Exponentially Scaled Point Processes and Data Classification

Marcel Jiřina

DRAFT

***Abstract*—We use a measure for distances of neighbors' of a given point that is based on l_p metrics and a scaling exponent. We show that if the measure scales with scaling exponent mentioned, then distribution function of this measure converges to Erlang distribution. The scaling of distances is used for designing a classifier. Three variants of classifier are described. The local approach uses local value of scaling exponent. The global method uses the correlation dimension as the scaling exponent. In the IINC method indexes of neighbors of the query point are essential. Results of some experiments are shown and open problems of classification with scaling are dicussed.**

***Keywords*—Multivariate data, nearest neighbor, Erlang distribution, multifractal, scaling exponent, classification, IINC.**

I. Introduction

The goal of this study is to analyze the distances of nearest neighbors from given point (location) in a multidimensional spatial point process in R^d with exponential scaling [5]. The result is that when using scaled measure for distance of the k-th neighbor, the distance can have the Erlang distribution of order k. We show here that scaling leads to simple polynomial transformation $z = r^q$. A classifier can be designed with the use of this transformation.

In this paper we use a model involving an underlying process while some events occur in the process. We suppose that events occur randomly and independently of one another. The only information we have is d-dimensional data arising from events, i.e. by (often rather approximate) measurement or sampling.

Important notion is a scaling characterized by scaling exponent denoted also as fractal dimension. This dimension q is lesser than space dimension d, and usually is not an integer. The space dimension d is often called embedding dimension using concept that fractal is a q-dimensional formation plunged into a larger d-dimensional space [16]. This concept can be applied to volume V of a ball of radius r. There is $V = c_q r^q$ for q-dimensional ball in d-dimensional space; c_q is a constant dependent on q and the metrics used. Usually $q = d$ but the same holds for integer $q < d$, e.g. two dimensional circle in three dimensional Euclidean space. Keeping the concept consistent, q need not be an integer but there is no intuition how, say, 2.57-dimensional ball looks like.

The final version published: Proceedings of the 2014 International Conference on Pure Mathematics, applied Mathematics, Computational methods (PMAMCM2014), Santorini Island, Greece, July 17-21, 2014, pp. 179-186 (Paper MATH-27.pdf). ISBN paper proceedings: 978-1-60804-240-8, CD proceedings: 978-1-60804-245-3.

The work was supported by the Ministry of Education of the Czech Republic under INGO project No. LG 12020.

M. Jiřina is with the Institute of Computer Science AS CR, Pod Vodarenskou vezi 2, 182 07 Praha 8, Czech Republic (e-mail: marcel@cs.cas.cz)

II. Multidimensional point processes and fractal behavior

A. Point processes

Let there be an "underlying process" U_P. This process is sampled randomly and independently so that random d-dimensional data

$$P = x_1, x_2, \ldots, x_i \in X \subset R^d \tag{1}$$

arose. These data (without respect to time or order in which individual samples x_i was taken) form spatial point process in R^d and individual samples x_i are called points, in applications often events [6], samples, patterns etc.

We are interested in distances from one selected fixed point x to others; especially distance to the k-th nearest neighbor. From now we use numbering of points according to their order as neighbors of point x; x_k being the k-th nearest neighbor of point x. To distance l_k from x to its k-th nearest neighbor a probability is assigned. There is introduced

$$S_k(l) = Pr\{l < l_k\} = Pr\{N(l_k) < k\}$$

i.e. probability that a distance to the k-th nearest neighbor is larger than l that is equal to probability of finding k-1 points within distance l_k [4]. For k = 1 it is called avoidance probability and often denoted P_0. Function

$$F_k(l) = 1 - S_k(l)$$

is the distribution function of distance l to the k-th neighbor.

A scaling function is a real-valued function $c : R^d \rightarrow R_+$, that satisfies a self-similarity property with respect to a group of affine transformations [20]. There are several types of scaling functions, shifting, scaling, eventually reflections. General equation for scaling can have the form

$$\mu(\vec{x} + \vec{a}) = c_\theta(\vec{x})$$

and in a less general (fractal) case of exponential scaling

$$\mu(\vec{x} + \vec{a}) = a^{h(\vec{x})}$$

Here θ is l-dimensional parameter vector. When the scaling is location dependent, we speak about locally dependent point process.

B. Fractal behavior

We admit that an "underlying process" U_P shows exponentially scaled characteristics. Let there be data in R^d, see (1).

One can introduce a distance between two points of P using l_p metrics, l_{ij} = $||x_i$ - $x_j||_p$, $x_i, x_j \in P$. In a bounded region $W \in R^d$ a cumulative distribution function of l_{ij}

$$C_I(l) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N-1} h(l - l_{ij}) \quad ,$$

is denoted as correlation integral; $h(.)$ is the Heaviside step function. Grassberger and Procaccia [10] introduced correlation dimension ν as limit

$$\nu = \lim_{l\to 0} \frac{C_I(l)}{l} \quad .$$

Having empirical data on P, distances between any two points of P is the only information yielded exactly with the use of a relatively simple computation.

