Classification Using Zipfian Kernel *

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

[1] Institute of Computer Science AS CR, v.v.i., Pod vodárenskou věží 2, 182 07 Prague 8 – Libeň, Czech Republic, marcel@cs.cas.cz

[2] Faculty of Biomedical Engineering, Czech Technical University in Prague, Nám. Sítná 3105, 272 01, Kladno, Czech Republic, jirina@fbmi.cvut.cz

Contents

Abstract
We propose to use the Zipfian distribution as a kernel for design of a nonparametric classifier in contrast to the Gaussian distribution used in most of kernel methods. We show that the Zipfian distribution takes into account multifractal nature of data and gives a true picture of scaling properties inherent in data. We also show that this new look at data structure can lead to a simple classifier that can, for some tasks, outperform more complex systems.

Keywords: kernel machine, Zipfian kernel, multivariate data, correlation dimension, harmonic series, classification.

* Final vrsion published: Jiřina, Marcel - Jiřina jr., Marcel: Classification Using Zipfian Kernel. Journal of Classification (Springer), Vol. 32, 12 April 2015, pp 305-326. ISSN 0176-4268.

I. INTRODUCTION

The proper selection of the kernel and window width are essential for good classification in classification tasks [19], [20]. This problem is solved solely from statistical point of view. However, the role of spatial correlations and the effective data dimensionality are not considered in kernel methods.

Here we show that a suitable alternative to the standard kernel functions can be the Zipfian distribution [15], [21] which can take into the account a fractal nature of data.

Furthermore, this classification method is truly nonparametric as there is no need to set up the window width common in most kernel methods. A new kind of nonparametric classifier of multivariate data is proposed.

In our method we center Zipfian kernel to each point x_i of learning data set. At point x the kernel gives value (probability density, in fact probability mass as Zipfian distribution is discrete) $1/i$; x_i is i-th nearest neighbor of x. In the two class classification problem summing up these values for all points x_i of class 1 gives S_1, for points of class 2 gives S_2. Estimate of probability that point x is of class 1 is

$$\hat{p}(c=1|x)=\frac{S_1(x)}{S_1(x)+S_2(x)} \ .$$

Then if there is given threshold θ and if the estimate above is greater than θ then we say that point x is of class 1 else it is of class 2.

We prove here that the kernel method with the Zipfian kernel gives an unbiased approximation of the probability of class of the given point. For proof we use here or necessarily redefine some notions from the multifractals theory. Singularity exponents, also scaling exponents are widely used in multifractal chaotic series analysis and can be related to data that do not form a series. It was shown already by Mandelbrot [14] that any data may possess a fractal or multifractal nature. We use these exponents for proof of our method of classification. As there is no time scale, even no ordering of samples, one cannot use such a tool as wavelet functions.

Our results demonstrate that the kernel method can be related to the fractal nature of data and to the harmonic series [15], [18] via the Zipfian distribution [21].

This work can be a starting point for more detailed description of local behavior of multivariate and not exactly self-similar fractal data, and for the development of new approaches to data analysis including classification problems.

II. KERNEL ESTIMATOR

A. Data and the learning set

Let the learning set U of total N samples be given. Each sample $x_t=\{x_{t1}, x_{t2},\dots x_{tn}\}$; $t = 1, 2, \dots N$, $x_{tk} \in R$; $k = 1, 2, \dots, n$ 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 $T: R^n \to \{1, 2, \dots C\}$: $T(x_t) = c$ is introduced. With the class function the learning set U is decomposed into disjoint classes $U_c = \{x_t \in U \mid T(x_t) = c\}$; $c \in \{1, 2, \dots, C\}$, $\bigcup_{c=1}^{C} U_c$, $U_c \cap U_d = \emptyset$; $c, d \in \{1, 2, \dots, C\}$; $c \neq d$. Cardinality of set U_c let be N_c; $\sum_{c=1}^{C} N_c = N$.

As we need to express which sample is closer or further from some given point x, we can rank points of the learning set according to distance r_i of point x_i from point x. Therefore, let points of U be indexed (ranked) so that for any two points $x_i, x_j \in U$ there is $i < j$ if $r_i < r_j$; $i, j = 1, 2, \dots N$, and class $U_c = \{x_i \in U \mid T(x_i) = c\}$. Of course, the ranking depends on point x and eventually metrics of M_n. We use Euclidean (L_2) and absolute (Manhattan, L_1) metrics here. In the following indexing by i means ranking just introduced.

B. Zipfian kernel

The Zipfian distribution (Zipf's law) [15][21] predicts that out of a population of N elements, the frequency of elements of rank i, $f(i;s,N)$, is given by probability mass function

$$f(i;s,N) = \frac{1/i^s}{\sum_{t=1}^{N} 1/t^s}, \tag{1}$$

where N is the number of elements, i is their rank, s is the value of the exponent characterizing the distribution. The law may also be written as:

$$f(i;s,N) = \frac{1}{i^s H_{N,s}},$$

where $H_{N,s}$ is the Nth generalized harmonic number.

