UDC 519.95

# BINARY DECISION TREE SYNTHESIS: SPLITTING CRITERIA AND THE ALGORITHM LISTBB

© V. I. Donskoy

Tavrida National University
Mathematics and Computer Science Department
Vernadsky av., 4, Simferopol, ARC, 95007, Ukraine
e-mail: *donskoy@tnu.crimea.ua*

**Abstract**. In our days, interest to the class of inductors on the basis of decision trees does not weaken, especially in the context of Data Mining paradigm . At the same time most widespread Quinlan algorithms ID3 and C4.5, as we show in the paper, are not the best. It is therefore possible to see the successful attempts of creation another heuristic splitting criteria for the algorithms of synthesis of decision trees. Comparative definition of different splitting criteria used for the synthesis of binary decision trees is the purpose of the paper. We included the criteria $D$, $\Omega$, $Z_1$ and other which were developed by the author yet at 1979-80 years. These criteria define combined splitting principle which is used in the algorithm LISTBB.

## Introduction

The idea to use decisions trees for machine learning and recognition appeared in the articles of Hunt and Hoveland at the end of 50th past century. But the central work came into notice of mathematicians and programmers to this scientific direction all over the world there was the book of Hunt, Marine, and Stone published in 1966 [8]. In the Soviet union the scientific direction related to the decision trees began to develop approximately at the same time at A. Blokh [25] scientific school. From numerous works of this school it is necessary to pay the special attention to the paper of V. Orlov [39]. In this Orlov's paper, yet at the beginning of 70th last century — more than on 10 years before J. R. Quinlan — an entropy splitting criterion and the algorithm for decision tree synthesis was presented, which on principle did not differ from the widely in-use algorithm ID3.

In our days, interest to the class of inductors on the basis of decision trees does not weaken, especially in the context of paradigm of Data Mining. At the same time most widespread Quinlan's algorithms ID3 and C4.5 , as possible to see below, are not the best. It is therefore possible to find out the successful attempts of creation another heuristic algorithms for synthesis decision trees by precedent information [15, 14].

The aim of the present paper is comparative description of the different splitting criteria used for the synthesis of binary decision trees (BDT), including the criteria developed by the author yet in 1979-80 years which underlay the algorithm LISTBB.

The name used for the algorithm LISTBB is explained to those, that it was first realized on the basis of list presentation (LIST); branching (B) — designate splitting, and second B — designate the case of Boolean variables.

Algorithm LISTBB and its modifications LISTD and LISTBB(P) were repeatedly used in practical tasks. These algorithms were used for development programm systems RADIUS-222, TRIOL, INTMAN [31, 34, 33]. The main feature of algorithm LISTBB consists of that he is "sharpened" exactly to minimization of the number of leaves of BDT inductor. Such approach gives results better then another approaches to splitting in the average (by a set of the tests).

BDT Synthesis, general speaking, consists of two stages: a) choice of feature predicates and b) decision tree construction. These stages can be joined as it used when decision tree corresponds to the partition of real feature space by hyper parallelepipedes. We will further suggest that two stage approach to the BDT synthesis is used, the set of feature predicates is given, and logical sample table is given as well.

Each inner node of BDT corresponds some fixed feature predicate. Each inner node has two outgoing edges which corresponds "zero" and "one" values of this fixed feature predicate. Any branch of BDT has no the same predicates in it nodes and ends by the leave marked by the class number. Recognizing algorithm which defined by the BDT uses this class to identify all the objects (points of the feature space) hits into partition element according to the branch.

It is well known that the number of the inner nodes of BDT is equal to $\mu-1$, where $\mu$ is the leaves number. Then minimization of the leaves number is equal to minimization of the inner nodes number or the number of the tests executed in the inner nodes.

The length of the branch is the number of the nodes contained in this branch. The height of BDT is the length of its branch with the most nodes number. A tree is called uniform (balanced) if all its branches has equal length.

We will identify $n$ feature predicates given for BDT synthesis with the Boolean variables $x_1, ..., x_n$.

The class of the Boolean function which is representable by BDT is complete: by mean some BDT the algorithm for realization any Boolean function can be realized. This important property can be easy proved by consecutive Shannon expansion by one variable. But the class of Boolean functions defined by BDT with the number of leaves bounded by the constant $\mu$ is enough narrow [32].

The expansion by $r$ variables along any BDT branch defines the interval of the rank $r$ in the partition of the set $B^n$ of the vertexes of unit n-dimensional cube. Any element of this partition is marked by the class number contained in the corresponded BDT

branch leave. We can say that BDT classifies the intervals of the partition. The codes of the interval is the set of values of predicates, placed in the inner nodes of the branch. Dimension of the interval of rank $r$ is $n-r$ and such interval contains $2^{n-r}$ points.

When we consider the branching process as the sequential partition of $B^n$ to intervals we use the set-theoretic approach in Boolean algebra defined by Yu. I. Zhuravlev [36]. This very fruitful approach stimulated development of the splitting criteria based on the concept of separability [37] presented in this paper. BDT synthesis with minimum leaves number is equal to synthesis of shortest orthogonal covering which is correct relatively the sample points distribution by partition intervals.

The leaves number $\mu$ of BDT is natural measure of its comlexity because the number of the inner nodes $\mu-1$ defines the number of the same type executable steps in process of "steady raising" synthesized BDT.

Let $q$ be number of the classes; $\mathscr{D}(n,q,\mu)$ be the family of BDT with exactly $\mu$ leaves. The exact formula for the number $d(n,q,\mu)=|\mathscr{D}(n,q,\mu)|$ is unknown. Arbitrary Boolean function is presented by BDT, generally speaking, not uniquely.

In the paper [32] the asymptotic estimation on condition that $n\to\infty$ is obtained:

$$d(n,q,\mu)\sim(\mu-1)![q(q-1)]^{\mu-1}n(n-1)^{\mu-2},$$

and it is proved that the number $b(n,2,\mu)$ of Boolean functions which can be presented by BDT with exactly $\mu$ leaves satisfy inequality

$$b(n,2,\mu)<(\mu-1)!2^{\mu-1}n^{\mu-1}.$$

Use of the pVCD method [29, 30] allows to find Vapnik-Chervonenkis Dimension (VCD) of the finite class $\mathscr{B}(n,2,\mu)$ of decision functions presentable by BDT with the leaves number not exceeding $\mu$ in the case of two classes [4]:

$$VCD(\mathscr{B}(n,2,\mu))<(\mu-1)(\log(n+1)+\log\mu+1). \tag{1}$$

## 1. Estimation methods for Decision Trees as Empirical Inductors

Machine Learning by sample (by precedents), we speak about in our paper, realizes empiric induction principle which consists in synthesis of decision by mean of generalization of particular cases to their common features. We consider such case when common features, which is found as result of machine learning, is represented as the set of concepts (by E. Hunt). These concepts are presented in the form of BDT. Conditional features or concepts are the conjunctions which corresponds to branches of the BDT. These conjunctions define the set of decision Boolean functions. Notice, machine learning must be organized such the way that common features were true on the as most as possible

examples which were not used to correct BDT in the learning process. So, if we have $l$ examples in the sample, and we synthesize step by step BDT, we graft the Tree in case of error. Another words, we correct the Tree if example is recognized incorrectly. If $r$ is such number of examples which is used for correction BDT, then the number $l - r$ examples must be as more as possible and these $l - r$ examples must be correct recognized by BDT. Then we can speak with confidence that learnability takes a place.

The aim of this paragraph is to ground that the problem of BDT synthesis must be stated as the problem of searching BDT with minimal leaves number which classifies correctly as more as possible number of examples. There are at least three approaches to ground this state.