It is apparent that scaling of distances between any two points of P also holds for near neighbors' distances distribution. Let $F_k(l)$ be the distribution function of distance from some point x to the k-th neighbor. Let us define another function, the function $D(x,\ l)$ of neighbors' distances from one particular point x as follows [13], [14].

Definition

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

In bounded region $W \subset P$ when using a proper rescaling, the DMF is, in fact, a cumulative distribution function of distances from given location $x \in W \subset P$ to all other points of P in W. We call it also near neighbors' distance distribution function. We use $D(x,\ r)$ mostly in this sense. It is easily seen that DMF can be written in the form

$$D(x,l) = \lim_{N\to\infty} \frac{1}{N-1} \sum_{j=1}^{N-1} h(l - l_j).$$

The correlation integral can be decomposed into set of DMFs each corresponding to particular point $x_{0i} \in W \subset P$ as follows [14]

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

that means

$$C_I(l) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} D(x_{0i}, l) \quad .$$

Thus, the correlation integral is a mean of probability distribution mapping functions for all points of $W \subset P$.

We introduce a local scaling exponent q according to the following definition.

Definition

Let there be a positive q such that $\frac{D(x,l)}{l^q} \to const$ for $l \to 0+$. We call function

$$z(l) = l^q$$

a power approximation of the probability distribution mapping function and q is a distribution mapping exponent.

C. Common interesting behavior

It is common that measure $l(A)$ on R^d is usually a Lebesgue measure or based on it. Thus $l(A)$ depends on integer dimensionality d. Our intention is to deal with some q, $d \geq q > 0$ not necessary an integer.

Here we contract metric space $(X,\ \rho)$ to $(R^d,\ l_p)$, where l_p is Lebesgue p-norm. Let $q \in (0,\ d]$. We define measure $\mu(.)$ of neighbors distances so that for S = (a line between x_i and x_j) there is $\mu(S) = l_p^q(x_i - x) - l_p^q(x_j - x)$, $l_p(x_i - x) \geq l_p(x_j - x)$, $\mu(O) = 0$, $\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2)$; $S_1 \cap S_2 = O \quad a.s.$

It is easily seen that $\mu(.)$ is a measure; it is nonnegative, it equals zero for the empty set and for $x_i = x_j$, and is countable additive.

Then the theorem that is a special and useful case of more general results about point processes [4], [5], [20] holds.

Theorem 1: Let there be a point process P and bounded region $x \subset R^d$, where there is given point x and Nnearest neighbors of x. Let $D(x, l)$ scales with exponent q. Let process P in bounded region $W \subset R^d$ be mapped (by mapping M_{Ppx}) to process p in bounded interval $w \subset R+$. Then one-dimensional point process p in $w \subset R+$ is a homogenous Poisson process with intensity $\lambda = \lim_{N\to\infty} N/z_N$.

Proof: It is omitted here.

Theorem 1 can be applied to all points $x_0 \in P$. Supposing monofractal underlying process U_P and by point process P induced measure $\mu_{p\nu}(.)$ with correlation dimension ν as one of its parameters, the ν scales also the DMF of all points of P and then q = ν.

Corollary 1: Let there be a point process P and bounded region W, where there are given location x and N nearest neighbors of x. Let DMF $D(x, l)$ scale with exponent q. Then probability distribution of μ_k = $\mu_{pq}(x_k - x)$ of the k-th nearest neighbor x_k of the given location x is Erlang distribution Erl($\mu_k,\ k,\ \lambda$), i.e.

$$F(\mu_k) = 1 - \exp(-\lambda\mu_k) \sum_{j=0}^{k-1} \frac{(\lambda\mu_k)^j}{j!}$$

$$f(\mu_k) = \frac{\lambda^k}{k!} (\mu_k)^{k-1} \exp(-\lambda\mu_k) \quad .$$

Proof: It is omitted here.

We have found that when one can find a scaling of neighbors' distances measure, in the form $z = r^q$, q is the distribution mapping exponent, then one can find a "Poisson process-like" behavior, i.e. Erlang distribution of neighbors' distances measure. Usually, a measure is considered that may depend on the embedding space dimension d (integer), while we use more general distribution mapping exponent q that is a positive real number.

III. Classification using scaling

Here we present the basic idea of multidimensional data classification using scaling and three variants of this approach.

