Correlation Dimension-Based Classifier

Jiřina, Marcel
2006

cAk

CENTRUM APLIKOVANÉ KYBERNETIKY

České vysoké učení technické v Praze - fakulta elektrotechnická

Correlation Dimension-Based Classifier

Technical report

Marcel Jiřina and Marcel Jiřina, jr.

www@c-a-k.cz

2006

Institute of Computer Science

Academy of Sciences of the Czech Republic

Correlation Dimension-Based Classifier

Marcel Jiřina and Marcel Jiřina, jr.

Technical Report No. V-945

January 2006

Abstract

An innovative approach that utilizes the correlation dimension (CD) both for probability density estimate of data and consequently for classification is suggested. The idea of correlation dimension classifier directly follows the principle of classifier, which uses the distribution mapping exponent (DME). The basic difference between these two approaches is that DME is a local feature while CD is global feature. A new "hyperbolic" approximation of the correlation integral for the correlation dimension estimation is suggested. It is shown that CD-based classifier outperforms DME classifier and many other classifiers published on Machine Learning Repository pages.

Keywords:

Multivariate data; Correlation dimension; Nearest neighbors; Distribution mapping exponent; Power approximation, Classification, ML Repository.

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: +420 266 051 111, fax: +420 286 585 789

e-mail: ics@cs.cas.cz

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 Biomedical Engineering, Czech Technical University in Prague, Zikova 4,
166 36, Prague 6, Czech Republic

jirina@fbmi.cvut.cz

1 Introduction

The correlation dimension [4], [7] is a feature of the fractal or the data generating process and thus more accurately expresses the nature of fractals or processes which generate data we wish to separate. Grassberger and Procaccia introduced the basic idea in [7]. There are many papers that utilize the correlation dimension estimation in different tasks, [2], [4], [6]. Also application of the correlation dimension for classification has been published, e.g. in [15].

The correlation dimension is impossible to calculate in an analytically closed form and therefore there are many sophisticated methods that estimate the correlation dimension. The basic approach is, of course, given in Grassberger-Procaccia paper [7]. In [11] a variant of this method is presented and it is also mentioned an estimation by L.A. Smith (1988) that for the estimation of fractal – correlation dimension v to be accurate within 5 %, the cardinality of the data set should be 42^v . Grassberger and Procaccia have also shown that correlation dimension estimated by the use of their algorithm grows systematically with number of random points of ten-dimensional data set and approaches to 10 for very large data set. One of most cited is Taken's estimator [2], [14]. Another estimation of correlation dimension is given in [6], where the estimation should compensate the edge (boundary) effects biasing the estimation of correlation dimension.

In this paper we innovatively deal with direct application of the correlation dimension for probability density estimate and consecutively for classification. The idea of correlation dimension (CD) classifier directly follows the principle of classifier, which uses the distribution mapping exponent (DME) [8], [9], [10]. The basic difference is that DME is a local feature depending on the position of the query point and on the number of points of the learning set while CD is not. Here we suggest a correlation dimension-based classifier that utilizes a new heuristic procedure for the correlation dimension estimation. It will be shown that for some data sets CD-based classifier outperforms DME classifier and many other classifiers.

2 Probability distribution mapping function

Here two important notions, the *probability distribution mapping function* and the *distribution density mapping function* are introduced [8], [9]. To understand these terms we give a brief example that demonstrates them.

Let us have a ball in an n-dimensional space containing points over its volume. Let us divide the ball on concentric "peels". Using the formula $r_i = S(n)/2^n.\sqrt[n]{V_i}$ we obtain a quite interesting succession of radii corresponding to the individual volumes. The symbol $S(n)$ denotes the volume of a ball with unit radius in E_n; note $S(3) = {}^4/_3\pi$. The V_i denotes volume corresponding to radius r_i. The higher space dimension is considered the more similar values of the radii approaching the outer ball radius are obtained. The inner part of the ball is thus nearly empty; this is the "Theater Effect" [10].