The simplest case of Zipf's law is a "1/f function" arising when $s = 1$. Given a set of Zipfian distributed frequencies of occurrence of some objects, sorted from the most common to the least common, the second most common frequency will occur ½ as often as the first. The third most common frequency will occur 1/3 as often as the first, and so on. Over fairly wide ranges, and to a fairly good approximation, many natural phenomena obey Zipf's law. Note that in the case of a "1/f function", i.e. $s = 1$, N must be finite and its denominator is equal to H_N, the so-called harmonic number, i.e. the sum of truncated harmonic series [18]; otherwise the denominator is a sum of harmonic series, which is divergent. This is not true if exponent s exceeds 1, $s > 1$, then the series is convergent,

$$\zeta(s) = \sum_{t=1}^{\infty} \frac{1}{t^s} < \infty,$$

where ζ is Riemann's zeta function.

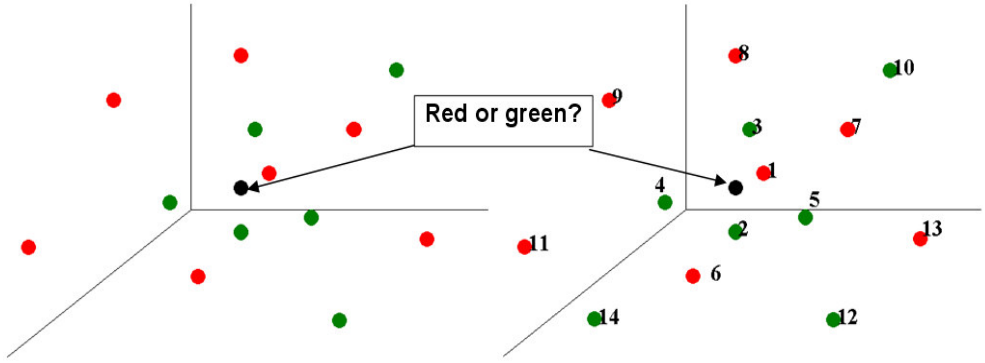
III. THE METHOD

A. Intuitive basis

The method of probability estimation proposed is based on the following illustrative example. Let us consider partial influences of individual points to the probability that point x is of class c; we consider two classes only here. Both classes have the same cardinality. 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 $c = \{1, 2\}$ is the class mark. This influence is the larger the closer the point considered is to point x and vice versa. This observation is based on the finding of [5] that the first nearest neighbor has the largest influence on proper estimation to what class point x belongs. Suppose that the influence on the probability that point x is of class c of the nearest neighbor of class c is 1, the influence of the second nearest neighbor is ½, the influence of the third nearest neighbor is 1/3 etc. Just these values are related to Zipfian distribution.

From kernel methods point of view, we center Zipfian kernel to each point x_i of data set. At point x the kernel gives probability mass proportional to $1/i$ (we use exponent $s = 1$); x_i is i-th nearest neighbor of x. Summing up these values for points x_i of class 1 gives number S_1, for points of class 2 number S_2. Estimate of probability that point x is of class 1 is

$$\hat{p}(c=1|x)=\frac{S_1(x)}{S_1(x)+S_2(x)}\ .$$



samples ordered according to the distance from the query point

1	2	3	4	5	6	7	8	9	10	11	12	13	14	S or S_C	p(r/g|x)
1	0.5	0.33	0.25	0.20	0.17	0.14	0.13	0.11	0.10	0.09	0.08	0.08	0.07	3.251562	
1					0.17	0.14	0.13	0.11		0.09		0.08		1.713467	0.526967
	0.5	0.33	0.25	0.20					0.10		0.08		0.07	1.538095	0.473033

reciprocals of order numbers for class of red points

reciprocals of order numbers (without respect to class - color)

sum of rows

probability that the point belongs to the red class

$$0.526967 = \frac{1.713467}{3.251562}$$

Figure 1. Illustration of classification procedure for the simplest case of two classes and of the same number of samples of both classes.

The classification procedure is depicted in Fig. 1. The problem is: What is color of given point x depicted in black at the left upper part of picture? First we rank points of the learning set according to their distances from point x as shown at the right upper part of picture. There are 14 points here, 7 red, 7 green as shown in the upper lines in the table below pictures. Reciprocals of rank numbers are in the third line. In the fourth and fifth line there are reciprocals of ranks of points x_i from sets $U_{c=red}$ and $U_{c=green}$. In the rightmost two columns of table are corresponding sums and estimated probabilities that point x is red (0.526967) or green (0.473033). Setting threshold $\theta = 0.5$ we can state that point x is red.