A. Data

Let the learning set U of total N samples be given. Each sample $x_t = \{x_{t1}, x_{t2}, x_{td}\}; t = 1, 2, ...N, x_{tk} \in R; k = 1, 2, ..., d$ corresponds to a point in d-dimensional metric space M_d, where d is the sample space dimension. For each $x_t \in U$ a class function $T : R^d \rightarrow \{1, 2, ...C\} : T(x_t) = c$ is introduced. With the class function the learning set U is decomposed into disjoint classes $U_c = \{x_t \in U | T(x_t) = c\}; c \in \{1, 2, ..., C\}, U = \bigcup_{c=1}^{C} U_c, \quad U_c \cap U_b = \emptyset; c, b \in 1, 2, ..., C; c \neq b$. Let the cardinality of set U_c be N_c. 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 $r_i < r_j; i, j = 1, 2, ...N$, and class $U_c = \{x_i \in U | T(x_i) = c\}$. Of course, the ranking depends on point x and eventually metrics of M_d. We use Euclidean (L_2) and absolute (Manhattan, L_1) metrics here. In the following indexing by i means ranking just introduced.

B. The DME method

This classifier uses the distribution mapping exponent already introduced.

1) Intuitive explanation: 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 $c = 0, 1$ is the class mark. Suppose that this contribution is the 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 and there is $p(c|x, 1) \simeq \frac{1}{S_q r_1^q}$, where we use the distribution mapping exponent q instead of the data space dimensionality d; S_q is proportionality constant dependent on the dimensionality and metrics used. For the second point i = 2 there is $p(c|x, 2) \simeq \frac{1}{S_q r_2^q}$ ... And so on; generally for point No. i $p(c|x, i) \simeq \frac{1}{S_q r_i^q}$.
We add the partial contributions of individual points together by summing up

$$p(c|x) \simeq \sum_{x_i \in U_c} p(c|x, i) = \frac{1}{S_q} \sum_{x_i \in U_c} 1/r_i^q$$

(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 $p(0|x) + p(1|x) = 1$ and from it $S_q = \sum_{i=1}^{N} 1/r_i^q$ Thus, we get the form suitable for practical computation

$$\hat{p}(c|x) = \frac{\sum_{x_i \in U_c} 1/r_i^q}{\sum_{i=1}^{N} 1/r_i^q} \tag{2}$$

(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 ($i = 1$) usually. It can be found that its influence is more negative than positive on the probability estimate here.

2) Theory: Here we proceed from the assumption that the best approximation of the probability distribution of the data is closely related to the uniformity of the data space around the query point x. In cases of uniform distribution - at least in the neighborhood of the query point - the best results are usually obtained. Therefore, we approximate (polynomially expand) the true distribution so that at least in the neighborhood of the query point the distribution density mapping function appears to be constant.

Now a question arises why influences of individual points of a given class to the final probability that point x is of the class are inversely proportional to the $z = r_i^q$. Let there be Z, the largest of all z for a given class. We have shown that variable $z = r^q$ has uniform distribution with some density p_z. It holds that $Zp_z = 1$ because the integral of the distribution density function over its support $(0, Z)$ equals one. If support would be $(0, Z_1), Z_1 < Z$, then the density must be larger proportionally to Z/Z_1. It means that shift of each point closer to point x will enlarge the density so that it will be inversely proportional to the distance of a point from point x.

Theorem 2: Let the task of classification into two classes be a given. Let the size of the learning set be N and let both classes have the same number of samples. Let q , $1 < q < d$ be the distribution mapping exponent, let i be the index of the i-th nearest neighbor of point x (without respect to class), and $r_i > 0$ its distance from point x. Then,

$$p(c|x) = \lim_{N \rightarrow \infty} \frac{\sum_{x_i \in U_c} 1/r_i^q}{\sum_{i=1}^{N} 1/r_i^q} \tag{3}$$

(the upper sum goes for all points of class c only) is probability that point x belongs to class c.

Proof: can be found in [13]

3) Generalization: Up to now we reckoned with two classes only and the same number of samples of both classes in the learning set. Formula (3) must be completed for general number of C classes and the different number of the samples $N_1, N_2, ...N_C$ of individual classes. In fact, the latter is only a recalculation of the relative representation of the different number of the samples in classes.

$$p(c|x) = \frac{\lim\limits_{N\to\infty}(1/N_c \sum\limits_{x_i\in U_c} 1/r_i^q)}{\sum\limits_{k=1}^{C}\lim\limits_{N\to\infty}(1/N_k \sum\limits_{x_i\in U_k} 1/r_i^q)} \quad (4)$$

4) The DME classifier construction: This method represents a direct use of formula (2), eventually formula (4) in the form

$$\hat{p}(c|x) = \frac{1/N_c \sum\limits_{x_i\in U_c} 1/r_i^q}{\sum\limits_{k=1}^{C}(1/N_k \sum\limits_{x_i\in U_k} 1/r_i^q)} \quad (5)$$

Note that the convergence of sums above is faster the larger DME q is. Usually, for multivariate real-life data the DME is also large (and the correlation dimension as well). Figs. 1 and 2 illustrate the convergence of the sum in the numerator above for one query point for the well-known "vote" data, see [1]. The task is to find whether a president elected will be republican or democrat. The data are 15-dimensional of two classes that have a different number of samples. In the learning set, there are 116 times republican and 184 times democrat. The distribution mapping exponent q varies between 4.52 and 14 with the mean value 10.22.

