Correlation Integral Decomposition for Classification*

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

[1] Institute of Computer Science AS CR, Pod vodarenskou vezi 2,
182 07 Prague 8 – Liben, Czech Republic
marcel@cs.cas.cz
http://www.cs.cas.cz/~jirina

[2] Faculty of Biomedical Engineering, Czech Technical University,
Zikova 4, 166 36, Prague 6, Czech Republic
jirina@fbmi.cvut.cz

Contents

Abstract. In this paper we introduce a new approach that utilizes the correlation dimension (CD) both for probability density estimate of data and consequently for classification. It will be shown that just a classifier utilizing CD exhibits significantly better behavior (classification accuracy) then other kinds of classifiers.
Correlation dimension is used for complexity estimation of fractals or any other data generating processes. Basic idea was introduced in the well known paper by Grassberger and Procaccia and there are papers dealing with the correlation dimension estimation. Some experiments that use the correlation dimension for classification has been published too.
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 depending on the position of the query point and on the number of points of the learning set while CD is not. It is shown that CD-based classifier outperforms DME classifier and many classifiers published on Machine Learning Repository pages.

* International Journal of Information Technology and Intelligent Computing, 2006, Vol. 1, No. 3, pp. 547-557. ISSN 1895-8648. Held: The 8th International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, June 25-29, 2006..

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. The basic idea was introduced by Grassberger and Procaccia [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 [3].

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 . They have also shown that correlation dimension estimated by the use of Grassberger-Procaccia's 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 [1], [2]. 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] derived from the k-nearest neighbor method [5], [12]. 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. 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 an example of a ball in an n-dimensional space containing points distributed over its volume. Let us divide the ball on concentric "peels" of the same volume.

A mapping between the mean density in an i-th peel ρ_i and its radius r_i is $\rho_i = p(r_i)$, where $p(r_i)$ is the mean probability density in the i-th ball peel with radius r_i. The probability distribution of points in the neighborhood of a query point x is thus simplified to a function of a scalar variable. We call this function a probability distribution mapping function $D(x, r)$ and its partial derivation according to r the distribution density mapping function $d(x, r)$. 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 [8], [9].

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 as a function of variable $z = r^q$ in neighborhood of origin, i.e. in the neighborhood of the query point. The distribution density mapping function $d(x, r)$ as function of variable $z = r^q$ 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.

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 [11].

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

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

In more comprehensive form one can write

$$C(r) = \Pr(\left\| X_i - X_j \right\| < r).$$

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 $\nu \le \sigma \le D_h$, 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 $\nu < n$ (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 can be rewritten in form [11]

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

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 proper value of the correlation dimension. One of most cited is Taken's estimator [2], [1].

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 as shown in the theorem below..

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 $D(i, r_{ik})$:

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

Proof

Let $h(x)$ be Heaviside step function. Then

$$C(l_{ij}) = \frac{1}{m_T (m_T - 1)} \sum_{i=1}^{m_T} \sum_{i=1}^{m_T - 1} D(i, l_{ij}) \,. \tag{2}$$

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

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

$$D(i, d) = \frac{1}{m_T - 1} \sum_{k=1}^{m_T - 1} h(d - l_{ik}) \,. \tag{3}$$

Comparing (2) and (3) we get directly (1). □

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

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

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. We come from the observation that 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 which 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.

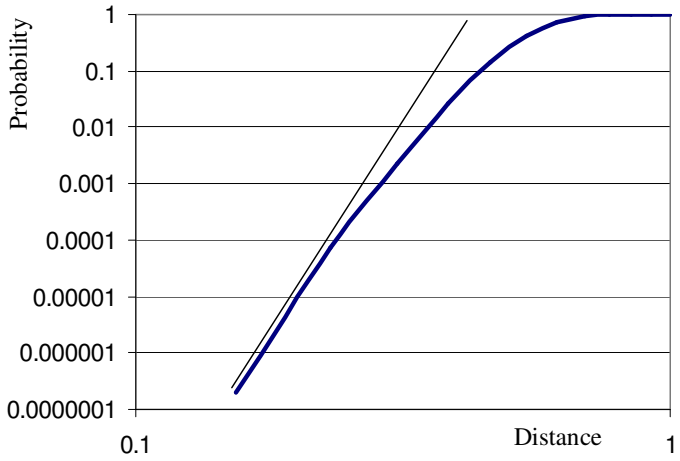

Fig. 1. Asymptote to the DMF or correlation integral.

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 $\Delta \ln r_i = \dfrac{a}{-\ln p_i}$, 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.

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^v instead of r.

Let U be a learning set composed of points (patterns, samples) x_i, where i is the index of point without respect to class $c = \{0, 1\}$ to which it belongs; x_i is the i-th nearest neighbor of point x. By symbol $i(c)$ we denote those indexes i that point $x_{i(c)}$ belongs to class c.

In the k-NN method the resulting estimation of probability is dependent on the number of points k inside the ball of radius r_k. It doesn't matter how the points inside the ball are distributed. Points can be concentrated in the center or spread to the surface of the ball, the result is the same. Let us consider partial influences of individual points to the probability that point x is of class c. Each point of class c in the neighborhood of point x adds a little to the probability that point x is of class c, where $c = \{0, 1\}$ is the class mark. This influence is the larger the closer the point considered is to point x and vice versa.

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

for the second point $i = 2$ $$p(x \mid c,2) = \frac{1}{S_n r_2^n},$$

and so on, generally for point No. i $$p(x \mid c,i) = \frac{1}{S_n r_i^n}.$$

Here S_n is constant dependent on dimensionality n and metrics used.

Then partial influences of individual points we add together by summing up

$$p(x \mid c,k) = \sum_{i=1(c)}^{k} p(x \mid c,i) = \frac{1}{S_n} \sum_{i=1(c)}^{k} 1/r_i^n \quad . \tag{4}$$

(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 i from point x will influence the probability that point x is of class c.

Let us compare this formula with formula for the k-NN method $p(x \mid c, k-NN) = \frac{i_c}{S_n r_k^n}$.

Here i_c denotes the number of points of class c from k nearest points to point x. In practical computation there is usually $p(x \mid c, k-NN) = \frac{i_c}{k}$.

In similar way we can rewrite Eq. (4) in more suitable form for practical computation.

$$p(x \mid c) = \frac{\sum_{i=2(c)}^{m_T} 1/r_i^n}{\sum_{i=2}^{m_T} 1/r_i^n} \quad .$$

(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 m_T points of the learning set are used instead of some number k. Moreover we do not use the nearest point ($i = 1$). It can be found that its influence is more negative than positive on the probability estimate.

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

$$p(x \mid c,k) = \sum_{i=1(c)}^{k} p(x \mid c,i) = \frac{1}{S_n} \sum_{i=1(c)}^{k} 1/r_i^{\nu} .$$

Again, if the sum of series $1/r_i^{\nu}$ converges with the size of r_i, we can use all points in the learning set excluding the nearest neighbor.

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 $1/r_i^{\nu}$ 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	0.261	**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 we used nonlinearly transformed data 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 (1M684077004), and No. MSM6840770012 Transdisciplinary Research in the Field of Biomedical Engineering II.

References

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

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

[3] 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.

[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