From another point of view, let $\Pr(T(x)=c\,|\,T(x_i)=c)$ be the probability that the given point x is of class c if neighbor point number i is of the same class as point x. Note that points of the learning set U are indexed so that for any two points $x_i, x_j \in U$ there is $i<j$ if $r_i < r_j$; $i, j = 1, 2, ..., N$. In the following K is a constant that is used to normalize the probability that point x belongs to a class to 1:

For the first (nearest) point $i = 1$ $\quad \Pr(T(x)=c\,|\,T(x_1)=c)=K\frac{1}{1}$,

for the second point $i = 2$ $\quad \Pr(T(x)=c\,|\,T(x_2)=c)=K\frac{1}{2}$,

and so on, generally for point No. i $\quad \Pr(T(x)=c\,|\,T(x_i)=c)=K\frac{1}{i}$.

Individual points are independent and then we can sum up these probabilities. Thus, we add the partial influences of k individual points together by summing up

$$\hat{p}(c \mid x) = \sum_{x_i \in U_c} \Pr\big(T(x) = c \mid T(x_i) = c\big) = K \sum_{x_i \in U_c} 1/i \,. \tag{2}$$

Note that the sum goes over indexes i for which the corresponding samples of the learning set are of class c, $c = 1, 2, \ldots C$, where C is the number of classes.

Let

$$S_c = \sum_{x_i \in U_c} 1/i \,.$$

Then there is

$$\sum_{c=1}^{C} S_c = H_N \,,$$

where H_N is the N-th harmonic number. The estimation of the probability that the given point x belongs to class c is

$$\hat{p}(x \mid c) = \frac{S_c}{H_N} \,.$$

This approach is based on hypotheses that the weight of a neighbor is proportional just to reciprocal of its order number as to its distance from the given point, see Theorem 1.

It can be seen that $K\dfrac{1}{i}$ is also the value (at point x) of kernel function with its center at point x_i and having form of Zipfian probability mass function (1) for $s = 1$. Summing up these values over all centers that belong to class U_c gives (2).

B. The Classification Procedure

The probability estimation above can be used for classification. Let the samples of the learning set (i.e. all samples irrespective of the class) be sorted according to their distances from the given point x. Let indexes be assigned to these points so that 1 is assigned to the nearest neighbor, 2 to the second nearest neighbor of the given point x etc.

Let us compute sums $S_c(x) = \dfrac{1}{N_c} \sum_{x_i \in U_1} 1/i$, i.e. the sums of reciprocals of the indexes of samples from each class c separately; N_c is the number of samples of class c (cardinality of U_c) and $N_1 = N_2 = \ldots = N_C$.
The estimate of the probability that point x belongs to class c is

$$\hat{p}(c \mid x) = \frac{S_c}{\sum_{k=1}^{C} S_k} \tag{3}$$

In the end, the formula above is nothing else than Bayes formula.
Usually we say that point x is of class k if $\hat{p}(k \mid x)$ is the largest of all $\hat{p}(c \mid x)$;
In the case of two classes, when some discriminant threshold θ is chosen then if $p(c = 1 \mid x) \geq \theta$ point x is of class 1, else it is of class 2. This is the same procedure as in other classification approaches where the output is estimation of probability (naïve Bayes) or any real valued variable (neural networks). The value of the threshold can be optimized with respect to loss function.
For classification into more than two classes we use this formula for all classes and we assign to the given point x a class c for which $p(c \mid x)$ is the largest.

Formally we can rewrite (3) into more comprehensive form. For two class problem with different number of samples of one and the other class formula we have

$$\hat{p}(c \mid x) = \frac{\dfrac{1}{N_c} \sum_{x_i \in U_c} 1/i}{\dfrac{1}{N_1} \sum_{x_i \in U_1} 1/i + \dfrac{1}{N_2} \sum_{x_i \in U_2} 1/i} \,.$$

It is seen here the introduction of the relative representation of different numbers of samples of one and the other class, i.e. introducing a priori probabilities.

For C classes there is

$$\hat{p}(c \mid x) = \frac{\frac{1}{N_c} \sum_{x_i \in U_c} 1/i}{\sum_{k=1}^{C} \frac{1}{N_k} \sum_{x_i \in U_k} 1/i}.$$

IV. Approximation of Probability of Class

A. Zipfian kernel approach

Let indexes i be assigned to points (samples) of the learning set without respect to a class so that $i = 1$ is assigned to the nearest neighbor of point x, $i = 2$ to the second nearest neighbor etc. We have a finite learning set of size N samples and N_c samples of individual class.