• Class of BDT which is used for decisions making becomes narrower, when becomes smaller the parameter $\mu$ which bounds number of leaves. In that case VCD of this class becomes smaller and learnability has a place in accordance with statistical Vapnik-Chervonenkis theory.

• Another statistical estimations of the statistical reliability BDT, which doesn't use VCD, as well become better when the parameter $\mu$ becomes smaller.

• The description length of the BDT becomes shorter, when the number of his leaves is less, that determines reliability of recognition on the basis of principle of MDL – minimum description length.

We will describe these three approaches briefly.

1. Difficulty of estimation of BDT probability errors is explained thus empirical error rate, which is found by numbers of errors obtained by the count on the sample, are biased. But finiteness of VCD of the class of BDT is sufficient condition for uniform convergence of empirical empirical error rate to the error probability. The less VCD the higher uniform convergence and the less examples we need to achieve adequate accuracy. The estimation (1) manifests the following conclusion: the smaller leaves number $\mu$ the smaller VCD of the class $\mathscr{B}(n, 2, \mu)$ of BDT. So, minimization of the leaves number $\mu$ allows to achieve learnability.

2. The accuracy of BDT as empirical inductor can be estimated by the check sample. In this case the following probability scheme is used. Elements from the check sample are drawn out from universe accidentally and independently of one another. The check sample is correct and has no examples which are contained in the learning sample. Then error rate of BDT on the check sample will be unbiased.

We consider Boolean variables and suppose that source feature space maps into $B^n = \{0,1\}^n = \{\tilde{x} : \tilde{x} = (x_1, ..., x_n), x_i \in \{0,1\}\}$. Let $P$ is probabilistic measure on $B^n$;

$\sum_{\tilde{x}\in B^n} P(\tilde{x}) = 1$. Let $P(E)$ be the error probability of arbitrary BDT with $\mu$ leaves when arbitrary $\tilde{x} \in B^n$ will be recognized.

If BDT classifier has $\mu$ leaves, the length of the check sample is $l_c$, and $\delta_c$ is the number of errors of this BDT on $l_c$ tests, $0 \le \delta < 1$, then for any $\varepsilon : 1 > \varepsilon > \delta$

$$\mathbf{Pr}(P(E) \ge \varepsilon) \le \frac{\mu}{4l_c(\varepsilon - \delta)^2};$$

$$\mathbf{Pr}(P(E) \ge \varepsilon) < exp\{-\frac{(\varepsilon - \delta)^2 l_c}{\mu}\}$$

(look Appendix I). From this inequalities we can conclude that statistical reliability of BDT as higher as the leaves number is smaller.

3. Not losing community, we consider the case when the number of classes is equal two. Let us programme the binary word $p$ which can be used to decompress any BDT with $\mu$ leavs with a goal to estimate Kolmogorov complexity of such BDT. We present any BDT inner node $(1, \dots, \mu - 1)$ by the atom word which consists from two parts: cod of variable number $(1, \dots, n)$–prefix — and concatenated code of the number of the next atom or class value (0 or 1) — ending. Atom prefix has $n + 1$ possible values. 0 and 1 reserved for class numbers; $2, 3, \dots, n + 1$ used for feature numbers $1, \dots, n$. Atom ending has $\mu$ possible values: 0 and 1 reserved for class numbers as well as in the prefix, and the rest $\mu - 2$ values reserved for the pointers to BDT nodes (atoms). We use the list of atoms to present and describe BDT. According this list we can estimate description length or prefix Kolmogorov complexity $KP(BDT_\mu)$ of BDT with $\mu$ leaves:

$$KP(BDT_\mu) < 2(\lceil \log\log n \rceil + \lceil \log\log \mu \rceil) + (\mu - 1)(\lceil \log(n + 1) \rceil + \lceil \log \mu \rceil),$$

$$KP(BDT_\mu) \approx 2(\log\log n + \log\log \mu) + (\mu - 1)(\log(n + 1) + \log \mu).$$

It is evident the less leaves number the shorter BDT classifier description or its prefix Kolmogorov complexity.

## 2. Splitting criteria

The choice of variable or predicate for the splitting is the main element of all BDT synthesis algorithms. Splitting is equal to partition of some Boolean interval $N_t$ into two intervals $N_t^1$ and $N_t^2$ so that $N_t^1 \cup N_t^2 = N_t$, $N_t^1 \cap N_t^2 = \emptyset$, where $t$ is step splitting number. Partitionable interval must necessarily contains examples from different classes.

Let $k$ be the number of variable which is chosen for the partition of the interval $N_t$. Then we denote two intervals of partition as $N_t^1(k)$ and $N_t^2(k)$. We define $A(k) = N_t^1(k) \cap T_{l,n}$ — the set of examples (points) from the learning sample (learning table) $T_{l,n}$ which hits to the interval $N_t^1(k)$. Analogously $B(k) = N_t^2(k) \cap T_{l,n}$.

The table $T_{l,n}$ consists of $l$ rows (examples) and $n$ columns — values of variables $x_1, ..., x_n$. Additionally any row from $T_{l,n}$ is marked by the class number. Let $|A(k)| = m_1(k)$ and $|B(k)| = m_2(k)$.

We will talk that the predicate $S(k)$ is splitting criterion when variable $x_k$ is chosen for splitting when $S(k) = 1(True)$.

Let us consider the following criteria.

**$S_2$ *criterion*** (*of complete separability*). $S_2(k) = 1$ iff the set $A(k)$ contains examples of only one class and $B(k)$ contains examples of only one class, and the classes of the examples from $A(k)$ and $B(k)$ are different. Else $S_2(k) = 0$.

**$S_1$ *criterion*** (*of partial separability*) [35].

$S_1(k) = 1$ iff the set $A(k)$ contains examples of only one class *or* $B(k)$ contains examples of only one class; else $S_1(k) = 0$. It is evidently that $[S_2(k) = 1] \Longrightarrow [S_1(k) = 1]$.

**$Z_1$ *criterion*** (*of maximum partial separability*) [35].

Let $\exists k : S_1(k) = 1$ and $Z_1(k)$ is the number of points from the interval $N_k^1$ or $N_k^2$ which belong only one class. Then the variable with the number $k_* = arg \max\limits_k Z_1(k)$ must be chosen for the splitting.

**D *criterion*** (*of uniform pairs separability*) [35]. Let $T_{m_t,n} = T_{l,n} \cap N_t$ is subset of points from the learning sample contained in the interval $N_t$; $K_t(k)$ — the number of pairs of examples of different classes in the subset $T_{m_t,n}$ which are different by the value of variable $x_k$. We will talk that **D** criterion is used iff $k^* = arg \max\limits_k K_t(k)$ and variable $x_{k^*}$ is used for the splitting.

**D *criterion properties*.**

$1^o$ Let the number of points in the interval $N_t$ which are to be shatter is fixed. Let any allocations of these points and their class number marks in the partitionable interval $N_t$ are possible. To the value of $D(k^*) = \max\limits_k K_t(k)$ be maximum possible (when interval $N_t$ is shattered) it is necessary and sufficient the following two conditions simultaneously:

(i) the class of any point from $A(k^*)$ is different from the class of any point from $B(k^*)$ and

(ii) The partition of $N_t$ is uniform: $m_1(k^*) = m_2(k^*)$ when $m_{1,2}$ is even and $|m_1(k^*) - m_2(k^*)| = 1$ when $m_{1,2}$ is odd, where $m_{1,2} = m_1(k^*) + m_2(k^*)$ is the number of points contained in the interval $N_t$.