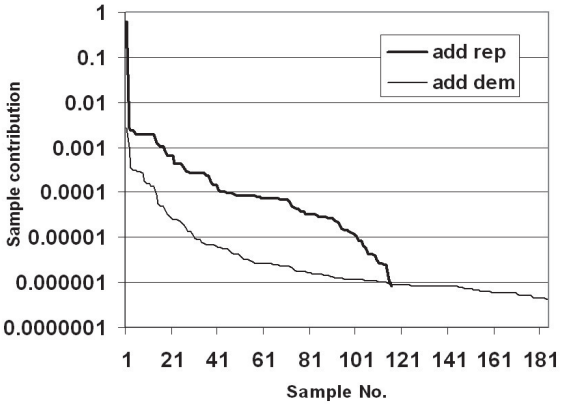

Fig. 1. Sample contribution to the sum in the numerator of (5) for the 15 dimensional data vote and one particular query point; $q = 7.22$. The upper line corresponds to the republican, the lower line to the democrat. Samples are sorted according to the distance r, i.e. also to the size of the sample contribution to the sum for one class. There are different numbers of samples of one and the other class in the learning set.

The classification procedure is rather straightforward. First, compute the distribution mapping exponent q for the query point x by standard linear regression, see the next section. Then, we simply sum up all the components excluding the nearest point.

In our approach, a true distribution is mapped to the uniform distribution. For uniform distribution, it holds that the i-th neighbor distance from a given point has an Erlang distribution of i-th order. For an Erlang distribution of i-th order, the relative statistical deviation, i.e. the statistical deviation divided by the mean, is equal to $1/\sqrt{i}$. Then, the relative statistical

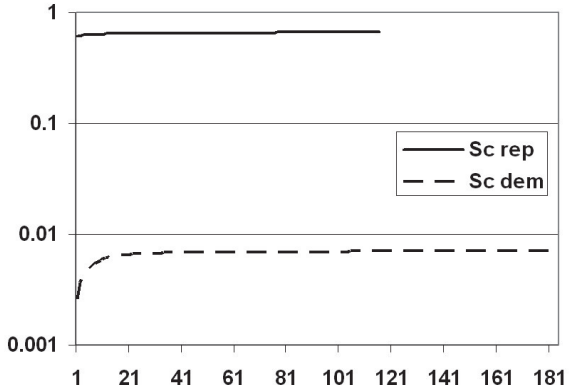

Fig. 2. The size of the total sum in the numerator of (5) for the 15-dimensional data "vote" and one particular query point; q = 7.22. The upper line corresponds to the republican, the lower line to the democrat. The samples are sorted according to the distance r, i.e. also to the size of the sample contribution to the sum for one class.

deviation diminishes with the index of the neighbor and for the nearest neighbor is equal to 1, which also follows from the fact that the Erlang(1) distribution is exponential distribution. So, there is a large relative spread in the positions of the nearest neighbor and, at the same time, its influence is the largest. In practice, it seems better to eliminate the influence of the first nearest neighbor. Theorems for DME as well for the CD method remain valid.

This is made for classes, simultaneously getting C sums for all classes. Then, we can get the Bayes ratio or a probability estimate that point x belongs to class. The class that has largest probability estimate is taken as an estimated class of query point x. Eventually these probabilities can be weighted in the same way as in other classifiers.

5) Distribution mapping exponent estimation: An important issue of this method is the procedure how to determine the distribution mapping exponent.

To estimate the distribution mapping exponent q we use a similar approach, nearly identical, to the approach of Grassberger and Procaccia [10] for the correlation dimension estimation.

This is the task of estimating the slope of a straight line linearly approximating the graph of the dependence of the neighbor's index as a function of distance in log-log scale. Grassberger and Procaccia [10] proposed a solution by linear regression. Dvorak and Klaschka [7], Guerrero and Smith [11], Osborne and Provenzale [18] later proposed different modifications and heuristics. Many of these approaches and heuristics can be used for the distribution mapping exponent estimation, e.g. use of the square root of N_c nearest neighbors instead of N_c to eliminate the influence of a limited number of the points of the learning set. 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.

The approach described here has two other variants.

