is

$$c(x) = c(x_1). \tag{4}$$

Note. Equation (3) can be written in form

$$\hat{p}(c|x) = \frac{S_c}{S_c + S_{\bar{c}}} = \frac{S_c}{H_N}, \tag{5}$$

where

$$S_c = \sum_{i:x_i \in U_c} 1/i$$

and H_N is the N-th harmonic number.

We consider the same a priori probability of both classes here. It means, in practice, the same number of patterns of all classes in the learning set. If it is not so, the imbalance problem arises. This problem can be solved using "recomputation to one pattern", i.e. by dividing individual S_c's by the number of patterns N_c of this class. In this way, we get formula for any number of classes numbered $0, 1, \ldots C-1$, where C is the number of classes

$$\hat{p}(c|x) = \frac{\frac{1}{N_c} S_c}{\sum_{i=0}^{C-1} \frac{1}{N_i} S_i}. \tag{6}$$

Bear in mind that the denominator is not a harmonic number here.

It can be proven that the IINC represents the best neighbor class weighting scheme among all the k-NN rules. The proof shows that any deviation of weight of the k-th neighbor leads to an enlargement of the classification error. Representation of rank of a neighbor by the sum of the Heaviside step functions of differences of neighbors distances is used in the proof. For the Heaviside step function a smooth approximation is used. Then a formula for the classification error of the IINC rule and the rule with slightly modified weight are compared. From this comparison it follows that any change of weight of a neighbor - positive or negative - leads to an enlargement of the classification error.

2.5 The distance function and conditions for optimization

For optimization of the distance function we use the standard procedure of setting up the first derivative of the function minimized, the error function, to zero. Naturally, unless the assumption of continuous second partial derivatives and concave form in the neighborhood of the minimum, i.e. positive second derivative.

The classification error can be defined as

$$E_c = \frac{1}{||V||} \int_{x \in V} \left[h(\hat{p}(c|x) - 1/2) - c_x \right]^2 f(x) dV, \tag{7}$$

where $h(.)$ is the Heaviside step function, c_x is the class of pattern x, and $p(c|x)$ is the classifier's estimate of the probability that pattern x is of class c_x; we write $\hat{p}$ for short later. This is a functional over the function that gives an error of one query pattern times local density $f(x)$,

$$L = [h(\hat{p} - 1/2) - c_x]^2 f(x). \tag{8}$$

In the following we omit the local density. The Heaviside step function can be approximated by

$$h(x) = \frac{1}{2}\left(1 + \frac{x}{\sqrt{\epsilon^2 + x^2}}\right), \tag{9}$$

where ϵ is the smoothing factor. The derivative is

$$\frac{\partial h}{\partial x} = \frac{\epsilon^2}{\sqrt{(\epsilon^2 + x^2)^3}} = \epsilon^2 D, \tag{10}$$

where

$$D = \frac{1}{\sqrt{(\epsilon^2 + x^2)^3}}. \tag{11}$$

We also write

$$S_c = \sum_{k:x_k \in U_c} \frac{1}{r_k}, \tag{12}$$

where r_k is the rank of the pattern No. k from the query point x and it holds

$$r_k = \sum_{j:x_j \in U} h[\rho(x, x_k) - \rho(x, x_j)]. \tag{13}$$

(To get rank of x_k we sum out all points x_i that are closer to point x than is the x_k. Then the distance $\rho(x, x_j) < \rho(x, x_k)$ and the Heaviside step function is equal to one, otherwise zero.)

With the use of substitutions we get function L as a function of vectors $x_k = \{^1x_k, ^2x_k, \ldots ^dx_k\}, x_l = \{^1x_l, ^2x_l, \ldots ^dx_l\}$, i.e. function of $2d$ scalar variables. These variables affect the classification error via the distance function $\rho(.,.)$. Depending on x_k or x_l we have "two" distance functions $\rho_k = \rho(x, x_k)$ and $\rho_l = \rho(x, x_l)$ that we have to differentiate. We will need some derivatives

$$\frac{\partial \hat{p}}{\partial\, ^rx_k} = \frac{1}{H_N} . \frac{\partial S_c}{\partial\, ^rx_k}, \tag{14}$$

$$\frac{\partial S_c}{\partial\, ^rx_k} = \sum_{k:x_k \in U} \frac{-1}{r_k^2} . \frac{\partial r_k}{\partial\, ^rx_k}, \tag{15}$$

at the same time

$$\frac{\partial S_c}{\partial\, ^rx_j} = \sum_{k:x_k \in U} \frac{-1}{r_k^2} . \frac{\partial r_k}{\partial\, ^rx_j}, \tag{16}$$

because r_k is function of x_k as well as x_j. Now we need the derivative of the r_k wrt. ^rx_k. There is

$$\frac{\partial r_k}{\partial\, ^rx_k} = \frac{\epsilon^2}{2} \sum_{j:x_j \in U} D.(\rho_k - \rho_j) . \frac{\partial \rho_k}{\partial\, ^rx_k}, \tag{17}$$

where

$$D = \frac{1}{\sqrt{[\epsilon^2 + (\rho_k - \rho_j)^2]^3}}. \tag{18}$$

Similarly,

$$\frac{\partial r_k}{\partial\, ^rx_k} = \frac{\epsilon^2}{2} D.(\rho_k - \rho_j) . \frac{\partial \rho_j}{\partial\, ^rx_j}, \tag{19}$$

where there is no sum. Now we can write the derivative of L by ^rx_k and ^rx_j:

$$\frac{\partial L}{\partial\, ^rx_k} = \frac{-\epsilon^2}{2H_N r_k^2} \sum_{j:x_j \in U} D.(\rho_k - \rho_j) . \frac{\partial L}{\partial p} . \frac{\partial \rho_k}{\partial\, ^rx_k}. \tag{20}$$

Here we have one fixed k and one fixed index r, and j is such that x_j runs over the whole learning set U. In the following:

$$\frac{\partial L}{\partial\, ^rx_j} = \frac{\epsilon^2}{2H_N} \sum_{k:x_k \in U_c} \frac{D.(\rho_k - \rho_j)}{r_k^2} . \frac{\partial L}{\partial p} . \frac{\partial \rho_j}{\partial\, ^rx_j} \tag{21}$$

there j and index r are fixed, and k is such that x_k runs over set U_c of all patterns of class c from the learning set U.

2.6 Conditions for a good distance function

Equation (21) should be fulfilled for any pattern x. Unfortunately, only the constant function can have zero derivatives anywhere. We must limit ourselves to situations of most probable error. One can suppose that it arises where the class probability estimation $\hat{p}(c|x)$ is close to the decision threshold θ, usually $\theta = 1/2$ in two class problems. This conjecture can be verified by comparison of distribution functions of classifier's output before thresholding. The distribution functions for badly recognized patterns lie below the distribution function for well recognized ones. For badly recognized patterns the mean is close to 0.5 whereas the mean for well recognized patterns is substantially larger. It holds also for multiclass problems. Then $\hat{p}(c|x) - 1/2 = 0$ characterizes where we need to minimize the number of errors and then zero derivatives. So we have to discuss $\frac{\partial \rho_k}{\partial r_k^r} = 0$ and $\frac{\partial \rho_j}{\partial r_j^r} = 0$. From (20) and (21) it follows

$$\frac{-\epsilon^2}{2H_N r_k^2} \sum_{j:x_j \in U} D.(\rho_k - \rho_j).\frac{\partial \rho_k}{\partial\, ^r x_k} = 0 \tag{22}$$

and also

$$\frac{\epsilon^2}{2H_N} \sum_{k:x_k \in U_c} \frac{D.(\rho_k - \rho_j)}{r_k^2}.\frac{\partial \rho_j}{\partial\, ^r x_j} = 0. \tag{23}$$

for $k : x_k \in U_c, j : x_j \in U$, and index $r = 1, 2, \ldots, d$. Then there are $\forall k$ and $\forall j$ total $2d$ expressions. Term $D.(\rho_k - \rho_j)$ is never zero, then there must be $\frac{\partial \rho_j}{\partial\, ^r x_j} = 0$ and $\frac{\partial \rho_k}{\partial\, ^r x_k} = 0$, i.e. $\nabla \rho_j = 0, \nabla \rho_k = 0$.

