Correlation Dimension-Based Classifier

Marcel Jirina[1] and Marcel Jirina, jr.[2]

[1] Institute of Computer Science, Pod vodarenskou vezi 2,
182 07 Prague 8 – Liben, Czech Republic

marcel@cs.cas.cz

[2] Faculty of Information Technology, Czech Technical University in Prague,

Thakurova 9, 160 00 Prague 6, Czech Republic

jirina@fbmi.cvut.cz

Abstract

Correlation dimension, singularity exponents, also scaling exponents are widely used in multifractal chaotic series analysis. Correlation dimension and other measures of effective dimensionality are used for characterization of data in applications. A direct use of correlation dimension to multidimensional data classification has not been hitherto presented. There are observations that the correlation integral is a distribution function of distances between all pairs of data points, and that by using polynomial expansion of distance with exponent equal to the correlation dimension this distribution is transformed into locally uniform. The classifier is based on consideration that the "influence" of neighbor points of some class on the probability that the query point belongs to this class is inversely proportional to its distance to the correlation dimension - power. New classification approach is based on summing up all these influences for each class. We prove that a resulting formula gives an estimate of probability of class – not a measure of membership to a class only – to which the query point belongs. For this assertion to be valid it is necessary that exponent of the polynomial transformation must be the correlation dimension. We also propose an "averaging approach" that speeds up computation of the correlation dimension especially for large data sets. It is demonstrated that the correlation dimension based classifier can outperform more sophisticated classifiers.

Final version published in:

IEEE Transactions on Cybernetics, Vol. 44, No. 12 (2014), pp. 2253-2263. ISSN 2168-2267; Published online 5 March, - (2014). ISSN 2168-2267.

1 Introduction

Correlation dimension introduced by Grassberger and Procaccia [1] is a means for characterizing the nature of fractals. The correlation dimension (CD) can be used for the characterization of very general data sets usually described by some stochastic characteristics.

Each point of multivariate data including fractals can be mathematically described by a vector or point in the so-called embedding space of some dimension. Neither data nor fractal fill the multivariate space fully, and the measure of this "filling" is measured by effective dimensionality of the given data set or fractal especially by correlation dimension that is derived from correlation integral. It is a common license to say that a data set is a fractal. In fact, a finite or countable data set cannot be fractal; fractal is – or can be – a measure on it. A measure considered here is a distance because the correlation dimension is defined as a function of distances between points of the data set. Singularity exponents, also scaling exponents are widely used in multifractal chaotic series analysis. In applications it can be found that effective dimensionality, scaling exponents, and correlation dimension are used for characterization of data in different ways before a classification procedure is employed.

However, a direct application of correlation dimension to the approximation of probability of class at a given point and for classification [4], [5], [6], [10] has not been presented up to now.

Here we show that the correlation dimension can be useful for this approximation and for the construction of a new classifier. The correlation dimension characterizes the correlation integral and the correlation integral is, in fact, a distribution function of distances between all pairs of data points. Thus, the approach presented here is closely related to the nearest neighbor-based methods [4], [9]. For design of a new classifier we use or necessarily redefine some notions from the multifractals theory. We found that the correlation integral can be decomposed to a set of newly defined probability distribution mapping functions. The probability distribution mapping function maps the distribution of points in the neighborhood of fixed point with respect to distance from that point. Moreover the distribution-mapping function can be approximated by simple polynomial function of distance r in the same way as correlation integral ($C_I(r) = Cr^{\nu}$ with correlation dimension ν), i.e. in the form Cr^q, where q is a distribution-mapping exponent and C is a constant. We show that distribution-mapping exponent q is very close to the correlation dimension, and that the correlation dimension is equal to mean distribution-mapping exponent. We consider here that the "influence" of neighbor points of some class on the probability that the query point belongs to this class is inversely proportional to r^{ν}. Thus, weighting these influences we design a classification approach based on summing up all these influences for each class. At this point, the method reminds a kernel method with a rather strange kernel that has a singularity in its center and is not fulfilling condition to have a finite integral. The sums are corrected by class priors in cases of different numbers of points of different classes. We prove that a resulting formula gives an estimate of probability of class to which the query point belongs. It is an interesting difference to other classifiers where output variable is a measure of membership to a class, but not a probability. An important fact for the assertion to be valid is that exponent ν must be the correlation dimension.

A related problem is an effective method for correlation dimension estimation. Unlike other needs of correlation dimension estimation oriented to exactness of the estimate, we need a fast approach. For correlation dimension estimation we used the Grassberger-Procaccia approach [1] and Takens' estimator [7] together with an "averaging approach" proposed here that speeds up computation especially for large data sets. We found that when the correlation integral is decomposed to a set of probability distribution mapping functions in the form Cr^q the correlation dimension can be estimated by mean distribution mapping exponent q.

We tested the new classifier on various real-life multivariate data sets. Our results demonstrate that the polynomial projection with correlation dimension as an exponent can convert a complex multivariate data distribution into a more tractable form.

Our results show that the decomposition of correlation integral to local functions, which are approximated by simple polynomial, can be used for approximation of the probability of class at a given point and thus can be used for constructing a new type of classifier, which can, for some data, outperform some more complex classifiers.

This study can lead to a more detailed analysis of the relation between fractal dimension and probability density, and also for the development of new approaches to data analysis including classification problems.

Next Chap. 2 describes the data space transformation that forms the basis of the method proposed and describes the new classifier. The transformation is parameterized by correlation dimension as shown above. Therefore, Chap. 3 deals with this particular detail, i.e. correlation dimension estimation, and can be considered as a "step aside". It can be omitted in the first reading supposing that there are some ways in which the correlation dimension can be estimated. Description of some tests and discussion conclude the paper.

2 Probability of Class and Correlation Dimension

The main goal of this paper is to show that the approximation of probability of class at a given point can be expressed as a particular dependence on the correlation dimension. In this section, we proceed from the assumption that the best approximation of the probability distribution of the data is closely related to the uniformity of the space around the query point x. This uniformity is reached by the use of expanded distances, i.e. by the use of r^{ν} instead of distance r; ν is the correlation dimension. First, we point out the notion of correlation dimension and introduce the transformation mentioned.

CORRELATION DIMENSION

The correlation dimension was introduced in [1] as a characteristic measure of *strange attractors*, which allows distinguishing between deterministic chaos and random noise. The authors of [8] consider the set $\{X_i,\ i = 1, 2, .. N\}$ of points of the attractor. Most pairs (X_i, X_j) with $i \neq j$ are dynamically uncorrelated pairs of essentially random points [1]. The points lie, however, on the attractor. Therefore, they will be spatially correlated. This spatial correlation is measured by correlation integral $C_I(r)$, where r has the meaning of a distance, defined according to

$$C_I(r) = \lim_{N\to\infty} \frac{1}{N^2} \times \{number of\ \ pairs(i,j) : \left\| X_i - X_j \right\| < r\}. \tag{1}$$

In a more comprehensive form one can write

$$C_I(r) = \Pr(\left\| X_i - X_j \right\| < r). \tag{2}$$

In [1] it is shown that for small r the $C_I(r)$ grows like a power $C_I(r) \sim r^{\nu}$ and that "correlation exponent" ν can be taken as a most useful measure of the local structure of *strange attractor.* The authors also mention that correlation exponent (dimension) ν seems to be more relevant in this respect than the Hausdorff dimension [3] D_h of the attractor. In general, there is $\nu \le \sigma \le D_h$, where σ is the information dimension, and it can be found that these inequalities are rather tight in most cases, but not in all. Given an experimental signal and $\nu < n$ (degree of freedom or dimensionality or so-called embedding dimension), we can conclude that the

signal originates from deterministic chaos rather than random noise, since random noise will always result in $C_I(r) \sim r^n$.

The correlation integral (1) or (2) can be rewritten in the form [8]

$$C_I(r) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{1\le i<j\le N} h(r - \|X_j - X_i\|), \tag{3}$$

where $h(.)$ is Heaviside step function equal to one for positive argument and equal to zero otherwise. From it

$$\nu = \lim_{r\to\infty} \frac{\ln C_I(r)}{\ln r}. \tag{4}$$

DATA SPACE TRANSFORMATION

There are known facts and some simple considerations as follows.

