Journal of Classification 30:3-29 (2013)
DOI: 10.1007/s00357-013-9121-z

Utilization of Singularity Exponent in Nearest Neighbor Based Classifier

Marcel Jirina

Institute of Computer Science, Liben, Czech Republic

Marcel Jirina, Jr.

Czech Technical University in Prague, Czech Republic

Abstract: Classifiers serve as tools for classifying data into classes. They directly or indirectly take a distribution of data points around a given query point into account. To express the distribution of points from the viewpoint of distances from a given point, a probability distribution mapping function is introduced here. The approximation of this function in a form of a suitable power of the distance is presented. How to state this power—the distribution mapping exponent—is described. This exponent is used for probability density estimation in high-dimensional spaces and for classification. A close relation of the exponent to a singularity exponent is discussed. It is also shown that this classifier exhibits better behavior (classification accuracy) than other kinds of classifiers for some tasks.

Keywords: Multivariate data; Probability density estimation; Classification; Probability distribution mapping function; Probability density mapping function; Power approximation.

This work was supported by the Institute of Computer Science of the Czech Academy of Sciences RVO: 67985807 and by the Czech Technical University in Prague: CZ68407700.

Authors' Addresses: M. Jirina, Institute of Computer Science, Pod vodarenskou vezi 2, 182 07 Prague 8 – Liben, Czech Republic, e-mail: marcel@cs.cas.cz; M. Jirina, Jr., Faculty of Biomedical Engineering, Czech Technical University in Prague, Nam. Sítna 3105, 272 01, Kladno, Czech Republic, email: jirina@fbmi.cvut.cz

Published online 16 January 2013

1. Introduction

In classification problems, the only known fact is the learning set, i.e. the set of points each of known class. The problem is how to estimate the probability to which class a query point x of the data space belongs. The different approaches to classification can be divided into parametric and nonparametric methods. Parametric methods include e.g. artificial neural networks of different kinds (Haykin 1998), decision trees or forests. Another well-known method is the CART method (Breiman, Friedman, Stone and Olshen 2006) and many more. Nonparametric methods are mostly based on the Bayesian approach (Silverman 1986; Gama 2003), and the k nearest neighbors (k-NN) method (Cover and Hart 1967; Duda, Hart and Stork 2000; Petridis and Kaburlasos 2003; Hinnenburg, Agarwal and Keim 2000; Paredes and Vidal 2006; Zuo, Wang, Zhang and Zhang 2007).

It can be found that chaos theory provides some useful background and tools that could be utilized for estimating the probability mentioned and consequently to use them for classification tasks. The chaos theory is focused on chaotic processes that are described by time series. Therefore, the order of values of variables plays a significant role. There is a lot of data in practical tasks that do not form a series. In spite of that some elements of chaos theory could be used for processing of such a data that do not form series (Mandelbrot 1982). Such data are for example the well-known iris data on three species of iris flower, Fisher (1936). Each individual sample describes one particular flower but neither flowers nor data about them form series. There is a set of flowers as well as a set of data without any ordering. The task is to state to what species a flower belongs according to measured data. We use or necessarily redefine some notions from multifractals theory for a classification task here, especially correlation dimension (Grassberger and Procaccia 1983), which is a measure of the dimensionality of the space occupied by a set of random points. Here, data is considered as a set of points in a multidimensional space. The correlation dimension characterizes also a scaling, i.e. a transformation that enlarges (increases) or shrinks (diminishes) objects by a scale factor. While the correlation dimension is a global feature of the data set as a whole, a singularity exponent has the same role but locally.

Singularity exponents, and also scaling exponents are widely used in multifractal chaotic series analysis. We try here to use these exponents for classification problems. In classification, the task is to properly recognize to which class a presented multivariate sample belongs. This task usually has nothing to do with time series but as shown already by Mandelbrot (1982) any data may posses a fractal or multifractal nature. A problem is that there no time scale even no ordering of samples of the learning (refer-

ence) set exist. Therefore one cannot use such tool as wavelet functions (Lashermes 2004).

Comparing different classification methods applied to different sets of data, one can find a very interesting fact that sometimes a simple approach outperforms the other methods, including those which are very sophisticated. This is the case of the *k*-NN method and its particular version 1-NN method, for example. These methods are deeply elaborated since Cover and Hart (1967) and lots of variants have appeared since, see e.g. Paredes and Vidal (2006) and Zuo et al. (2007).

The *k*-NN method uses the ratio k/V, where k is the number points from the training set in an n-dimensional ball of volume V with its center at point x and a proper radius that corresponds to a distance of the furthest k-th point, see e.g. Silverman (1986), Duda, Hart and Stork (2000). For probability density estimation by the *k*-nearest-neighbor method in E_n, the best value of k must be carefully tuned to find the optimal results. The often used rule of thumb is that k equals the square root of the number of samples of the learning set. As mentioned the nearest neighbor based methods exhibit sometimes surprisingly good results, see e.g. Paredes and Vidal (2006).

Understanding data distribution in a multivariate (multidimensional) space is the first step in the analysis of the data and its features in a multivariate space and will further lead to the design of a powerful yet simple classier suggested in this paper. Multivariate space, even with Euclidean metrics, is much different than the three-dimensional space we live in. Three-dimensional space seems quite natural to us and we have no problems in understanding some geometrical facts. Consider the relation of a unit ball inside a unit cube, for example. The unit ball occupies approximately 56 % of the volume of the cube and 44 % of the volume is in the eight corners. In ten-dimensional Euclidean space the unit ball occupies 0.25 % of unit cube volume. At the same time a ten-dimensional unit cube has 1024 corners and 99.75 % of the volume of the cube is in these corners and nearly nothing is in the central part given by the unit ball inside. Most methods mentioned above do not reflect features of the multidimensional space. Standard methods use individual coordinates of multivariate data space as they are. On the other hand, Beyer et al. (1999) and Pestov (2000a) found that with increasing dimensionality, the distance to the nearest data point approaches the distance of the farthest data point of the data set. In other words, in high-dimensional space, all data points seem to be almost at the same distance from any given point.

This is a special case of much broader theory about a concentration phenomenon or concentration of measure described in detail especially in Chapter 2 of the book by Steele (1997) and later studied e.g. by Pestov (2000b). We do not deal with this general problem here, but note only that

we make war against this phenomenon with the use of "expanded distances" $z = r^q$ ($q > 1$) instead of standard distances r. Exponent q, a distribution mapping exponent, is smaller than but usually comparable with data dimensionality and thus from small difference in r a large difference in z arises.

However, the strangest thing about the k-NN method is the fact that the true distribution of points inside the ball has no influence on the probability estimation. At the same time, only distances are taken into account, but geometric features of multivariate space not. Points can be concentrated in the center or spread along the surface of the ball, the result is the same and thus a part of the information about the distribution of points is lost. There are attempts at taking this effect into account somehow. The simplest is the optimization of k, in fact the size of the ball (Silverman 1986). Other approaches try some way of weighting; e.g., Dudani (1976) has suggested using the weight given by the formula below for classification

$$w_j = \frac{d_k - d_j}{d_k - d_1} \quad \text{for} \quad d_k \neq d_j, \quad \text{and} \quad w_j = 1 \text{ for } d_k = d_j.$$

where d_k is a distance of k-th nearest neighbor from the query point and d_1 a distance of the first nearest neighbor from the query point. These weights are computed for all classes and the class with the largest sum of weights is associated to the query point. Hastie and Tibshirani (1996) use local discriminant analysis for each query point to estimate an effective metrics for searching neighborhoods. In fact, the local discriminant analysis linearly shrinks the original ball neighborhood in directions orthogonal to the local decision boundaries and after that, the k-NN method is used for classification. The method by Paredes and Vidal (2006) is based on the idea of weighting classes or features in the learning set so that the overall classification error on the training set is minimized.