2.7 Other findings

It can be found that some distance functions behave better than the L_2 metric. We study here the Pearson, Hassanat, and Orloci distance functions.

The Pearson distance function (a covariance dissimilarity) is given by

$$\rho_P = \frac{1}{d} \sum_{i=1}^{d} (x_i - \bar{x})(y_i - \bar{y}). \tag{24}$$

It is nearly a metric. When testing with random points, one can find less than 10% violations of the triangle inequality. In the Pearson distance $\rho_P = 0$ if $x_1 = x_2 = \ldots = x_d$ or $y_1 = y_2 = \ldots = y_d$ even if $x \neq y$, i.e. the distance of a point to the point that has all coordinates the same is equal to zero. In two dimensions a "circle" with center at 0 has form of hyperbolas $y = \pm 2r/x$, where r is the radius. The area of a circle is infinite.

The Hassanat metric [13] is defined as follows. Let $M_i = max(x_i, y_i)$ and $m_i = min(x_i, y_i)$. The metric is given by the formula

$$\rho_H = \sum_{j=1}^{d} d_i \tag{25}$$

where

$$d_i = 1 - \frac{1 + m_i}{1 + M_i}$$

for $m_i >= 0$, and

$$d_i = 1 - \frac{1 + m_i + |m_i|}{1 + M_i + |m_i|}$$

that is also

$$d_i = 1 - \frac{1}{1 + M_i + |m_i|}$$

for $m_i < 0$.

It has been proved to be a metric on R^d. The value of a distance is limited to d. If average is used instead of the sum in (25), then the distance is limited to 1. A "circle" with radius r in two dimensions has the form of hyperbolas as shown in Fig. 1. This Figure shows “circles” with radii 0.4, 0.5, 0.8, 0.9, and 1.0. At the top right one hyperbolic branch of a circle with radius 0.5 is depicted as a bold dashed line. Horizontal and vertical dashed lines indicate its asymptotes. It can be seen that "circles" with small radii have a finite area, while for large radii the area is infinite. For small radii the form reminds of the circles in the Manhattan metric, where the four edges are straight lines. A question arises as to what the ball looks like in multidimensional space. Fig. 1 leads us to compare it to the sphere in the Manhattan metric. In the Manhattan metric a d-dimensional sphere is actually a cube “standing on one corner”. This means that each of the d main diagonals is parallel to one coordinate axis. The ball in the Hassanat metric can be compared with a ball in the L_1 metric. The difference lies in the fact that ball in the Hassanat metric has all the edges and all the planes bent in a hyperbolic way. The larger the radius of the sphere, the greater the deflection.

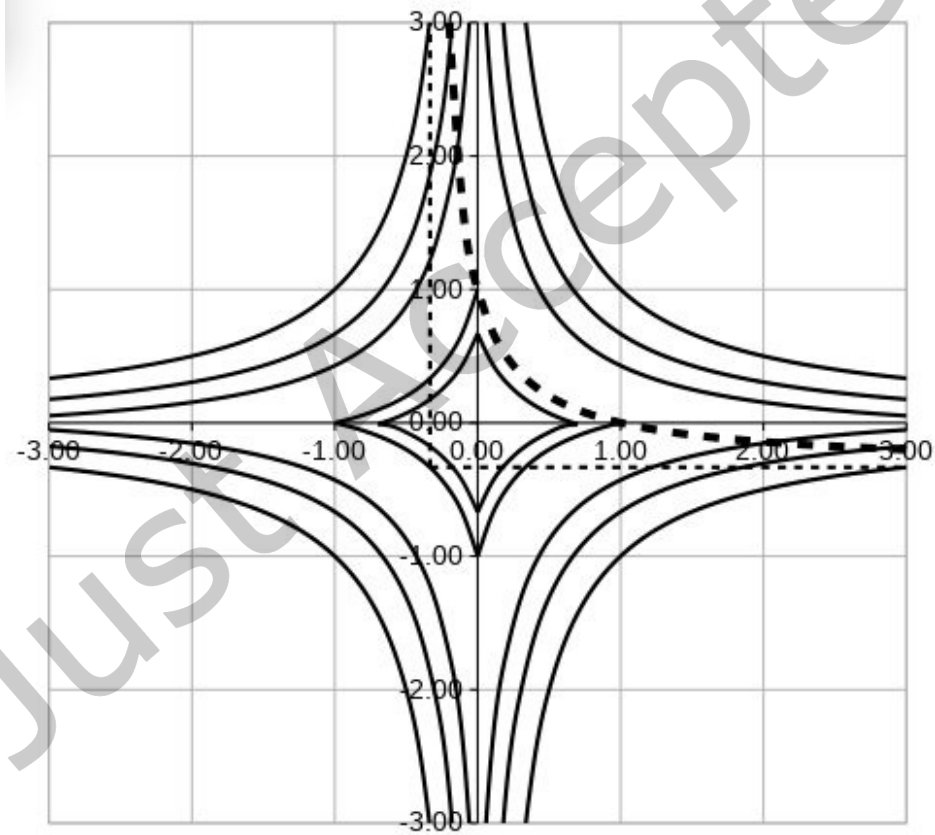

Fig. 1. Twodimensional balls (circles) in the Hassanat metric. Radii of balls from smallest to largest are 0.4, 0.5, 0.8, 0.9, and 1.0. At the top right one hyperbolic branch of a circle with radius 0.5 is depicted as a bold dashed line. Horizontal and vertical dashed lines indicate its asymptotes.

The Orloci metric (chord distance) is given by

$$\rho_O = \sqrt{2\left(1 - \frac{(x.y)}{||x||_2||y||_2}\right)}, \tag{26}$$

where (x, y) is the scalar product and $||.||_2$ denotes the L_2 norm. In the Orloci metric the value of distance is limited to $\sqrt{2}$. It also holds $\rho_O(x, y) = 0$ if $x_1 = x_2 = \ldots = x_d$ and $y_1 = y_2 = \ldots = y_d$ even if $x \neq y$. Moreover in two dimensions a "circle" has the form of two straight lines that cross each other at point 0.