C. CD method - correlation dimension based approach

In this method it is supposed that distribution mapping exponents for individual query points differ only slightly and that one can use the value of correlation dimension ν instead. Computation has then two steps, in the first step the estimate of correlation dimension ν is computed using any known suitable method, and then one uses formulas (2) or (5), where ν instead of q is used.

Again, as in Section III-B, we exclude the first nearest neighbor of the query point. The convergence of sums is equally fast as in the DME method.

A relative advantage of this approach is that the estimate of the correlation dimension is more exact than the estimate of the distribution mapping exponent and that computation of the correlation dimension is done only once unlike the DME that must be computed for each query point anew.

1) Correlation dimension estimation: For the approximation of probability of class at a given point and classification described above, a fast and reliable method for correlation dimension estimation is needed. Methods for the estimation of correlation dimension differ by approaches used and also by some kind of heuristics that usually optimize the size of radius r to get a realistic estimation of correlation dimension [17], [3], [25], as mentioned above.

Averaging method

The basic problem of correlation dimension estimation is the large number of pairs that arise even for a moderate learning set size. The idea of the correlation dimension estimation described below is based on the observation that distances between all pairs of points can be divided into groups, each group associated with one (fixed) point of the learning set.

Theorem 3: Let there be a learning set of N points (samples). Let the correlation integral be $C_I(r)$ and let $D(x,r)$ be the distribution mapping function corresponding to point x. Then, $C_I(r)$ is a mean of $D(x,r)$ for all points of U

Proof: For proof see [15].

We have found that for sufficiently good estimation of the correlation dimension one can use part of the data set only, for each point to estimate the distribution mapping exponent, and take the average. The part of the data set may be some number of points randomly selected from the data set. It suffices to use 100 points. The method of averaging need not be limited to the Grassberger-Procaccia algorithm. We use it analogically for Takens' algorithm [25] as well.

D. IINC method - the inverted indexes of neighbors classifier

1) Intuitive basis: In a similar way as in Section III-B1 let us assume 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 1/2, the influence of the third nearest neighbor is 1/3 etc. Again we add the partial influences of individual points together by summing up

$$\hat{p}(c|x) = \sum_{x_i \in U_c} p_1(c|x, r_i) = K \sum_{x_i \in U_c} 1/i.$$

The sum goes over indexes i for which the corresponding samples of the learning set are of class c. The estimation of the probability that the query point x belongs to class c is

$$\hat{p}(c|x) = \frac{\sum\limits_{x_i \in U_c} 1/i}{\sum\limits_{i=1}^{N} 1/i}.$$

In the denominator is the so-called harmonic number H_N, the sum of truncated harmonic series. The hypothesis above is equivalent to the assumption that the influence of individual points of the learning set is governed by the Zipfian distribution (Zipf's law) [27], [23]. There is an interesting fact that the use of $1/i$ has a close connection to the correlation integral and correlation dimension and, thus, to the dynamics and true data dimensionality of processes that generate the data we wish to separate.

2) Theory:

Theorem 4: Let the task of classification into two classes be given. Let the size of the learning set be N and let both classes have the same number of samples. Let i be the index of the i-th nearest neighbor of point x (without considering the neighbor's class) and r_i be its distance from point x. Then,

$$p(c|x) = \lim_{N \to \infty} \frac{\sum\limits_{x_i \in U_c} 1/i}{\sum\limits_{i=1}^{N} 1/i} \tag{6}$$

(the upper sum goes over indexes i for which the corresponding samples are of class c) is the probability that point x belongs to class c.

Proof: For proof see [15].

In the formula above it is seen that the approach is, in the end a kernel approach with rather strange kernel as compared with the kernels usually used [12], [22].

It is easily seen that

$$\sum_{c=1}^{C} p(c|x) = \sum_{c=1}^{C} \lim_{N \to \infty} \frac{\sum\limits_{x_i \in U_c} 1/i}{H_N} = 1$$

and $p(c|x)$ is a "sum of relative frequencies of occurrence" of points of a given class c. A "relative frequencies of occurrence" of point i , i.e. of the i-th neighbor of query point x, is

$$f(i; 1, N) = \frac{1/i}{H_N}$$

In fact, $f(i; 1, N)$ is a probability mass function of the Zipfian distribution (Zipf's law). In our case, $p(c|x)$ is a sum of probability mass functions for all appearances of class c. Theorem 4 above was formulated from these considerations.

