Using Singularity Exponent in Distance Based Classifier *

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. The paper deals with using so called singularity exponent in a classifier that is based on ordered distances of patterns to a given (classified) pattern. The approximation of probability distribution mapping function of the distribution of points from the viewpoint of distances from a given point in a form of a suitable power (exponent) of a distance is presented together with a way how to state it. A classifier utilizing knowledge about explored data distribution in a space and a suggested expression of the exponent is presented. Experimental results on both synthetic and real-life data show interesting behavior (classification accuracy) of the classifier in comparison with other well-known classifiers.

---

* Final version published: Marcel Jirina and Marcel Jirina, jr.: Using Singularity Exponent in Distance Based Classifier. Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA2010) November 29 - December 1, 2010 Cairo Egypt, paper No. 1011, pp. 220-224, ISBN: 978-1-4244-8135-4..

I Introduction

In this paper we use or slightly redefine if needed some notions from multifractals theory for a classification task. One of the most important elements of chaos theory are singularity exponents (also called scaling exponents). They are used in multifractal chaotic series analysis. We try here to use these exponents for a classification problem. In classification the task is to properly recognize to which class a presented multivariate sample belongs. This task usually has nothing to do with time series but as shown already by Mandelbrot, 1982, [1] any data may posses a fractal or multifractal nature.

In classification problems, the only known fact is the learning set, i.e. the set of points each of known class. The problem is how to estimate the probability to which class a query point x of the data space belongs. The different approaches to classification can be divided into parametric and nonparametric methods. Parametric methods include neural networks of different kinds [2], decision trees or forests and many more. Nonparametric methods are mostly based on the Bayesian approach [3] and the k nearest neighbors (k-NN) method [3] - [6].

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

A core notion in this transformation is a slightly redefined singularity or scaling exponent to fit notion of distance between points. 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 [7] than to box-counting or other fractal or multifractal dimension definitions [1], [8].

We remind three notions introduced or used in [9]-[13]. The *probability distribution mapping function* is a mapping of the probability distribution of points in n-dimensional space to the distribution of points in one-dimensional space of the distances. The *distribution density mapping function* (DDMF) is a one-dimensional analogy to the probability density function. The power approximation of the *probability distribution mapping function* in the form of $(\text{distance})^q$ is introduced, where the exponent q we call the *distribution mapping exponent* (DME).

These notions are local, i.e. are related to a particular (query) point. We show that the distribution mapping exponent q is something like a local value of the correlation dimension according to Grassberger and Procaccia, [8]. It can be viewed also as the local dimension of the attractor by Froehling [14] or singularity eventually scaling exponent ("exponent") in the sense of Stanley and Melkin, [8].

II. Probability Density Estimation

A. Probability Distribution Mapping Function

To study a probability distribution of points (patterns) in the neighborhood of a query point x in n-dimensional Euclidean space E_n, let us introduce two definitions.

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

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

B. Power Approximation of the Probability Distribution Mapping Function

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

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

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

The distribution mapping exponent (DME) reminds one of the so-called correlation dimension by Grassberger and Procaccia [7], and corresponds to generally used definitions of power scaling laws especially to singularity exponent. It can be seen that the correlation integral is a distribution function of distances between all pairs of points of the data points given. The probability distribution mapping function is a distribution function of the distances from one fixed point x. In the case of finite number of points N, there are $N(N - 1)/2$ distances between pairs of points and from them one can construct an empirical correlation integral. Similarly, for each point there are $N - 1$ distances and from these $N - 1$ distances one can construct an empirical probability distribution mapping function. There are exactly N such functions and the mean of these functions gives the correlation integral. This is also valid for N going to infinity.

C. Distribution Mapping Exponent Estimation

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

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

To estimate the distribution mapping exponent q we use a similar approach to the approach of Grassberger and Procaccia, [8], for the correlation dimension estimation.