Here we show the possibility of using a suitable transformation (distortion) of the data space so that the distribution of points, which is generally non-uniform, looks uniform-like in the transformed space, at least locally, i.e. in the neighborhood of the query point. It is generally accepted that all classifiers exhibit very good behavior just in cases of a uniform distribution of data.

A core notion in this transformation is a slightly redefined singularity or scaling exponent. The scaling considered here is related to distances between pairs of points in a multivariate space. Thus it is closer to the correlation dimension by Grassberger and Procaccia (1983) than to box-counting or other fractal or multifractal dimension definitions (Barabási and Stanley 1995).

Three new notions are introduced here. The probability distribution mapping function is a mapping of the probability distribution of points in n-

dimensional space to the distribution of points in one-dimensional space of the distances. The *distribution density mapping function* (DDMF) is a one-dimensional analogy to the probability density function. We show that the DDMF is a distribution function of all distances between a particular point (a query point) and all points of a set considered. The power approximation of the *probability distribution mapping function* in the form of $(\text{distance})^q$ is introduced, where we call the exponent q the *distribution mapping exponent* (DME). These notions are local, i.e. are related to a particular point (a query point). We also show that the distribution mapping exponent q is something like a local value of the correlation dimension according to Grassberger and Procaccia (1983). It can be viewed also as the local dimension of the attractor by Froehling, Crutchfield, Farmer, Packard, and Shaw (1981) or simply "exponent" in the sense of Stanley and Melkin (1988).

We found that the contribution of a point of class c to the probability that a query point x is of class c is inversely proportional to the q-th power, where q is the distribution mapping exponent, of the distance between a point of class c and the query point x. Summing up these partial influences for one and the other class we get the numbers S_0 and S_1 and the estimation of the probability that the query point x belongs to class c is S_c/S, where $S = S_0 + S_1$. This is a case of a two-class classifier. A generalization to an arbitrary number of classes is possible and it is straightforward, as we will show later. A welcomed feature of the classifier is that it does not have tuning parameters.

We suppose that the approach presented here can be a starting point for other methods based on the summation of the partial influences of the individual points around the query point x. Thus, finer information about the distribution of the points in the neighborhood of the query point can be taken into account than in the 1-NN and k-NN methods and their modifications. Then it should generally lead to a better classification than these methods.

2. Probability Density Estimation

2.1 Probability Distribution Mapping Function

To study a probability distribution of points (samples, patterns) in the neighborhood of a query point x in n-dimensional Euclidean space E_n, let us build n-dimensional balls with their centers at point x and with volumes V_i, $i = 1, 2, \ldots$ The individual balls are in one another, the $(i\text{-}1)$-th inside the i-th like peels of an onion. Then, the mean density of the points in the i-th ball containing m_i points is $\rho_i = m_i/V_i$. Thus, we have constructed a mapping between the mean density ρ_i of points in the i-th ball and its radius r_i. This

way a complex picture of the distribution of the points in the neighborhood of a query point x can be simplified to a function of a scalar variable – the density of points in a given volume.

Definition. Let the probability distribution mapping function $D(x, r)$ of the query point x in E_n be the function $D(x,r) = \int_{B(x,r)} p(z)dz$, where $p(z)$ is the probability density of the points at z; r is the distance from the query point x and $B(x, r)$ is the ball with center x and radius r.

Definition. Let the distribution density mapping function $d(x, r)$ of the query point x in E_n be the function $d(x,r) = \frac{\partial}{\partial r} D(x,r)$, where $D(x, r)$ is a probability distribution mapping function of the query point x with radius r.

Note. The functions $D(x, r)$ and $d(x, r)$ for a fixed x are one-dimensional analogs to the standard well-known probability distribution function and the probability density function, respectively. In fact, $D(x, r)$ is the probability distribution of a random variable $||r - x||$ for a fixed x, and $d(x, r)$ is the corresponding probability density.

2.2 Power Approximation of the Probability Distribution Mapping Function

Now we propose a transformation with the aim to somehow distort the distribution of the points to look uniform-like because it is generally accepted that all classifiers exhibit very good behavior in cases of a uniform distribution of data.

Let us try to transform the true distribution of points so that the distribution density mapping function is constant, at least in the neighborhood of the query point.

Definition. The power approximation of the probability distribution mapping function $D(x, r)$ is the function r^q such that $\frac{D(x,r)}{r^q} \to const$ for $r \to 0+$. The exponent q is the distribution mapping exponent.

The distribution mapping exponent (DME) reminds so-called correlation dimension by Grassberger and Procaccia (1983). Generally, it can be seen that the correlation integral is a distribution function of all pairwise distances among the data points given. The probability distribution mapping function is a distribution function of the distances from one fixed point x. In the case of finite number of points N, there are $N(N - 1)/2$ pairwise distances and from them one can construct an empirical correlation inte-

gral. Similarly, for each point there are N - 1 distances and from these N - 1 distances one can construct an empirical probability distribution mapping function. There are exactly N such functions and the mean of these functions gives the correlation integral. This definition remains valid for the number of points N going to infinity.

2.3 Indexing Data

Let U be a learning set composed of points (patterns, samples) $x_{cs,}$ where $c = \{0, 1\}$ is the class mark and $s = 1, 2, \ldots, N_c$ is the index of the point within class c; N_c is the number of points in class c and let $N = N_0 + N_1$ be the learning set size. Points x_{cs} of one class are ordered so that index $s = 1$ corresponds to the nearest neighbor, index $s = 2$ to the second nearest neighbor, etc. In Euclidean metrics, $r_s = \|x - x_{cs}\|$ is the distance of the s-th nearest neighbor of class c from point x.

This is the usual way of identifying data samples. For our purpose, we also use a simple form of numbering for the neighbors of point x: U is a learning set composed of points (patterns, samples) $x_{i,}$ where i is the index of the point without respect to which class it belongs; x_i is the i-th nearest neighbor of point x. By the symbol $i(c)$, we denote such index i that point $x_{i(c)}$ belongs to class c.

2.4 Distribution Mapping Exponent Estimation

In this section, we suggest a procedure how to determine the distribution mapping exponent for a classifier, which classifies into two classes. The extension to many classes will be then straightforward.

To estimate the distribution mapping exponent q we use a similar approach, nearly identical, to the approach of Grassberger and Procaccia (1983) for the correlation dimension estimation.

We look for exponent q so, that r_s^q is proportional to index s, i.e.

$$r_s^q = ks \text{ , } s = 1, 2, ..., N_c, \; c = 0 \text{ or } 1, \tag{1}$$

where k is a proportionality constant, which will be eliminated later, so we need not bother with it. Pragmatically said, we are looking for arbitrary q that fulfils the above equation (1) and consequently we show that the q found leads to a uniformity of the transformed space. Using a logarithm we get

$$q \ln(r_s) = \ln(k) + \ln(s) \text{ , } s = 1, 2, ..., N_c \text{ .} \tag{2}$$

This way of finding the optimal exponent is a task of estimating the slope of a straight line by linearly approximating the graph of the dependence of the neighbor's index s as a function of distance in log-log scale. It

is the same problem as in the correlation dimension estimation where equations of the same form as (1) and (2) arise. Grassberger and Procaccia (1983) proposed a solution by linear regression. Dvorak and Klaschka (1990), Guerrero and Smith (2003), Osborne and Provenzale (1989) later proposed different modifications and heuristics. Many of these approaches and heuristics can be used for the distribution mapping exponent estimation, e.g. use of the square root of N_c nearest neighbors instead of N_c to eliminate the influence of a limited number of the points of the learning set. The accuracy of the distribution mapping exponent estimation is the same problem as the accuracy of the correlation dimension estimation. On the other hand, one can find that a small change of q does not essentially influence the classification results.