$2^o$ **D** criterion can be specialized and used for any types feature spaces and any separating predicates.

**DKM *criterion*** (**D***ietterich,* **K***earns,* **M***ansour*) [10]. This criterion is meant for two classes of examples. If the first interval $N_t^1(k)$ of the partition contains $s_{11}$ examples of the first class and second interval $N_t^2(k)$ of the partition contains $s_{22}$ examples of the

second class then $DKM(k) = 2\sqrt{\frac{s_{11}s_{22}}{m_{1,2}}} = 2\sqrt{\hat{p}_{11}\hat{p}_{22}}$, where $\hat{p}_{11}$ and $\hat{p}_{22}$ are the empirical estimations of probability of examples of the first class will appear in the interval $N_t^1(k)$ and examples of the second class will appear in the interval $N_t^2(k)$. In the paper [10] it is shown that the DKM criterion is more preferable than E criterion and G (Gini) criterion (see below).

**DKM *criterion properties*.**

$1^o$ $DKM(k) = 1$ iff any interval of the partition contains examples which belong to only one class and $s_{11} = s_{22}$.

$2^o$ DKM criterion possesses the uniformity property as well as D criterion.

$3^o$ But D criterion has the preference in comparison with DKM criterion because it can be used when the number of classes is greater than 2.

**TWO(Twoing) *criterion*.**

Let we have two classes of examples and two intervals of the partition: $N_t^1(k)$ and $N_t^2(k)$. The interval $N_t^1(k)$ contains $s_{11}$ points of the first class and $s_{21}$ points of the second class; $N_t^2(k)$ — $s_{12}$ points of the first class and $s_{22}$ points of the second class; $m_1 = s_{11} + s_{21}$, $m_2 = s_{12} + s_{22}$. So, $m_{1,2}$ points are shattered. The Twoing criterion is defined be the following expression:

$$TWO = \frac{m_1 m_2}{m_{1,2}^2}\left(\left|\frac{s_{11}}{m_1} - \frac{s_{12}}{m_2}\right| + \left|\frac{s_{21}}{m_1} - \frac{s_{22}}{m_2}\right|\right)^2,$$

$$TWO = \hat{p}\hat{q}\Big(|\hat{p}_{11} - \hat{p}_{12}| + |\hat{p}_{21} - \hat{p}_{22}|\Big)^2,$$

where $\hat{p} = \frac{m_1}{m_{1,2}}$, $\hat{q} = \frac{m_2}{m_{1,2}}$, $\hat{p} + \hat{q} = 1$. When the partition is correct $s_{12} = s_{21} = 0$ then $TWO = 4\hat{p}\hat{q}$. If addition to correctness the partition is uniform i.e. $\hat{p} = \hat{q} = 0.5$ then $TWO = 1$.

**TWO *criterion properties*** are mainly closely with the properties of DKM criterion.

**$\Omega$ *criterion*** [35]. Let variable $x_k$ was used for the partition and the interval $N_t^1(k)$ contains points of $J_1(k)$ various classes and $N_t^2(k)$ contains points of $J_2(k)$ various classed. We denote $\Omega(k^*) = \min\limits_k(J_1(k) + J_2(k))$. Then if the variable $x^*$ is used for splitting and there is exists a pair *of different classes* points $\hat{\alpha}$ and $\hat{\beta}$ in the intervals of the partition, i.e. $\hat{\alpha} \in N_t^1(k^*)$ and $\hat{\beta} \in N_t^2(k^*)$, we will say that $\Omega$ criterion is used.

**$\Omega$ *criterion properties*.**

$1^o$ $(\Omega(k) = 2) \Leftrightarrow (S_2(k) = 1)$.

$2^o$ If the value of $\Omega(k) = q$, where $q$ is initial given number of classes in the solvable task, and variable $x_k$ is used for splitting, then examples of any class contained in only one interval. We name this property *hierarchical separation sensitiveness.*

**E *criterion*** (entropic).

Let $s_{i,j}$ is the number of points of the class $i$ in the interval $N_t^j(k)$,$j = 1, 2$, which is gained as the result of the partition when variable $x_k$ is chosen for splitting. In the general case $m_{1,2}$ points of learning sample will be distributed to the the pair intervals of the partition as shown on the table 1:

*Table* 1. Partitioning into two intervals

| $N_t^1(k)$ | $N_t^2(k)$ |
|---|---|
| contains $m_1(k)$ points; | contains $m_2(k)$ points; |
| $s_{1,1}$ points attributed to the class 1 | $s_{1,2}$ points attributed to the class 1 |
| $s_{2,1}$ points attributed to the class 2 | $s_{2,2}$ points attributed to the class 2 |

The probability of belonging of arbitrary point from the interval $N_t^j(k)$ to the class $i$ can be estimated as $\hat{p}_{i,j} = s_{i,j}/m_j(k)$ where $m_j(k)$ is the number of points from learning sample which hit into the interval $N_t^j(k)$. Notice, that $\hat{p}_{i,j}$ is biased estimator.

Estimator of entropy of the interval $N_t^j(k)$ is $I_j(k) = -\sum\limits_i \hat{p}_{i,j} \log \hat{p}_{i,j}$. The estimator of average entropy by two intervals $N_t^1(k)$ and $N_t^2(k)$ will be $E(k) = \frac{m_1(k)}{m_{1,2}(k)} I_1(k) + \frac{m_2(k)}{m_{1,2}(k)} I_2(k)$ because of $\frac{m_j(k)}{m_{1,2}(k)}$ is the estimator of probabilistic measure of interval $N_t^j(k)$. So, $E(k)$ is an average statistical estimator.

The $E$ criterion of choice of splitting variable consists of use the variable with the number

$$k_* = arg \min_k E(k).$$

This choice corresponds to minimization of uncertainty as a result of current interval splitting.

**E *criterion properties***.

$1^o$ The entropy criterion $E$ is not sensitive to uniformity of partition – it can give out equal values in the cases when the numbers of examples in the intervals is equal and when these values are different even through these values are 1 and $m_{1,2} - 1$.

Really, if some interval $j$ contains examples from the only one class $i$ then probability estimation $\hat{p}_{i,j} = s_{i,j}/m_j(k)$ will be equal to 1 regardless of the value $m_j(k)$. In particular, let's consider two tables (Fig. 1):

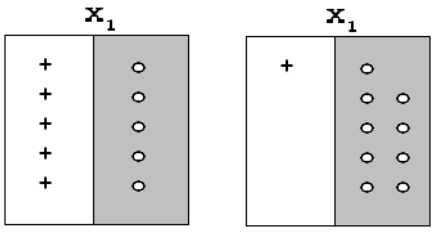


*Fig.* 1. *Nonuniform example distribution by the intervals of partition*

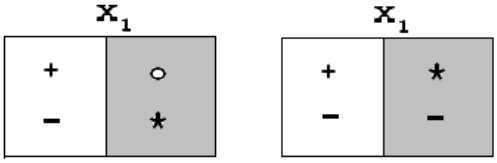


*Fig.* 2. *Two cases when E criterion values are agree and equal to 1*

In both cases (Fig. 1) E criterion value is equal to 0. Notice, the $D$ criterion in these cases takes values 25 and 9.

$2^o$ The $E$ criterion is not sensitive to the preference of hierarchical classification structure. This property is illustrated on the Fig.2.

**IGain *criterion*** (Information Gain) is meant for the choice of the splitting variable based on entropic approach.This criterion is improved to estimate average increase of information (gain) as result of branching step.