A mapping between the mean density in an i-th peel ρ_i and its radius r_i is a probability distribution mapping function $D(x, r)$ and its partial derivation according to r the distribution density mapping function $d(x, r)$ [8], [9]. Functions $D(x, r)$ and $d(x, r)$ for x fixed are one-dimensional analogs to the probability distribution function and the probability density function, respectively.

A need of the distribution that is uniform in the vicinity of the query point for the best probability density estimation is formulated in [8], [9]. To achieve it a parabolic function in the form $D(x, r) = \text{const.}r^q$ that both reduces dimensionality from E_n to E_1 and makes the picture of distribution more uniform was introduced. It is called a power approximation of the probability distribution mapping function $D(x, r)$. This approximating function is tangent to the horizontal axis in the origin and let it be going through some characteristic points of the distribution. The exponent q is a distribution-mapping exponent.

Using this approximation of the probability distribution mapping function $D(x, r)$ we, in fact, linearize this function in the neighborhood of the query point x. The distribution density mapping function $d(x, r)$ as function of variable (r^q) is approximately constant in vicinity of the query point. This constant includes true distribution of probability density of points as well as influence of boundary effect.

Important finding is that the distribution-mapping exponent reminds the Grassberger-Procaccia's correlation dimension [7]. There are three essential differences. First, the distribution-mapping exponent is a local feature of the data set because it depends on a position of the query point, whereas the correlation dimension is a feature of the whole data space. Second, the distribution mapping exponent is related to data only. Third, the distribution mapping exponent is influenced by boundary effect.

LINEAR REGRESSION PROCEDURE FOR DME ESTIMATION

A procedure for the distribution mapping exponent determination for a two-class classifier was suggested in [8], [9]. Let U be a learning set composed of points (patterns, samples) $x_{cs,}$ where $c = \{0, 1\}$ is the class and $s = 1, 2, \ldots, N_c$ is the point index within class c. N_c is the number of points in the class c. Let point $x \notin U$ be given and let points x_{cs} of one class be sorted so that index $i = 1$ corresponds to the nearest neighbor, index $i = 2$ to the second nearest neighbor, etc. In the Euclidean metrics, $r_i = \|x - x_{ci}\|$ is the distance of the i-th nearest

neighbor of class c from point x. From the definition of the distribution mapping exponent it follows that r_i^q should be proportional to index i, i.e.

$$r_i^q = ki \text{ , } i = 1, 2, ..., N_c, \; c = 0 \text{ or } 1, \tag{1}$$

where k is a suitable constant. Using a logarithm we get

$$q \ln(r_i) = k' + \ln(i) \text{ , } i = 1, 2, ..., N_c. \tag{2}$$

On one hand we exaggerate distances nonlinearly to make small distance differences appear much larger for the purposes of density estimation. On the other hand, in (2) there is a logarithm of distance, which decreases large influences of small noise perturbations on the final value of q. It can be also found that small change of q does not essentially influences classification results.

The system of N_c equations (2) with respect to unknown q can be solved using standard linear regression for both classes. Thus we get two values of q, q_0 and q_1. To get a single value of q we use the weighted arithmetic mean, $q = (q_0 N_0 + q_1 N_1)/ (N_0 + N_1)$.

3 Correlation dimension

The correlation dimension was introduced in [7] as a characteristic measure of *strange attractors*, which allows distinguishing between deterministic chaos and random noise. The authors of [11] consider the set $\{X_i, i = 1, 2, .. N\}$ of points of the attractor, obtained e.g. from time series with fixed time increment. Most pairs (X_i, X_j) with $i \neq j$ are dynamically uncorrelated pairs of essentially random points [7]. The points lie however on the attractor. Therefore they will be spatially correlated. This spatial correlation is measured by correlation integral $C(r)$ defined according to

$$C(r) = \lim_{N \to \infty} \frac{1}{N^2} \times \{number of \;\; pairs(i, j) : \left\| X_i - X_j \right\| < r\}. \tag{3}$$

In more comprehensive form one can write

$$C(r) = \Pr(\left\| X_i - X_j \right\| < r) \text{ .} \tag{4}$$

In [7] it is shown that for small r the $C(r)$ grows like a power $C(r) \sim r^{\nu}$ 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 Hausdorff dimension D_h of the attractor. In general there is $\nu \leq \sigma \leq D_h$, where σ is the information dimension, and it can be found that this inequalities are rather tight in most cases, but not all. Given an experimental signal and $\nu < n$ (n is degree of freedom or dimensionality or so-called embedding dimension) then we can conclude that the signal originates from deterministic chaos rather than random noise, since random noise will always result in $C(r) \sim r^n$.