We solve the system of N_c (or $\sqrt{N_c}$ as mentioned above) equations (2) with respect to an unknown q by the use of standard linear regression for both classes. Thus, for two classes we get two values of q, q_0 and q_1. To get a single value of q we use the arithmetic mean, $q = (q_0 + q_1)/2$. For more classes, the arithmetic mean of the q's for the individual classes can also be used.

At this point, we can say that the distribution mapping exponent q is something like a local effective dimensionality of the data including the true distribution of the points of both classes. The value of q is related to each particular point x and thus varies from one point x to another. Note that our notion of the distribution mapping exponent differs significantly from the local intrinsic dimensionality by Fukunaga and Olsen (1971), Froehling et al. (1981), see also e.g. Costa, Girotra, and Hero (2005), which is defined as an integer representing a rank of the data matrix for the data points within a local region.

2.5 The Method

We come from the assumption that the best estimation of the probability distribution of the data is closely related to the uniformity of the data around the query point x. This uniformity is reached by the use of the transformed distances, i.e. by the use of r^q instead of r.

Informally, let us consider the partial influences of the individual points to the probability that point x is of class c. Each point of class c in the neighborhood of point x adds a little to the probability that point x is of class c, where $c \in \{0, 1\}$ is the class mark. Suppose that this contribution is larger the closer the point considered is to point x and vice versa. Let $p(c|x, i)$ be a partial contribution of the i-th nearest point to the probability that point x is of class c. Then:

For the first (nearest) point $i = 1$ $\quad p(c \mid x,1) \cong \frac{1}{S_n r_1^q}$,

where we use the distribution mapping exponent q instead of the data space dimensionality n; S_n is proportionality constant dependent on the dimensionality and metrics used.

For the second point $i = 2$ $\quad p(c \mid x,2) \cong \frac{1}{S_n r_2^q}$.

And so on; generally for point No. i $\quad p(c \mid x,i) \cong \frac{1}{S_n r_i^q}$.

We add the partial contributions of individual points together by summing up into estimate

$$\hat{p}(c \mid x) \cong \sum_{i=1(c)}^{k} p(c \mid x,i) = \frac{1}{S_n} \sum_{i=1(c)}^{k} 1 / r_i^q . \tag{3}$$

(The sum goes over the indexes i for which the corresponding samples of the learning set are of class c). For both classes there is $\hat{p}(0 \mid x) + \hat{p}(1 \mid x) = 1$ and from it $S_n \cong \sum_{i=1}^{k} 1 / r_i^q$.

Thus we get the form suitable for practical computation

$$\hat{p}(c \mid x) = \frac{\sum_{i=2(c)}^{N} 1 / r_i^q}{\sum_{i=2}^{N} 1 / r_i^q} \tag{4}$$

(The upper sum goes over the indexes i for which the corresponding samples of the learning set are of class c).

At the same time all N points of the learning set are used instead of some finite number as in the k-NN method. Moreover, we do not use the nearest point ($i = 1$) because its influence is more negative than positive on the probability estimate here.

A more exact elicitation for the two class classification and the same number of samples for both classes of the learning set is given in the next section. We show that the generalization is straightforward later.

Theorem 1. *Let the task of classification into two classes be given. Let the size of the learning set be N and let both classes have the same number of*

samples. Let q > 1 be the distribution mapping exponent, let i be the index of the i-th nearest neighbor of point x (without respect to class), and $r_i > 0$ its distance from point x. Then

$$p(c \mid x) = \lim_{N\to\infty} \frac{\sum_{i=2(c)}^{N} 1/r_i^q}{\sum_{i=2}^{N} 1/r_i^q} \tag{5}$$

is a probability that point x belongs to class c.

Proof. For each query point x one can state the probability distribution mapping function $D(x, r_i, c)$. We set this function so that it holds (C is a constant)

$$D(x, r_i^q, c) = C r_i^q$$

in the neighborhood of point x. Using derivation, according to variable $z = r_i^q$, we get $d(x, r_i^q, c) = C$. By the use of $z = r_i^q$, the space is mapped ("distorted") so that the distribution density mapping function is constant in the neighborhood of point x for any particular distribution. The particular distribution is characterized by the particular value of the distribution mapping exponent q at point x. In this mapping, the distribution of the points of class c is uniform.

Let us consider sum $\sum_{i=2}^{N} d(x, r_i^q, c)/r_i^q$. For this sum we have

$$\lim_{N\to\infty} \sum_{i=2}^{N} d(x, r_i^q, c)/r_i^q = p(c \mid x) \lim_{N\to\infty} \sum_{i=2}^{N} 1/r_i^q \quad , \tag{6}$$

because $d(x, r_i^q, c) = d(x, z, c) = p(c|x)$ for all i (uniform distribution has a constant density).

Given the learning set, we have the space around point x "sampled" by the individual points of the learning set. Let $p_c(r_i)$ be an a-posteriori probability that point i at distance r_i from the query point x is of the class c. Then $p_c(r_i)$ is equal to 1 if point x_i is of class c and $p_c(r_i)$ is equal to zero, if not, i.e. if the point is of the other class. Then, the particular realization of $p(c \mid x)\sum_{i=2}^{N} 1/r_i^q$ is sum $\sum_{i=2(c)}^{N} 1/r_i^q$. Using this sum we can rewrite (6) into the form

$$p(c \mid x) \lim_{N\to\infty} \sum_{i=2}^{N} 1/r_i^q = \lim_{N\to\infty} \sum_{i=2(c)}^{N} 1/r_i^q \,.$$

Dividing this equation by the limit of the sum on the left hand side we get

$$p(c \mid x) = \frac{\lim\limits_{N\to\infty} \sum\limits_{i=2(c)}^{N} 1/r_i^q}{\lim\limits_{N\to\infty} \sum\limits_{i=2}^{N} 1/r_i^q},$$

and due to the same limit transition in the numerator and in the denominator we can rewrite it in form (5).

■

Note that the convergence of $S_c = \sum\limits_{i=2(c)}^{N} \frac{1}{r_i^q}$ is faster the larger DME q is. Usually, for multivariate real-life data the DME is also large (and the correlation dimension as well).

Theorem 1 states that probability density is proportional to $1/r_i^q$ and formula (3) uses the sum of these ratios supposing to get a reasonable number for probability density estimation. So it is supposed that for a number of samples going to infinity, the sum would be convergent.

Theorem 2. *Let there exist a mapping of probability density of points of class c in E_n, $E_n \to E_1$: $p(x_{ci}) = p(r_{ci}^q)$ so that*

$$K/r_{c1}^q = p(x_{c1}),\quad K/(r_{c2}^q - r_{c1}^q) = p(x_{c2}),\quad \cdots K/(r_{cNc}^q - r_{c(Nc-1)}^q) = p(x_{cNc}), \qquad (7)$$

where K is a fixed constant that has the same value for both classes. Let there exist a constant $\varepsilon > 0$ and index $k > 2$ so that for each $i > k$ it holds

$$p(x_{ci}) \le \frac{p(x_{c2})}{(1+(i-k)\varepsilon)^{i-k}}. \qquad (8)$$

Then

$$S_c = \sum_{i=2}^{N_c} \frac{1}{r_{ci}^q} = p(x_{c2})K(1+C_c), \qquad (9)$$

where K and C_c are finite constants.