It is also remarkable that distance functions that have a limited value to a margin have smaller errors than distance functions where the value of distance is unlimited, typically L_p metrics. Note that the larger p the worse; the preferable one is the L_1 metric. This hypothesis leads to the "bounded L_p" distance functions, especially bouded L_1 and bounded L_2 distance functions given by:

$$L_{\overline{1}}(x, y) = \sum_{j=1}^{d} A_j, \text{ where } A_r = \begin{cases} |x_j - y_j| & \text{for} \quad |x_j - y_j| < 1 \\ 1 + \alpha|x_j - y_j| & \text{for} \quad |x_j - y_j| \geq 1 \end{cases} \tag{27}$$

and

$$L_{\overline{2}}(x, y) = \sqrt{\sum_{j=1}^{d} A_j}, \text{ where } A_r = \begin{cases} (x_j - y_j)^2 & \text{for} \quad (x_j - y_j)^2 < 1 \\ 1 + \alpha(x_j - y_j)^2 & \text{for} \quad (x_j - y_j)^2 \geq 1 \end{cases} \tag{28}$$

The terms with α (≈ 0.0001) serve for differentiation of distances of points with very large coordinate differences $|x_j - y_j|$. Otherwise, all such points would appear to be at the same distance. These distance functions are not metrics, but there is a small percentage of the triangle inequality violations. It can be found that the classification results with bounded distance functions are better than with original ones, i.e. the L_p metrics.

In summary, there are conditions and findings

(1) $\nabla\rho_j = 0, \nabla\rho_k = 0$ for x for which $\hat{p}(c|x) - 1/2 = 0$ holds.
(2) The distance function should be a metric or nearly a metric, i.e. with small probability of the triangle inequality violations.
(3) The distance function should be limited to a (relatively) fixed value.
(4) A "circle" in two dimensions can have infinite area and should have the form as discussed above.

3 RESULTS

3.1 New distance functions

A metric can be derived in the following way. In a vector space there is a relation between norm and metric. It holds that if (X, p) is a normed vector space with norm p then $\rho : X * X \rightarrow R$ defined by $\rho(x, y) = ||x - y||$ is a metric on X. Moreover, a metric associated with a norm has additional properties, the translation invariance and homogeneity. The Hassanat metric on R^d is not absolutely homogenous, then we cannot use construction above to derive a norm. Proceeding purely formally, we get formula as follows (a simplified Hassanat metric; we call it h2 metric.)

$$\rho_{h2}(x, y) = \sum_{j=1}^{d}(1 - \frac{1}{1 + |x_i - y_i|}). \tag{29}$$

Note. It can be easily seen that due to the use of the absolute value in this formula there is no zero partial derivative according to any of coordinates $x_j, y_j, j = 1, 2, \ldots d$ if $x_j = y_j = 0$. On the other hand, any "trick" to smooth the break of the absolute value will lead to zero derivative at the minimum. Thus, this condition can be easily fulfilled.

3.2 Features of the h2 metric

3.2.1 *Metric.*

THEOREM 3.1. *Function* (29) *is a metric in* R^d.

PROOF. First, to prove the nonnegativity it is sufficient to prove the nonnegativity of each summand. If

$$1 - \frac{1}{1 + |x_j - y_j|} \geq 0$$

then

$$|x_j - y_j| \geq 0.$$

Second, $\rho(x, y) = 0 <=> x = y$. In our case from

$$1 - \frac{1}{1 + |x_j - y_j|} = 0 \tag{30}$$

it follows $|x_j - y_j| = 0$ and then $x_j = y_j$ for every $i = 1, 2, \ldots d$. And from $x_j = y_j$ it directly follows that (30) holds. Third, the symmetry is apparent.
Fourth, the triangle unequality has the form

$$\frac{1}{d}\sum_{j=1}^{d}(1 - \frac{1}{1 + |x_j - z_j|}) \leq \frac{1}{d}\sum_{j=1}^{d}(1 - \frac{1}{1 + |x_j - y_j|}) + \frac{1}{d}\sum_{j=1}^{d}(1 - \frac{1}{1 + |y_j - z_j|}).$$

For short, we use symbol XY for term $|x_j - y_j|$ and analogously symbols XZ and YZ. The equation above can be rewritten in the form

$$\sum_{j=1}^{d}((1 - \frac{1}{1 + XY}) + (1 - \frac{1}{1 + YZ}) + (1 - \frac{1}{1 + XZ})) \geq 0.$$

Using a common denominator, there is

$$\sum_{j=1}^{d} \frac{XY + YZ - XZ + 2.XY.YZ + XY.XZ.YZ}{(1 + XY).(1 + YZ).(1 + XZ)} \geq 0. \tag{31}$$

Now it suffices to prove that $XY + YZ - XZ \geq 0$ for every $j = 1, 2, \ldots d$. Here $XY = |x_j - y_j| = \rho_1(x_j, y_j)$ and similarly XZ and YZ, then ρ_1 is an L_1 metric on R. Then $XY + YZ - XZ \geq 0$ and (31) is a triangle unequality. Thus, (29) is a metric on R^d. □

3.2.2 *Continuity.*

In (29) it is seen that it is continuous in R^d. Due to the absolute value in the denominator the derivative is discontinuous in set $S = \{x, y \in R^d : x_i = y_i, i = 1, 2, \ldots, d\} \subset R^d \times R^d$. In $R^d \times R^d - S$ all derivatives of (29) are continuous. See also Note in Chap. 3.1.

3.2.3 *A circle in* R^2.

Let a "ball" (circle) $\mathcal{B}(0, r)$ in R^2 be centered at the origin. In the first quadrant the distance of a point (x, y) from point $(0, 0)$ is equal to r and it holds

$$\frac{1}{1 + x} + \frac{1}{1 + y} = 2(1 - r),\ 0 < r < 1.$$

A ball has the form of four hyperbolas of type $y = 1/x$ with asymptotes parallel with coordinate axes. Due to symmetry in other quadrants this ball looks similar. It can be also found that hyperbolas have horizontal and vertical asymptotes parallel to coordinate axes at the distance given by

$$a = \frac{1}{2(1 - r)} - 1\ .$$

Thus, for radius $r = 0.5$ asymptotes are identical with coordinate axes. For $r > 0.5$ the distance of asymptotes is positive and the hyperbola in the first quadrant is shifted by this value to the right and up. For $r < 0.5$ the asymptotes lie by this shift left and below the coordinate axes, and therefore the hyperbola crosses the coordinate axes in the distance $\frac{2r}{1-2r}$ from the origin. An equation for a circle $\mathcal{B}(0, r)$, $r < 1$ in R^2 is

$$y = \pm\left(\frac{1}{2(1-r) - \frac{1}{1+|x|}} - 1\right). \tag{32}$$

Such circles for several values of radius are depicted in Fig. 2. For radii $r \geq 0.5$ the area ("volume") is infinite, for $0 \leq r \leq 0.5$ the area V is finite and it holds

$$V = 4\left(\ln\frac{|bs + b - 1|}{|b-1|} + \frac{s}{b} - s\right), \tag{33}$$

where $b = 2(1-r)$ and the length of a "ray" of the star is $s = \frac{2r}{1-2r}$ These data in numbers are shown in Table 1.