The correlation integral (3) or (4) can be rewritten in form [11]

$$C(r) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{1\le i<j\le N} h(r - \|X_j - X_i\|), \tag{5}$$

where $h(.)$ is Heaviside step function. From it

$$\nu = \lim_{r\to\infty} \frac{\ln C(r)}{\ln r}.$$

There are methods for estimation of correlation dimension ν, but the problem is that they are either too specialized for one kind of equation or they use some kind of heuristics that usually optimize the size of radius r to get the correlation dimension.

CORRELATION DIMENSION ESTIMATION

In [11] a method for establishing the *intrinsic dimension* – in fact the correlation dimension - of data is given. The method in its first step assumes that *intrinsic dimension* of data representing points, which are uniformly distributed in an n-dimensional cube, is equal to dimensionality n. This is not true due to the boundary effect, as it has been shown in [10]. Thus the method has systematic error, which grows with dimensionality.

There are other approaches to correlation dimension estimation. One of most cited is Taken's estimator [2], [14].

$$\nu_T(r) = (N_p - 1) \Big/ \left(N_p \log r_p - \sum_{p=1}^{N_p} \log r_p\right),$$

where N_p is the number of pairs considered, r_p are distances between randomly chosen points which are smaller than r. In our tests we found that results are underestimated and it was reported, that it might behave poorly [14].

Another estimation of correlation dimension is given in [6]. The estimation is given by

$$\nu_{DK} = \frac{d \log C_N(r)}{d \log[r(2-r)]}$$

and authors mention that it should compensate the edge (boundary) effects biasing the estimation of correlation dimension.

THE DISTRIBUTION MAPPING EXPONENT AND THE CORRELATION DIMENSION

The difference between DME and CD is obvious: Grassberger-Procaccia algorithm uses distances of all pairs of points of the data set. For the distribution mapping exponent estimation distances from the query point to the first, the second, ... neighbor of one particular point x are used. At the same time in the Grassberger-Procaccia algorithm it is supposed that the number of data points is large enough to describe features of the attractor or of the other data generating process adequately. Data at hand are considered in distribution mapping exponent estimation only and thus the influence of boundaries can effect the estimation.

The basic idea of this section is to use the distribution mapping estimate for the correlation dimension estimate. Particularly, we propose to use the mean value of the distribution mapping exponent as an estimate of the correlation dimension. Moreover, to get closer correlation dimension estimate we propose the "hyperbolic approximation". This approximation uses a heuristics based on observation, that the graph of probability vs. distance in log-log coordinates often looks like approaching asymptotically to some limit.

Let us use Camastra-Viciarelli's [11] form of correlation integral (3) or probability form (4). In any case for finite set of data with cardinality m_T we get an approximation $C_{m_T}(r)$ of the correlation integral as

$$C_{m_T}(r) = \frac{1}{m_T(m_T - 1)} \sum_{1 \le i < j \le m_T} h(r - \|X_j - X_i\|) \cdot \tag{6}$$

Plotting this function in logarithmic scales we usually get a curve with rather straight left hand part. The steepness of this left part, gives an estimation of the correlation dimension for given set of data, see Fig. 1.

Remind definition of the probability distribution mapping function $D(x, r)$ [8], [9]. In the neighborhood of the query point x there is $D(x,r) = \int_{B(x,r)} p(z)dz$, where r is distance from the query point and $B(x, r)$ is a ball with center x and radius r. It can be easily seen that this is nearly the same definition as (4). The essential difference is in the fact that the correlation integral is related to the attractor as a whole, and the probability distribution mapping function is related to particular point, the query point x.