Proof. First we arrange (9) in the form

$$S_c = \sum_{i=2}^{N_c} \frac{1}{r_{ci}^q} = \frac{1}{r_{c2}^q} + \sum_{i=3}^{N_c} \frac{1}{r_{c2}^q + \Delta_{c3} + \Delta_{c4} + \ldots + \Delta_{ci}}.$$

Then using mapping (7) we get

$$S_c = Kp_{c2} + K\sum_{i=3}^{Nc} \frac{1}{\frac{1}{p_{c2}} + \frac{1}{p_{c3}} + \ldots + \frac{1}{p_{ci}}}$$

$$= p_{c2}K(1 + \sum_{i=3}^{Nc} \frac{1}{1 + \frac{p_{c2}}{p_{c3}} + \ldots + \frac{p_{c2}}{p_{ci}}}) \equiv p_{c2}K(1 + \sum_{i=3}^{Nc} P_i) \ . \qquad (10)$$

For individual elements p_{c2} / p_{cj} $(j = 2,3,...,i)$ in denominators of fractions in the sum it holds

$$\frac{p_{c2}}{p_{cj}} = \frac{p_{c2}(1+(i-k)\varepsilon)^{i-k}}{p_{c2}} = (1+(i-k)\varepsilon)^{i-k} \ .$$

Using condition (8) the summed elements P_k, P_{k+1}, … in (10) have the form

$$P_k = \frac{1}{C}, \quad P_{k+1} = \frac{1}{C+1+\varepsilon}, \quad P_{k+2} = \frac{1}{C+1+\varepsilon+(1+\varepsilon)^2}, \ ,$$

$$P_{k+i} = 1/[C + (1+\varepsilon) + (1+2\varepsilon)^2 + \ldots + (1+i\varepsilon)^i] \ .$$

Then according to d'Alembert's criterion

$$\frac{P_{k+i+1}}{P_{k+i}} = \frac{C + (1+\varepsilon) + (1+2\varepsilon)^2 + \ldots + (1+i\varepsilon)^i}{C + (1+\varepsilon) + (1+2\varepsilon)^2 + \ldots + (1+i\varepsilon)^i + (1+(i+1)\varepsilon)^{i+1}} < 1 \ ,$$

$\forall i > 0$ and $\forall \varepsilon > 0$. Then the series is convergent and thus K and C_c are finite constants.

■

Note. In the statement of the theorem the sum need not start just by index $i = 2$. We can start with the nearest neighbor ($i = 1$) or other neighbors ($i > 2$).

Figure 1 and Figure 2 illustrate the convergence of the sum S_c above for one query point for the well-known "vote" data, see Asuncion and Newman (2007). The task is to find whether a president elected will be republican or democrat. The data is 15-dimensional of two classes, republican and democrat, and has a different number of samples. In the learning set, there are 116 times republican and 184 times democrat. The distribution mapping exponent q varies between 4.52 and 14 with the mean value 10.22.

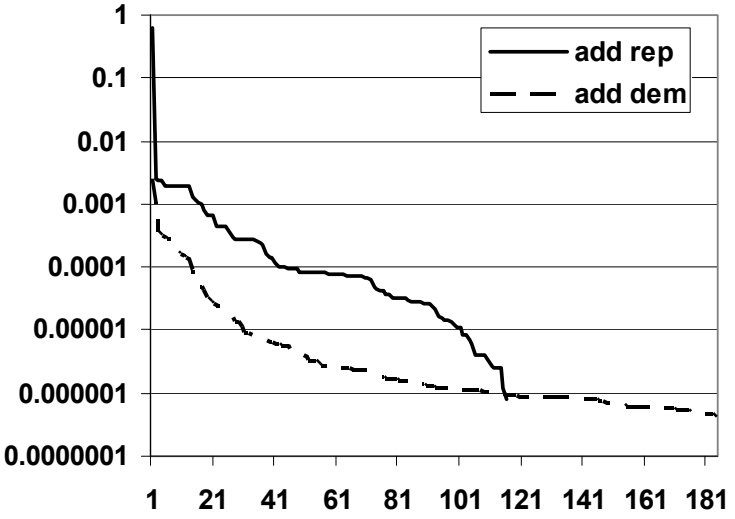

Figure 1. Sample contribution to the sum S_c for the 15-dimensional data "vote" and one particular query point; $q = 7.22$. The upper line corresponds to the republican, the lower line to the democrat. Samples are sorted according to the distance r, i.e. also to the size of the sample contribution to the sum S_c. There are different numbers of samples of one and the other class in the learning set.

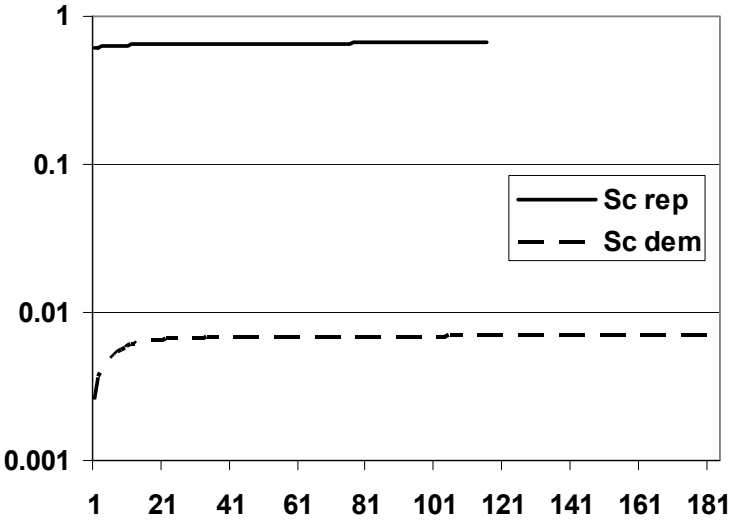

Figure 2. The size of the total sum S_c for the 15-dimensional data "vote" and one particular query point; $q = 7.22$. The upper line corresponds to the republican, the lower line to the democrat. The samples are sorted according to the distance r, i.e. also to the size of the sample contribution to the sum S_c.

2.6 Classifier Construction

In this section, we show how to construct a classifier that incorporates the idea of the distribution mapping exponent. First, compute the distribution mapping exponent q using (2) by linear regression for the query point x. Then, we simply sum up all the components $1/r_i^q$ excluding the nearest point. The sum is calculated class-wise, simultaneously getting numbers S_0 and S_1 for both classes. Then we can get the Bayes ratio or a probability estimate that point $x \in E_n$ belongs to class 1 from the Equations

$$R(x) = \frac{S_1}{S_0} \qquad \text{or} \qquad p_1(x) = \frac{S_1}{S_1 + S_0} \ .$$

Then for a threshold (cut) θ chosen, if $R(x) > \theta$ or $p_1(x) > \theta$ then x belongs to class 1 or else to class 0.

Note that for the different number N_0 and N_1 of the samples of one and the other class formula (4) has the form

$$\hat{p}(c \mid x) \cong \frac{\frac{1}{N_c} \sum_{i=2(c)}^{N} 1/r_i^q}{\frac{1}{N_0} \sum_{i=2(0)}^{N} 1/r_i^q + \frac{1}{N_1} \sum_{i=2(1)}^{N} 1/r_i^q} \ .$$

It is only a recalculation of the relative representation of the different number of the samples of one and the other class.