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

$$r_s^q = ks \text{ , } s = 1, 2, ..., N_c, \quad c = 0 \text{ or } 1, \qquad (1)$$

where k is a proportionality constant, which will be eliminated later, so we need not bother with it. Using a logarithm we get

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

This is a task of estimating the slope of a straight line linearly approximating the graph of the dependence of the neighbor's index s as a function of distance in log-log scale. This is the same problem as in the correlation dimension estimation where equations of the same form as (1) and (2) arise. Grassberger and Procaccia [7], proposed a solution by linear regression. Other authors proposed different modifications and heuristics later. Many of these approaches can be used for the distribution mapping exponent estimation, e.g. the use of $N_v < N_c$ nearest neighbors instead of N_c eliminates the influence of a limited number of the points of the learning set. N_v may be equal e.g. to one half or the square root of N_c. The accuracy of the distribution mapping exponent estimation is the same problem as the accuracy of the correlation dimension estimation. On the other hand, one can find that a small change of q does not essentially influence the classification results.

We solve the system of N_v equations (2) with respect to an unknown q by the use of standard linear regression for both classes. Thus, for two classes we get two values of q, q_0 and q_1. To get a single value of q we use the arithmetic mean, $q = (q_0 + q_1)/2$. For more classes, the arithmetic mean of the q's for the individual classes is used.

III. The Method

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

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

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

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

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

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

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

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

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

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

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

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

IV. Classifier Construction

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

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

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

Note that for the different number N_0 and N_1 of the samples of one and the other class formula (1) has the form

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

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

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

$$\hat{p}(c \mid x) = \frac{\frac{1}{N_c} \sum_{i=2(c)}^{N} 1/r_i^q}{\sum_{k=1}^{M} \frac{1}{N_k} \sum_{i=2(c)}^{N} 1/r_i^q} \qquad (7)$$

V. Experiments

The presented classification method based on DME with respect to other well-known classification algorithms is compared on both synthetic and real-life data.

A. Synthetic Data

Synthetic data according to Paredes and Vidal [5] is two-dimensional and consists of three two-dimensional normal distributions with identical a-priori probabilities. μ denotes the vector of means and $\boldsymbol{C}_m$ is the covariance matrix

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

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

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

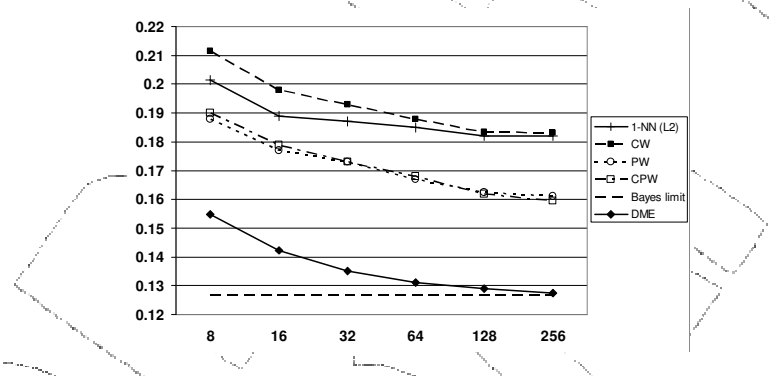

Figure 1. Comparison of classification errors of the synthetic data for the different approaches. In the legend, 1-NN (L2) means the 1-NN method with Euclidean metrics, CW, PW, and CPW are three variants of the method by Paredes and Vidal, 2006, [5].

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

B. Data from Machine Learning Repository

Tasks from UCI Machine Learning Repository – Comprehensive Tests.

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

We used real-life tasks mainly from the UCI Machine Learning Repository, see Asuncion and Newman, 2007, [15]. DNA data can be found in Paredes, 2009, [6]. 24 databases have been used for the classification task into two to 26 classes. The number of attributes not including the class mark differs from 4 to 180.

Classification methods compared

The best results obtained with five different classification methods are shown in Table 1. We used five classification methods as follows. Notation corresponds to columns in Table 1.

- Bayes – the naïve Bayes method that uses 10 bins histograms (e.g. [3]).
- 1-NN – standard nearest neighbor method (Cover and Hart, 1967, [4]).
- ParedBest – the best results obtained by three variants of method by Paredes and Vidal [5], [6].
- SVMbest – the best results obtained with support vector machine (Joachims [16]) using four types of kernels.
- DMEbest – the best results obtained with the method presented here with different DME estimations.