- The correlation integral is a distribution function of distances between all pairs of data points.
- Grassberger and Procaccia [1] have shown that correlation integral grows like a power $C_I(r) \sim r^{\nu}$.
- When a new variable $z = r^{\nu}$ is introduced, the correlation integral is transformed to the distribution function of random variable z.
- This distribution function of random variable z grows linearly (for small values of z).
- Its derivative according to z is the distribution density function of random variable z. This distribution density function is constant (for small values of z).
- Then the probability distribution of random variable z is uniform (for small values of z).

Thus, a complex multivariate distribution of points in n-dimensional space is transformed to a uniform (for small values of z) distribution of a scalar variable. We use this fact when designing a new method of approximation of probability of class at a given point and in proof of Theorem 1. One could even say that in the following we "measure" the distance by r^{ν}.

When using a notion of distance, we, in fact, use a simple transformation from n-dimensional to one-dimensional space. By the use of any measure of distance (instead of all coordinates in n-dimensional space), the problems with dimensionality are eliminated at the loss of information on the true distribution of points in the neighborhood of the query point.

THE METHOD

Let the learning set U of total N samples (points, patterns) be given. Each sample $x_t=\{x_{t1}, x_{t2},\ldots\ x_{tn}\}$; $t = 1, 2, \ldots N$, $x_{tk} \in R$; $k = 1, 2, \ldots, n$ corresponds to a point in n-dimensional metric space M_n, where n is the sample space dimension. For each $x_t \in U$ a class function $T: R^n \to \{1, 2, \ldots C\}: T(x_t) = c$ is introduced; C is the number of classes. With the class function the learning set U is decomposed into disjoint classes $U_c = \{x_t \in U \mid T(x_t) = c\}$; $c \in \{1, 2, \ldots, C\}$, $\bigcup_{c=1}^{C} U_c$, $U_c \cap U_d = \emptyset$; $c, d \in \{1, 2, \ldots, C\}$; $c \neq d$. Let the cardinality of set U_c be N_c ; $\sum_{c=1}^{C} N_c = N$.

For the purpose of this paper we denote learning samples x_i, where i is the index of point without respect to class to which it belongs; x_i is the i-th nearest neighbor of point x. The distance of point x_i and query point x is r_i.

In the *k*-NN method the resulting estimation of probability that a query point belongs to a class is dependent on the number of points *k* inside the ball of radius r_k. It does not matter how the points inside the ball are distributed. Points can be concentrated in the center or spread along the surface of the ball, the result is the same.

To intuitively describe the method presented let us consider partial influences of individual points to the probability that point *x* is of class *c*. Suppose, for simplicity, the same priors for all *C* classes. Each point of class *c* in the neighborhood of point *x* adds a little to the probability that point *x* is of class *c*. This influence is the larger the closer the point considered is to point *x* and vice versa. With respect to the transformation introduced above it depends also on exponent equal to correlation dimension ν.

For the first (nearest) point $i = 1$ $$p_1(c \mid x,1) = \frac{1}{Sr_1^{\nu}}, \tag{5}$$

for the second point $i = 2$ $$p_1(c \mid x,2) = \frac{1}{Sr_2^{\nu}}, \tag{6}$$

and so on, generally for point No. i $$p_1(c \mid x,i) = \frac{1}{Sr_i^{\nu}}. \tag{7}$$

Here *S* is constant dependent on dimensionality *n* and metrics used.

Then, we add the partial influences $p_1(c \mid x,i)$ of individual points of class *c*, i.e. points of U_c, together by summing up

$$\hat{p}(c \mid x,k) = \sum_{x_i \in U_c}^{k} p_1(c \mid x,i) = \frac{1}{S} \sum_{x_i \in U_c}^{k} 1/r_i^{\nu}. \tag{8}$$

(The sum goes over indexes *i* for which the corresponding samples of the learning set are of class *c*.) It can be seen that any change of distance r_i of any point x_i of class *c* from point *x* will influence the probability that point *x* is of class *c*.

Let us compare formula (8) with the formula for the *k*-NN method $\hat{p}(c \mid x,kNN) = \frac{i_c}{Sr_k^n}$. Here i_c denotes the number of points of class *c* from *k* nearest points to point *x*. In practical computation there is usually

$$\hat{p}(c \mid x,kNN) = \frac{i_c}{k}. \tag{9}$$

In a similar way, we can rewrite (8) into a more suitable form for practical computation

$$\hat{p}(c \mid x) = \frac{\sum_{x_i \in U_c} 1/r_i^{\nu}}{\sum_{i=1}^{N} 1/r_i^{\nu}}. \tag{10}$$

(The upper sum goes over 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 number *k*.

The denominator $S = \sum_{i=1}^{N} \frac{1}{r_i^{\nu}}$ is, in fact, a sum of a series $\{1/r_i^{\nu}\}$. Terms $1/r_i^{\nu}$ of this series are reciprocals of distances between the query point and points of the learning set to the $-\nu$ power.

The numerator $S_c = \sum_{x_i \in U_c} \frac{1}{r_i^{\nu}}$ (eventually S_1, S_2) is the sum of series selected from $\{1/r_i^{\nu}\}$ so that it contains only terms corresponding to class *c*.

The approach described relies on the knowledge of the correlation dimension. This problem is discussed in Section 3.

APPROXIMATION OF PROBABILITY OF CLASS AT A GIVEN POINT

The approximation of the probability is often used in classification tasks [4], [7], [9], [10], [11]. The decision that a pattern is of a given class is based on finding a probability with which the pattern (sample, point or query point) belongs to a given class. The highest probability usually corresponds to the appropriate class.

Theorem 1

Let the task of classification into two classes be a given. Let the size of the learning set be N and let both classes have the same number of samples. Let ν, $1 < \nu < n$ be the correlation dimension, and let the correlation integral have the form of polynomial function $C_I(r, c) = k r^{\nu}$, where k is a constant. Let $r_i > 0$ be the distance of point x_i from point x. Then,

$$\lim_{N\to\infty} \frac{\sum_{x_i \in U_c} 1/r_i^{\nu}}{\sum_{i=1}^{N} 1/r_i^{\nu}} = p(c \mid x) \quad . \tag{11}$$

Proof:

Let us consider one class.. Let us use a new variable $z = r^{\nu}$. Then, $C_I(z, c) = kz$ is a linear function. By the use of $z = r^{\nu}$, the space is mapped ("distorted") so that the correlation integral, in fact the distribution function of distances between all pairs points of class *c* of the learning set, is linear as a function of variable *z*. Thus, the corresponding distribution density function $d(z, c)$ is constant (as a function of z) for any particular distribution of points of class *c* of the learning set.

Let us consider a query point *x*. Let the distance of a point *i* of class *c* of the learning set be r_i. Let us consider sum $\sum_{x_i \in U_c} d(r_i^{\nu}, c)/r_i^{\nu}$. For this sum we have

$$\lim_{N\to\infty} \sum_{i=1}^{N} d(r_i^{\nu}, c)/r_i^{\nu} = \lim_{N\to\infty} \sum_{x_i \in U_c} p(c \mid x)/r_i^{\nu} = p(c \mid x) \lim_{N\to\infty} \sum_{x_i \in U_c} 1/r_i^{\nu} \tag{12}$$

because there is a constant $d(z, c) = p(c \mid x)$ for all *i* (uniform distribution has a constant density).

Given the learning set, we have the space around point *x* "sampled" by individual points of the learning set. Let $p_c(r_i)$ be an a-posteriori probability that point *i* in 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 the point is of the other class. Then, the particular realization of $p(c \mid x)\sum_{i=1}^{N} 1/r_i^{\nu}$ is sum $\sum_{x_i \in U_c} 1/r_i^{\nu}$ (the sum here goes over indexes of class c only). Using this sum, we can rewrite the right-hand side of (12) into the form

$$p(c \mid x) \lim_{N\to\infty} \sum_{i=1}^{N} 1/r_i^{\nu} = \lim_{N\to\infty} \sum_{x_i \in U_c} 1/r_i^{\nu} . \tag{13}$$

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

$$\frac{\lim_{N\to\infty} \sum_{x_i \in U_c} 1/r_i^{\nu}}{\lim_{N\to\infty} \sum_{i=1}^{N} 1/r_i^{\nu}} = p(c \mid x) \tag{14}$$

and due to the same limit transition in the numerator and in the denominator we can rewrite it in the form (11).