For M classes, $M \geq 2$ the formula above has the form

$$\hat{p}(c \mid x) = \frac{\frac{1}{N_c} \sum_{i=2(c)}^{N} 1/r_i^q}{\sum_{k=1}^{M} \frac{1}{N_k} \sum_{i=2(c)}^{N} 1/r_i^q} \ . \tag{11}$$

Final algorithm proposed here consists of the following steps.

Input:

- The learning set with samples of C classes and N_c samples of each class, $c = 1, 2, ..., C$, and total N samples.
- The query point x.
- Threshold θ (for a two class classification).

Output:

- Estimates of probabilities $\hat{p}(c \mid x)$ that point x belongs to class c, c = 1, 2, ..., C.
- Class k to which the query point x most probably belongs.

For query point x do {
Sort all samples of the learning set according to the distance from the query point x.

Assign indexes i to sorted samples of the learning set without respect to class so that $i = 1$ is assigned to the nearest neighbor, $i = 2$ to the second nearest neighbor etc.

For $c = 1, 2, \ldots C$ {
Estimate value of the distribution mapping exponent q.
Compute probability $\hat{p}(c \mid x)$ according to (11).
}
// now we have estimates of probabilities that the query point // x belongs to individual classes.)

For a classification task do {
if $C = 2$ then {// two class problem
if $\hat{p}(0 \mid x) > \theta$ then point x belongs to class $k = 0$.
else point x belongs to class $k = 1$.
}
else {
$k = \arg\max_{c=1}^{C}(\hat{p}(c \mid x))$ is the estimated class to which point x belongs.
}
}
}

3. Experiments

We demonstrate the features and the power of the classifier both on synthetic and real-life data.

3.1 Distribution Mapping Exponent Estimation

An essential part of the algorithm and experiments is the distribution mapping exponent estimation. As said above, many approaches and heuristics for correlation dimension estimation can be used for this task. In experiments below we used algorithms of distribution mapping exponent estimation as follows.

- Linear regression over the whole learning set. This systematically underestimates q by factor 2 and then it is corrected by this factor.
- Linear regression over the nearer half of points of the learning set.
- t-score robust regression (Fabian and Vajda 2003) over the nearer half of points of the learning set and over the nearer square root of number of points of the learning set
- Takens estimator (Takens 1985) over the nearer half of points of the learning set and over the nearer square root of number of points of the learning set.

It will be seen below that spread of results with different algorithms and the part of the learning set really used is generally relatively low. A small change of q does not essentially influence the classification results.

3.2 Synthetic Data

Synthetic data according to Paredes and Vidal (2006) is two-dimensional and consists of three two-dimensional normal distributions with identical a-priori probabilities. If μ denotes the vector of the means and $\boldsymbol{C}_m$ is the covariance matrix, there is

Class A: $\mu = (2, 0.5)^t$, $\boldsymbol{C}_m = (1, 0; 0, 1)$ (identity matrix)
Class B: $\mu = (0, 2)^t$, $\boldsymbol{C}_m = (1, 0.5; 0.5, 1)$
Class C: $\mu = (0, -1)^t$, $\boldsymbol{C}_m = (1, -0.5; -0.5, 1)$.

Figure 3 shows the results obtained by the different methods for the different learning sets sizes from 8 to 256 samples and a testing set of 5000 samples all from the same distributions and independent. Each point in the figure was obtained by averaging over 100 different runs. For other methods, i.e. the 1-NN method with L2 metrics and variants of the LWM method by Paredes and Vidal (2006), the values were estimated from literature cited. It is seen that in this synthetic experiment, the DME based method presented here reliably outperforms all other methods shown and for a large number of samples fast approaches the Bayes limit. For the distribution mapping estimation the linear regression over the whole learning set was used.

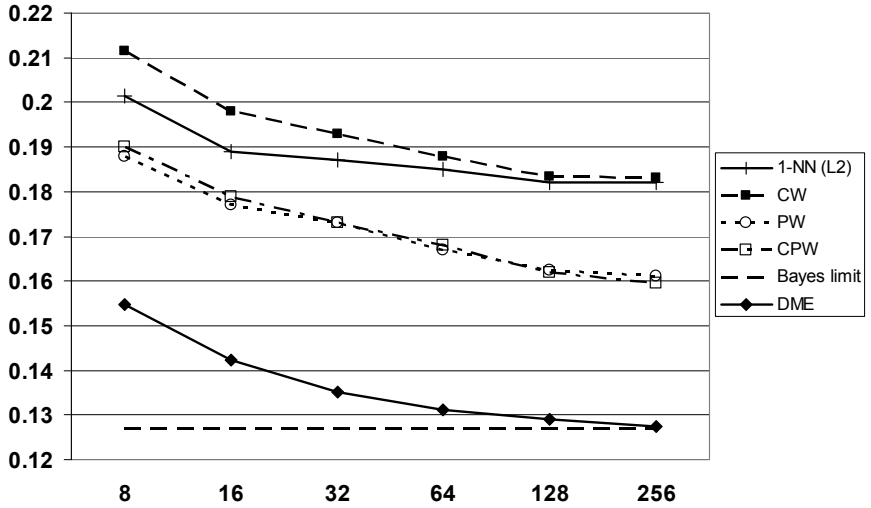

Figure 3. Comparison of the classification errors of the synthetic data for the different approaches. In the legend, 1-NN (L2) means the 1-NN method with Euclidean metrics, CW, PW, and CPW are three variants of the method by Paredes and Vidal (2006); the points are estimated from the reference cited. DME means the method presented here.

Note that in this test, the error of the DME estimation is combined with numerical errors, and with a negative influence of the low number of the samples giving the results presented in Figure 3.

3.3 Data from Machine Learning Repository

Tasks from UCI Machine Learning Repository – Comprehensive Tests

The testing should show the classification ability of the DME method for some tasks and also shows the classification ability relative to the other published methods and the results for the same data sets.

We used real-life tasks from the UCI Machine Learning Repository; see Asuncion and Newman (2007). 24 databases have been used for the classification task into two to 26 classes. The number of attributes not including the class mark differs from 4 to 180. Basic characteristics of data sets are summarized in Table 1. Data originally from the UCI Machine learning repository (Asuncion and Newman, 2007) were gained mostly from (Paredes, 2008) (denoted by P in column Source in the table). These data sets are ready for a run with a classifier. We used all data sets in this corpus. Each task consists of 50 pairs of training and testing sets corres-

Table 1. Characteristics of data sets basically from the UCI Machine learning repository [MLR] gained from or modified according to different sources. Abbreviations for sources: P – Paredes (2008); P2 – Paredes (2009); UCI MLR - Asuncion and Newman (2007). Note (1): Iris data are used without Setosa class, i.e. two classes Versicolor and Virginica only according to Friedman (1994).

Dataset	Dimen-sion (#att-ributes)	Number of classes	Total samples	Learning set size	Test set size	Cross valida-tion	Source
Australian	42	2	690	551	139	50	P
Balance	4	3	625	499	126	50	P
Cancer	9	2	683	546	137	50	P
Diabetes	8	2	768	614	154	50	P
DNA	180	3	31186	2000	1186	1	P2
German	24	2	1000	800	200	50	P
Glass	9	6	215	169	46	50	P
Heart	25	2	270	216	54	50	P
Ionosphere	34	2	351	280	71	50	P
Iris (1)	4	2 (3)	100 (150)	90	10	10	UCI MLR
Led17	24	10	2000	1595	405	50	P
Letter	16	26	20000	16000	4000	1	UCI MLR
Liver	6	2	345	276	69	50	P
Monkey1	17	2	556	444	112	50	P
Phoneme	5	2	5404	4322	1082	50	P
Satimage	36	7	6435	4435	2000	1	UCI MLR
Segmen	19	7	2310	1848	462	50	P
Sonar	60	2	208	165	43	50	P
Vehicle	18	4	846	675	171	50	P
Vote	16	2	435	347	88	50	P
Vowel	10	11	528	418	110	50	P
Waveform21	21	3	5000	3998	1002	50	P
Waveform40	40	3	5000	3999	1001	50	P
Wine	13	3	178	141	37	50	P

ponding to 50-fold cross validation. For DNA data (Paredes 2009), Letter data (Letter recognition (Asuncion and Newman 2007)), and Satimage (Statlog Landsat Satellite (Asuncion and Newman,2007)) the single partition into training and testing sets according to specification in (Asuncion and Newman 2007) was used. We also added the popular Iris data set. Iris

data were taken from (Asuncion and Newman 2007) but we use them without Setosa class, i.e. we used two classes Versicolor and Virginica only according to Friedman (1994) and then we have split remaining data into 10 pairs for ten-fold cross validation.