Initial average quantity of information needed to define the class of arbitrary point is

$$Info(T) = -\sum_{j=1}^{q} \frac{s_j}{l} \log \frac{s_j}{l} = -\sum_{j=1}^{q} \hat{p}_j \log \hat{p}_j,$$

where $T$ is learning sample; $l$ — the number of examples (points) in T; $q$ — the number of classes; $s_j$ the number of points in $T$ marked by class $j$; $\hat{p}_j$ — the estimator of appearance probability of the class $j$.

$IGain$ criterion of maximum information gain is $IGain(k) = Info(T) - E(k)$, where $E(k)$ is the value of defined above E criterion — the average entropy by intervals of the partition obtained by choice of the variable $x_k$.

**MEE *criterion*** (**M***inimum* **E***rror* **E***ntropy*)[14].

Let's first consider the case of two classes — $\omega_1$ and $\omega_2$. Let $x_k$ is variable — candidate for splitting; $\omega_1$ — the class number — candidate for the mark of interval $N_t^1(k)$ (the left

branch) if variable $x_k$ is used. Then the right branch (and interval $N_t^2(k)$) is admittedly marked which the class $\omega_2$. If we suppose such splitting is correct then any point from the learning sample which hits in $N_t^1(k)$ but marked in the sample by the class $\omega_2$ will be classified incorrectly. We denote correspondingly $r_{12}$ and $r_{21}$ the numbers of such incorrectly classified points in the intervals $N_t^1(k)$ and $N_t^2(k)$. Then estimators of error probabilities sort of "mixed up classis"in the shattered interval $N_t = N_t^1 \cup N_t^2$ is $\hat{P}_{12} = \frac{r_{12}}{m_{1,2}}$ and $\hat{P}_{21} = \frac{r_{21}}{m_{1,2}}$, where $m_{1,2}$ is the number of points of the sample hit in $N_t$. The value $1 - \hat{P}_{12} - \hat{P}_{21}$ is the estimation of probability of the correct computation of the class number by the node with the predicate (variable) $x_k$ and edges marked $\omega_1$ and $\omega_2$.

The MEE formula based the *error entropy* $EE$:

$$EE = EE(N_t, k, \hat{P}_{12}, \hat{P}_{21}) = \hat{P}_{12}\log\hat{P}_{12} - \hat{P}_{21}\log\hat{P}_{21} - (1 - \hat{P}_{12} - \hat{P}_{21})\log(1 - \hat{P}_{12} - \hat{P}_{21}).$$

The rule of splitting by the $MEE$ criterion consists in the choice of variable $x_k^*$ where

$$k^* = arg\min_{N_t,k} EE(N_t, k, \hat{P}_{12}, \hat{P}_{21}).$$

**MEE *criterion properties*.**

$1^o$ The minimum value of EE is equal to 1 when all examples are correctly classified by the partition created with splitting variable $x_{r^*}$. The maximum EE=1 is when examples are completely mixed up classis — when $\hat{P}_{12} = \hat{P}_{21} = 1/2$.

$2^o$ With mixed up classis increases the EE estimator increases too. Notice, the value of $\Omega$ criterion increases in this case as well.

$3^o$ When the partial separability takes a place $(S_1(k) = 1)$, for example, when $\hat{P}_{12} = 1/2$, then $EE = 1$. Therefore $MEE$ criterion sometimes can't detect the difference between cases of the partial and complete separability.

**G *criterion*** (based on Gini Index).

Gini Index of the interval $N_t^j(k)$ is

$$g(N_t^j(k)) = 1 - \sum_i \hat{p}_{i,j}^2 = 1 - \sum_i (s_{i,j}/m_j(k))^2.$$

Squares of estimators of conditional probabilities of all classes in the interval $N_t^j(k)$ are summed. If the interval $N_t^j(k)$ contains points of only one class then Gini Index reaches its minimum value equal to 0. $G$ criterion defined by formula

$$G(k) = g(N_t^1(k)) + g(N_t^2(k)).$$

The splitting variable number is $k_* = arg\min_k G(k)$.

**G** ***criterion properties***.

$1^o$ If the interval contains poins of only one class then its index is equal to 0, therefore G criterion is enable recognize the partial separability.

$2^o$ $(G(k) = 0) \Leftrightarrow (S_2(k) = 1)$ what means ability of G criterion to recognize complete separability.

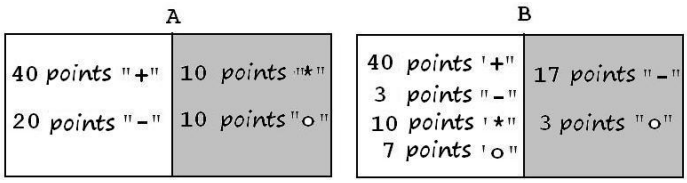


*Fig.* 3. *Two cases of points distribution*

In the paper [21], page 7, it is shown that Gini criterion is disable recognize hierarchial separability of classes and the explanatory example is done (Fig. 3). On the Fig. 3 two cases of points displacement. The case A correspondents to the completely separability of two classes **(+)∪(-)** and **(*)∪(o)**. But G criterion makes more preferable the partition B.

## 3. Comparison of the criteria

**Example 1.** Let the interval of dimension 5 is given with 9 points distribution as shown on Fig. 4. These points classes denoted by symbols $+, -, *$. The values of splitting criteria when variable $x_i$ is chosen, $x_i \in x_1, ..., x_5$ are presented on the Fig. 5. The comparison of the criteria values shows that all criteria except $S_1$ and $G$ criterions are concordant: they define a choice of the same variable $x_5$. Criterions $S_1$ and $G$ for one's turn put are concordant each other and pick out the case of partial separability.

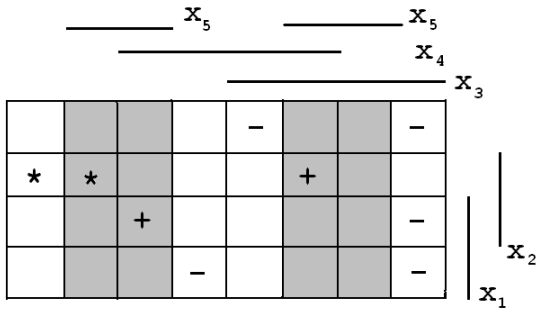


*Fig.* 4. *The sample points distribution*

If variables are ordered by the E criterion decrease then D criterion values will be increase but the monotonicity of the growth as it seen from the Table 2 and Fig. 6 is violated. For the variable $x_3$ increased value $D(3) = 15$ is explained the more sensitivity of $D$ criterion to the partition separability in comparison with $E$ criterion.