For finite data set with cardinality m_T we get $D(x, r)$ in form (x is one point of the data set, its index let be m_T)

$$D(x,r) = \frac{1}{m_T - 1} \sum_{1 \le i \le m_T - 1} h(r - \|x - X_i\|) \tag{7}$$

and it is seen that in (7) selected points of (6) are used.

From it also follows that $C_{m_T}(r)$ is an average of distribution mapping functions for all m_T points of the data set Ω. Thus the approximation of the correlation integral follows directly from (6) and (7)

$$C_{m_T}(r) = \frac{1}{m_T} \sum_{x \in \Omega} D(x,r). \tag{8}$$

Averaging method for correlation dimension estimation

A significant result of this paper that we show in this section is that the correlation integral is the mean of the distribution mapping functions and that the correlation dimension can be approximated by the mean of distribution mapping exponents.

Theorem

Let there be a learning set of m_T points (samples). Let empirical correlation integral, i.e. empirical probability distribution of pair-wise distances l_{ij} of points from the learning set, be $C(l_{ij})$ and let $D(i, r_{ik})$, where r_{ik} is the distance of k-th neighbor from point i, be the empirical distribution mapping function corresponding to point i. Then $C(l_{ij})$ is a mean value of $D(i, r_{ik})$:

$$C(l_{ij}) = \frac{1}{m_T} \sum_{i=1}^{m_T} D(i, r_{ik}) \tag{9}$$

Proof

The correlation integral can be written in form

$$C(l_{ij}) = \frac{1}{m_T (m_T - 1)} \sum_{i=1}^{m_T} \sum_{i=1}^{m_T - 1} D(i, l_{ij}) . \tag{10}$$

Let for each i $r_{ik} = l_{ij}$. The r_{ik}, k=1, 2,...m_T-1 can be ordered so that $r_{i1} <= r_{i2} <= ... <= r_{i(m\mathrm{T}-1)}$. Thus r_{ik} is the distance from point i to its k-th nearest neighbor. Let $h(x)$ be Heaviside step function. Then

$$D(i, d) = \frac{1}{m_T} \sum_{k=1}^{m_T} h(d - r_{ik})$$

$$D(i, d) = \frac{1}{m_T - 1} \sum_{k=1}^{m_T - 1} h(d - l_{ik}) . \tag{11}$$

Comparing (10) and (11) we get directly (9). □

It is clear that the correlation integral $C(d) = \lim_{m_T \to \infty} C(l_{ij})$ and the correlation dimension ν can be approximated as a mean of distribution mapping exponents q_i:

$$\nu = \frac{1}{m_T} \sum_{k=1}^{m_T} q_i .$$

HEURISTICS FOR CD AND DME COMPUTATION

Here we discuss some heuristics, which can be used for estimation of the correlation dimension and for distribution mapping exponent estimation. First, we briefly mention the well-known square root rule and introduce our own hyperbolic approximation to the correlation dimension estimation.

Correlation dimension as an average DME

It was shown that the correlation dimension is, in fact, an average value of all distribution mapping exponents computed for all points of the data set, see (8). When all points of the data

set are used, the number of pair-wise distances is the same as in the Grassberger-Procaccia algorithm of building the correlation integral. One can see that for good estimation of the correlation dimension not all points are necessary. 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 randomly selected e.g. the square root of the number of points or any other part of the data set. We use 100 randomly selected points.

The square root rule

One very often used heuristics is to use the square root of the number of all points, i.e. all data points. For example, it is a good rule in nearest-neighbors based methods. This is also rather good rule for estimating distribution mapping exponent q. For correlation dimension one should take shortest distances and their number should be the square root of all pairs. From it the number of pairs used is $\sqrt{\frac{m_T(m_T-1)}{2}} \approx \frac{m_T}{\sqrt{2}}$. Even if this rule is used, the value of correlation dimension as well as the distribution mapping exponent is usually underestimated when the linear regression is used.