Classification Methods Compared

The best results obtained with six different classification methods are shown in Table 2. We used six classification methods as follows. Notation corresponds to columns in Table 3.

- Bayes – the naïve Bayes method that uses 10 bins histograms (Silverman 1986). A short but comprehensive description one can find in paper by Gama (2003).
- 1-NN – standard nearest neighbor method (Cover and Hart 1967).
- ParedBest – the best results obtained by three variants of method by Paredes and Vidal (2006) using original software available from (Paredes 2009). Detailed results for three variants of this method are shown in the Appendix.
- SVMbest – the best results obtained with support vector machine (Joachims 1999; Tsochantaridis, Joachims, Hofmann, and Altun 2005) using four types of kernels (default values for other parameters) and software available at (Joachims 2008). Detailed results for four different kernels are shown in the Appendix.
- MLP – the well-known Multilayer Perceptron artificial neural network (Haykin 1998).
- DMEbest – the best results obtained with variants of the method presented here. Variants mean different approaches to DME estimation. Detailed results for different algorithms for stating the distribution mapping exponent and the part of the learning set used are shown in Table 3.

In Table 2 in each row the best result is denoted by bold numerals. Details of results for the method presented here are shown in Table 3. The method is the same and follows the algorithm shown in Chapter 2.6. Six variants in Table 3 differ in approach to the distribution mapping estimation and are described in legend in the table caption.

4. Discussion

Our model utilizing a transformation of the data space in the form $(distance)^q$ comes from the demand to have a uniform distribution of

Table 2. Condensed comparison of six types of classification methods including DME method presented here. The best result for each particular data set are shown in bold.

Dataset	Bayes	1-NN	ParedBest	SVMbest	MLP	DMEbest
Australian	14.88%	34.29%	31.91%	35.99%	15.12%	**14.20%**
Balance	15.17%	22.05%	13.68%	33.17%	**3.85%**	24.85%
Cancer	**2.68%**	4.83%	3.41%	16.32%	4.71%	3.69%
Diabetes	25.19%	32.76%	29.60%	29.64%	22.92%	24.75%
DNA	6.66%	23.44%	3.71%	**0.00%**	12.14%	28.33%
German	**24.97%**	33.74%	29.79%	27.25%	38.80%	27.64%
Glass	47.37%	30.81%	**30.75%**	32.63%	35.85%	34.47%
Heart	18.44%	41.48%	38.15%	37.22%	17.98%	**17.96%**
Ionosphere	9.26%	14.07%	**5.87%**	18.52%	18,39%	15.58%
Iris	9.82%	5.91%	**4.91%**	5.55%	5,26%	5.91%
Led17	**0.00%**	24.92%	0.02%	11.52%	0.00%	0.32%
Letter	28.98%	4.35%	3.25%	**2.68%**	3.26%	5.73%
Liver	39.42%	39.25%	38.14%	**35.54%**	32.56%	40.09%
Monkey1	28.01%	29.47%	**0.04%**	2.94%	1.44%	8.22%
Phoneme	21.47%	**11.50%**	11.60%	14.39%	22.24%	16.49%
Satimage	19.15%	10.55%	**9.25%**	24.30%	14.61%	11.95%
Segmen	9.85%	4.30%	**3.76%**	34.27%	6.09%	6.48%
Sonar	31.46%	22.62%	19.42%	**19.67%**	42.31%	24.25%
Vehicle	38.40%	35.08%	29.95%	**26.23%**	21.80%	29.37%
Vote	9.70%	8.13%	**5.35%**	22.64%	14.81%	9.28%
Vowel	26.64%	1.37%	**1.33%**	8.54%	12.12%	6.66%
Waveform21	19.26%	21.91%	18.30%	26.34%	15.16%	**15.05%**
Waveform40	20.31%	23.34%	24.55%	32.25%	17.04%	**16.49%**
Wine	5.50%	27.05%	19.46%	8.85%	**0.00%**	5.04%

points, at least locally. There is an interesting relationship between the correlation dimension and the distribution mapping exponent. The former is a global feature of the fractal or data generating process. The latter is a local feature of the data set and is closely related to a particular query point. On the other hand, if linear regression were used, the computational procedure is almost the same in both cases. Moreover, it can be found that the values of the distribution mapping exponent lie sometimes in a narrow,

Table 3. Error rates for 24 data sets with the DME classifier with different approaches to DME estimation. Legend for columns headings:

DMEL2 - Linear regression over the whole learning set.
DME1/2L2 - Linear regression over the nearer half of points of the learning set.
DME1/2FabiL2 - t-score robust regression (Fabian and Vajda 2003) over the nearer half of points of the learning set
DMEsqrFabiL2 - t-score robust regression (Fabian and Vajda 2003) over the nearer square root of number of points of the learning set
DME1/2TakeL2 - Takens estimator (Takens 1985) over the nearer half of points of the learning set.
DMEsqrTakeL2 - Takens estimator (Takens 1985) over the nearer square root of number of points of the learning set.

Dataset	DMEL2	DME1/2L2	DME1/2FabiL2	DMEsqrFabiL2	DME1/2TakeL2	DMEsqrTakeL2
Australian	17.34%	14.83%	14.78%	14.20%	15.84%	15.03%
Balance	25.17%	24.85%	24.86%	24.99%	25.01%	25.12%
Cancer	3.70%	3.69%	3.70%	3.98%	3.72%	3.79%
Diabetes	25.39%	24.95%	24.97%	24.75%	25.18%	24.82%
DNA	25.04%	31.70%	31.70%	30.69%	33.31%	28.33%
German	29.20%	27.65%	27.64%	27.90%	27.74%	28.14%
Glass	32.95%	34.52%	34.47%	35.20%	34.57%	35.58%
Heart	19.00%	18.15%	18.15%	17.96%	18.15%	18.26%
Ionosphere	15.41%	16.09%	16.09%	16.29%	16.44%	15.58%
Iris	5.91%	5.91%	5.91%	5.91%	5.91%	5.91%
Led17	3.62%	0.43%	0.43%	0.32%	0.87%	1.82%
Letter	5.05%	6.65%	6.68%	10.30%	6.55%	5.73%
Liver	39.68%	40.09%	40.17%	40.26%	40.12%	40.23%
Monkey1	6.72%	9.19%	9.21%	10.42%	8.22%	10.42%
Phoneme	13.37%	16.93%	17.00%	19.25%	16.49%	20.17%
Satimage	10.65%	12.85%	12.85%	13.90%	13.05%	11.95%
Segmen	5.25%	6.48%	6.54%	8.23%	6.61%	6.81%
Sonar	23.81%	25.65%	25.65%	27.43%	24.92%	24.25%
Vehicle	28.86%	30.65%	30.67%	31.51%	30.79%	29.37%
Vote	8.50%	9.88%	9.90%	10.11%	9.49%	9.28%
Vowel	5.67%	10.21%	10.28%	13.81%	6.86%	6.66%
Waveform21	15.82%	15.06%	15.05%	15.29%	15.35%	15.60%
Waveform40	17.94%	16.49%	16.49%	16.83%	16.78%	17.27%
Wine	4.99%	5.76%	5.76%	6.39%	5.04%	5.38%

sometimes in a rather wide interval around its mean value. Not surprisingly, the mean value of the distribution mapping exponent over all samples is not far from the correlation dimension. A question arises what the relation of the distribution mapping exponent statistic is to the overall accuracy of the classification. A direct use of the correlation dimension for the classification is another question. Introducing the notion of the distribution mapping exponent and the transformation of the distances may be a starting point for a more detailed description of the local behavior of the multivariate data and for the development of new approaches to the data analysis, including classification problems.