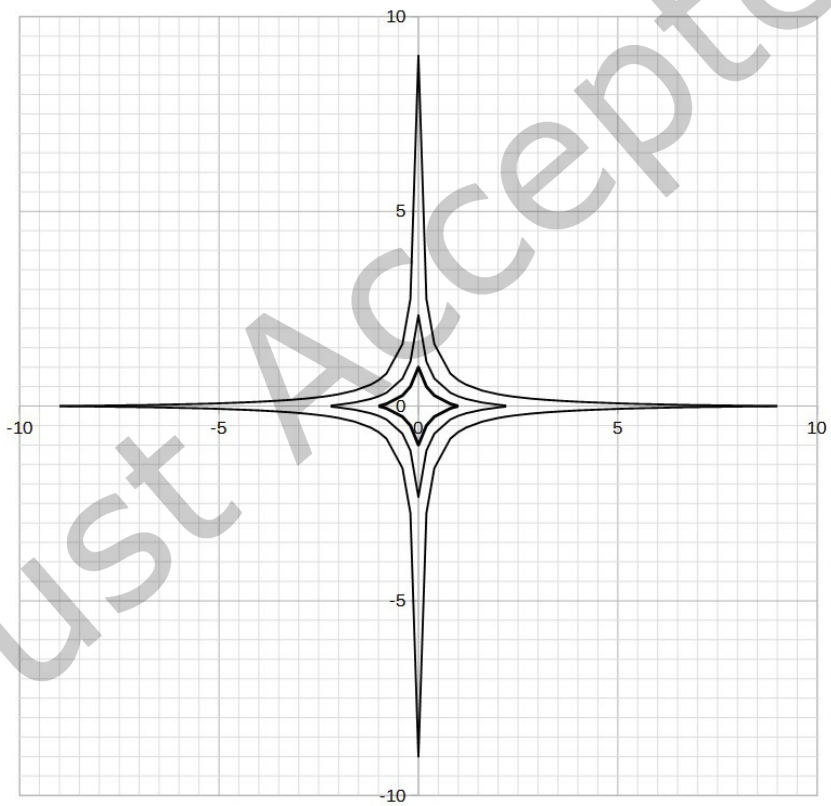

Fig. 2. Balls (circles) in R^2 with radii 0.25, 0.35, and 0.4; h2 metric.

Table 1. Volume (area) of a ball in R^2 and length of "ray" of the "star" for different radii r of a ball; the h2 metric.

r	area	length of "ray"
0.1	0.107	0.25
0.2	0.596	0.667
0.3	2.026	1.5
0.4	6.275	4
0.45	11.951	9
0.49	26.238	49
0.499	45.535	499
≥ 0.5	∞	∞

3.3 A product semimetric

It seems that distance functions in which balls in R^2 have the form of hyperbolas gave good results when used in classification tasks. The idea behind this distance function is to construct a distance function that would have a ball in the form of exact hyperbolas. Then we define a distance function as a product of features differences.

$$\rho_{h3}(x,y) = \left(\prod_{j=1; x_j \neq y_j}^{d} |x_j - y_j| \right)^{1/d} \tag{34}$$

if at least for one j there is $x_j \neq y_j$, and

$$\rho_{h3}(x,y) = 0 \tag{35}$$

otherwise.

3.4 Features of product semimetric

3.4.1 Semimetric. To demonstrate that (34), (35) is a semimetric suffice to show that it is, first, nonnegative, which is apparent from (34). Second, $\rho(x,y) = 0 <=> x = y$ according to (35). Third, the symmetry is apparent. Note that the triangle inequality does not hold but it does not hold in a small percentage of cases.

3.4.2 Continuity. In (34) it is seen that it is continuous in R^d. Due to the absolute value of the coordinates differences $x_j - y_j$ the derivative is discontinuous in set $S = \{x, y \in R^d : x_i = y_i, i = 1, 2, ..., d\} \subset R^d \times R^d$. In $R^d \times R^d - S$ all derivatives of (34) are continuous. See also Note in Chap. 3.1.

3.4.3 A circle in R^2. Let a "ball" (circle) $\mathcal{B}(0,r)$ in R^2 be centered at the origin. In the first quadrant the distance of a point (x_1, x_2) from point $(0,0)$ is equal to r and it holds

$$r^2 = |x_1|.|x_2|,$$

eventually, when a point is given as (x,y)

$$r^2 = |x|.|y|,$$

Generally in a d-dimensional space

$$r^d = \prod_{j=1}^{d} |x_j|.$$

Table 2. Table of basic characteristics of tasks from the UCI Machine Learning Repository modified by sources cited. Abbreviations for sources: P - [28]; P2 - [27]; UCI MLR - [10]. Note (1): Iris data are used without Setoza class, i.e. two classes Versicolor and Virginica only. The last column gives the application, ev. the scientific field from which the data were derived.

Dataset	Dimens.	Classes	Samples tot.	Learn.	Test set	Cross	Source	Appl. field
Australian	42	2	690	551	139	50	P	Finance
Balance	4	3	625	499	126	50	P	Psychology
Cancer	9	2	683	546	137	50	P	Medicin
Diabetes	8	2	768	614	154	50	P	Medicin
DNA	180	3	31186	2000	1186	1	P2	Life Sci.
German	24	2	1000	800	200	50	P	Finance
Glass	9	6	215	169	46	50	P	Forensic
Heart	25	2	270	216	54	50	P	Medicin
Ionosphere	34	2	351	280	71	50	P	Meteorology
Iris (1)	4	2 (3)	100 (150)	90	10	10	UCI MLR	Life Sci.
Led 17	24	10	2000	1595	405	50	P	Computer
Letter	16	26	20000	16000	4000	1	UCI MLR	Computer
Liver	6	1	345	276	69	50	P	Medicin
Monkey1	17	2	556	444	112	50	P	Zoology
Phoneme	5	2	5404	4322	1082	50	P	Speech R.
Satimage	36	7	6435	4435	2000	1	UCI MLR	Geology
Segmen	19	7	2310	1848	462	50	P	Image Proc.
Sonar	60	2	208	165	43	50	P	Naval-Milit.
Vehicle	18	4	846	675	171	50	P	Pattern R.
Vote	16	2	435	347	88	50	P	Politics
Vowel	10	11	528	418	110	50	P	Speech R.
Waveform21	21	3	5000	3998	1002	50	P	Signal Proc.
Waveform40	40	3	5000	3999	1001	50	P	Signal Proc.
Wine	13	3	178	141	37	50	P	Agriculture

A circle in R^2 has the form of four hyperbolas with asymptotes identical with coordinate axes. Equation for a circle $\mathcal{B}(0, r)$, $r < 1$ in R^2 is

$$y = \pm \frac{r^2}{|x|}.$$

The area ("volume") is infinite for each radius r.

4 APPLICATION IN CLASSIFICATION PROBLEMS

The influence of the distance function used is demonstrated in computational experiments employing 24 tasks from the Machine Learning Repository [10]. Characteristics of these tasks are given in Table 2. The number of attributes not including the class mark ranges from 4 to 180. There are two to 26 classes. Data originally from the UCI Machine Learning Repository were gained mostly from [28] (denoted by P in the column Source in the Table). These data sets are ready for run with a classifier. Most of the tasks consist of 50 pairs of training and testing sets corresponding to 50-fold cross validation. For the DNA, Letter and Satimage data a single partition into training and testing sets according to specification in the UCI MLR was used. We also added the popular Iris

data set. We use them without Setoza class, i.e. with two classes Versicolor and Virginica only, and the remaining data were split into 10 pairs for ten-fold cross validation.

New distance functions can be useful for a better functioning of the distance-based classifiers. According to the study [14], the best results with different metrics were reached for the IINC algorithm. Therefore, we present results for the IINC here.