3) The Classifier construction: Let samples of the learning set (i.e. all samples regardless of the class) be sorted according to their distances from the query point x. Let indexes be assigned to these points so that 1 is assigned to the nearest neighbor, 2 to the second nearest neighbor etc. This sorting is an important difference to both methods described before

that need no sorting when distribution mapping exponent or cortrelation dimension are known. Let us compute sums $S_0(x) = \frac{1}{N_0} \sum_{x_i \in U_0} 1/i$ and $S_1(x) = \frac{1}{N_1} \sum_{x_i \in U_1} 1/i$, i.e. the sums of the reciprocals of the indexes of samples from class $c = 0$ and from class $c = 1$. N_0 and N_1 are the numbers of samples of class 0 and class 1, respectively, $N_0 + N_1 = N$. The probability that point x belongs to class 0 is

$$\hat{p}(c = 0|x) = \frac{S_0(x)}{S_0(x) + S_1(x)} \tag{7}$$

and, similarly, the probability that point x belongs to class 1 is

$$\hat{p}(c = 1|x) = \frac{S_1(x)}{S_0(x) + S_1(x)} \tag{8}$$

When some discriminant threshold θ is chosen, then if $\hat{p}(c = 1|x) > \theta$ point x is of class 1 else it is of class 0. This is the same procedure as in other classification approaches where the output is an estimation of probability (naive Bayes) or any real valued variable (neural networks). The value of threshold can be optimized with respect to minimal classification error. The default value of the discriminant threshold here is $\theta = 0.5$.

4) Generalization: The formulas above hold for two class problem with equal number of samples of both classes in the learning set. For larger number of classes and a different number of samples of classes the formula has the form similar to (5):

$$\hat{p}(c|x) = \frac{1/N_c \sum_{x_i \in U_c} 1/i}{\sum_{k=1}^{C} (1/N_k \sum_{x_i \in U_k} 1/i)} \tag{9}$$

It is only a recalculation of the relative representation of different numbers of samples of one and the other class. For classification into more than two classes we use this formula for all classes and we assign to the query point x a class c for which $\hat{p}(c|x)$ is the largest.

IV. Experiments

We demonstrate the features and the power of the classifier both on synthetic and real-life data.

A. Synthetic Data

Synthetic data according to Paredes and Vidal [19] are two-dimensional and consist of three two-dimensional normal distributions with identical a-priori probabilities. If μ denotes the vector of the means and C_m is the covariance matrix, there is

Class $A : \mu = (2, 0.5)^t, C_m = (1, 0; 0, 1)$ (identity matrix)
Class $B : \mu = (0, 2)^t, C_m = (1, 0.5; 0.5, 1)$
Class $C : \mu = (0, -1)^t, C_m = (1, -0.5; -0.5, 1)$.

Fig. 3 shows the results obtained by the different methods for the different learning sets sizes from 8 to 256 samples and a testing set of 5000 samples, all from the same distributions and all independent. Each point in the figure was obtained by averaging over 100 different runs. 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 to the Bayes limit.

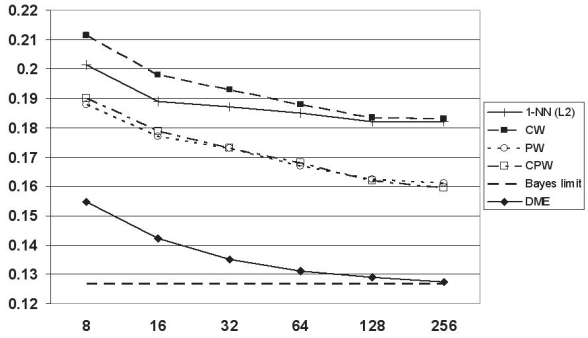

Fig. 3. Comparison of the classification errors of the synthetic data for the different approaches in dependence on the size of the learning set. In the legend, 1-NN(L2) means the 1-NN method with Euclidean metrics, CW, PW, and CPW are three variants of the method by Paredes and Vidal [19]; the points are estimated from this reference. Bayes means the Bayes limit, DME means the basic method presented here.

Note that in this test, the error of the DME estimation is combined with numerical errors, and with a negative influence of the low number of the samples giving the results presented in Fig. 3.

B. Data from Machine Learning Repository

The classification ability of the algorithm (DME) was tested using real-life tasks from the UCI Machine Learning Repository; see [1]. Seven databases have been used for the classification task, see Table 1.

TABLE 1. Classification mean square errors for four different methods including DME.

Data set	Attributes	Cross validation	$\sqrt{N}$ -NN	1-NN	Bayes	DME
Mushroom	22	1	0.0207	0	0.00764	0
Shuttle (Statlog)	9	1	0.00828	0.00259	0.01294	0.00207
Iris (see Friedman, 1994)	4	10	0.0488	0.0609	0.0854	0.0488
Congressional Voting ("Vote")	16	1	0.0602	0.1053	0.0977	0.0752
Spambase	57	1	0.113	0.0997	0.143	0.0886
Heart (Statlog)	13	9	0.158	0.245	0.182	0.166
Molecular Biology (Splice)	61	10	0.372	0.404	0.287	0.297

For the Shuttle data, the learning and testing sets are directly at hand and were used as they are. For smaller data sets a

cross validation of 10 or 9 was used. The Iris data set was modified into a two-class problem excluding the iris-setoza class according to Friedman [9]. The methods for comparison are

- 1-NN - standard nearest neighbor method
- Sqrt-NN - the k-NN method with k equal to the square root of the number of samples of the learning set
- Bayes - the naive Bayes method using ten bins histograms

For the k-NN, Bayes, and our method the discriminant thresholds were tuned accordingly. The testing shows the classification ability of the DME method for some tasks compared to the other published methods and results for the same data sets.