*Table* 2. *E and D criterions comparison*

| Критерии | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $E$ | 1.206 | 1.068 | 0.984 | 0.846 | 0.739 |
| $D$ | 13 | 15 | 14 | 16 | 17 |

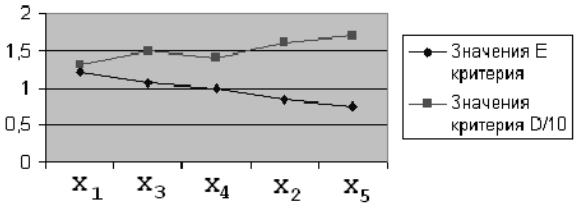


*Fig.* 5. *E and D criterions comparison*

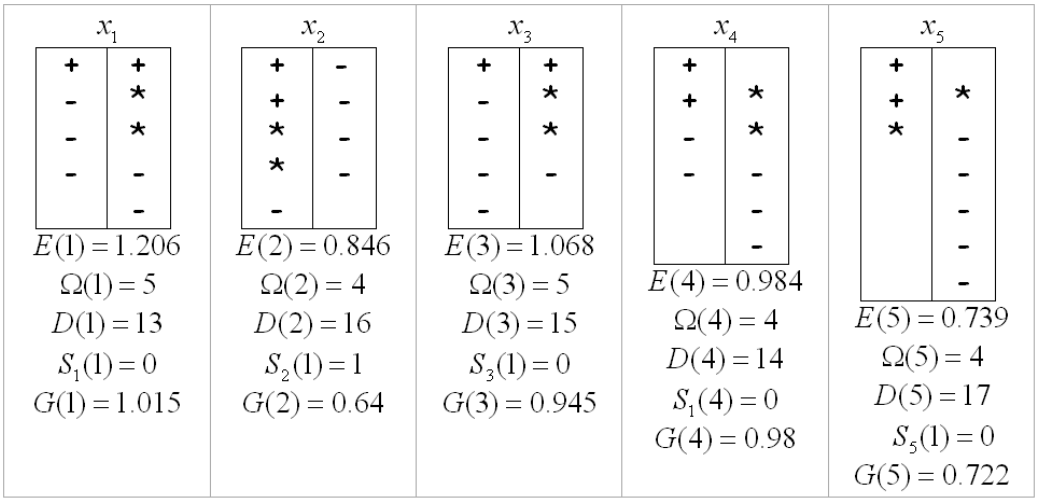


*Fig.* 6. *Criteria values for the various point distribution*

**Example 2.** Let the interval of dimension 4 is given which contains 10 points of 5 classes (Fig. 7)

The values of $E$ and $D$ criterions are concordant each step of splitting this example. We give their values only for the first step (Table 3).

*Table* 3. *The first step E and D criterions values*

| Creteria | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $D$ | 25 | 20 | 21 | 22 |
| $E$ | 1.246 | 1.565 | 1.922 | 1.551 |

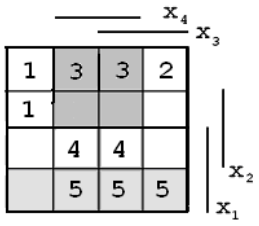


*Fig.* 7. Points distribution

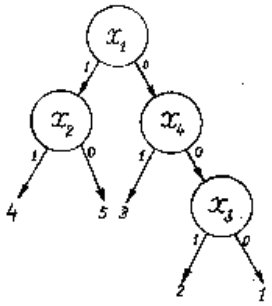


*Fig.* 8. Optimal BDT

It's easy to see criterion $E$ gives value 0 when each interval of the partition contains points of only one class. And in this case $E$ and $D$ criteria values coincide.

According to multiple experiment computations with various splitting criteria the comparative data are presented in the paper [15]. In particular, the number of leaves of BDT were compared as result of synthesis. The comparison carried out on 36 real tasks. In the table 4 it is shown how many times the use of each from 5 criteria reduced to acquisition the BDT with the least leaves number (the best result or win) and the most ones (the worst result or loss) in comparison with all other algorithms.

*Table* 4. *Comparison of the win numbers*

| Algorithms | Gini | Info Gain | Twoing | C4.5 | MEE |
|---|---|---|---|---|---|
| Win number | 11 | 9 | 8 | 1 | 18 |
| Loss numder | 4 | 3 | 3 | 24 | 7 |

The data presented in the table 4 confirm first of all that *it's impossible to pick out a criterion which gives the best result all the cases for any admissible learning samples.* Nevertheless according to the table 4 the MEE algorithm wins at the minimum twice frequently in comparison with other. It's rather unexpectedly that the algorithm C4.5 was the worst in these experiments in spite of it's widely used in applications.

In the paper [35]experimental researches of BDT synthesis algorithms were carried out. In these experiments the points of $\{0,1\}^n$ — examples — were generated according to the equally probable distribution with $n = 25$. As well random number was designated to each generated point. Result of these experiments are presented in the table 5.

*Table* 5. *Comparison of the win numbers*

| Algorithms | Average by 15 experiments leaves number | | |
|---|---|---|---|
| | 5 classes 50 examples | 2 classes 50 examples | 5 classes 100 examples |
| LISTBB | 23.1 | 11.3 | 44.7 |
| LISTD | 24.5 | 14.1 | 46.7 |
| LISTB | 44.9 | 34.9 | - |

The best in these experiments algorithm $LISTBB$ (see below) is the hybrid procedure of situational choice of splitting criterion which depends from the initial value of the $\Omega$ criterion and of presence the partly or complete separability. The algorithm $LISTD$ uses only $D$ criteria. The algorithm $LISTB$ uses random order of variables for the splitting.

Notice the algorithm $LISTBB$ first computes the $\Omega$ criterion value which most closely to the $MEE$ criterion.

## 4. The stopping rules and BDT branche reducing

The BDT is called *correct* relatively the learning sample if all examples of this sample are classified by the BDT correctly. The feature space partition which is generated by the correct BDT is such that each terminal set included in this partition contains the points belonging to only one class. These terminal sets correspond to the BDT leaves and inherit leave numbers.

*The rule 1.* A branching process of the BDT synthesis is continued as long as this BDT becomes correct. This is possible if and only if predicate descriptions of of all pair sample examples are different.

*The rule 2.* A branching process of the BDT synthesis is stopped when the leaves number reaches established threshold.

*The rule 3.* A branching process of the BDT synthesis is stopped when Information gain can't be increased by adding a new inner node.

*The rule 4.* A branching process of the BDT synthesis is stopped when the lengthes of all BDT branches reaches some given value.

*The rule 5.* A branching process of the BDT synthesis is stopped when all terminal sets which must be shattered contains the point numbers which are less then given threshold.

*The rule 6.* Stopping of the BDT synthesis is defined on the base of Minimum Description Length principle which is in accord with the choice of more probabilistic hypothesis by the Bayes rule [22] — one of formalization of "Occam's razor": *the best hypothesis is that which minimize sum of hypothesis description length (of the model)and data (relatively this hypothesis)length.* This stopping rule for the BDT synthesis is described in detail in [29].

*The rule 7.* A branching process of the BDT synthesis is stopped by the rule "*Plus five*" based on the class VCD with restricted $\mu$ leaves estimator [4].

*The rule 8.* A branching process of the BDT synthesis is stopped on the base of the theoretical estimator of error probability when adding any additional none to BDT doesn't result to this error probability decrease. Such approach is described in many papers for example in [29].

The last two rules are more theoretical grounded.

Any stopping rule listed above can be used together with another one ore jointly with some splitting criteria collection to obtain the new BDT synthesis algorithm. It can be seen in some publications dedicated to BDT inductors synthesis.

*The pruning (reducing) rules* define the maximum possible length of BDT branches. If some branch has the length more than the given bound then it is pruned and the last inner node is replaced by the class label. This label most often is defined by the class of the most quantity examples contained in the shattering interval which is correspond to the pruning branch.

Pruning must be used when synthesis of the correct BDT leads to its unjustified complexity.

## 5. The binary decision tree synthesis algorithms

***The CLS algorithm*** (**C***oncept* **L***earning* **S***ystem*). This is classical Hunt's algorithm [8] which is the base of most BDT algorithm synthesis methods. The algorithm CLS shatters cyclically the learning sample into subsets in accordance with its most separating capability. A shattering is ended when all obtained subsets contains poins of onle one class. When in use shattering the BDT is synthesized.