4.1 Selection of distance functions for comparison

The main source of not too common distance functions is book by E. Deza and M. M. Deza [7], ev. [8]. This is a great piece of work. Our conditions for selection were first, if it is applicable to R^d, and second, if it is computationally simple. It means without integral enumerations, solving equations, taking supremum etc. This allows (with some licence) to use Cayley-Klein-Hilbert metric or Weierstrass metric ([7], p. 122), but excludes distance functions of type Harnack metric, Apollonian metric ([7], p. 123) and many others.

Into our selection we included common metrics, Euclidean, Manhattan (L_1), L_{10} that simulates L_∞, Mahalanobis distance and class dependent Mahalanobis distance. The last two are rather common but they do not fulfill the condition of a simple computation. They need evaluation of the inverse covariance matrix, eventually as many covariance matrices as there are classes. Moreover difficulty arises in high dimensional tasks. To avoid these problems we use algorithm ainvl adapted from Matlab code according to [4].

Another distance functions sometimes considered in classification or machine learning tasks is the Orloci (chord) distance and very similar angular semimetric. Five distance functions were derived from various correlation coefficients as $1 - correlation\ coefficient$. This way we got Pearson, jacknife, Goodman-Kruskal, Kendall, and Spearman distances. Remaining eight distance functions were selected from [7] according to criteria mentioned already.

4.2 Results of tests

Summary of measurements is shown in Table 3. Rows in the table show results for individual tasks, the last line gives the mean. 23 columns give classification errors for all the tasks for 23 distance functions, mostly metrics, with the IINC classifier. These columns are ranked according to the mean classsification error, the best leftmost. In each row there is one entry in bold showing the minimal classification error and the best distance function for a task. The last column of the Table gives classification errors for the SVM (support vector machine), implementation by T. Joachims [19], [18]. Data for other classifiers, the 1-NN and k-NN type with or without learning can be found e.g. in [29], [14].

Note. In Table 3 one can find entries larger than 50%. To make it lesser than 50% one could use a complement to one, that means to interchange classes in the case of two-class problems. Since we think this is unfair we left things as they are. The cases of such a large classification error say simply that the task is hard for the classifier and the distance function used. It can be also found that this does not appear for the left (upper) half of the table, i.e for distance functions that can be considered "better".

4.3 Statistical evaluation

For evaluation of distance functions we use statistical tests, the Friedman aligned test and the Quade test, and also simple criteria, such as the mean classification error and quartiles, esp. median. In statistical tests we follow the methodology and recommendations according to [6]. That is why we do not describe these methods here. Note only that in both the Friedman aligned test and the Quade test one ranks all the entries of the table of classification errors (Tables 3 and 4). Both tests serve for multiple comparisons. The Friedman aligned test considers all the tasks equally important, whereas the Quade test takes into account the fact that some tasks are more difficult

than others. In the Quade test each problem is scaled, depending on the differences observed in classification performances. In this way, the Quade test gives a weighted ranking.

Ranks for each distance function follow from each of the criteria and tests. These ranks are summarized in Table 5. In this Table the distance functions are arranged in the same order as in Tables 3 and 4, i.e. according to mean classification error, the smallest mean error first.

Table 5 shows how much rankings depend on the criteria chosen. There are six columns showing ranks according to six methods of evaluation of classification errors. It is seen here that ranking of distance functions depends on the criterion to a considerable extent.

Table 3. Classification errors of the IINC classifier with 23 distance functions for 24 tasks - part I. The columns (distance functions) are ranked according to the mean classification error, the best (with smallest error) first (leftmost).

Distance function	Hassanat	h2	L1	h3	Pearson	Orloci	L2	L10	CD-Mahal.	Spear-man	Weier-strass	Lorentz
Task	1	2	3	4	5	6	7	8	9	10	11	12
australian	12.90%	12.58%	13.31%	12.96%	15.16%	15.17%	14.75%	22.28%	17.96%	12.71%	15.90%	15.09%
balance	31.44%	34.25%	32.55%	31.14%	10.71%	23.71%	30.50%	30.18%	30.01%	26.83%	29.64%	17.61%
cancer	3.25%	3.65%	3.28%	3.98%	6.25%	2.88%	3.48%	3.94%	4.01%	9.87%	4.57%	6.20%
diabetes	27.06%	26.75%	26.21%	25.78%	34.49%	26.83%	25.52%	27.06%	25.97%	35.71%	36.85%	38.80%
DNA	27.49%	27.49%	27.82%	28.25%	16.86%	16.61%	31.03%	34.82%	17.62%	15.26%	30.94%	26.98%
german	29.54%	29.92%	30.91%	30.09%	32.62%	32.69%	31.13%	31.19%	33.38%	31.51%	29.69%	31.26%
glass	30.39%	30.75%	33.01%	34.16%	32.47%	32.48%	35.18%	37.90%	46.93%	31.51%	46.97%	46.22%
heart	17.89%	17.93%	17.96%	17.52%	18.37%	18.48%	17.93%	21.89%	18.67%	18.78%	19.26%	18.78%
ionosphere	8.57%	8.15%	10.82%	10.96%	12.16%	12.39%	14.81%	14.05%	14.81%	11.59%	32.36%	14.24%
iris	7.91%	7.91%	7.91%	5.91%	3.91%	4.91%	4.91%	4.91%	4.00%	32.82%	4.00%	9.91%
led17	0.46%	0.46%	0.46%	0.46%	4.51%	5.08%	0.45%	0.08%	10.43%	17.05%	0.53%	0.76%
letter	5.10%	5.50%	4.85%	5.25%	6.78%	6.33%	4.98%	7.23%	7.15%	12.03%	12.68%	38.90%
liver	36.99%	37.88%	38.29%	40.43%	36.75%	36.70%	39.13%	41.94%	35.74%	37.07%	50.81%	50.52%
monkey1	4.79%	4.81%	4.81%	4.76%	4.88%	6.10%	4.79%	6.39%	9.57%	23.75%	7.99%	4.86%
phoneme	16.73%	16.75%	17.60%	17.37%	19.35%	18.63%	18.06%	18.40%	18.65%	30.10%	32.87%	32.28%
satimage	11.30%	11.45%	11.00%	11.65%	15.70%	11.60%	11.55%	13.45%	13.30%	19.45%	15.95%	35.30%
segmen	3.68%	4.08%	4.12%	4.55%	5.00%	5.72%	5.05%	6.72%	5.05%	7.03%	15.99%	26.45%
sonar	20.94%	20.83%	19.89%	19.56%	23.86%	22.74%	22.85%	29.28%	34.69%	22.71%	22.23%	20.19%
vehicle	28.83%	28.86%	29.40%	29.23%	28.88%	29.68%	29.34%	30.97%	36.51%	29.64%	31.23%	40.43%
vote	8.86%	8.86%	8.52%	8.73%	9.28%	9.65%	8.89%	10.45%	13.85%	9.03%	8.93%	8.84%
vowel	2.84%	2.93%	2.73%	3.31%	3.86%	3.82%	2.74%	4.18%	12.99%	10.41%	17.23%	30.19%
waveform21	16.68%	16.78%	16.15%	16.27%	17.91%	18.26%	16.38%	16.88%	21.00%	17.79%	16.04%	18.18%
waveform40	17.73%	17.89%	17.59%	17.26%	21.15%	21.30%	18.08%	23.38%	22.01%	20.94%	17.56%	17.98%
wine	4.88%	4.04%	4.24%	4.89%	5.52%	6.45%	5.66%	8.29%	6.92%	6.94%	6.22%	5.79%
MEAN	15.68%	15.85%	15.98%	16.02%	16.10%	16.18%	16.55%	18.58%	19.22%	20.44%	21.10%	23.16%

Table 4. Classification errors of the IINC classifier with 23 distance functions for 24 tasks - part II. The last column shows results obtained using SVM with best kernel for each task. NA means not available; SVM does not work for this task.