We do not use the first nearest neighbor in our formulas. In our approach, a true distribution is mapped to the uniform distribution. For uniform distribution, it holds that the *i*-th neighbor distance from a given point has an Erlang distribution of *i*-th order. For an Erlang distribution of *i*-th order, the relative statistical deviation, i.e. the statistical deviation divided by the mean, is equal to $1/\sqrt{i}$. Then the relative statistical deviation diminishes with the index of the neighbor and for the nearest neighbor is equal to 1 which also follows from the fact that Erlang(1) distribution is just exponential distribution. So, there is a large relative spread in the positions of the nearest neighbor and, at the same time, its influence is the largest. It appears better to eliminate the influence of the first nearest neighbor. Theorem 1 remains valid when the first neighbor is included, as well as if more than one neighbor is excluded. In any case the influence of the nearest neighbors is large as illustrated in Figures 1 and 2 while farther neighbors have nearly no influence. One could say that the distribution mapping exponent approach somehow automatically controls the size of neighborhood, which influences the estimation.

We have shown that the distribution mapping exponent (DME) can be computed by similar approaches as with the correlation dimension and that the polynomial transformation which uses the distribution mapping exponent as the exponent leads to a classification algorithm.

By the use of the notion of distance, i.e. a simple transformation $E_n \rightarrow E_1$, the problems with the curse of dimensionality are easily eliminated. The curse of dimensionality (Bellman 1961, Pestov 2000a) means that the computational complexity grows exponentially with the dimensionality *n*, while complexity here grows only linearly. In any case there is a loss of information on the true distribution of the points in the neighborhood of the query point which is fortunately not fundamental for distance based methods. On the other hand, the distribution mapping exponent method uses more information than the 1-NN and *k*-NN methods as it takes the individual distances of all points of the learning set into account.

Our experiments demonstrate that the simplest classifier based on the ideas introduced here can outperform other methods for some data sets. We

compared our method with standard methods as naive Bayes and 1-NN method as well as with two powerful and complex methods the Learning Weighted Metrics (LWM) by Paredes and Vidal (2006) and the Support Vector Machine (SVM), see (Joachims 1999, Tsochantaridis et al. 2005). Each method is good for some group of tasks as seen in Table 2 but the basic 1-NN method is apparently outperformed by any other method studied here.

On the other hand, the target of this paper was to present a basically new approach to probability density estimation and classification. Some refinement of this approach can bring better results in the future. There is an observation that the distribution mapping exponent (DME), in fact a redefined scaling exponent, shows an inherent multifractal nature of data. A multifractal system (Stanley-Melkin 1988) is a generalization of a fractal system in which a single exponent (the fractal dimension) is not sufficient to describe its dynamics; instead, a continuous spectrum of exponents (the so-called singularity spectrum) is needed or, as presented here, a stating of exponent to each particular point. Further research may try to use another description of multifractal phenomena to get better probability density estimation and thus better classification.

Appendix

Details on classification accuracy of the LWM and SVM methods computed using software available from Paredes (2009) and Joachims (2008).

Table 4. Classification accuracy for 26 datasets for three variants of the LWM method by Paredes and Vidal (2006). The software package by Paredes (2009) was used with default control parameters.

Dataset	CW	PW	CPW
Australian	31.91%	36.43%	33.88%
Balance	18.44%	13.68%	18.01%
Cancer	3.75%	4.06%	3.41%
Diabetes	30.62%	31.15%	29.60%
DNA	4.30%	40.98%	3.71%
German	29.87%	33.59%	29.79%
Glass	31.40%	30.75%	31.60%
Heart	39.22%	39.63%	38.15%
Ionosphere	7.92%	6.24%	5.87%
Iris	5.91%	4.91%	5.91%

Dataset	CW	PW	CPW
Led17	0.02%	24.10%	0.02%
Letter	3.25%	4.23%	3.25%
Liver	39.25%	38.14%	38.58%
Monkey1	0.04%	23.35%	0.04%
Phoneme	12.61%	11.60%	12.30%
Satimage	10.95%	9.25%	9.25%
Segmen	3.76%	4.30%	3.82%
Sonar	19.42%	21.19%	19.94%
Vehicle	30.05%	35.93%	29.95%
Vote	5.70%	8.16%	5.35%
Vowel	1.33%	1.38%	1.41%
Waveform21	22.15%	21.00%	18.30%
Waveform40	24.55%	35.56%	30.83%
Wine	19.79%	27.84%	19.46%

Table 5. Classification accuracy for 26 datasets and for four kernels of the Support vector machine (SVM) (Joachims 1999, Tsochantaridis et al. 2005). The software packages by Joachims (2008) were used with default control parameters.

Dataset	SVMlin	SVMpoly	SVM-RBF	SVMsigmo
Australian	35.99%	40.69%	41.33%	41.33%
Balance	33.17%	47.00%	33.56%	33.56%
Cancer	16.34%	16.32%	17.08%	17.08%
Diabetes	29.64%	29.77%	32.74%	32.74%
DNA	NA	NA	NA	NA
German	27.25%	27.94%	29.64%	29.64%
Glass	32.63%	33.81%	46.62%	46.62%
Heart	38.89%	37.22%	37.22%	37.22%
Ionosphere	22.75%	NA	18.52%	18.52%
Iris	6.55%	8.55%	5.55%	6.55%
Led17	11.52%	21.23%	16.97%	16.97%
Letter	2.68%	NA	3.98%	2.68%
Liver	37.68%	37.57%	35.54%	35.54%
Monkey1	23.54%	NA	2.94%	2.94%

Phoneme	21.71%	17.63%	14.39%	14.39%
Satimage	44.85%	NA	24.30%	44.85%
Segmen	34.27%	NA	46.48%	46.48%
Sonar	26.58%	22.72%	19.67%	26.58%
Vehicle	26.23%	NA	28.23%	28.23%
Vote	22.64%	22.78%	23.54%	23.54%
Vowel	8.54%	NA	13.64%	13.64%
Waveform21	26.34%	NA	26.94%	26.94%
Waveform40	32.30%	NA	32.25%	33.07%
Wine	41.13%	8.85%	27.77%	41.13%

References

AGARWAL, S., GRAEPEL, T., HERBRICH, R., HAR-PELED, S., and ROTH, D. (2005), "Generalization Bounds for the Area Under the ROC Curve", *Journal of Machine Learning Research, 6,* 393–425.

ASUNCION, A., and NEWMAN, D.J. (2007), *UCI Machine Learning Repository*, Irvine, CA: University of California, School of Information and Computer Science, cited January 26, 2008, available at http://www.ics.uci.edu/~mlearn/MLRepository.html .