The hyperbolic approximation

In the log-log graph of the correlation integral or of the distribution mapping function it can be seen that it looks like the curve approaches to some asymptote for small distances., see Fig. 1. Let us suppose that such an asymptote exists and by its direction the correlation dimension or DME are given.

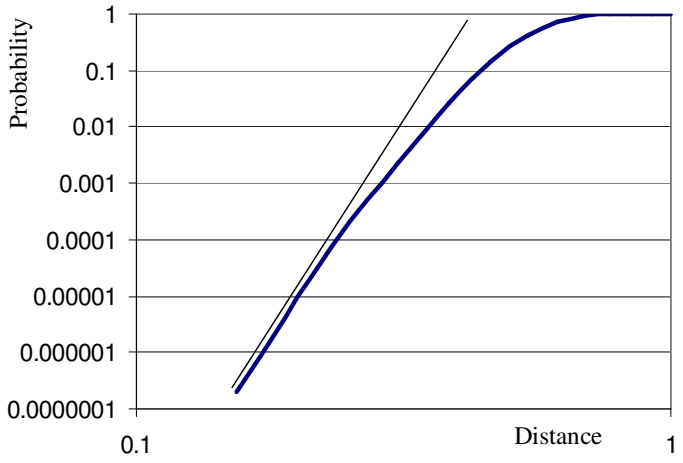

Fig. 1. Asymptote to the DMF or correlation integral.

Let the asymptote be given by $\ln p = k + q \ln r$. Each point on the curve in Fig. 1 has the coordinates $\ln p_i$ and $\ln r_i$. For the same value of $\ln p_i$ there is a point on the asymptote with

coordinate $\ln r_{0i}$. Then $\ln p_i = k + q \ln r_{0i}$. From it $\Delta \ln r_i = \ln r_i - \ln r_{0i}$. We choose a hyperbolic dependence $\Delta \ln r_i = \frac{a}{-\ln p_i}$, where a is a constant. From it we get

$$\ln r_i - \frac{1}{q} \ln p_i + q \ln r_{0i} = -\frac{a}{\ln p_i}.$$

This is a linear equation of three unknown variables, namely k/q, $1/q$, and a. The regression equation is

$$\begin{bmatrix} \sum 1 & \sum -\ln p_i & \sum 1/\ln p_i \\ \sum -\ln p_i & \sum (\ln p_i)^2 & \sum -1 \\ \sum 1/\ln p_i & \sum -1 & \sum (1/\ln p_i)^2 \end{bmatrix} \cdot \begin{pmatrix} k/q \\ 1/q \\ a \end{pmatrix} = \begin{pmatrix} \sum -\ln r_i \\ \sum \ln p_i \ln r_i \\ \sum -\ln r_i / \ln p_i \end{pmatrix}.$$

(All sums go from $i = 1$ to some number equal to or less than m_T according to another heuristics used; we use the square root of m_T.) We are interested in $1/q$ only and it is possible to get not too complex formulas for computation. This approach gives usually larger value of q than direct linear regression. Using the averaging method we get estimation of the correlation dimension ν.

4 Probability Density Estimation

The main goal of this paper is to find a classifier that would exhibit better features then other ones. The better estimation of the probability distribution of the data the better classifier. In this section we come from assumption that the best estimation of the probability distribution of the data is closely related to 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^q instead of r.

UNIFORMITY ASSUMPTION

It is known [1] that for larger dimensions something like local approximation of real distribution by uniform distribution does not exist. On the other hand it has been assumed that the best case for the distribution density estimation is a case of uniform distribution [3], [8], [9], [5], [12]. In such cases – at least in the neighborhood of the query point - the best results are usually obtained. This is a basis for all nearest neighbor-based methods [3]. Therefore the true distribution was approximated (polynomially expanded) so that at least in the neighborhood of the query point the distribution density mapping function is constant.

It was also shown [8], [9] that the probability distribution mapping function $D(x, r)$ can be approximated by simple polynomial as a function in the form const.r^q. The proper setting of the exponent q can lead to a sufficiently good transformation of the real distribution density mapping function to the function which is constant in the neighborhood of the query point x.