***The ID3 algorithm*** was offered by Hunt's student J. Ross Quinlan [17] (1986). ID3 was based on Hunt's CLS algorithm and used gain ratio as splitting criteria. The synthesis was stopped when the BDT became correct or when the further splitting didn't give the information gain increase.

***The C4.5 algorithm*** This algorithm developed by Ross Quinlan (1993)[18] is improved variant of the ID3. It uses the gain ratio for the splitting. The synthesis is stopped when the BDT becomes correct or when the points number for the shattering becomes less then given threshold.

***The CART algorithm***. The abbreviature CART is given from "**C**lassification and **R**egression **T**rees". The algorithm is intended for BDT regression synthesis as well as classification trees and uses the Twoing criterion. Regression trees have in its terminal nodes (leaves)some real numbers instead the class labels. The splitting is realized by the mean-square error minimum.

***The CHAID algorithm*** [9] (**CH**isquare-**A**utomatic-**I**nteraction-**D**etection on the base of $\chi^2$ criterion ). The applications applied statistics methods for the BDT splitting obtained its development in the 70-s last century. The *CHAID* is the evolution of the ***AID algorithm***[20] (**A**utomatic **I**nteraction **D**etection). The *CHAID* is destined for the choice of variable groups for the splitting in the following way. For each variable such pairs it values is found which are slightly changed with a changing of the goal feature (class number). Depending on types of features-variables slightness of such changing is estimated by the Pearson criterion $\chi^2$ (for the nominal variables), by the Fisher criterion (for the continuous variables), by the likelihood ration test (for the rank variables). The statistical significantly distinguishable pairs of values are joined in the homogeneous group of values, and the process is reiterated while distinguishable pairs are found. The variable dividing the groups of the homogeneous values is chosen for the splitting. Stopping of the BDT synthesis takes place when any from the following conditions is holds:

1) The given maximum tree depth is reached;

2) Any terminal node contains smaller points number then given threshold points number.

Missing variable values (if such ones exist) is joined in the individual groups.

***The QUEST algorithm*** [13] (**Q**uick **U**nbiased **E**fficient **S**tatistical **T**ree). For the splitting, a connection between each input variable and the goal variable is estimated on the base F-criterion ANOVA (Analysis Of Variances)or on the base the Levene test [11] of the dispersion homogeneity of the order or continuous variables, and on the base $\chi^2$ criterion for the nominal variables. For the multiclass goal variables, cluster analysis is used to join in two superclass. For the splitting, the variable with the largest estimator of the statistical connection with the goal feature is chosen. A cross validation is used for the pruning. This gives grounds to speak about unbiasedness of the statistical estimations. Notice, we described only some part of the QUEST algorithm concerning the variable choice for the splitting. As a hole the QUEST can be classified as a complex system of data analysis which gives ability to analyze various variants of predictors and use optimization procedures to choose them.

***The SLIQ algorithm*** [16] (**S**upervized **L**earning **I**n **Q**UEST ). This algorithm is intended for Data Mining applications whith the big size raw data. Gini Index and quick sort are used for the splitting.

***The PUBLIC algorithm*** [19] (**P**runing and **B**uilding **I**ntegrate **C**lassifier ). The classes distribution bar chart is used for the splitting. Each point of the distribution bar chart is considered as a candidate to define the branch threshold. The entropic splitting criterion is used for the thresholds and variables choices.

***The algorithms CAL5*** [24]***, FACT (early version of QUEST), LMDT*** [2]***, T1*** [7]***, MARS*** [6] and many others aren't principally different from above presented algorithms.

The Table 6 placed below presents comparative data of using of various BDT synthesis algorithms in medical applications.

*Table* 6. The usage of the algorithms in medical applications

| The algorithm | Usage (% ) |
|---|---|
| ID3 | 68 |
| C4.5 | 54.55 |
| CART | 40.9 |
| SLIQ | 27.27 |
| PUBLIC | 13.6 |
| CLS | 9 |

## 6. The Hybrid algorithm LISTBB based on the aggregate splitting criteria

| Procedure ***LISTBB***: a splitting variable choice |
|---|
| *Input*: The interval $N_t$ to be shattered and points from the learning sample which are contained in $N_t$<br>*Output*: The variable for $N_t$ splitting (for the current BDT growth) |
| 1: Compute the set of variable numbers for which the minimum of the $\Omega$ criterion is achieved: $\tilde{k}_\Omega = \{k_o : k_o = arg\min_k \Omega(k)\}$, where $k$ runs all numbers of free variables of the shattered interval<br>2: If $\lvert\tilde{k}_\Omega\rvert = 1$, i.e. the $\Omega$ criterion achieves the minimum for only one variable, then choose the variable $x_{k_o}$ for splitting and return from the procedure<br>3: If $\min_k \Omega(k) = q$, where $q$ is the initial number of classes, then choose any variable $k^*$ such that $k^* = arg\max_{k\in\tilde{k}_\Omega} D(k)$ and return from the procedure<br>4: If there is no partial separability, i.e. $\forall k \in \tilde{k}_\Omega(S_1(k) = 0)$, then choose for the splitting any variable $k^*$ such that $k^* = arg\max_{k\in\tilde{k}_\Omega} D(k)$ and return from the procedure<br>5: If the partial separability exists then choose for the splitting any variable $k_*$ by the maximum of partial separability: such that $k^* = arg\max_{k\in\tilde{k}_\Omega} Z_1(k)$ and return from the procedure |

To explain step 3 of the procedure $LISTBB$ the following example (Fig. 9) can be considered. Let six points in the shattering interval belong to the classes labeled by $+$, **-**, *****, **o**, , $\Delta$. Partitions by the variables $x_1$ and $x_2$ give $\Omega(1) = \Omega(2) = 5$, $D(1) = 8$ but $D(2) = 9$. This example proves then when values of $\Omega$ criterion is equal for some two variables, the $D$ criterion for these variables can be different.

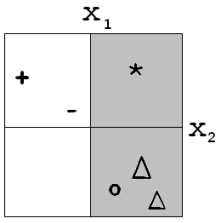


*Fig.* 9. Step 3 explaination

When the BDT is built, splitting steps are executed thus and so the leaves number grows. There exists a lower bound for the BDT leaves number which will be obtained when the synthesis stops. But while the BDT synthesis is run the lower bound can be changed. So we call such lower bound as current.

**Proposition 1.** When the procedure LISTBB is used, the value $\mu_t + \Omega(k^*) - 1$, where $\mu_t$ is the current BDT leaves number and $\Omega(k^*)$ is the maximum of the $\Omega$ criterion achieved for the variable $x_k^*$, is the current lower bound for the BDT leaves number which will be obtained when the synthesis stops.

*Proof.* Really, let's consider step $t$ of BDT synthesis. Let $\mu_t$ is the current leaves number, the interval $N_t$ must be shattered and splitting variable $x_k^*$ is chosen. If we shatter the interval then one leave is replaced by the new node pointing to the two intervals which contains $\Omega(k^*)$ various classes, consequently we will need to add at least $\Omega(k^*)$ leaves to reach correctness. So, we have the lower bound $\mu_t - 1 + \Omega(k^*)$. □

**Remark.** Easy to see that $q \leq \Omega(k) \leq 2q$, where $q$ is initial given number of example classes. Then if $q$ is small ($q = 2$ or 3 ) utility of the above lower bound is minor. But with $q$ increasing it becomes more and more.

**Proposition 2.** When the procedure LISTBB is used and on arbitrary step $t$ of the BDT synthesis the partial separability exists, then the following estimator of the increment leaves number $\Delta\mu_t$, which will be added to the BDT when the synthesis stops, is true:

$$\min_k \Omega(k^*) - 1 \leq \Delta\mu_t \leq m_{1,2} - Z_1(k^*),$$

where $k^*$ is the splitting variable number.