Using Zipfian probability mass function (1) as a kernel function, we have the kernel function in the form $K(\|x-x_i\|/h)=1/(i^s H_{N,s})$. At the same time, product Nh, i.e. the number of samples times the smoothing factor has no significance here and we set $Nh = 1$. Then we get approximation of probability that the given point x belongs to class c in the form

$$\hat{p}(c \mid x) = \frac{1}{H_{N,s}} \sum_{i \in U_c} \frac{1}{i^s}, \tag{4}$$

where the sum goes over indexes i for which the corresponding samples of the learning set are of class c. Summing up approximations of probability densities at point x for all classes, we get apparently $\frac{1}{H_{N,s}} \sum_{i=1}^{N} \frac{1}{i^s}$ that is equal to 1 and thus fulfills assumption that the given point belongs to some class.

Two classes only and the same number of samples of both classes are assumed without loss of generality in the theorem and the proof as follows.

Theorem 1. Let the task of classification into two classes be given and let the size of the learning set be N and let both classes have the same number of samples, i.e. there is the same a priori probability. Let i be the index (rank) of the i-th nearest neighbor x_i of point x (without considering the neighbor's class) and r_i be its distance from the point x. Then

$$\lim_{N \to \infty} \frac{\sum_{x_i \in U_c} 1/i}{H_N} = p(c \mid x) \quad . \tag{5}$$

where $p(c|x)$ is the probability that point x belongs to class c.

In the following proof we use some notions known from [10], [11], [12] and shortly summarized as follows.

1) Mapping the distribution

Let us have an example of a ball in an n-dimensional space containing uniformly distributed points over its volume. Let us divide the ball on concentric "peels" of the same volume. Using the formula $r_i = \sqrt[n]{V_i / S(n)}$, which is, in fact, inverted formula for volume V_i of n-dimensional ball of radius r_i, we obtain a quite interesting succession of radii corresponding to the individual volumes - peels. The symbol $S(n)$ denotes the volume of a ball with unit radius in E_n; note $S(5) = 4/3\pi$. A mapping between the mean density ρ_i in an i-th peel and its radius r_i is $\rho_i = p(r_i)$; $p(r_i)$ 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 given point x is thus simplified to a function $p(r_i)$ of a scalar variable r_i. We call this function a probability distribution mapping function $D(x, r)$ and its partial differentiation with respect to r the distribution density mapping function $d(x, r)$. Functions $D(x, r)$ and $d(x, r)$ for x fixed are, in fact, the probability distribution function and the probability density function of variable r, i.e. of distances of all points from the given point x. More exact definitions follow [11].

Definition 1. Probability distribution mapping function $D(x, r)$ of the given point x is function $D(x,r) = \int_{B(x,r)} p(z)dz$, where r is distance from the given point and $B(x, r)$ is ball with center x and radius r.

Definition 2. Distribution density mapping function $d(x, r)$ of the given point x is function $d(x,r) = \frac{\partial}{\partial r} D(x,r)$, where $D(x, r)$ is a probability distribution mapping function of the given point x and radius r.
Note. When it is necessary to differentiate class of point in distance r from point x, we write $D(x, r, c)$ or $d(x, r, c)$.

2) Correlation dimension

It is seen that for fixed x the function $D(x, r)$, $r > 0$ is monotonously non-decreasing from zero to one. 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. In fact, $D(x, r)$ is the distribution function of distances of points from the given point x and $d(x, r)$ is corresponding probability density function. So we can write $p(c|x, r) = d(x, r, c)$. Moreover, $D(x, r)$ reminds the correlation integral [6]. The correlation integral

$$C(r) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{i,j=1}^{N} h(r - |x_i - x_j|),$$

where x_i and x_j are points of the learning set without respect to class and $h(.)$ is a Heaviside's step function, can be written in the form [3], [4]

$$C(r) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} h(r - |x_i - x_j|).$$

It can be seen [3], [4] that correlation integral is a distribution function of distances between pairs of data points given. The probability distribution mapping function is a distribution function of distances from one fixed point. In the case of finite number of points N, there are $N(N - 1)/2$ pairs of points and then distances between them, and from them one can construct empirical correlation integral. Similarly, for each point there are $N - 1$ distances and from these $N - 1$ distances one can construct empirical probability distribution mapping function. There are exactly N such functions and mean of these functions gives empirical correlation integral. This is valid also in limit for number N of points going to infinity.
On the other hand there are essential differences. The probability distribution mapping function is a local feature dependent on the position of point x. Empirical distribution mapping function also includes boundary effects [2] of true data set. The correlation integral is a feature of the fractal or data generating process and should not depend on the position of the particular point considered or on the size of the data set at hand.
In a log-log graph of the correlation integral, i.e. the graph of the dependence of C on r, the slope gives the correlation dimension ν. In the log-log graph of the probability distribution mapping function $D(x, r)$ the curve is also close to a monotonously and nearly linearly growing function. The slope (derivative) is given by a constant parameter. Let us denote this parameter q and call it the distribution mapping exponent. This parameter is rather close but generally different from ν.
The linear part of the log-log graph means

$$\log C(r) = a + \log \nu,$$

where a is a constant, and then $C(r) = ar^{\nu}$. Thus, $C(r)$ grows linearly with variable $z = r^{\nu}$.
Similarly the probability distribution mapping function grows linearly with r^q at least in the neighborhood of point x. Its derivative, the distribution density mapping function, is constant there.