Note that the convergence of $S = \sum_{i=1}^{N} \frac{1}{r_i^{\nu}}$ and $S_c = \sum_{x_i \in U_c} \frac{1}{r_i^{\nu}}$ is the faster the larger correlation dimension ν is. Usually for multivariate real-life data correlation, dimension is large too; in any case larger than one. Theorem 1 states that probability of the class is proportional to $1/r_i^{\nu}$ and formula (3) uses the sum of these ratios assuming to attain a reasonable number for class probability estimation. So it is supposed that for a number of samples going to infinity, the sum would be convergent. Clearly, let distances r_i be reordered so that $r_i > r_{i+1}$, i=1, 2, …; then ratio $r^{\nu}_i / r^{\nu}_{i+1} < 1$ for any $\nu > 0$ and according to the d'Alembert criterion the series is convergent.

The question arises about the speed of diminishing the tail of the series. It can be found that condition that the distribution of random variable $1/r^{\nu}$ has the mean may suffice, as shown in the theorem below.

Theorem 2

Let $P(r)$ be the probability distribution function of neighbor distances and let there exist a mapping of probability density of points x_{ci} of class c in E_n, $E_n \to E_1$: $p(x_{ci}) = p(r_{ci}^{\nu})$ so that $\int_{r_{c1}}^{\infty} \frac{1}{r^{\nu}} dP(r) < \infty$. Then, for $\nu \geq 2$ $S_c = \sum_{i=1}^{N_c} \frac{1}{r_{ci}^{\nu}}$ converges for $N_c \to \infty$ as fast as $N_c^{-\nu/2}$.

For proof we use theory of U-statistics. Citing [19] let X_1, X_2, ... be independent observations on a distribution F. Consider a parametric function $\theta = \theta(F)$ for which there is an unbiased estimator. Let there be a function $h = h(x_1, \ldots x_m)$, called a "kernel". For any kernel h, the corresponding U-statistics for estimation of θ on the basis of sample $X_1, \ldots X_n$ of size $n \geq m$ is obtained by averaging the kernel h symmetrically over the observations:

$$U_n = U(X_1, \ldots X_n) = \frac{1}{\binom{n}{m}} \sum_{c} (X_{i_1}, \ldots X_{i_m})$$

For example, $\theta(F)$ = mean of $F = \int x dF(x)$ and $h(x) = x$ the corresponding U-statistics is $U_n = U(X_1,...X_n) = \frac{1}{n}\sum_{i=1}^{n} X_i = \overline{X}$, the sample mean. Let $E(.)$ means mean, and $E_F(.)$ means mean over specified distribution F. It holds

Lemma. Let r be a real number ≥ 2. Suppose that $E_F|h|^r < \infty$. Then,

$$E|U_n - \theta|^r = O(n^{-(r/2)r}), \quad n \to \infty.$$

Proof of Theorem 2.

Comparing Theorem 2 with Lemma, it is easily seen that $X_i = 1/r_i$, $i = 1, 2, ...,$ U-statistics is the S_c, and condition $E_F(h)^r < \infty$ holds according to assumption. Then, $S_c(N_c)$ converges for $N_c \to \infty$ as fast as $N_c^{-\nu/2}$.

Figs. 1 and 2 illustrate the convergence of sum S_c above for a particular query point for well-known "vote" data [12]. The task is to find whether a president elected will be Republican or Democrat. Data is 15-dimensional of two classes, Republican and Democrat, and classes have a different number of samples. In the learning set there are a Republican 116 times and a Democrat 184 times. Value 11.46 is the estimate of correlation dimension here.

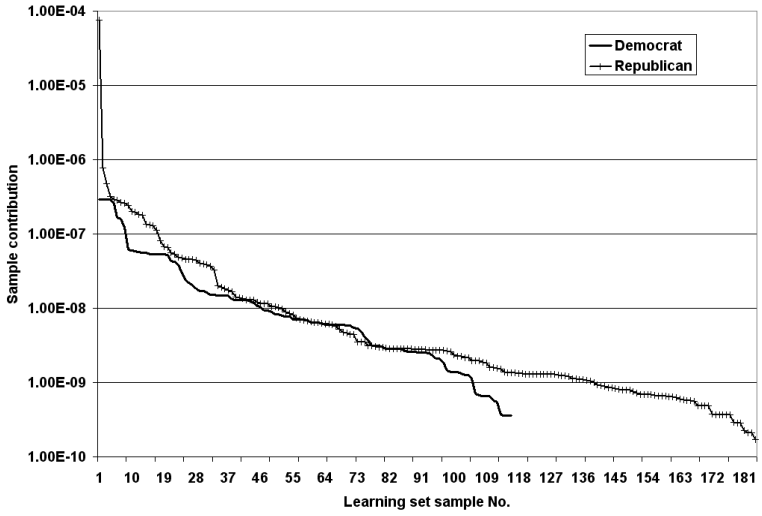

Fig. 1. Sample contribution to sum S_c for 15-dimensional data "vote" and one particular query point; correlation dimension estimate $\nu = 11.46$. The upper line corresponds to Republican, the lower line to Democrat. Samples are sorted according to distance r, i.e. also the size of sample contribution to the sum S_c. There are different numbers of samples of one and the other class in the learning set.

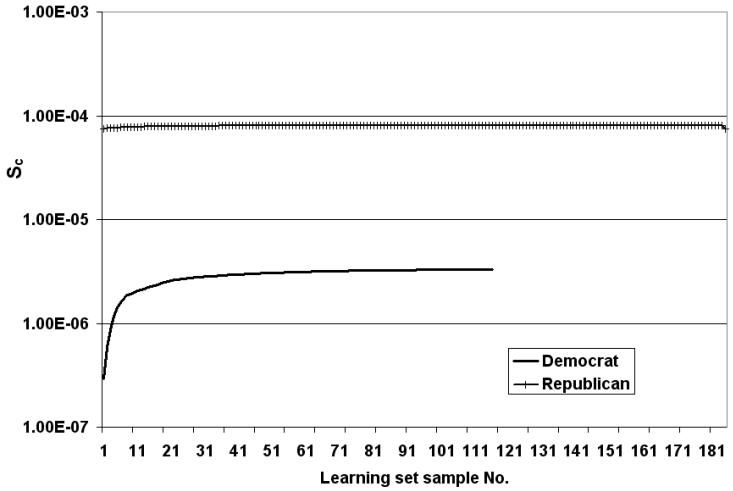

Fig. 2. Size of the total sum S_c for 15-dimensional data "vote" and one particular query point; correlation dimension estimate $q = 11.46$. The upper line corresponds to Republican, the lower line to Democrat. Samples are sorted according to distance r, i.e. also the size of sample contribution to the sum S_c

CLASSIFIER CONSTRUCTION

In this section we show how to construct a classifier that incorporates the idea of correlation dimension (including the approaches mentioned). First, we compute the correlation dimension v by the method discussed in the section dealing with correlation dimension estimation. Then, we simply sum up all components $1/r_i^v$. This is made for both classes separately getting numbers S_1 and S_2 for both classes. Then, we can get the Bayes ratio or a probability approximation that the point $x \in R_n$ (n-dimensional space of real numbers) belongs to class 1 from equations

$$R(x) = \frac{S_1}{S_2} \text{ or } p_1(x) = \frac{S_1}{S_1 + S_2} . \tag{15}$$

Then, for a threshold (cut) θ chosen, if $R(x) > \theta$ or $p_1(x) > \theta$, then x belongs to class 1, else to class 2.

Note that we have found in practice the influence of the first nearest neighbor usually more negative than positive. Therefore, the first nearest neighbor is excluded from practical computation. As above, we simply sum up all components $1/r_i^v$ excluding the nearest point without respect to its class.

Generalization

For a different number of samples of one and the other class formula (11) has the form

$$p(c \mid x) = \lim_{N \to \infty} \frac{\frac{1}{N_c} \sum_{x_i \in U_c} 1/r_i^v}{\frac{1}{N_1} \sum_{x_i \in U_1} 1/r^v + \frac{1}{N_2} \sum_{x_i \in U_2}^{N} 1/r_i^v} . \tag{16}$$

It is only a recalculation of the relative representation of different numbers of samples of one and the other class [10].