BARABÁSI, A.-L., and STANLEY, H.E. (1995), *Fractal Concepts in Surface Growth*, New York: Cambridge University Press.

BELLMAN, R.E. (1961), *Adaptive Control Processes*, New Jersey: Princeton University Press.

BEYER, K., GOLDSTEIN, J., RAMAKRISHNAN, R., and SHAFT, U. (1999), "When is "Nearest Neighbor" Meaningful?" in *Proceedings of the 7th International Conference on Database Theory, Jerusalem, Israel*, pp. 217–235.

BREIMAN, L., FRIEDMAN, J., STONE, C.J., and OLSHEN, R.A. (1984), *Classification and Regression Trees,* Boca Raton, Florida: Chapman and Hall/CRC.

COSTA, J.A., GIROTRA, A., and HERO, A.O. (2005), "Estimating Local Intrinsic Dimension with k-Nearest Neighbor Graphs", *IEEE Workshop on Statistical Signal Processing (SSP), Bordeaux*, pp. 417–422.

COVER, T.M, and HART, P.E. (1967), "Nearest Neighbor Pattern Classification", *IEEE Transactions on Information Theory, 13(1)*, 21–27.

DUDA, R.O, HART, P.E., and STORK, D.G. (2000), *Pattern Classification* (2nd ed.), New York: John Wiley and Sons, Inc.

DUDANI, S.A. (1976), "The Distance-Weighted K-Nearest Neighbor Rule", *IEEE Transactions on Systems, Man, and Cybernetics, 6,* 325–327.

DVORAK, I., and KLASCHKA, J. (1990), "Modification of the Grassberger-Procaccia Algorithm for Estimating The Correlation Exponent of Chaotic Systems with High Embedding Dimension", *Physics Letters A, 145(5),* 225–231.

FABIAN, Z., and VAJDA, I. (2003), "Core Functions and Core Divergencies of Regular Distributions", *Kybernetika, 39,* 29–42.

FISHER, R.A. (1936), "The Use of Multiple Measurements in Taxonomic Problems", *Annual Eugenics, 7(2),* 179–188.

FRIEDMAN, J.H. (1994), "Flexible Metric Nearest Neighbor Classification", Technical Report 113, Stanford University, Department of Statistics.

FUKUNAGA, K., and OLSEN, D.R. (1976), "An Algorithm for Finding Intrinsic Dimensionality of Data", *IEEE Transactions on Computers, 20(2),* 176–183.

FROEHLING, H., CRUTCHFIELD, J.P., FARMER, D., PACKARD, N.H., and SHAW, R. (1981), "On Determining the Dimension of Chaotic Flows*", Physica, 3D*, 605–617.

GAMA, J. (2003), "Iterative Bayes", *Theoretical Computer Science, 292*, 417–430.

GRASSBERGER, P., and PROCACCIA, I. (1983), "Measuring the Strangeness of Strange Attractors", *Physica, 9D,* 189–208.

GUERRERO, A., and SMITH, L.A. (2003), "Towards Coherent Estimation Of Correlation Dimension", *Physics Letters A, 318,* 373–379.

HAKL, F., HLAVÁČEK, M., and KALOUS, R. (2002), "Application of Neural Networks Optimized by Genetic Algorithms to Higgs Boson Search", in *Proceedings of the 6th World Multi-Conference on Systemics, Cybernetics and Informatics* (Vol. 11), eds. N. Callaos, M. Margenstern, and B. Sanchez, pp. 55–59.

HASTIE, T., and TIBSHIRANI, R. (1996), "Discriminant Adaptive Nearest Neighbor Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18,* 607–616.

HINNENBURG, A., AGGARWAL, C.C., and KEIM, D.A. (2000), "What is the Nearest Neighbor in High Dimensional Spaces*?", in Proceedings of the 26th VLDB Conference, Cairo, Egypt*, pp. 506–515.

HAYKIN, S. (1998), *Neural Networks: A Comprehensive Foundation* (2nd ed.), Englewood Cliffs, NJ: Prentice Hall.

JOACHIMS, T. (1999), "Making Large-Scale SVM Learning Practical", in *Advances in Kernel Methods - Support Vector Learning*, eds. B. Schölkopf , C. Burges and A. Smola, MIT-Press.

JOACHIMS, T. (2008), *Program Codes for SVM-Light and SVM-Multiclass*, cited 1.12.2008, available at http://svmlight.joachims.org/ .

KIM, H.Ch., and GHAHRAMANI, Z. (2006), "Bayesian Gaussian Process Classification with the EM-EP Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence,28(12),* 1948–1959.

LASHERMES, B., ABRY, P., and CHAINAIS, P. (2004), *Scaling Exponents Estimation for Multiscaling Processes*, CNRS, Physics Laboratory, Ecole Normale Supérieure, Lyon, France, available at http://perso.ens-lyon.fr/patrice.abry/MYWEB/VERSIONSPS/ lacicassp04.pdf.

MANDELBROT, B. (1982), *The Fractal Theory of Nature*, New York: W.H. Freeman and Co.

OSBORNE, A.R., and PROVENZALE, A. (1989), "Finite Correlation Dimension for Stochastic Systems with Power-Law Spectra", *Physica D, 35*, 357–381.

PAREDES, R., and VIDAL, E. (2006), "Learning Weighted Metrics to Minimize Nearest-Neighbor Classification Error", *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(7),* 1100–1110.

PAREDES, R. (2008), *Data Sets Corpora*, available at http://algoval.essex.ac.uk/data/vector/UCI/

PAREDES, R. (2009), *CPW: Class and Prototype Weights Learning,* available at http://www.dsic.upv.es/~rparedes/research/CPW/index.html.

PESTOV, V. (2000a), "On the Geometry of Similarity Search: Dimensionality Course and Concentration of Measure", *Information Processing Letters, 73,* 47–51.

PESTOV, V. (2000b), "The Concentration Phenomenon and Topological Groups", *Topology Atlas, 5,* 5–10.

S, V., and KABURLASOS, V.G. (2003), "FINkNN: A Fuzzy Interval Number k-Nearest Neighbor Classifier for Prediction of Sugar Production from Populations of Samples", *Journal of Machine Learning Research, 4,* 17–37.

SAUL, L.K. (2003), "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds", *Journal of Machine Learning Research, 4,* 119–155.

SILVERMAN, B.W. (1986), *Density Estimation for Statistics and Data Analysis*, London: Chapman and Hall.

STANLEY, H.E., and MELKIN, P. (1988), "Multifractal Phenomena in Physics and Chemistry (Review)", *Nature, 335*, 405–409.

STEELE, J.M. (1997), "Probability Theory and Combinatorial Optimization*", CBMS-NSF Regional Conference Series in Applied Mathematics, CB69*, Philadelphia, PA:Society for Industrial and Applied Mathematics (SIAM).

TAKENS, F. (1985), "On the Numerical Determination of the Dimension of the Attractor", in *Dynamical Systems and Bifurcations, Lecture Notes in Mathematics, Vol. 1125,* Berlin: Springer, pp. 99–106.

TSOCHANTARIDIS, I., JOACHIMS, T., HOFMANN, T., and ALTUN, Y. (2005), "Large Margin Methods for Structured and Interdependent Output Variables", *Journal of Machine Learning Research (JMLR), 6,* 1453–1484.

ZUO, W., WANG, K., ZHANG, H., and ZHANG, D. (2007), "Kernel Difference-Weighted k-Nearest Neighbors Classification", in *ICIC 2007*, eds. D.-S. Huang, L. Heutte, and M. Loog, Springer-Verlag LNAI 4682, pp. 861–870.