3) Proof of Theorem 1

There are c spatial distributions $p_c(x)$ of probability that any point x (on the support considered) is of class c. Then for each point x and C classes there is $\sum_{c=1}^{C} p_c(x) = 1$. For each given point x one can state the probability distribution mapping function $D(x, r_i, c)$. We approximate this function so that it holds (K is a constant)

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

in the neighborhood of point x. Using derivation, according to variable $z = r_i^q$, we get $d(x, r_i^q, c) = K$. It means that by the use of $z = r_i^q$, the space is mapped ("distorted") so that

the distribution density mapping function is constant in the neighborhood of point x for a particular distribution. Let us consider sum $\sum_{i=1}^{N} d(x, r_i^q, c) / r_i^q$. For this sum we have

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

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

By the use of $z_i = r_i^q$, the space is nonlinearly rescaled so that the distribution density mapping function $d(x, z_i, c)$ is constant in the neighborhood of point x. Then r_i^q is proportional to i, $r_i^q = k_1 i$; k_1 is a constant. Exponent q need not be a constant but can be a function $q = q(x, c)$; we write it for point x_i in form $q = q(i, c)$. Let $r_i^{q(i,c)} = k_1 i$ for all i of class c. (From the last formula one could derive the $q(i, c)$, but we need not do it.) We rewrite the equation (6) in the form

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

and then in the form

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

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 given point x is of the class c. Then $p_c(r_i)$ is equal to 1 if point 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 $p(c \mid x) H_N$ is sum $\sum_{x_i \in U_c} 1 / i$. Using this sum, we can write

$$p(c \mid x) \lim_{N\to\infty} H_N = \lim_{N\to\infty} \sum_{x_i \in U_c} 1 / i .$$

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

$$\frac{\lim_{N\to\infty} \sum_{x_i \in U_c} 1 / i}{\lim_{N\to\infty} H_N} = p(c \mid \mathrm{x})$$

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

B. Bayes Risk

We have shown that estimate (4) converges to true probability $p(c|x)$. Considering two-class classification with simple loss matrix $L(1, 1) = L(2, 2) = 0,\ L(1, 2) = L(2,1) = 1$ there is conditional Bayes risk of estimating a class of point x

$$R(x) = R(c \mid x) + R(\mathit{not}\ c \mid x) = 2(1 - p(c \mid x)) .$$

It is apparent that its estimate $\hat{R}(x) = 2(1 - \hat{p}(c \mid x))$ converges to $R(x)$ as $\hat{p}(c \mid x)$ converges to $p(c \mid x)$. The $\hat{R}(x)$ can be computed easily having classification error that is equal to $1 - \hat{p}(c \mid x)$ and can be found e.g. in Table 1, see Chap. 5.

C. Computational complexity

For total N samples and single given point x the procedure consists of three steps:

- Computation of distances; the computational complexity for one distance is proportional to dimensionality n, of all N distances nN.
- Sorting distances is proportional to $N\log N$.
- Summing up of reciprocals of indexes is proportional to N.

Then the total complexity is $anN + bN\log N + dN = N(an + b\log N + d)$, where a, b, d are implementation dependent constants. For larger learning data sets the complexity is governed by sorting. It is also seen that the computational complexity directly depends on the learning set size N and in small extend on dimensionality n.

V. Experiments

A. Tasks from UCI Machine Learning Repository – Comprehensive Tests

Data sets ready for a run with a classifier were prepared by Paredes and Vidal and are available on the net [13]. For small data sets in this corpus each task consists of 50 pairs of training and testing sets corresponding to 50-fold cross validation. For large data sets, i.e. DNA data [16], Letter data (Letter recognition [1]), and Satimage (Statlog Landsat Satellite [1]) the single partition into training and testing sets according to specification in [1] was used. We also added the popular Iris data set [1] with ten-fold cross validation.

In Table 1 the results obtained by different methods are summarized. The methods are as follows:

L2	The nearest neighbor method by [17]
1-NN L2	The nearest neighbor method computed by authors
sqrt-NN L2	The k-NN method with k equal to square root of the number of samples of the learning set computed by authors
Bayes 10	The Bayes naive method with ten bins histograms, computed by authors
CDM	The learning weighted metrics method with class dependent Mahalanobis by [17]
CW	The learning weighted metrics method with class dependent weighting by [17]
PW	The learning weighted metrics method with prototype dependent weighting by [17]
CPW	The learning weighted metrics method with class and prototype - dependent weighting by [17]
IINC L1	The method presented here with Manhattan L_1 metrics
IINC L2	The method presented here with Euclidean L_2 metrics

In Table 1 in each row the best result is denoted by bold numerals. Furthermore, in the last column, the values for IINC better with L_2 metrics than with L_1 metrics are shown in italics. There are five such cases out of a total of 24.