For more than two classes, say C classes, the equation is

$$p(c \mid x) = \lim_{N \to \infty} \frac{\frac{1}{N_c} \sum_{x_i \in U_c} 1/r_i^v}{\sum_{k=1}^{C} \frac{1}{N_k} \sum_{x_i \in U_k} 1/r_i^v} \,. \tag{17}$$

3 Correlation Dimension Estimation

For the approximation of probability of class at a given point and classification described above, a fast and reliable method for correlation dimension estimation is needed. Methods for the estimation of correlation dimension v try somehow estimating a limit (4). Methods differ by approaches used and also by some kind of heuristics that usually optimize the size of radius r to get a realistic estimation of correlation dimension [2], [13].

THE LINEAR REGRESSION

The oldest approach is based on the estimation of the slope of correlation integral in log-log coordinates [1]. First, a proper part of the correlation integral is selected, e.g. the leftmost half of the correlation integral graph. Then, standard "one dimensional" regression is used for the slope, i.e. correlation dimension, computation. The error of this method grows with dimensionality and lessens with the size of the learning set. The method proposed by Camastra and Vinciarelli [8] compensates for the influence of the limited size of the learning set at the cost of extensive computation.

The complexity of this approach follows from necessity

- To compute $N(N\text{-}1)/2$ distances, each representing n multiplications, n-1 additions. Square root is not necessary as one can work with distances squared.
- Sort $N(N\text{-}1)/2$ items
- To compute standard "two-dimensional" linear regression with $\eta N(N\text{-}1)/2$ points, in fact shortest distances. η is a fraction (typically ½ or 1/3) of shortest distances used.

Thus, the total complexity in the number of multiplications is $nN(N\text{-}1)/2 + (\eta N(N\text{-}1)/2)^3$ that is $O(N^6)$ for large N.

TAKENS' ESTIMATOR

One of the most cited estimators of the correlation dimension is Takens' estimator [7], [13]. It can be written in the form

$$\nu_T(r) = N_p \left(N_p \log r_{Np} - \sum_{p=1}^{N_p} \log r_p \right)^{-1} , \tag{18}$$

where N_p is the number of pairs considered, r_p are distances between randomly chosen points which are smaller than r, and r_{Np} is the largest of all r_p. As in the previous case, it means that we use some proper part of all pairs that have the shortest distances, and then we apply the formula above.

It was shown by Takens [7] that his estimation is unbiased and error converges to zero with $1/\sqrt{N_p}$. In our tests we have found that results are quite good.

The complexity of this approach follows from necessity for each class

- to compute $N(N\text{-}1)/2$ distances, each representing n multiplications, n-1 additions.
- sort $N(N\text{-}1)/2$ items; the number of (smallest) pairs considered $N_p = \eta N(N\text{-}1)/2$.
- to compute and sum up N_p times the $\log r_p$.

Thus, the total complexity in the number of multiplications is $nN(N\text{-}1)/2 + N_p(N_p \log(N_p)) = nN(N\text{-}1)/2 + (\eta N(N\text{-}1)/2)^2 \log(\eta N(N\text{-}1)/2)$ that is $O(N^4 \log(N))$ for large N.

AVERAGING APPROACH TO CORRELATION DIMENSION ESTIMATION

The basic problem of correlation dimension estimation is the large number of pairs that arise even for a moderate learning set size, as seen from the complexity considerations above. There is the obvious fact that the correlation integral is the probability distribution of distances of all pairs of points of the learning set. The idea of the correlation dimension estimation described below is based on the observation that distances between all pairs of points can be divided into groups, each group associated with one (fixed) point of the learning set. It appears that these distances between pairs of points are, in fact, distances of neighbors of that fixed point. We call the corresponding distribution function a probability distribution mapping function. We consider this function as a kind of map of probability distribution in the neighborhood of a fixed point, and it was introduced e.g. in [14], see definitions below. A core notion of a distribution mapping exponent in this mapping 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 [1] than to the box-counting or other fractal or multifractal dimension definitions [20].

Definition 1

The probability distribution mapping function $D(x, r)$ of the neighborhood of the query point x is the function $D(x,r) = \int_{B(x,r)} p(z)dz$, where r is the distance from the query point and $B(x, r)$ is a ball with center x and radius r.

Note: It can be seen that for a fixed x, the function $D(x, r)$, $r > 0$ grows monotonically from zero to one. Function $D(x, r)$ for a fixed x is one-dimensional analog to the probability distribution function.

One can write the probability distribution mapping function in the form

$$D(x,r) = \lim_{N\to\infty} \frac{1}{N-1} \sum_{j=1}^{N-1} h(r - r_j), \tag{19}$$

where $h(.)$ is the Heaviside step function.

For a finite number of points, we have the empirical probability distribution mapping function

$$\hat{D}(x,r) = \frac{1}{N-1} \sum_{j=1}^{N-1} h(r - r_j). \tag{20}$$

We show, in this section, that the correlation integral is the mean of the distribution mapping functions and that the correlation dimension ν can be approximated by the mean of the distribution mapping exponents q, as shown in the theorem below:

Theorem 3

Let there be a learning set of N points (samples). Let the correlation integral be $C_I(r)$ and let $D(x_i, r)$ be the distribution mapping function corresponding to point x_i. Then, $C_I(r)$ is a mean of $D(x_i, r)$:

$$C_I(r) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} D(x_i, r). \tag{21}$$

Proof

Let $h(x)$ be a Heaviside step function and l_{ik} be the distance of k-th neighbor from point x_i. Then, the correlation integral is

$$C_I(r) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N-1} h(r - l_{ij}) \tag{22}$$

and also

$$C_I(r) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{N-1} \sum_{j=1}^{N-1} h(r - l_{ij}) \right) . \tag{23}$$

Comparing (23) with (19), we get (21) directly.

The probability distribution mapping function can be – in analogy to correlation integral – approximated by a simple polynomial as follows.

Definition 2

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 (DME).

With respect to (4) and (21) the correlation dimension can be approximated by the mean of distribution mapping exponents q_i:

$$\nu = \frac{1}{N} \sum_{i=1}^{N} q_i \ . \tag{24}$$

Thus, the correlation dimension is, in fact, an average of all distribution mapping exponents computed for all points of the data set. When all points of the data set are used, the number of distances between pairs of points is the same as in the Grassberger-Procaccia algorithm [1] for assessing the correlation dimension. We have found that for sufficiently good estimation of the correlation dimension we can use part of the data set only, for each point to estimate the distribution mapping exponent, and take the average. The part of the data set may be some number of points randomly selected from the data set.

Now a problem arises how many points are necessary for an appropriate assessment of the correlation dimension. The distribution mapping exponent varies from point to point. Suppose a relative variation $\rho = \sigma / \nu$, where ν is a mean, i.e. the correlation dimension.

The central limit theorem states that, under fairly common conditions, the sum of a large number of random variables will have an approximately normal distribution. Then, suppose that $X_1 = q_1 - \nu$, …, $X_n = q_n - \nu$ be independent and identically distributed random variables, all with the same arbitrary distribution, with zero mean and variance σ^2; and that Z is their mean scaled by $\sqrt{n}$, that is,

$$Z = \sqrt{n} \left(\frac{1}{n} \sum_{i=1}^{n} X_i \right).$$

Then, as n increases, the probability distribution of Z will tend to the normal distribution with zero mean and variance σ^2. From it follows that variable $z = Z / \sqrt{n}$ has variance $\sigma_z^2 = \sigma^2 / n$, and random variables q_i have the relative standard deviation $\sigma_q / \nu = (\sigma / \nu) / \sqrt{n}$. Now

supposing relative standard deviation of the distribution mapping exponent to be $\rho = \sigma/\nu = 0.1$ and after $n = 100$ trials we have mean within ±1 % around value of the correlation dimension estimated by Grassberger-Procaccia's algorithm with probability 68 % or within ±3 % around this value with probability 99.7 %. The value $\rho = \sigma/\nu = 0.1$ will be discussed at the end of the third paragraph of the next Chapter. The method of averaging need not be limited to the Grassberger-Procaccia algorithm. We use it analogically for Takens' algorithm as well.

4 Performance Analysis

COMPLEXITY ESTIMATION

Learning

Learning represents approximation of the correlation dimension. When learning k_s samples are selected from a learning set and for each of them the distribution mapping exponent is computed. For each such computation the learning set is searched once, distances are computed, and then sorted and the slope, i.e. the distribution mapping exponent, is computed. Thus, there are nN multiplications, $N\ln N$ exchange operations, computations of logarithms and solving the regression equation. Supposing multiplication as the most frequent and the most time-consuming operation the computational complexity of learning is roughly proportional to $k_s nN$.