PROBABILITY DENSITY ESTIMATION

Let us consider the uniform distribution first. Let there be a ball with center at point x and radius r_i sufficiently large to contain just i points nearest to point x. The volume of the ball is

V_i = const.r_i^n in E_n. For each ball with index i and having just i points inside it, the probability density estimate can be given by formula (K is a constant)

$$p(x,i) = K\frac{i}{V_i} \ .$$

For probability density estimation at the point x we take average values of i/V_i for several i´s. Let us use i = 2, 3, ..., k, excluding, in fact, the influence of the nearest neighbor as its influence is most unreliable. Having in mind nonequidistant (nonequivolumous) sizes of individual balls of volumes V_i, it seems more appropriate to use the true distance of point i instead of some "weight" expressed by numerator i in each fraction i/V_i. Thus if K_1 and K_2 are constants independent of class then

$$\overline{p}_c(x) = \frac{K_1}{k-1}\sum_{i=2}^{k}\frac{r_i}{V_i} = \frac{K_2}{k-1}\sum_{i=2}^{k}1/r_i^{n-1} \ . \qquad (12)$$

Under the assumption that series $1/r_i^{n-1}$ converges with the size of r_i for $n > 1$ we have no reason to limit ourselves to nearest k points and we can use all points in the learning set using k equal to the learning set size. Note that only the distances from the query point x play role here and the n-th power of these distances r^n have the uniform distribution.

General case. Using the correlation dimension ν we, in fact, use true dimensionality of the data space so that variable r^ν has the uniform distribution (at least in the vicinity of the query point x). In the same way as in (12) we estimate the distribution density in point x by

$$\overline{p}_c(x) = \frac{K_3}{k-1}\sum_{i=2}^{k}1/r_i^{\nu-1} \ ,$$

where K_3 is a constant. Again, if the sum of series $1/r_i^{\nu-1}$ converges with the size of r_i, we can use all points in the learning set using $k = N_c$, $c = \{0, 1\}$.

CLASSIFIER CONSTRUCTION

In this section we show how to construct a classifier that incorporates the idea of correlation dimension (including approaches mentioned). First, we compute the correlation dimension ν as a mean of distribution mapping exponents q_i of 100 randomly selected points of the learning set. Individual q_i are computed using square root rule and hyperbolic approximation. Then we simply sum up all components $1/r_i^{\nu-1}$ excluding the nearest point because its influence is most unreliable. This is made for both classes simultaneously getting numbers S_0 and S_1 for both classes. Then we can get the Bayes ratio or a probability estimation that the point $x \in E_n$ belongs to class 1 from equations

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

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

5 Results - testing classification ability

The algorithm for classification to two classes based on the correlation dimension was written in C++ . The classification ability of this program was tested using four real-life tasks from UCI Machine Learning Repository [13]. Four databases, namely "Adult", "German", "Heart", and "Ionosphere" have been used for the classification task into two classes.

We do not describe these tasks in detail here as all can be found in [13]. For each task the same approach to testing and evaluation was used as described in [13]. In Table 1 results are shown together with results for other methods as given in [13]. For each task methods are sorted according to the classification error, the method with the best behavior – the smallest error – first.

Table 1. Comparison of the classification error of the program (CD) which implements the method described here for different tasks with results of other classifiers as given in [13].