Table 1. Classification error rates for different datasets and different approaches. Empty cells denote not available data. For legend see text above.

\Method	L2	1-NN L2	sqrt-NN	Bayes 10	SVM	CDM	CW	PW	CPW	**IINC L1**	**IINC L2**
Australian	34.37	20.73	15.50	13.88	35.99	18.19	17.37	16.95	16.83	**13.31**	14.75
Balance	25.26	23.61	32.06	15.17	45.48	35.15	17.98	**13.44**	17.6	32.58	*30.80*
Cancer	4.75	5.07	3.25	**2.68**	16..34	8.76	3.69	3.32	3.53	3.28	3.48
Diabetes	32.25	29.48	26.46	**24.19**	29.64	32.47	30.23	27.39	27.33	26.21	*25.52*
DNA	23.4	25.72	34.06	6.66		15	4.72	6.49	**4.21**	27.82	31.03
German	33.85	32.76	30.90	**24.97**	27.25	32.15	27.99	28.32	27.29	30.91	31.13
Glass	**27.23**	32.72	42.10	47.37		32.9	28.52	26.28	27.48	33.01	35.18
Heart	42.18	25.11	16.89	17.44	38.89	22.55	22.34	18.94	19.82	17.96	***17.93***
Ionosphere	19.03	14.05	14.70	**9.26**						10.82	14.81
Iris	6.91	5.91	7.91	9.82	6.55					7.91	***4.91***
Led17	20.5	11.50	**0.12**	0.00						0.46	*0.45*
Letter	4.35	4.80	18.70	28.98	40.53	6.3	3.15	4.6	**4.2**	4.85	4.98
Liver	37.7	39.59	41.48	39.42	37.68	39.32	40.22	**36.22**	36.95	38.29	39.13
Monkey1	2.01	2.01	9.27	28.01	23.54					4.79	4.79
Phoneme	18.01	11.83	20.71	21.47	21.71					17.55	18.06
Satimage	10.6	10.65	15.20	19.15	44.85	14.7	11.7	**8.8**	9.05	11.00	11.55
Segmen	11.81	3.81	11.41	9.85						**4.12**	5.05
Sonar	31.4	18.37	32.51	31.46						**19.89**	22.85
Vehicle	35.52	30.51	31.51	38.40		32.11	29.38	**29.31**	28.09	29.40	*29.34*
Vote	8.79	8.74	9.60	9.70		6.97	6.61	5.51	**5.26**	8.52	8.89
Vowel	1.52	**1.19**	46.68	26.64		1.67	1.36	1.68	1.24	2.73	2.74
Waveform 21	24.1	23.73	**14.71**	19.26						16.15	16.38
Waveform 40	31.66	28.22	**16.24**	20.31						17.59	18.08
Wine	24.14	5.42	6.15	4.50		2.6	1.44	1.35	**1.24**	4.24	5.66

B. Standalone Real-life Comprehensive Classification Task

This data set was available for tests described in [7] as one of many simulation studies for data processing relating ATLAS experiment at CERN, Geneva, Switzerland. For the description of the particle physics problem we cite [7] verbatim in Table 2 as follows:

Table 2. Problem formulation from the point of view of physics.

Identification of hadronic τ decays will be the key to the possible Higgs boson discovery in the wide range of the MSSM parameter space [1]. The $h/H/A \to \tau\tau$ and $H^{\pm} \to \tau\nu$ are promising channels in the mass range spanning from roughly 100 GeV to 800 GeV. The sensitivity increases with large $\tan\beta$ and decreases with rising mass of the Higgs boson. The $H \to \tau\tau$ decays will give access to the Standard Model and light Minimal Supersymmetric Standard Model Higgs boson observability around $m_H = 120$ GeV, with Higgs boson produced by vector-boson fusion [2]. The hadronic τ identification is also very important in searching for supersymmetric particles, particularly at high $\tan\beta$ values [3]. ... The same signal and background samples, as discussed in [4], are used to evaluate performance of the proposed methods. As signal, we consider reconstructed candidates from tau decays in $pp \to W \to \tau\nu$ and $pp \to Z \to \tau\tau$ events. As background, we consider candidates from QCD shower in the same $pp \to W \to \tau\nu, pp \to Z \to \tau\tau$ events and in QCD dijet events (sample with $p_T^{hard} > 35$ GeV). (Note that references relate to [7].)