The value of k_s must be set up in advance. We have found $k_s = 100$ sufficient. One can change it to any value up to N. In the latter case, the computational complexity of learning is proportional to nN^2.

Thus, the computational complexity is much lesser than computational complexity of linear regression and Takens' approaches to correlation dimension computation especially for k_s small, as discussed above. Sensitivity of classification error to error in correlation dimension estimation is rather low, as discussed below.

Recall – class estimation

Computation for one sample given consists of computing according to the formula (15) and its variants (16) and (17). In the end, it is a sum of N elements. Each element is a reciprocal of the ν-th power of distance, and computation of the distance takes n multiplications. On the whole, the complexity of one sample recall is proportional to nN, i.e. to the size of the learning set.

SENSITIVITY OF CLASSIFICATION ERROR TO ERROR IN CD ESTIMATION

For error sensitivity to the value of correlation dimension no particular threshold θ was used. Instead, we use a more general classification quality measure here, the size of area under the ROC (Receiver operating characteristics [15]) curve (the AUC) of dependence of "sensitivity", i.e. the acceptance of class 1 samples (often called signal) on "specificity", i.e. on the suppression of class 0 samples (often called background, i.e. background error). It holds that the larger the area under the curve (AUC, classification efficiency) the better classification in a general sense. The ideal case is unit area, i.e. ROC curve going through point (0, 1), which means 100 % sensitivity (signal efficiency) and 100 % specificity, i.e. zero background error.

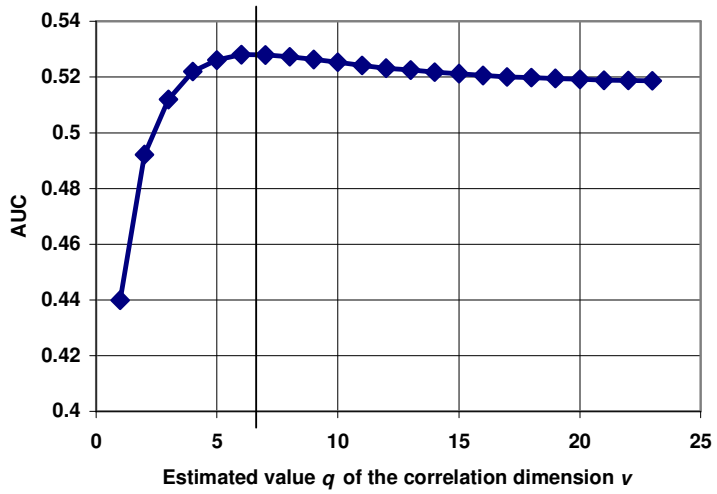

Fig. 3. General classification efficiency AUC as the function of the estimated value q of correlation dimension ν for data "Higgs". Data dimensionality is 23, note that 6.5 is value of the correlation dimension, i.e. the optimal value of q.

In Fig. 3 the classification efficiency as the function of the value q of estimated correlation dimension ν for data "Higgs" [16] is shown. These data have estimated correlation dimension $\nu = 6.5$. For this value of q, the minimal error is 0.472, the value of error is 0.481 for $q = n = 23$. This is rather a small difference, only 1.93 %, showing that error in correlation dimension estimate need not be critical.

THE CORRELATION DIMENSION AND SPREAD OF THE DME

In Table 1 and Fig. 4 features of six different data sets and corresponding distribution-mapping exponents are summarized. Data sets originate from the UCI Machine Learning Repository [12]. Note that mean distribution-mapping exponent is, in fact, the correlation dimension. It can be seen that

- Mean DME (in fact, an estimate of correlation dimension) is much smaller than dimension for all data varying from a little more than 6.2 % (data Ionosphere) to nearly 49.5 % (data RKB).
- DME of a data set lies in a rather narrow band; normalized mean squared variation, sigma/mean, σ/μ varies from 7.357 % to less than 19 %.
- Note that lines for Heart, German, and Higgs data look suspiciously similar but these data come from very different independent sources.

Table 1.

Parameters of DME distribution for different data sets and color notation for Fig. 4. Data sets are from UCI MLR [12].

Data	Higgs	German	Heart	Adult	RKB	Ionosphere
Color	Red	Aquamarine	Violet	Green	Blue	Coral
Entries	6508	1000	270	15037	6341	151

Dimension	23	20	13	14	10	33
Max DME	2.17589	3.2249	2.75272	7.66802	6.54623	2.40971
min DME	0.782672	2.0181	1.79976	2.88812	1.68963	1.47318
Mean DME	1.750632	2.713477	2.416463	5.27807	4.944528	2.056234
sigma DME	0.171495	0.213795	0.177784	0.835889	0.915318	0.285874
sigma/Mean	**0.09796**	**0.07879**	**0.07357**	**0.1584**	**0.1851**	**0.1390**
Mean ratio of DME to dimension	0.07611	0.1357	0.1859	0.3770	0.4945	0.06231

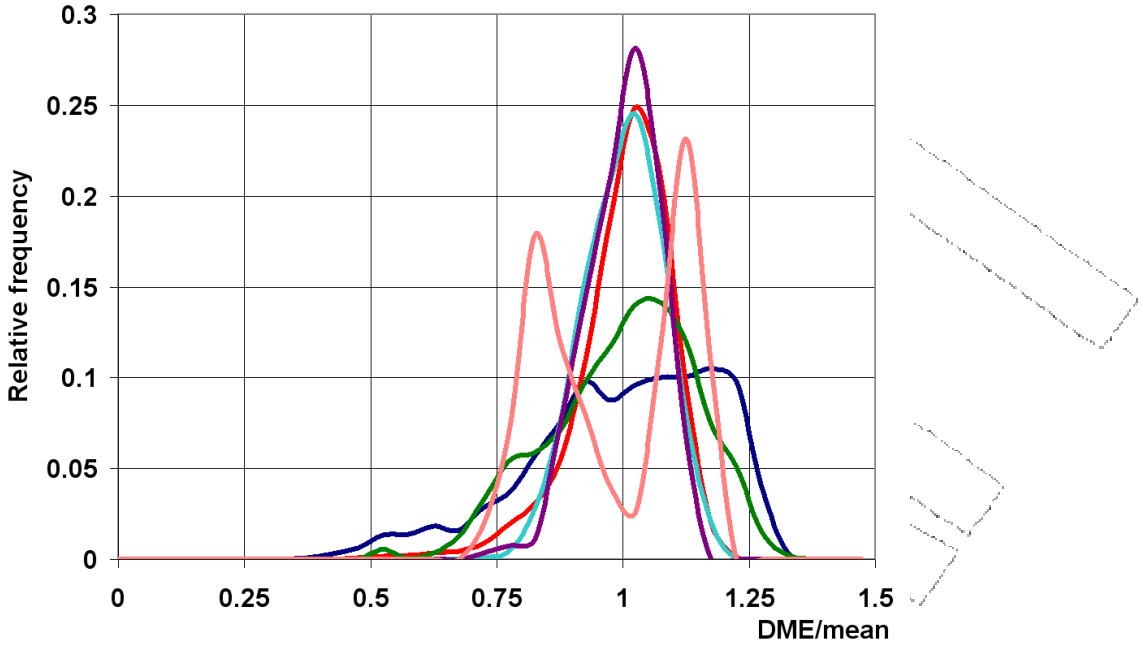

Fig. 4. Histograms of distribution mapping exponent for six different data sets. Histograms are normalized by mean value of the DME and have a unit area. For legend see Table 1.

Here we cannot conclude that data are generally multifractals as the scaling exponent, DME varies from point to point of the set. These variations usually lie in a rather narrow band and thus mean DME, i.e. the correlation dimension, may suffice for characterization of the fractal nature of data.

In Fig. 4 it is seen that relative standard deviation of the DME does not exceed 25 %, and typical value can be estimated as 15 %. From analysis of the averaging method of correlation dimension estimation then follow estimates for given numbers of random trials, as stated in Table 2.

Table 2. Error of CD estimation by averaging method as function of relative standard deviation σ/ν of DME and reliability level given by 1 to 3 sigma.