"German"		**"Heart"**		**"Adult"**		**"Ionosphere"**	
Algorithm	Error	Algorithm	Error	Algorithm	Error	Algorithm	Error
CD	0.07	**CD**	0.16	FSS Naive Bayes	0.1405	IB3	0.0330
Discrim	0.535	Bayes	0.374	NBTree	0.1410	backprop	0.0400
LogDisc 0.0600	0.538	Discrim	0.393	C4.5-auto	0.1446	Ross Quinlan's C4	0.0600
Castle	0.583	LogDisc	0.396	IDTM (Decision table)	0.1446	**CD**	0.0667
Alloc80	0.584	Alloc80	0.407	HOODG	0.1482	nearest neighbor	0.0790
Dipol92	0.599	QuaDisc	0.422	C4.5 rules	0.1494	"non-linear" perceptron	0.0800
Smart	0.601	Castle	0.441	OC1	0.1504	"linear" perceptron	0.0930
Cal	0.603	Cal5	0.444	C4.5	0.1554		
Cart	0.613	Cart	0.452	Voted ID3 (0.6)	0.1564		
QuaDisc	0.619	Cascade	0.467	CN2	0.1600		
KNN	0.694	KNN	0.478	Naive-Bayes	0.1612		
Default	0.700	Smart	0.478	Voted ID3 (0.8)	0.1647		
Bayes	0.703	Dipol92	0.507	T2	0.1684		
IndCart	0.761	Itrule	0.515	**CD**	0.1781		
Back Prop	0.772	Bay Tree	0.526	1R	0.1954		
BayTree	0.778	Default	0.560	Nearest-neighbor (4)	0.2035		
Cn2	0.856	BackProp	0.574	Nearest-neighbor (2)	0.2142		

6 Conclusion

An innovative new method for classification based on the notion of the correlation dimension and its estimate was suggested. Features of the correlation dimension can be easily and properly utilized for a classification task. The correlation dimension cannot be expressed in an analytical form but must be estimated. There are several methods that estimate it, for example the well-known Takens estimator. However, this method has some negative features, mainly underestimates the correlation dimension. Therefore, we introduced here the hyperbolic approximation that behaves better.

It is evident that the better estimation of probability distribution of data is at hand the better classification can be achieved. It has been found that a uniform distribution of data implies better results as well. Therefore nonlinearly transformed data was used to achieve it. By using a notion of distance, i.e. a simple transformation $E_n \rightarrow E_1$, the problems with dimensionality are easily eliminated at a loss of information on the true distribution of points in the neighborhood of the query point. The assumption of at least local uniformity in the neighborhood of a query point is fulfilled by the use of simple polynomial expansion where the exponent is equal to the correlation dimension.

The classification method has no tuning parameters and there is no true learning phase. In the "learning phase" normalization constants and an estimate of the correlation dimension are computed. It seems that it can outperform much sophisticated classification algorithms in some cases.

Acknowledgment

This work was supported by the Ministry of Education of the Czech Republic under project Center of Applied Cybernetics No. 1M0567, and No. MSM6840770012 Transdisciplinary Research in the Field of Biomedical Engineering II.

References

[1] Arya,S., Mount,D.M., Narayan,O.: Accounting for boundary effects in nearest neighbor searching, Discrete and Computational Geometry, Vol. 16 (1996) 155-176.

[2] Guerrero,A., Smith,L.A.: Towards coherent estimation of correlation dimension. Physics Letters A 318 (2003) , pp. 373–379.

[3] 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.

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

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

[6] Dvořák, I., Klaschka,J.: Modification of the Grassberger-Procaccia Algorithm for Estimating the Correlation Exponent of Chaotic Systems with High Embedding Dimension. Physics Letters (A), Vol. 145, 1990, No. 5, pp. 225-231 (ISSN: 0375-9601).

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

[8] Jirina,M., Jirina,M.jr.: Boundary Phenomenon in Multivariate Data. In: Pinker, J. (Ed.): Proceedings of the Applied Electronics 2004 International Conference Pilsen, 8-9 September 2004, ISBN 80-7043-274-8, University of West Bohemia in Pilsen, September 2004, pp.97-100.

[9] 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.

[10] Jirina,M., Jirina,M.jr.: Features of Neighbors Spaces. In: Peter Van Emde Boas, Jaroslav Pokorný, Mária Bieliková, et al.: SOFSEM 2004: Theory and Practice of Computer Science, Lecture Notes in Computer Science, vol. 2932/2003, pp. 241 – 248.

[11] Camastra,P., 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.

[12] Silverman,B.W.: Density estimation for statistics and data analysis, (Chapman and Hall, London, 1986).

[13] UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLSummary.html

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

[15] Buchala,S. et al.: Analysis of Linear and Nonlinear Dimensionality Reduction Methods for Gender Classification of Face Images. To be published in: International Journal of Systems Science, 2005.