The data set consists of 7 dimensional vectors of real numbers and class mark, which differentiates between signal samples (events) and background samples. The data set is split into learning and testing set, each of 3279 samples.

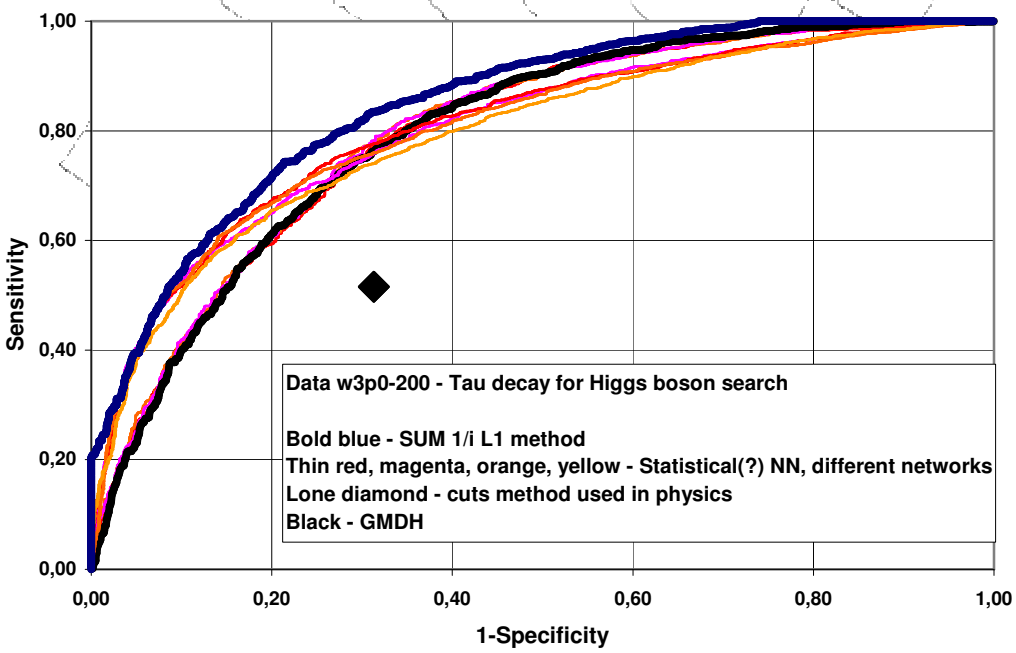


Figure 2. ROC curves for different separation/classification tools including the “cut” method.

In Fig. 2 well-known ROC curves are shown for different separation/classification tools including the “cut” method popular in physics studies.
The result obtained with “cuts” method is depicted by the black diamond.
The result obtained by GMDH-MIA algorithm is depicted by the lower bold black line
The results obtained by STATISTICA Neural Networks are depicted by two sets of red, magenta, orange and yellow lines. Each set corresponds to four best results out of ten networks generated. The set going more to the left at level 0.4 or 0.6 of sensitivity (signal acceptance) corresponds to its being set as a classifier; the other set (closer to the black line of GMDH-MIA) corresponds to its being set as an approximator.

The upper bold blue line was obtained by the IINC method described in this paper with L_1 metrics.

VI. Discussion

We have proved that the probability density approximation can be based on the Zipfian kernel. In the proof we have shown a close relation of the Zipfian distribution (and of the selected harmonic series as well) to the local fractal nature of data. Especially the use of $1/i$ has a close connection to the scaling exponent, eventually to correlation integral, and thus to the dynamics of the processes that generate data we wish to separate. We have shown, in fact, that the influence to the probability that point x is of class c is $1/i$ if the i-th nearest neighbor is of class c. We sum up these influences so that the sum goes over the indexes i for which the corresponding samples of the learning set are of class c. In the case of two classes we get two numbers S_1 and S_2 which together give the sum of N first terms of harmonic series $H_N = 1 + ½ + ⅓ + ¼ + \ldots + 1/N$. ($N$ is the size of the learning set.)

An interesting finding is that the method proposed here and the proof of the theorem uses the notion of distance but no explicit metrics need to be specified.

The method designed has no parameters to be tuned. There is also no problem with the convergence and the curse of dimensionality. The computational complexity grows at most linearly with the dimensionality and quadratically or less with the learning set size depending on the sorting algorithm used.

The main merit of the new method presented here is a new view on the data space. This view is based on a strange geometry with polynomially expanded distances in dependence on the local dimensionality of data denoted as the distribution mapping exponent. This leads to the use of reciprocals of the neighbor indexes and finally to the probability density estimation. The reciprocals of the neighbor indexes can be understood as "weights" of the learning set samples. It means, in fact, that the probability that the i-th neighbor and the given point are of the same class is given by the Zipfian distribution. In this context the Zipfian distribution gets a much broader role than its use in linguistics and psychology.