No. of trials	10			100			1000		
σ/ν	1σ	2σ	3σ	1σ	2σ	3σ	1σ	2σ	3σ
Probable margins 15%	4.74%	9.49%	14.23%	1.50%	3.00%	4.50%	0.47%	0.95%	1.42%
Largest margins 25%	7.91%	15.81%	23.72%	2.50%	5.00%	7.50%	0.79%	1.58%	2.37%

TESTS WITH SYNTHETIC DATA

Synthetic data are two dimensional and consist of three two dimensional normal distributions with identical a-priori probabilities. If μ denotes vector of 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: $\mu = (0, -1)^t$, $\boldsymbol{C}_m = (1, -0.5; -0.5, 1)$.

Fig. 5 shows results obtained by different methods for different learning sets of sizes from 8 to 256 samples and a testing set of 5000 samples all from the same distributions and mutually independent. Each point was obtained by averaging over 100 different runs. For 1-NN method with L_2 (Euclidean) metrics and variants of the LWM method by Paredes and Vidal [11] in Fig. 5 the values were adopted from the literature cited.

In Fig. 5 it is seen that the use of the method presented here outperforms all other methods shown and for large number of samples approaches fast to the Bayes limit.

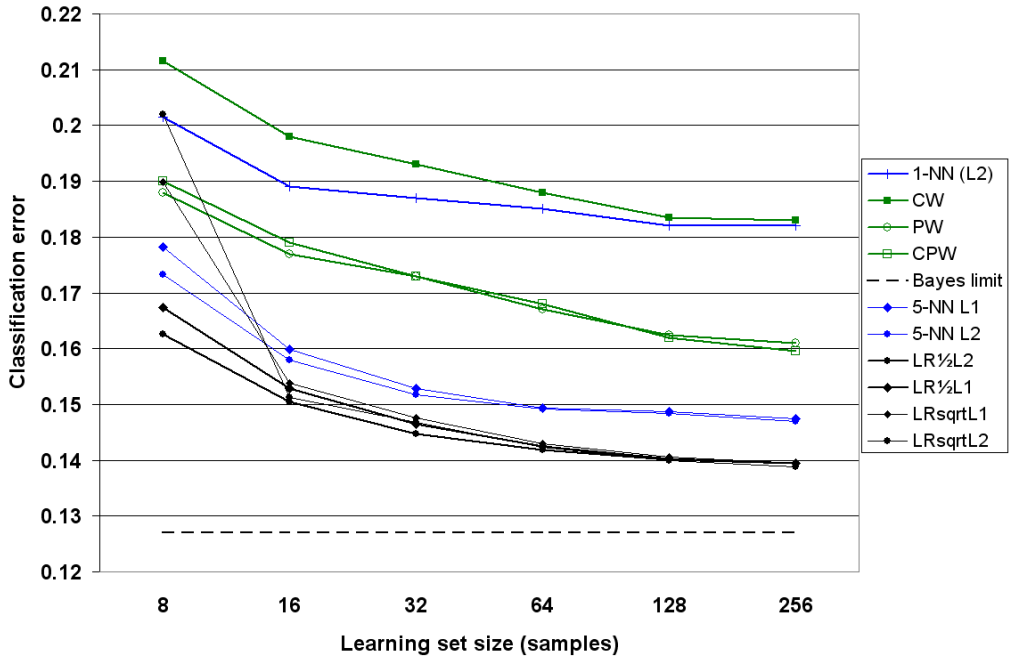

Fig. 5. Comparison of classification errors of synthetic data for different approaches. On horizontal axis there is learning set size, on vertical axis classification error. In legend 1-NN (L2) means 1-NN method with Euclidean metrics, CW, PW, and CPW are three methods by Paredes and Vidal [11]. "Bayes" means the Bayes limit. 5-NN means k-NN method with five nearest neighbors. Black lines mean the method presented here. LR means standard linear regression, ½ means the use of the first half of the samples; sqrt means that the square root of the number of samples is used for correlation dimension estimation. L1 and L2 denote Manhattan and Euclidean metrics used.

Note that L_1 (Manhattan) or L_2 (Euclidean) metrics does not give significantly different results. Also selection of a part of DMF – in fact the part of nearest neighbors from all possible neighbors – used for correlation dimension estimation (one half and of the square

root of number of samples) does not result in a significant difference for 16 and more samples of the learning set.

TESTS OF CLASSIFICATION ACCURACY WITH REAL-LIFE DATA FROM UCI MLR

Experiments described below follow procedures described by Paredes and Vidal [11] as truly thorough tests. Paredes and Vidal prepared a corpus of data sets suitable for use with any classifier. The data sets are available on the Internet [17] and originate from the Machine Learning Repository, see [12]. We used all the data sets of this corpus. Each task consists of 50 pairs of training and testing sets corresponding to 50-fold cross validation.

In Tables 3 and 4 classification accuracy is given for different tasks and different classifiers. Table 3 corresponds to eight variants of the method presented here. In the headings of these eight double-columns LR means standard linear regression, TA means Takens' estimator. ½ means the use of the first half of the samples; sqrt means that the square root of the number of samples is used for correlation dimension estimation. L1 and L2 denotes metrics used – Manhattan or Euclidean. The columns with heading σ show the standard deviation of the error estimate at the left column.

In Table 4 the first five double columns give mean errors and standard deviations for the 1-NN method, the *k*-NN method with *k* equal to the square root of the number of samples of the learning set, the Bayes method with ten bins histograms, perceptron neural network implemented in Statistica-12 system, and SVM according to Joachims [21], [22], respectively. The heading SVM best means the best result obtained with one of four kernels, linear, polynomial, Gaussian, and RBF. The last four columns in Table 4 correspond to four variants (CDM, CW, PW, and CPW) of the Learning Weighted Metrics (LWM) and show results by [11], [18]. Here data for some tasks and standard deviations are not available (N/A).

Table 3. Classification accuracy of eight variants of the method described here for different tasks. Results can be compared with results of other classifiers shown in Table 3. For explanation of columns headings see text.

Dataset	LR ½ L2	σ	LR ½ L1	σ	LRsqrtL2	σ	LRsqrtL1	σ	TA½L2	σ	TA½L1	σ	TAsqrtL2	σ	TAsqrtL1	σ
Australian	18.23%	0.0266	16.79%	0.0285	14.20%	0.0286	12.54%	0.0286	22.66%	0.0395	16.57%	0.0280	18.90%	0.0298	18.25%	0.0329
Balance	25.17%	0.0700	26.82%	0.0689	24.99%	0.0717	27.27%	0.0717	26.05%	0.0701	26.93%	0.0716	26.22%	0.0697	26.88%	0.0719
Cancer	3.85%	0.0172	4.53%	0.0194	3.98%	0.0198	4.78%	0.0198	3.90%	0.0181	4.63%	0.0205	3.73%	0.0143	4.20%	0.0473
Diabetes	25.40%	0.0332	25.63%	0.0350	24.73%	0.0341	24.45%	0.0341	25.83%	0.0327	25.77%	0.0324	26.08%	0.0340	26.02%	0.0332
DNA	32.12%	0.0467	25.63%	0.0436	30.69%	0.0441	26.98%	0.0441	32.29%	0.0468	25.55%	0.0436	27.15%	0.0445	26.90%	0.0444
German	29.71%	0.0287	31.16%	0.0226	27.90%	0.0242	28.16%	0.0242	29.66%	0.0269	30.96%	0.0241	30.73%	0.0277	31.64%	0.0249
Glass	32.50%	0.0869	29.96%	0.0811	35.24%	0.0649	32.18%	0.0649	28.59%	0.0359	25.88%	0.0628	31.80%	0.0320	30.33%	0.0386
Heart	19.56%	0.0517	20.63%	0.0551	17.96%	0.0526	18.70%	0.0526	19.48%	0.0529	20.52%	0.0553	20.67%	0.0487	21.11%	0.0543
Ionosphere	18.06%	0.0399	13.49%	0.0446	16.21%	0.0431	15.44%	0.0431	18.17%	0.0403	13.52%	0.0458	16.84%	0.0371	12.33%	0.0397
Iris	5.91%	0.0694	7.91%	0.1031	5.91%	0.1057	7.91%	0.1057	5.91%	0.0839	6.91%	0.1057	5.91%	0.0839	6.91%	0.1057
Led17	2.59%	0.0083	5.77%	0.0119	0.32%	0.0128	0.42%	0.0128	2.43%	0.0099	5.70%	0.0121	4.45%	0.0120	6.91%	0.0136
Letter	5.03%	0.0218	4.80%	0.0214	10.18%	0.0331	11.03%	0.0331	4.98%	0.0217	4.75%	0.0213	11.78%	0.0322	12.40%	0.0329
Liver	39.68%	0.0697	38.55%	0.0727	40.17%	0.0609	39.80%	0.0609	39.83%	0.0595	38.26%	0.0645	39.68%	0.0593	38.87%	0.0579
Monkey1	6.74%	0.0733	6.34%	0.0694	10.39%	0.0847	10.85%	0.0847	6.79%	0.0726	6.29%	0.0698	6.30%	0.0675	7.76%	0.0963
Phoneme	13.24%	0.0141	12.92%	0.0141	19.19%	0.0189	18.71%	0.0189	13.15%	0.0143	12.86%	0.0141	15.76%	0.0146	14.86%	0.0178
Satimage	9.80%	0.0297	9.35%	0.0291	13.90%	0.0290	13.45%	0.0290	9.95%	0.0299	9.60%	0.0295	9.35%	0.0291	9.30%	0.0290
Segmen	5.14%	0.0138	4.23%	0.0101	8.17%	0.0137	6.48%	0.0137	27.61%	0.0239	48.21%	0.0197	27.58%	0.0240	27.64%	0.0249
Sonar	24.41%	0.0820	23.70%	0.0908	27.43%	0.1232	24.63%	0.1232	42.28%	0.0423	37.67%	0.1210	46.58%	0.0063	45.85%	0.0183
Vehicle	28.54%	0.0274	29.02%	0.0297	31.49%	0.0353	30.53%	0.0353	28.89%	0.0274	28.96%	0.0319	47.17%	0.0540	32.99%	0.0366
Vote	9.19%	0.0336	7.69%	0.0287	10.11%	0.1490	9.65%	0.1490	9.12%	0.0332	7.72%	0.0295	17.15%	0.1699	17.16%	0.1572
Vowel	5.12%	0.0202	4.18%	0.0176	13.79%	0.0173	12.81%	0.0173	4.81%	0.0189	3.98%	0.0175	10.76%	0.0273	6.59%	0.0194
Waveform21	15.93%	0.0120	15.69%	0.0125	15.29%	0.0122	15.32%	0.0122	15.31%	0.0116	15.21%	0.0122	17.85%	0.0127	17.42%	0.0122
Waveform40	16.48%	0.0095	15.85%	0.0085	16.83%	0.0082	16.15%	0.0082	16.61%	0.0092	16.00%	0.0087	18.67%	0.0104	17.18%	0.0088
Wine	4.99%	0.0312	4.03%	0.0280	6.39%	0.0259	5.09%	0.0259	5.10%	0.0320	3.59%	0.0266	6.89%	0.0324	3.71%	0.0244