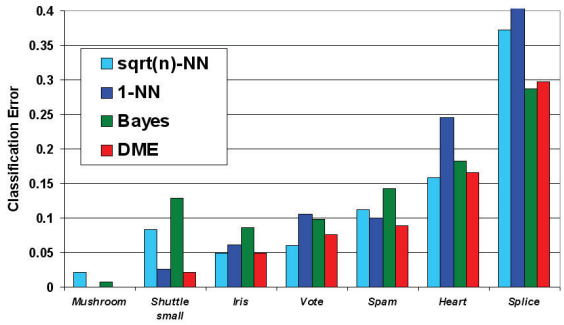

Fig. 4. Comparison of the classification errors for four different methods including the DME. Note that for the Mushroom data, both the 1-NN and DME algorithms, give zero error. For the Shuttle data, the errors are ten times enlarged.

V. Open problems

We have shown the use of scaling for data classification. The three classifiers presented here were tested and can be used as they are similar to the one that sometimes uses the 1-NN or k-NN methods. On the other hand, preprocessing like data editing or some kind of learning may essentially enhance classifier‘s behavior.

A. Editing

This is a way of learning data set modification that tries to enhance especially borders between classes to make class recognition easier. The original idea of editing (or preclassification) [26] is to classify a sample of the learning set by the method for which edited learning data will be used. If classification result does not correspond to the sample class, remove this sample from the learning set. After this is done, use the edited learning set for data classification by the standard way. There are other ingenious methods that modify originally simple methods with the help of learning, e.g. the learning weigting method [19] modifies the learning set by weighting classes and features, and then uses simple 1-NN method similarly as [26].

B. Crossing phenomenon

The basic notion used here is the distribution mapping function. Depicted in the log-log coordinates, it is approximately linearly growing function. When there are two classes we may have two such lines in one graph for a point x. If one line lies under the other, point x belongs to class of the lower line. But what if the lines cross? And is the crossing point an essential issue?

C. Scaled point processes but not exactly exponentially

The exponential scaling used here is a special case of more complex scaling functions. Transformation $z = r^q$ may have another form, depending on the scaling function used. The main problem is scaling function identification [20], [21]. One can suppose that the use of more realistic scaling function than exponential may lead to modification of the methods presented here and to improving their behavior.

VI. Discussion

We have found that when one can find a scaling of neighbors' distances measure, in the form $z = r^q$, q is the distribution mapping exponent, then one can find a "Poisson process-like" behavior, i.e. the Erlang distribution of neighbors' distances measure. Usually, a measure is considered that may depend on the embedding space dimension d (integer), while we use more general distribution mapping exponent q that is a positive real number.

Because the Erlang distribution converges to the Gaussian distribution for index $k \to \infty$, the result according to Theorem 1 also relates to some results of e.g. [2], [8], [24] about convergence of near-neighbor distances.

The correlation dimension, eventually multifractal dimension, singularity (or Hölder) exponent or singularity strength, is often used for characterization of one dimensional or two-dimensional data, i.e. for signals and pictures. Our results are valid for multidimensional data that need not form a series because, in this respect, data are considered as individual points in a multidimensional space with proper metrics.

Our model of the polynomial expansion of the data space comes from the demand to have a uniform distribution of points, at least locally. 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 is used, the computational procedure is almost the same in both cases. Moreover, it can be found that the values of the distribution mapping exponent lie sometimes in a narrow, sometimes in a rather wide interval around its mean value. Not surprisingly, the mean value of the distribution mapping exponent over all samples is not far from the correlation dimension. Introducing the notion of the distribution mapping exponent and the polynomial expansion of the distances may be a starting point for a more detailed description of the local behavior of the multivariate data and for the development of new approaches to data analysis, including classification problems.

Our experiments demonstrate that the simplest classifier based on the ideas introduced here can outperform other methods for some data sets. In all, the tasks presented here, the distribution-mapping-exponent-based method outperforms or is comparable to the 1-NN algorithm and in six of the seven tasks outperforms naive Bayes algorithm being only slightly worse for the Splice data. All of these comparisons include an uncertainty in the computation of the distribution mapping exponent. By the use of the notion of distance, i.e. a simple transformation $E_n \rightarrow E_1$, the problems with dimensionality are easily eliminated at the loss of information on the true distribution of the points in the neighborhood of the query point, which does not seem to be fundamental.