The other question is how the method presented here can be further improved. We suspect e.g. that data of one and the other class can be similarly distributed in the space even if data have different intrinsic dimensionality. Data often lie in clusters, which is a fact not tackled here. For given points outside these clusters or on boundaries of clusters the sum of reciprocals of the neighbor indexes of the opposite class may prevail, thus causing misclassification. This is a theme for further research in this field.

Acknowledgments

This work was supported by Technology Agency CR under project of series ALFA No. TA01010490 and by the Czech Technical University in Prague: RVO: 68407700. We also thank the Institute of Computer Science of the Czech Academy of Sciences for its support in submitting application of patent [9] for the classifier described.

References

[1] A. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository [online], [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science, 2011.

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

[3] Camastra, F.: Data dimensionality estimation methods: a survey. *Pattern recognition* Vol. 36 (2003), pp. 2945-2954.

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

[5] Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. *IEEE Transactions in Information Theory,* Vol. IT-13, No. 1, January 1967, pp. 23-27.

[6] Grassberger, P., Procaccia, I.: Measuring the Strangeness of Strange Attractors. *Physica* Vol. 9D (1983), pp. 189-208.

[7] Hakl, F., Jirina, M., Richter-Was, E.. Hadronic tau's identification using artificial neural network. ATLAS Physics Communication, ATL-COM-PHYS-2005-044, CERN, Geneve http://documents.cern.ch/cgibin/ setlink?base=atlnot&categ=Communication&id= com-phys-2005-044, (2005)

[8] Herbrich, R.: Learning Kernel Classifiers. Theory and Algorithms. The MIT Press, Cambridge, Mass., London, England, 2002.

[9] Jiřina, M., Jiřina, jr., M.: Apparatus for assessing a control value. Patent pending under number PV 2008-245; Z 7576 submitted on 22 April 2008 to the Industrial Property Office, Prague, Czech Republic.

[10] Jiřina, M., Jiřina jr., M. Correlation Integral Decomposition for Classification. In Artificial Neural Networks - ICANN 2008 Part II.. Berlin : Springer, 2008. S. 62-71. ISBN 978-3-540-87558-1. [ICANN 2008. International Conference on Artificial Neural Networks /18./, Prague, 03.09.2008-06.09.2008, CZ].
[11] Jiřina, M., Jiřina jr., M.: Utilization of Singularity Exponent in Nearest Neighbor Based Classifier. Journal of Classification. In print, 2012.
[12] Jiřina, M., Jiřina jr., M.: Classification by the Use of Decomposition of Correlation Integral. /Foundations of Computational Intelligence/. Berlin: Springer, 2009 - (Abraham, A.; Hassanien, A.; Snášel, V.) S. 39-55. ISBN 978-3-642-01535-9. - (Studies in Computational Intelligence. 205).
[13] Lucas, S. M., Algoval (2008). Algorithm Evaluation over the Web, \online], 2008, \cited November 23, 2008]. Available: _http://algoval.essex.ac.uk/data/vector/UCI/>
[14] Mandelbrot, B.B.: The Fractal Theory of Nature. W. H. Freeman and Co., New York, 1982.
[15] Maslov, V.P.: On a General Theorem of Set Theory Leading to the Gibbs, Bose–Einstein, and Pareto Distributions as well as to the Zipf–Mandelbrot Law for the Stock Market. Mathematical Notes, vol. 78, no. 6, 2005, pp. 807–813.
[16] Paredes, R. (2008). CPW: Class and Prototype Weights learning, \online], 2008, \cited November 23, 2008]. Available:_http://www.dsic.upv.es/~rparedes/research/ CPW/index.html>
[17] Paredes, R., Vidal, E. (2006). Learning Weighted Metrics to Minimize Nearest Neighbor Classification Error. IEEE Transactions on Pattern Analysis and M achine Intelligence, pp. 1100-1110, Vol. 20, No. 7
[18] Schmuland, B: Random Harmonic Series, American Mathematical Monthly 110, 407-416, May 2003.
[19] Schölkopf, B., Smola, A.J.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, Cambridge, Mass., London, England, 2002.
[20] Scott, D.W.: Multivariate Density Estimation. Theory, Practice, and Visualization. John Wiley and Sons, New York, 1992.
[21] Zipf, G.K.: The Psycho-Biology of Language. An Introduction to Dynamic Philology. The MIT Press, 1968. (Eventually: http://en.wikipedia.org/wiki/Zipf's_law)
[22] Zuo, W., Wang, K., Zhang, H., and Zhang, D.: Kernel Difference-Weighted k-Nearest Neighbors Classification In: D.-S. Huang, L. Heutte, and M. Loog (Eds.): ICIC 2007, LNAI 4682, pp. 861–870, 2007, Springer-Verlag Berlin Heidelberg 2007.