VI. Discussion

Our model of the polynomial expansion of the data space comes from the demand to have a uniform distribution of points, at least locally. We introduced the distribution mapping exponent as redefinition of the singularity or scaling exponent from the point of view of distances of near points. There is an interesting relationship between the correlation dimension and the distribution mapping exponent. The former is a global feature of the fractal or data generating process; the latter is a local feature of the data set and is closely related to a particular query point. On the other hand, if linear regression were used, the computational procedure is almost the same in both cases. Not surprisingly, the mean value of the distribution mapping exponent over all samples is not far from the correlation dimension. Our experiments demonstrate that the simplest classifier based on the ideas introduced here can outperform other methods for some data sets. On the other hand, the target of this paper was to present basically new approach to probability density estimation and classification.

TABLE I. CONDENSED COMPARISON OF FIVE TYPES OF METHODS INCLUDING DME METHOD PRESENTED HERE. IN BOLD THE BEST RESULT (CLASSIFICATION ERROR) FOR EACH PARTICULAR DATA SET IS SHOWN.

Dataset	Bayes	1-NN	ParedBest	SVMbest	DMEbest
Australian	14.88%	34.29%	31.91%	35.99%	**14.20%**
Balance	15.17%	22.05%	**13.68%**	33.17%	24.85%
Cancer	**2.68%**	4.83%	3.41%	16.32%	3.69%
Diabetes	25.19%	32.76%	29.60%	29.64%	**24.75%**
DNA	6.66%	23.44%	3.71%	**0.00%**	28.33%
German	**24.97%**	33.74%	29.79%	27.25%	27.64%
Glass	47.37%	30.81%	**30.75%**	32.63%	34.47%
Heart	18.44%	41.48%	38.15%	37.22%	**17.96%**
Ionosphere	9.26%	14.07%	**5.87%**	18.52%	15.58%
Iris	9.82%	5.91%	**4.91%**	5.55%	5.91%
Led17	**0.00%**	24.92%	0.02%	11.52%	0.32%
Letter	28.98%	4.35%	3.25%	**2.68%**	5.73%
Liver	39.42%	39.25%	38.14%	**35.54%**	40.09%
Monkey1	28.01%	29.47%	**0.04%**	2.94%	8.22%
Phoneme	21.47%	**11.50%**	11.60%	14.39%	16.49%
Satimage	19.15%	10.55%	**9.25%**	24.30%	11.95%
Segmen	9.85%	4.30%	**3.76%**	34.27%	6.48%
Sonar	31.46%	22.62%	19.42%	**19.67%**	24.25%

Vehicle	38.40%	35.08%	29.95%	**26.23%**	29.37%
Vote	9.70%	8.13%	**5.35%**	22.64%	9.28%
Vowel	26.64%	1.37%	**1.33%**	8.54%	6.66%
Waveform21	19.26%	21.91%	18.30%	26.34%	**15.05%**
Waveform40	20.31%	23.34%	24.55%	32.25%	**16.49%**
Wine	5.50%	27.05%	19.46%	8.85%	**5.04%**

Acknowledgements

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

References

[1] B. Mandelbrot: The Fractal Geometry of Nature, W. H. Freeman & Co; ISBN 0-7167-1186-9 (1982).

[2] S. Haykin. *Neural Networks: A Comprehensive Foundation*, (2nd Edition) Prentice Hall, USA, (1998).

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

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

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

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

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

[8] H.E.Stanley, P. Melkin: *Multifractal phenomena in physics and chemistry*. (Review) Nature Vol. 335, 29 Sept. 1988, pp. 405-409.

[9] Jiřina, Marcel. Dimensionality Reduction and Classification using the Distribution Mapping Exponent. In ESANN'2004. Everedside, 2004. s. 169-174. ISBN 2-930307-04-8. [ESANN'2004. European Symposium on Artificial Neural Networks /12./, Bruges, 28.04.2004-30.04.2004, BE].