*Proof.* The left part of the inequality is proven (Proposition 1). The right part becomes evident if we note that $m_{1,2}$ points contained in the shattering interval and worse case each point may be separated by individual leave. But when the partial separability exists, $Z_1(k^*)$ point will be separated into one correct interval, and another — second interval of the partition — will contain $m_{1,2} - Z_1(k^*)$ points. □

According to propositions 1, 2, the algorithm LISTBB (in spite it is heuristic) is directed to the choice of splitting variable by such the way to minimize both lower and upper bounds of leaves number increment. But algorithm's LISTBB "bias" and its "drive for" partial separability can put to the cases when $Z_1(k^*)$ is very small, for example $Z_1(k^*) = 1$, and then the choice of variable based on partial separability may be unprofitable.

The parametric version LISTBB(p) contains the parametr $p$, which modifies step 5 by following:

$5_{(p)}$: If the partial separability exists and $Z_1(k^*) > p$ then choose for the splitting any variable $k^*$ by the maximum of partial separability: such that $k^* = arg\max\limits_{k\in\tilde{k}_\Omega}(Z_1 k)$ else choose for the splitting any variable $k^*$ such that $k^* = arg\max\limits_{k\in\tilde{k}_\Omega} D(k)$ and return from the procedure.

The LISTBB algorithm (splitting)stops when a) correct partition is obtained or b) the list number exceeds the specified threshold. This threshold at fist was an heuristic parameter, but now it is defined on the base of MDL principle [29].

## 7. Appendix I

We consider the case when variables are Boolean because we suppose that arbitrary feature space is mapped on $\{0,1\}^n$ predicates values space. We denote $P(E)$ error probability of arbitrary BDT with $\mu$ leaves when admissible object described as $\tilde{x} \in \{0,1\}^n$ is recognized. At length, $\mathbf{Pr}(P(U)$ is the probability of fulfilling some condition $U$.

**Theorem 1.** *If BDT $\mu$ leaves classifier made $\delta l_c$ errors on the check sample of length $l_c$ where $0 \le \delta < 1$ then for any $\varepsilon : 1 > \varepsilon > \delta$ the following inequality takes a place:*

$$\mathbf{Pr}(P(E) \ge \varepsilon) \le \frac{\mu}{4l_c(\varepsilon - \delta)^2}.$$

*Proof.* Let's denote BDT leaves labels as $\omega_1, ..., \omega_s, ..., \omega_\mu$ so that label $\omega_s$ defines the class of points hits in the interval $N_s$ which corresponds to the BDT branch number $s$. So, this branch ends by leave $\omega_s$. The probabilistic measure of intrval $N_s$ is denoted as $P(N_s) = \mathbf{Pr}(\tilde{x} \in N_s)$. For simplicity we will denote $N_s$ the event "$\tilde{x} \in N_s$" as well as the interval, and $\omega_s$ — the event of appearance of the point of the class $\omega_s$. The intervals $N_1, ..., N_s, ..., N_\mu$ correspond the partition of $B^n = \{0,1\}^n$ thus

$$\sum_{s=1}^{\mu} P(N_s) = 1;\ P(E) = \sum_{s=1}^{\mu} P(E|N_s)P(N_s);$$

$$P(E|N_s) = 1 - P(\omega_s|N_s);\ P(\omega_s, N_s) = P(\omega_s|N_s)P(N_s);$$

$$P(E) = \sum_{s=1}^{\mu}(1 - P(\omega_s|N_s))P(N_s) = \sum_{s=1}^{\mu} P(N_s) - \sum_{s=1}^{\mu} P(\omega_s, N_s) = 1 - \sum_{s=1}^{\mu} P(\omega_s, N_s).$$

For each interval of the partition frequency of events

$$\nu(\omega_s, N_s) = \frac{n(\omega_s, N_s)}{l_c}$$

are defined by the numbers $n(\omega_s, N_s)$ of points from the learning sample hits in interval $N_s$ and classified as $\omega_s$. These points are classified by the BDT correctly. Let's denote the number of points from the learning sample which hit in interval $N_s$ but classified wrong as $k_s$. Then

$$\sum_{s=1}^{\mu}(n(\omega_s, N_s) + k_s) = l_c; \ \sum_{s=1}^{\mu}\frac{n(\omega_s, N_s)}{l_c} + \sum_{s=1}^{\mu}\frac{k_s}{l_c} = 1;$$

$$\sum_{s=1}^{\mu}\nu(\omega_s, N_s) + \delta = 1,$$

where $\delta = \frac{1}{l_c}\sum\limits_{s=1}^{\mu} k_s$ is the errors quota on the learning sample. Let's substitute the left part of the equality in lieu of 1 in the formula which defines the BDT error:

$$P(E) = 1 - \sum_{s=1}^{\mu}P(\omega_s, N_s) = \sum_{s=1}^{\mu}\nu(\omega_s, N_s) - \sum_{s=1}^{\mu}P(\omega_s, N_s) + \delta.$$

The event "$P(E) \geq \varepsilon$" is equal to the event

$$\sum_{s=1}^{\mu}\nu(\omega_s, N_s) - \sum_{s=1}^{\mu}P(\omega_s, N_s) \geq \varepsilon - \delta.$$

Mathematical average and variance of the random quantity $\zeta = \sum\limits_{s=1}^{\mu}\nu(\omega_s, N_s)$ of the sum of independent random quantities are

$$\mathbf{M}[\zeta] = \sum_{s=1}^{\mu}\mathbf{M}[\nu(\omega_s, N_s)] = \sum_{s=1}^{\mu}P(\omega_s, N_s);$$

$$\mathbf{D}[\zeta] = \sum_{s=1}^{\mu}\mathbf{D}[\nu(\omega_s, N_s)] = \sum_{s=1}^{\mu}\mathbf{M}[(\frac{n(\omega_s, N_s)}{l_c} - P(\omega_s, N_s))^2] =$$

$$= l_c^{-2}\sum_{s=1}^{\mu}\mathbf{M}[(n(\omega_s, N_s) - l_cP(\omega_s, N_s))^2],$$

where $l_cP(\omega_s, N_s)$ is mathematical average and $\mathbf{M}[(n(\omega_s, N_s) - l_cP(\omega_s, N_s))^2$ is variance of number of cases when the point hits in the interval $N_s$ and its class is $\omega_s$. From the inequality

$$l_cP(\omega_s, N_s)(1 - P(\omega_s, N_s)) \leq \frac{l_c}{4}$$

we get

$$\mathbf{D}[\zeta] \leq \frac{\mu}{4l_c}.$$

By the Chebyshev inequality

$$(\forall\varepsilon > 0)\mathbf{Pr}(|\xi - \mathbf{M}[\xi]| \geq \varepsilon) \leq \mathbf{D}[\xi]/\varepsilon^2$$

we get

$$\mathbf{Pr}(P(E) \geq \varepsilon) \leq \frac{\mu}{4l_c(\varepsilon - \delta)^2}.^{1}$$