Table 4. Results of other classifiers. For explanation of columns headings see text. N/A in some entries denotes that corresponding data is not available from the reference [11]. Note also, that standard deviation σ is not available for CDM, CW, PW, and CPW classifiers.

Dataset	1-NN	σ	Sqrt-NN	Σ	Bayes	σ	Neur Net	σ	SVMbest	σ	CDM	CW	PW	CPW
Australian	20.73%	0.0297	15.50%	0.0232	13.88%	0.0249	14.88%	0.0288	35.99%	0.0804	18.19%	17.37%	16.95%	16.83%
Balance	23.61%	0.0545	32.06%	0.0861	15.17%	0.0398	5.65%	0.0340	33.17%	0.1768	35.15%	17.98%	13.44%	17.60%
Cancer	5.07%	0.0161	3.25%	0.0110	2.68%	0.0121	3.30%	0.0125	16.34%	0.1634	8.76%	3.69%	3.32%	3.53%
Diabetes	29.48%	0.0302	26.46%	0.0336	24.19%	0.0315	23.95%	0.0402	29.64%	0.0646	32.47%	30.23%	27.39%	27.33%
DNA	25.72%	0.0437	34.06%	0.0474	6.66%	0.0249	5.73%	0.0233	N/A	N/A	15.00%	4.72%	6.49%	4.21%
German	32.76%	0.0268	30.90%	0.0318	24.97%	0.0289	25.37%	0.0297	27.25%	0.0405	32.15%	27.99%	28.32%	27.29%
Glass	32.72%	0.0811	42.10%	0.0980	47.37%	0.0651	39.94%	0.0761	32.63%	0.0920	32.90%	28.52%	26.28%	27.48%
Heart	25.11%	0.0540	16.89%	0.0496	17.44%	0.0519	19.12%	0.0587	37.22%	0.0581	22.55%	22.34%	18.94%	19.82%
Ionosphere	14.05%	0.0385	14.70%	0.0382	9.26%	0.0353	10.99%	0.0356	18.52%	0.1655	N/A	N/A	N/A	N/A
Iris	5.91%	0.0962	7.91%	0.0787	9.82%	0.0923	8.00%	0.0919	6.55%	0.1437	N/A	N/A	N/A	N/A
Led17	11.50%	0.0158	0.12%	0.0015	0.00%	0.0000	0.18%	0.0030	11.52%	0.1001	N/A	N/A	N/A	N/A
Letter	4.80%	0.0214	18.70%	0.0390	28.98%	0.0454	25.88%	0.0438	2.68%	0.0161	6.30%	3.15%	4.60%	4.20%
Liver	39.59%	0.0597	41.48%	0.0595	39.42%	0.0601	30.91%	0.0534	35.54%	0.0697	39.32%	40.22%	36.22%	36.95%
Monkey1	2.01%	0.0385	9.27%	0.0878	28.01%	0.1090	0.57%	0.0103	2.94%	0.0548	N/A	N/A	N/A	N/A
Phoneme	11.83%	0.0132	20.71%	0.0173	21.47%	0.0218	16.84%	0.0223	14.39%	0.0199	N/A	N/A	N/A	N/A
Satimage	10.65%	0.0308	15.20%	0.0359	19.15%	0.0393	14.75%	0.0354	24.30%	0.0429	14.70%	11.70%	8.80%	9.05%
Segmen	3.81%	0.0123	11.41%	0.0375	9.85%	0.0258	5.46%	0.0155	46.48%	0.3389	N/A	N/A	N/A	N/A
Sonar	18.37%	0.0695	32.51%	0.0756	31.46%	0.0916	27.98%	0.0865	19.67%	0.0593	N/A	N/A	N/A	N/A
Vehicle	30.51%	0.0263	31.51%	0.0264	38.40%	0.0301	19.76%	0.0304	28.23%	0.1631	32.11%	29.38%	29.31%	28.09%
Vote	8.74%	0.0269	9.60%	0.0334	9.70%	0.0347	6.05%	0.0264	22.64%	0.1777	6.97%	6.61%	5.51%	5.26%
Vowel	1.19%	0.0107	46.68%	0.0425	26.64%	0.0517	26.94%	0.0506	13.64%	0.0976	1.67%	1.36%	1.68%	1.24%
Waveform21	23.73%	0.0125	14.71%	0.0113	19.26%	0.0086	15.54%	0.0116	26.94%	0.1517	N/A	N/A	N/A	N/A
Waveform40	28.22%	0.0147	16.24%	0.0098	20.31%	0.0092	15.93%	0.0098	32.25%	0.2068	N/A	N/A	N/A	N/A
Wine	5.42%	0.0290	6.15%	0.0413	4.50%	0.0308	5.12%	0.0373	27.77%	0.0805	2.60%	1.44%	1.35%	1.24%

5 Discussion

The main goal of this paper was to show that the correlation dimension of the approximation of probability of class at a given point could be expressed as a particular dependence on correlation dimension. We used the assumption that the best approximation of the probability distribution of the data is closely related to the uniformity of the space around the query point x. This uniformity is reached by the use of expanded distances, i.e. by the use of r^{ν} instead of distance r ; ν is the correlation dimension.

The other distance-based or kernel-based approaches have to tune weights of distances – if possible – or to tune parameters of kernels used to get optimal results. Based on our theory, the classifier proposed needs no tuning because we have found that it is a correlation dimension as a suitable exponent in polynomial transformation of distances. In most of classifiers the output variable corresponding to a class is a measure of the membership of the query point to the class. In our case, the output variable that expresses a class is an estimate of probability of the membership of the query point to the class.