VII. Conclusion

This work was motivated by the observation that near neighbors distances in homogenous Poisson processes in R^d have, in fact, the Erlang distribution modified so that independent variable is substituted by term Kr^d, where K is a constant, r the distance of the neighbor and d the space dimension. This is the scaling function in exponential form. Here we answer the question, what if point process has arisen from underlying process with scaling exponent smaller than space dimension d.

This problem is solved by introducing of a distribution mapping function and its power approximation. It has been shown that the distribution mapping exponent of the power approximation is very close to the scaling exponent known from the theory of fractals and multifractals. When simplified, it leads, in the end, to a strange scale measured by scaling exponent-power of neighbors' distances. It was then found that when using thus scaled measure for distance of the k-th neighbor one can construct simple and effective classifier; we have presented here three of its variants and discussed some open problems.

References

[1] Bache, K., Lichman, M.(2013) UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science, [online], 2013. Available: ¡http://archive.ics.uci.edu/ml¿.

[2] Bonetti M, Pagano M. (2005) The interpoint distance distribution as a descriptor of point patterns, with an application to spatial disease clustering. *Stat. Med.* Vol. 24, No. 5, pp. 753-773.

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

[4] Daley, D. J., Vere-Jones, D. (2005) *An Introduction to the Theory of Point Processes. Volume I, Elementary theory and methods. Second edition*, Springer.

[5] Daley, D.J., Verre-Jones, D. (2008) *An Introduction to the Theory of Point Processes. Volume II, General Theory and Structure. Second Edition*, Springer.

[6] Diggle, P.J. (2003) *A statistical Analysis of Spatial Point Processes*. Arnold, London.

[7] Dvorak,I., Klaschka,J. (1990) Modification of the Grassberger-Procaccia algorithm for estimating the correlation exponent of chaotic systems with high embedding dimension. *Physics Letters A,* Vol. 145, No. 5, pp. 225-231.

[8] Evans, D. (2008) A law of large numbers for nearest neighbour statistics. *Proc. R. Soc. A.* Vol. 464, pp. 3175–3192.

[9] Friedmann,J. H. (1994) Flexible Metric Nearest Neighbor Classification. *Technical Report, Dept. of Statistics, Stanford University*, 32 p.

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

[11] Guerrero, A., Smith, L.A. (2003) Towards coherent estimation of correlation dimension. *Physics letters A*, Vol. 318, pp. 373-379.

[12] Herbrich, R. (2002) *Learning Kernel Classifiers. Theory and Algorithms.* The MIT Press, Cambridge, Mass., London, England.

[13] Jiřina, M. (2013) Utilization of singularity exponent in nearest neighbor based classifier. *Journal of Classification (Springer)* Vol. 30, No. 1, pp. 3-29.

[14] Jiřina, M. (2014) Correlation Dimension-Based Classifier. *IEEE Transaction on Cybernetics* Vol. 44, in print.

[15] Jiřina, M., Jiřina, M., jr. (2014) Classification Using the Zipfian Kernel. *Journal of Classification (Springer).* Vol. 31, in print.

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

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

[18] Osborne,A. R., Provenzale,A.(1989) Finite correlation dimension for stochastic systems with power-law spectra. *Physica* D, Vol. 35, pp. 357-381, (1989).

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

[20] Prokešovà, M., Hahn, U., and Jensen, E. B. V. (2006) Statistics for locally scaled point processes. *In Baddeley, A., Gregori, P., Mateu, J., Stoica, R., and Stoyan, D. (editors), Case Studies in Spatial Point Process Modelling, Lecture Notes in Statistics*, Springer, New York, Vol. 185, pp. 99-123.

[21] Prokešovà, M. (2010) Inhomogeneity in spatisal Cox point processes - location dependent thinning is not the only option. *Image Anal Stereol* Vol.29, pp. 133-141.

[22] Schlkopf, B., Smola, A.J. (2002) *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, Cambridge, Mass., London, England.

[23] Schmuland, B.(2003) Random Harmonic Series. *American Mathematical Monthly* Vol 110, pp. 407-416, May 2003.

[24] Silverman, B. W. (1976) Limit theorems for dissociated random variables. *Advances in Applied Probability*, Vol. 8, pp.806-819.

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

[26] Wilson,D.L. (1972) Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans on System Man and Cybernetics*, Vol. SMC-2, No. 3, pp. 408-421. (July 1972)

[27] Zipf, G.K.(1968) The Psycho-Biology of Language. *An Introduction to Dynamic Philology* . The MIT Press, 1968.