Distance function	Mahala-nobis	Canberra	elliptic	Clark	Goodman Kruskal	Kendall	Bray-Curtis	hyper-bolic	Inter-section	Cayley-Klein	Jacknife	SVM-best
Task	13	14	15	16	17	18	19	20	21	22	23	–
australian	31.45%	16.73%	20.49%	44.71%	11.18%	15.45%	39.69%	44.72%	44.90%	46.13%	27.53%	35.99%
balance	30.59%	17.61%	23.63%	41.96%	45.12%	53.02%	15.74%	15.42%	15.56%	24.95%	64.63%	33.17%
cancer	3.98%	5.64%	34.27%	34.85%	43.88%	17.36%	14.07%	34.85%	34.85%	48.29%	3.29%	16.34%
diabetes	28.54%	39.10%	40.13%	49.88%	48.84%	45.87%	46.11%	40.32%	37.60%	47.72%	33.81%	29.64%
DNA	46.88%	26.90%	32.29%	23.61%	8.62%	14.08%	48.40%	49.16%	49.16%	26.56%	51.10%	NA
german	36.44%	29.33%	33.53%	36.40%	33.08%	35.26%	44.31%	37.20%	46.22%	36.25%	30.49%	27.25%
glass	65.48%	49.15%	38.75%	13.54%	32.45%	45.89%	37.00%	32.76%	32.76%	24.38%	59.47%	32.63%
heart	35.22%	20.07%	29.52%	47.78%	21.29%	17.48%	41.11%	46.67%	46.70%	48.26%	26.96%	37.22%
ionosphere	14.67%	9.65%	35.81%	35.90%	23.95%	14.92%	43.34%	41.61%	41.61%	36.29%	21.20%	18.52%
iris	4.00%	9.91%	37.64%	22.73%	63.74%	32.91%	27.55%	49.55%	48.55%	44.82%	22.82%	5.55%
led17	0.57%	79.87%	5.48%	10.43%	30.22%	20.85%	6.34%	10.86%	11.87%	10.43%	24.70%	11.52%
letter	7.33%	39.00%	15.15%	3.60%	42.26%	41.50%	3.98%	3.77%	3.77%	12.95%	88.13%	2.68%
liver	38.67%	50.90%	46.35%	49.04%	41.50%	42.26%	48.87%	45.22%	46.35%	43.22%	51.86%	35.54%
monkey1	20.93%	21.42%	2.47%	50.00%	20.85%	30.22%	49.23%	50.00%	49.35%	40.60%	46.71%	2.94%
phoneme	18.15%	32.27%	28.82%	43.32%	32.91%	63.74%	29.36%	29.35%	29.35%	40.92%	42.05%	14.39%
satimage	12.50%	35.40%	26.50%	19.45%	14.92%	23.95%	22.55%	19.85%	12.45%	15.70%	49.35%	24.30%
segmen	5.98%	26.52%	45.33%	14.29%	17.48%	21.29%	20.58%	14.29%	14.29%	29.11%	56.19%	34.27%
sonar	24.69%	35.95%	40.00%	46.63%	45.89%	32.45%	49.26%	46.63%	46.63%	46.63%	44.90%	19.67%
vehicle	44.41%	40.85%	38.16%	24.27%	35.26%	33.08%	28.02%	23.70%	24.01%	23.28%	62.27%	26.23%
vote	14.04%	8.31%	26.04%	39.07%	14.08%	8.62%	48.74%	40.45%	41.98%	45.82%	11.84%	22.64%
vowel	4.31%	30.21%	34.84%	12.61%	45.87%	48.84%	16.18%	9.09%	8.81%	21.14%	74.28%	8.54%
waveform21	20.47%	19.38%	37.69%	33.34%	17.36%	43.88%	33.70%	33.73%	33.78%	32.63%	29.45%	26.34%
waveform40	21.06%	27.80%	40.10%	33.72%	53.02%	45.12%	32.51%	33.23%	32.43%	33.72%	27.84%	32.25%
wine	29.02%	8.13%	33.95%	26.95%	15.45%	11.18%	30.90%	33.15%	33.15%	25.81%	21.25%	27.77%
MEAN	23.31%	28.34%	31.12%	31.59%	31.63%	31.63%	32.40%	32.73%	32.75%	33.57%	40.51%	22.84%

Table 5. The ranks of 23 distance functions according to six criteria and statistical tests. Rows in this table, i.e. distance functions, are ranked according to the mean classification error in the same way how they are ranked in Tables 3 and 4. Individual columns give ranks of distance functions according to six criteria. Note the remarkable similarities between the columns.

Ranks Dist.function	1 quartile	Mean	Median	3 quantile	Friedman aligned test	Quade test
Hassanat	3	1	4	8	1	2
h2	5	2	2	6	2	3
L1	1	3	3	5	3	4
h3	4	4	1	3	4	5
Pearson	6	5	5	1	5	1
Orloci	7	6	7	2	6	6
L2	2	7	6	4	7	7
L10	8	8	9	9	8	8
CDMahanobis	9	9	10	7	9	9
Spearman	12	10	11	10	10	12
Weierstrass	11	11	8	11	11	10
Lorentz	13	12	12	13	13	13
Mahalanobis	10	13	13	12	12	11
Canberra	14	14	14	14	14	16
ellipt	22	15	18	15	18	19
Clark	18	16	19	16	17	14
Goodm.-Kruskal	15.5	17.5	15.5	17.5	15.5	22
Kendall	15.5	17.5	15.5	17.5	15.5	21
Bray-Curtis	19	19	17	19	19	15
hyperb	20	20	20	20	21	18
Intersection	17	21	21	22	20	17
Cayley-Klein-Hilbert	21	22	22	21	22	20
Jacknife	23	23	23	23	23	23

In Table 5 the L_1, Hassanat, h2, and h3 distance functions form a group apparently better than the rest of the distance functions and the SVM. The leading role of these four distance measures may slightly change according to the set of tasks used for comparison. In any case, one of these distance functions will appear as best.

On the other hand, this may, but need not, hold for a particular task. The best classification error and best distance function for each task are shown in bold in Tables 3 and 4. It is seen here that the first five distance functions are each best for two tasks from the total 24. Also Clark and Goodman-Kruskal distance functions are the best for the two tasks. At the same time, there are eight bold entries in the second half of the Table. It shows that a generally not too good distance function may be the best for a particular task. If a "good" distance function is the best, then the classification error for a "bad" distance function may be even ten times larger. In contrast to the best result for a "bad" distance function, results with any "good" distance function are usually only a little bit worse. One can conclude that if a "good" distance function is used, the "danger" of much better results with another distance function is very low.