Designing a classifier, we consider partial influences of individual points to the probability that point x is of class c. We state here that the "influence" of neighbor points of some class on the probability that the query point belongs to this class is inversely proportional to r^{ν}, ν is the correlation dimension. Thus, weighting these influences, we design a classification approach based on summing up all these influences for each class. For example, in the case of two classes we get two sums, S_0, S_1. Ratio $S_0/(S_0 + S_1)$ is an estimate of probability that the query point belongs to class 0. The sums are corrected (multiplied) by class priors in cases of different numbers of points of different classes in the learning set as it is common in most of classifiers, and follows from Bayes theorem. At the point of summing up influences, the method reminds of a kernel method with rather strange kernel that has a singularity in its center and not fulfilling condition to have finite integral.

There are important findings. We have found that correlation dimension plays an essential role as an exponent in polynomial data space projection that finally allows handling with one-dimensional uniform distribution. This projection may be useful for solving different problems. We have shown here an application for approximation of probability of class at a given point and for the construction of a new classifier. The classifier has no true learning phase. In the "learning phase" an estimate of the correlation dimension is computed. When it is assumed that the correlation dimension is constant, the learning set may change dynamically or may be enlarged or updated without necessity relearn the classifier.

The crucial point of the idea of polynomial transformation of distances is the correlation dimension. Thus, the estimate of the correlation dimension is an essential part of the method. There is lot of papers dealing with correlation dimension estimation. We have shown in Chap. 4 that result, i.e. that classification quality is not too sensitive to this estimate. In the case of a small learning set the estimation of the correlation dimension by Grassberger-Procaccia or by Takens' approach are sufficiently fast. The complexity of these approaches grows quadratically with learning set size, and for large learning sets they are rather time-consuming. An approximate but fast averaging approach to correlation dimension estimation can be used with success in this case.

The averaging approach is based on finding that the correlation integral is a mean of distribution mapping functions, as proved in Theorem 3. Supported by this theorem and finding that the distribution mapping exponent has rather narrow spread, as shown in Fig. 4, we assume that also the correlation dimension is a mean of distribution mapping exponents for all points of the learning set. Using all points of the learning set, it is, in fact, the Grassberger-Procaccia method. To speed up computation we propose to use only 100 random points to state 100 distribution mapping exponents and use the mean as an estimate of the correlation dimension. The number 100 follows from observation (see Fig. 4) that ratio DME

to mean DME has standard deviation approx. 0.15 (max 0.25), and thus standard deviation of estimate of mean DME that approximates the correlation dimension ν is 0.15ν (0.25ν). Using 100 observations, the standard deviation of mean estimate lessens to 0.015ν (0.025ν). In practice, user may consider this standard deviation too large and use a larger number of points to diminish it. Comparing numbers according to Table 2 with Fig. 3, it can be seen that 100 trials suffice not to degrade the classification accuracy; even 10 trials may suffice in many cases.

The classifier presented here was tested with 24 data sets from the Machine Learning repository and it was shown in Tables 3 and 4 that the classifier outperforms all other methods in four cases from the 24 data sets mentioned. Note that there are four other classifiers (1-NN, Bayes, CPW, and NeurNet) that outperform others in four cases. The Sqrt-NN outperforms others in two cases, and PW and SVM in one case of all 24 data sets used for testing. The classification errors for the best and the second best classifier for a task differ usually a little; we found one exception – for task Balance the NeurNet has error 5.65 %, whereas PW 13.44 % and all others between 14 % and 35 %. At the right part of Table 3 it is seen that the use of one half of samples is generally a little better than the use of the square root of the number of samples. At the same time, L_1 metrics appears slightly better than L_2 metrics especially when Takens' estimator is used. On the other hand, due to small differences in most of the cases one need not see any special advantage of L_1 metrics over L_2 metrics.

As to accuracy of stating classification errors given in Table 3 and in Table 4, the error estimates are ratio of the number of badly recognized testing samples to total number of testing samples. Where possible, the standard deviation is given next right at the error estimate. It can be seen that the standard deviations of error estimates depend, to larger extent, on the task solved, and less on the type of classifier or on the corresponding value of error estimate. Testing the classifier on practical data we found that the influence of the first nearest neighbor is usually more negative than positive. It means that the classifier has a tendency to overestimate the class probability of the query point to the advantage of class of the nearest neighbor. It is also motivated by the fact that polynomial transformation used transforms general distribution of points around the query point to one-dimensional uniform distribution of variable $z = r^{\nu}$. This variable expresses the distance of the k-th neighbor. In one-dimensional uniform distribution it holds that the distribution of the k-th neighbor has Erlang distribution Erl(λ, k). For the first neighbor it is the exponential distribution that has relative standard deviation σ/μ equal to one, whereas for larger k it is equal to $1/\sqrt{k}$ and diminishes with k. Cases where r_1 is relatively very small making the "weight" of the first neighbor too large are rather frequent, and then the first nearest neighbor is excluded from practical computation, as mentioned in Chap. Classifier Construction.

The core of this paper is transformation $z = r^{\nu}$, i.e. transformation of distance r to a variable that is parameterized by exponent ν, the correlation dimension. The classifier proposed and averaging method for correlation dimension estimation demonstrates a practical power of this transformation. By this simple "expansion" of distance a distribution of points around a fixed point is transformed into uniform distribution that is easy to deal with. Here it was used for designing a classifier. The same transformation may also be used for study of other problems, e.g. complex problem of distribution function of neighbor's distances in point processes.

Acknowledgment

The work was supported by the Ministry of Education of the Czech Rep. under the INGO project No. LG 12020 and by the Czech Technical University in Prague RVO: 68407700.

References

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

[2] Mo, D., Huang, S.H.: Fractal-Based Intrinsic Dimension Estimation and Its Application in Dimensionality Reduction. IEEE Trans on Knowledge and Data Engineering. Vol. 24 no. 1, pp. 59-71 (2012).

[3] Mandelbrot, B.: The fractal geometry of nature. W.H. Freeman and Company, New York, 1982.

[4] Boyu Li, Yun Wen Chen, Yan Qiu Chen: The Nearest Neighbor Algorithm of Local Probability Centers. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 38, No. 1, Feb. 2008, pp. 141 - 154, (2008).

[5] Sang-Woon Kim, Oommen, B.J.: On Using Prototype Reduction Schemes to Optimize Kernel-Based Fisher Discriminant Analysis. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 38, No. 2, April 2008, pp. 564 - 570, (2008)

[6] Ghosh, A.K.: Kernel Discriminant Analysis Using Case-Specific Smoothing Parameters. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 38, No. 5, Oct. 2008, pp. 1413 - 1418, (2008)

[7] Takens, F.: On the Numerical Determination of the Dimension of the Attractor. in: Dynamical Systems and Bifurcations, in: Lecture Notes in Mathematics, Vol. 1125, Springer, Berlin, 1985, p. 99-106.

[8] Camastra, F., Vinciarelli, A.: Intrinsic Dimension Estimation of Data: An Approach based on Grassberger-Procaccia's Algorithm. Neural Processing Letters, Vol. 14 (2001), No. 1, pp. 27-34.

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

[10] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, Second Edition, John Wiley and Sons, Inc., (New York, 2000).

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

[12] Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science (2010).

[13] Camastra, F.: Data dimensionality estimation methods: a survey. Pattern Recognition, Vol. 6 (2003), pp. 2945-2954.

[14] Jirina, M.: Local Estimate of Distribution Mapping Exponent for Classification of Multivariate Data. Proceedings of EIS2004: Fourth International ICSC Symposium on Engineering of Intelligent Systems. February 29 - March 2, 2004, Island of Madeira, Portugal.

[15] Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers. Technical report, Palo Alto, USA: HP Laboratories, 38 pp. (2004).

[16] Hakl, F., Jirina, M., Richter-Was, E.: Hadronic tau's identification using artificial neural network. ATLAS Physics Communication, ATL-COM-PHYS-2005-044, last revision: 26 August 2005, 12 pp.

[17] Paredes. R.: Data sets corpora. [online] http://algoval.essex.ac.uk/data/vector/UCI/ (2008)

[18] Paredes, R.: CPW: Class and Prototype Weights Learning [online] http:// www.dsic.upv.es/~rparedes/research/CPW/index.html (2009).

[19] Serfling, R.J.: Approximation Theorems of Mathematical Statistics. John Wiley and Sons, New York, 1980. (Especially p. 185, Lemma A.)

[20] Lowen, S.B., Teich, M.C.: Fractal-Based Point Processes. Wiley, 2005.

[21] 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.

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