In this work we use rankings based on two thorough statistical methods, the Friedman aligned test and the Quade test, and rankings according to simple criteria (three quartiles and mean), see Table 5. The mean has a

known disadvantage in the fact that several large values may shift the mean to a larger value. On the other hand, quartiles, especially median, are robust estimators. The quartiles have also a simple quantitative interpretation. The rank in the first quartile says that a particular distance function is the "rank-best" for 1/4 of tasks. Here with the $L1$ metric there are minimal errors in 1/4 of tasks. With the $h3$ metric there are minimal errors in 50% of tasks. And with Pearson's metric there are minimal errors in 3/4, i.e. in 18 from 24 tasks.

Considering the first and the third quartile as too weak and too tough, the mean and the median remain. In first four places the Hassanat and h3 metrics interchange their ranks. According to the mean, the Hassanat metric is the best, according to the median the h3 pseudometric is the best.

It can also be seen that the ranking according to the Friedman test and according to a simple mean are the same for distance functions ranked 1 to 11. The rankings 12 to 23 are very similar here.

In contrast to this, there is a large difference between the Quade test and the Friedman aligned test, eventually the mean. The ranking according to the Quade test is closer to the ranking according to the median, eventually to the third quartile. Beside the advantages of the Quade test being thoroughly theoretically supported and taking task difficulties into account, it has a disadvantage in its difficult interpretation of the resulting ranking.

From the point of view of weighting tasks according to their difficulties, the mean and the Friedman aligned test do not seem to be the best choice. Therefore, a slightly more time-consuming Quade test should be preferred. In the end, it appears simpler to rank distance functions according to the median that has the simple interpretation mentioned above.

4.4 Practical example

Data we discuss here describe successes and faults in lending money. Clients are of two types: those to whom the money was provided and those to whom the loaned money was refused. Error in both case means loss. In the first case, money is simply lost; a person never pays or pays too late and does pay in full. In the latter case, the interest is lost. Moreover, almost certainly the client will not come again. The error in decision therefire has the same weight in both cases. The simplest criterion to minimize is the classification error.

The goal of automation is to minimize decision errors with help of machine learning. In this case, it is lazy learning, taking advantage of the entire learning set and updating immediately or at correct intervals as new data appears. Specifically, we present results with data where there are 2000 cases used as a learning set, and 31000 cases (samples) form a test set. Each client is characterized by seven parameters (features), e.g. money amount, city of residence, economic category, years of employment. Fig. 3 shows the comparison of the two distance-based classification methods and the ranking of distance measures described here. Distance functions are sorted by the IINC classification error. For this reason, the $1 - NN$ classification error is not a monotonous function. On the other hand, there is a match between good and bad distance functions in both methods. As mentioned, it is a common rule, but it may not be the case. Also in this example, these groups correspond to the finding given in Chap. 4.3 and shown in Tables 3 and 4.

5 CONCLUSION

The IINC as well as the k-NN are distance-based classifiers. Then, their function depends on the distance function used. We introduced here two new distance functions, a metric one, denoted as h2 here, and a pseudometric one that we denote h3.

We tested these distance functions in a set of 23 different distance functions, mostly metrics for the IINC classification algorithm. Some of the 23 distance functions are little known and look unusually. We mean that they do not conform to our intuition how a geometric object should look like or what features it should have. For example, we accept that a ball in L_1 metric has the form of a "diamond" and in the L_∞ metric the form of a cube with edges parallel with coordinate axes. These "balls" have a finite volume. But there are distance functions,

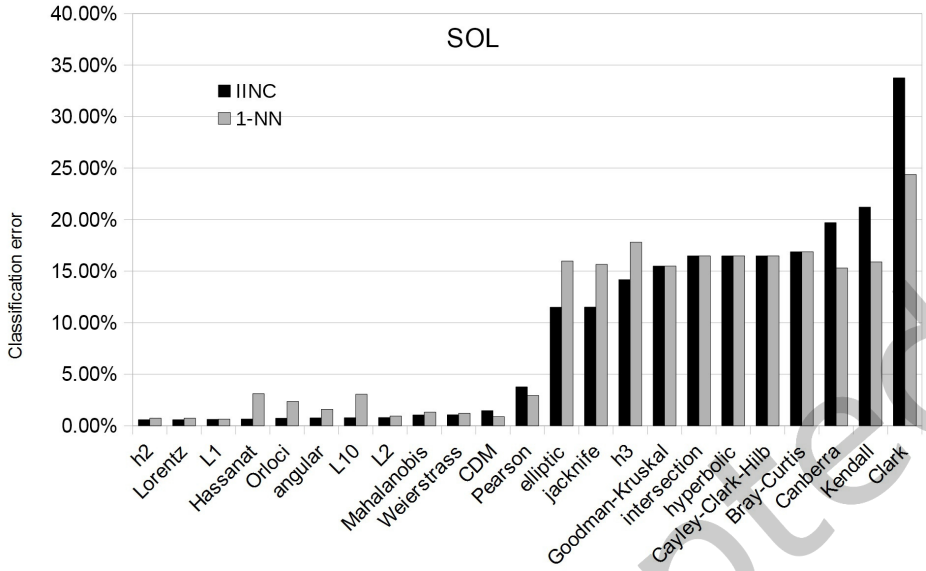

Fig. 3. Classification error as a function of the distance function. The distance functions are ranked according to the classification error for the IINC method.

even metrics, where a "ball" has infinite volume at least for some finite radii. On the other hand, some of the uncommon metrics give a very low classification error in some tasks when used in distance based classifiers.

When using a larger set of tasks, one can summarize classification errors of the IINC with various distance functions as shown in Tables 3 and 4. But a detailed inspection of this Table shows that some distance functions that are by far not the best in any ranking (see Table 5) can be the best for a particular task. Thus, to find the best distance function for a particular set of similar tasks one should test all the distance functions. One may doubt if the best distance function thus found is really the best. Fortunately, the difference in the classification errors for the best and the second best, eventually the third best, is usually very low. So the chance that an unknown distance function would be much better than the best found for a particular task is also very low.

It was shown in [14] that the behavior of other simple nearest neighbor rules (1-NN, k-NN) is very similar to behavior of the IINC classifier. Then, we can generalize that almost surely the influence of a distance function to the classification error is the same. It means that better distance function for a classifier is very probably a better distance function for other classifier. Thus, the new distance functions presented here can be used for the k-NN rules with success.

ACKNOWLEDGMENTS

The work was supported by the Czech Ministry of Education, Youth and Sports in project No. LM2015068 Cooperation on experiments at the Fermi National Laboratory, USA.

REFERENCES

[1] Alkasassbeh, M., Altarawnwh, G.A, Hassanat, A.B. (2015) On enhancing the performance of nearest neighbor classifiers using Hassanat distance metric. *Canadian Journal of Pure and Applied Science*, Vol. 9, No. 1, 6 pp.

[2] Angiulli, F., Fassetti, F. (2013) Nearest neighbor-based classification of uncertain data. *ACM Trans. Knowl. Discov. Data*, Vol. 7, No. 1, Article 1 (March 2013), 35 pp., DOI: 10.1145/2435209.2435210

[3] Ashraf, M., Le, K., Huang, X. (2011), Iterative weighted k-NN for constructing missing feature values in Wisconsin breast cancer dataset. *Proceedings of the 3rd International Conference on Data Mining and Intelligent Information Technology Applications (ICMiA)*, Macao, 24-26 Oct. 2011, pp. 23 - 27, ISBN: 978-1-4673-0231-9 (IEEE)

[4] Benzi,M., Cullum,J.K., Tůma,M.(2000) Robust Approximate Inverse Preconditioning for the Conjugate Gradient Method. *SIAM J. Sci. Comput.* Vol.22, pp. 1318–1332.

[5] Cover, T.M., Hart, P.E. (1967) Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* Vol. 13, No. 1, pp. 21-27.

[6] Derrac, S.G. et al. (2011), A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, Vol. 1, pp. 3-18.

[7] Deza, E., Deza, M.M. (2006) *Dictionary of distances.* Elsevier, Amsterdam, 391 pp.

[8] Deza, M.M., Deza, E. (2009) *Encyklopedia of distances.* Springer, Heildelberg, 590 pp.

[9] Domeniconi, C., Peng, J., Gunopulos, D. (2002), Locally adaptive metric nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 9, pp. 1281-1285.

[10] Dua, D. and Karra Taniskidou, E. (2017), UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. *online*,http://archive.ics.uci.edu/ml

[11] Duda, R., Hart, P., Stork, D.G. (2000) *Pattern Classification.* John Wiley and Sons, 2000.

[12] Grassberger, P., Procaccia, I. (1983) Measuring the strangeness of strange attractors. *Physica*, Vol. 9D, pp. 189-208.

[13] Hassanat, A.B. (2014), Dimensionality Invariant Similarity Measure. *Journal of American Science*, Vol. 10, No. 8, pp. 221-226.

[14] Hassanat, A.B.. Abbadi, M.A., Altarawneh, G.A., Alhasanat, A.A. (2014), Solving problem of K parameter in the KNN classifier using an ensemble learning approach. *Internationa Journal of Computer Science and Information Security (IJCSIS)*, Vol. 12, No. 8, pp. 33-39.

[15] Jiřina, M. and Jiřina, Jr., M. (2013), Utilization of Singularity Exponent in Nearest Neighbor Based Classifier. *Journal of Classification (Springer)*, Vol. 30, No. 1, pp. 3-29. ISSN 0176-4268.

[16] Jiřina, M. and Jiřina, Jr., M. (2014), Correlation Dimension Based Classifier. *IEEE Transactions on Cybernetics*, Vol. 44, pp. 2253-2263. ISSN 2168-2267.

[17] Jiřina, M. and Jiřina, Jr., M. (2015), Classification Using Zipfian Kernel. *Journal of Classification (Springer)*, Vol. 32, No. 2, pp. 305-326. ISSN 0176-4268.

[18] Joachims, T. (1999), Making Large-Scale SVM Learning Practical. *In: Advances in Kernel Methods - Support Vector Learning, eds. B. Scholkopf, C. Burges and A. Smola, MIT-Press.*

[19] Joachims, T. (2008), Program Codes for SVM-Light and SVM-Multiclass. *Available at http://svmlight.joachims.org/* .

[20] Kontorovich, A., Weiss, R. (2015) A Bayes consistent 1-NN classifier. *Proc. of the 18th Int. Conference on Artifficial Intelligence and Statistics* 2015, San Diego, USA, JMLR: W&CP vol. 38, pp. 480-488.

[21] B. Li, Y. W. Chen, Y. Q. Chen (2008), The Nearest Neighbor Algorithm of Local Probability Centers. *IEEE Trans. on Systems, Man, and Cybernetics / Part B: Cybernetics* Vol.38, No. 1, pp. 141-154.

[22] A. L uschow, C. Wartena (2017) Classifying Medical Literature Using k-Nearest-Neighbours Algorithm. *In: Proceedings of the 17th European Networked Knowledge Organization Systems Workshop co-located with the 21st International Conference on Theory and Practice of Digital Libraries 2017 (TPDL 2017)*, Mayr P., Tudhope D., Golub K., Wartena C., Luca E.W.D. (eds), CEUR-WS.org, CEUR Workshop Proceedings, Vol. 1937, pp. 26–38. Available at: http://ceur-ws.org/Vol-1937/paper3.pdf.

[23] Mandelbrot, B.B. (1982) The Fractal Geometry of Nature. *W. H. Freeman and Co.*, ISBN 0-7167-1186-9.

[24] Mishra, A. (2020) k-Nearest Neighbor (k-NN) for Machine Learning. *Data Science Foundation*, May 2020, 4 pp., online: https://datascience.foundation/datatalk/k-nearest-neighbor-k-nn-for-machine-learning

[25] Muja, M., Lowe, D.G. (2014), Scalable nearest neighbor algorithms for high dimensional data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 11, pp. 2227-2240.

[26] Noh, Y.-K., Zhang, B.T., Lee, D.D. (2018), Generative Local Metric Learning for Nearest Neighbor Classification *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 1, pp. 106 - 118.

[27] Paredes, R. (2008), CPW: Class and Prototype Weights learning. *[online]*, Available: http://www.dsic.upv.es/ rpare-des/research/CPW/index.html.

[28] Paredes. R. (2010), Data sets corpora. *[online]*, Available: http://algoval.essex. ac.uk/data/vector/UCI/, in fact, the primary source is S. M. Lucas, Algoval: Algorithm Evaluation over the Web.

[29] Paredes, R., Vidal, E. (2006), Learning Weighted Metrics to Minimize Nearest Neighbor Classification Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 7, pp. 1100-1110, (July 2006).

[30] Piryonesi, S. M.; El-Diraby, T. E. (2020) Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems. *Journal of Transportation Engineering*, Part B: Pavements. Vol. 146, No. 2. doi:10.1061/JPEODX.0000175.

[31] S. Ram irez-Gallego et al. (2017), Nearest Neighbor Classification for High-Speed Big Data Streams Using Spark. *IEEE Trans. on Systems, Man and Cybernetics. Systems* Vol. 47, No. 10, pp. 2727-2739.

[32] Samworth, B. J. (2012), Optimal weighted nearest neighbour classifiers. *The Annals of Statistics* Vol. 40, No. 5, pp. 2733-2763. doi:10.1214/12-AOS1049

[33] Silverman, B. W. (1986) *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

[34] Weinberger, K.Q., Saul, L.K. (2009), Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, Vol. 10, pp. 207-244.

[35] Xiong, F., Kam, M., Hrebien, L., Wang, B., Qi, Y. (2016) Kernelized information-theoretic metric learning for cancer diagnosis usinf high-dimensional moleculr profiling data. *ACM Trans. Knowl. Discov. Data* Vol. 10, No. 4, Article 38 (May 2016), 23 pp., DOI: 10.1145/2789212

[36] Yu, D., Yu, X., Wu, A (2011) Making the Nearest Neighbor Meaningful for Time Series Classification. *Proc. of the 4th International Congress on Image and Signal Processing*. pp. 2481-2485.

[37] Zhang, B., Srihari, S.N. (2004), Fast k-nearest neighbor classification using cluster-based trees *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, Vol. 4, pp. 525 - 528.

[38] Zhang, Q. et al. (2018), Integrating forest inventory data and MODIS data to map species-level biomass in Chinese boreal forests. *Canadian Journal of Forest Research*, Vol.48, pp. 461–479 (2018) dx.doi.org/10.1139/cjfr-2017-0346.