□

**Corollary.** *The less $\mu$ — leaves number of BDT — the more its statistical reliability.*

## References


1. Blumer A., Littlestont N. Learning faster than promise by the Vapnik-Chervonenkis dimension / Anselm Blumer, Nick Littlestone // Discrete Applied Mathematics, 1989. — Vol. 24. — Iss. 1–3. — P. 47–63.
2. Brodley C. E., Utgoff. P. E. Multivariate decision trees / C. E. Brodley, P. E. Utgoff // Machine Learning. — 1995. — Vol. 19. — P. 45–77.
3. Donskoy V. I. The Estimations Based on the Kolmogorov Complexity and Machine Learning from Examples / V. I. Donskoy // Proceedings of the Fifth International Conference "Neural Networks and Artificial Intelligence" (ICNNAI'2008). — Minsk: INNS. — 2008. — P. 292–297.
4. Donskoy V. I. Complexity of families of learning algorithms and estimation of the nonrandomness of extraction of empirical regularities / V. I. Donskoy // Cybernetics and System Analysis. — 2012. — Vol. 48. — No 2. — P. 233–241.
5. Floyd S., Warmuth M. Sample Compression, learnability, and the Vapnik-Chervonenkis dimension / Sally Floyd, Manfred Warmuth // J. Machine Learning, 1995. — Vol. 21. — Iss. 3. — P. 269–304.
6. Friedman J. H. Multivariate Adaptive Regression Splines / J. H. Friedman // The Annual of Statistics. — 1991. — Vol. 19. — P. 1–141.
7. Holte R. C. Very simple classification rules perform well on most commonly used datasets / R. C. Holte // Machine Learning. — 1993. — Vol. 11. — P. 63–90.
8. Hunt E. B. Experiments in Induction / Earl B. Hunt, Janet Marin, Philip J. Stone. — N. Y.: Academic Press, 1966. — 247 p.
9. Kass G. V. An exploratory technique for investigating large quantities of categorical data / G. V. Kaas // Applied Statistics. — 1980. — Vol. 29(2). — P. 119–127.
10. Kearns M. On the boosting ability of top-down decision tree learning algorithms / M. Kearns, Y. Mansour // Journal of Computer and Systems Sciences. — 1999. — Vol. 58 (1). — P. 109–128.
11. Levene H. Robust tests for equality of variances / H. Levene // Contributions to Probability and Statistics / Ed. I. Olkin, Palo Alto. — Stanford University Press: 1960. — P. 278–292.
12. Li M. Sharpening Occam's Razor / Ming Li, John Tromb, Paul M. B. Vitanyi. — Research Rep. CT-94–03. — Amsterdam: ILLC, 1994. — 13 p. http://www.illc.uva.nl/Research/Reports/CT-1994-03.text.pdf
13. Loh W.-Y. Split Selection Methods for Classification Trees / Wei-Yin Loh and Yu-Shan Shih // Statistica Sinica. — 1997. — Vol. 7. — P. 815–840.


[1]The use of Bernstein inequality allows to get the inequality

$$\mathbf{Pr}(P(E) \geq \varepsilon) < \exp\left(-\frac{(\varepsilon - \delta)^2 l_c}{\mu}\right).$$

14. Marques de Sa J. P. Tree Classifiers Based on Minimum Error Entropy Decisions / Joaquim P. Marques de Sa, Raquel Sebastiao and Joao Gama // Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition. — 2011. — Vol. 2. — № 3. — P. 41–55.
15. Marques de Sa J. P. New Results on Minimum Error Entropy Decision Trees / Joaquim P. Marques de Sa, Raquel Sebastiao and Joao Gama, Tanja Fontes // CIAPR'11 Proceedings of the 16th Iberoamerican Congress conference on Progress in Pattern Recognition, Image Analysis, Computer vision, and Applications. Chile, Pucon. — 2011. — P. 355–362.
16. Mehta M. SLIQ: A fast scalable classifier for data mining / Manish Mehta, Rakesh Agrawal, Jorma Rissanen / In Advances in Database Technology — EDBT'96. Avignon, France, March 1996 // Lecture Notes in Computer Science. — 1996. — Vol. 1057. — P. 18–32.
17. Quinlan J. R. Induction of decision trees // Machine Learning. — 1986. — Vol. 1. P. 81–106.
18. Quinlan J. R. C4.5: Programs for Machine Learning / John Ross Quinlan. — Morgan Kaufmann: 1993. — 302 c.
19. Rastogi R. PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning / Rajeev Rastogi, Kyuseok Shim // Proceedings of the 24th VLDB Conference August 1998, USA. — New York: 1998. — P. 404–415.
20. Sonquist J. A. Searching for structure (alias-AID-III) // John A. Sonquist, Elizabeth Lauh Baker, James N. Morgan.–Institute for Social Research, University of Michigan: 1971. – 287 P.
21. Taylor P. C. Block diagrams and splitting criteria for classification trees / P. C. Taylor, B. W. Silverman // Statistics and Computing. — 1993. Vol.3. — P. 147–161.
22. Vitanyi P. Ideal MDL and Its Relation to Bayesianism Bayesianism / Paul M. B. Vitanyi, Ming Li // In Proc. ISIS: Information, Statistic and Induction in Science. — Singapore: World Scientific, 1996. — P. 282–291.
23. Vitanyi P. Minimum description length induction, Bayesianism, and Kolmogorov complexity / Paul M. B. Vitanyi, Ming Li // IEEE Transactions on Information Theory, 2000. — Vol. 46. — No 2. — P. 446–464.
24. Muller W. Automatic construction of decision trees for classification / W. Muller, F. Wysotzki // Annals of Operations Research. — 1994. — Vol. 52. — P. 231—247.
25. Блох А. Ш. Об одном алгоритме обучения для задач по распознаванию образов / А. Ш. Блох // Вычислительная техника в машиностроении. — Минск: 1966. — № 10. — С. 37–43.
26. Вапник В. Н. Теория распознавания образов / В. Н. Вапник, А. Я. Червоненкис. — М.: Наука, 1974. — 416 с.
27. Донской В. И. Колмогоровская сложность классов общерекурсивных функций с ограниченной ёмкостью / В. И. Донской // Таврический вестник математики и информатики, 2005. — №1. — С. 25–34.
28. Донской В. И. Оценки ёмкости основных классов алгоритмов эмпирического обобщения, полученные pVCD методом / В. И. Донской // Ученые записки ТНУ им. В. И. Вернадского. Серия "Физико-математические науки", 2010. — Т. 23 (62). — № 2. — С. 56–65.
29. Донской В. И. Сложность семейств алгоритмов обучения и оценивание неслучайности извлечения эмпирических закономерностей / В. И. Донской // Кибернетика и системный анализ, 2012. — № 2. — С. 86–96.

30. Донской В. И. Колмогоровская сложность и ее применение в машинном обучении / В. И. Донской // Таврический вестник информатики и математики. — 2012. — № 2. — С. 4–35.
31. Донской В. И. О построении программного обеспечения распознающих систем / В. И. Донской // Программирование. — 1980. — № 2. — С. 87–90.
32. Донской В. И. Асимптотика числа бинарных решающих деревьев / В. И. Донской // Ученые записки Таврического национального ун-та им. В. И. Вернадского, серия "Информатика и кибернетика". — 2001. — № 1. — С. 36–38.
33. Донской В. И. Интеллектуализированная программная система IntMan поддержки принятия решений в задачах планирования и управления / В. И. Донской, В. Ф. Блыщик, А. А. Минин, Г. А. Махина // Искусственный интеллект. — 2002. — №2. — С. 406–415.
34. Донской В. И. О совместном использовании абдукции, аналогии, дедукции и индукции при синтезе решений / В. И. Донской // Искусственный интеллект. — № 2. — 2000. — С. 59–66.
35. Донской В. И. Исследование алгоритмов распознавания, основанных на построении решающих деревьев: автореф. дисс. на соиск. уч. степени канд. физ.-мат. наук: спец. 01.01.09 "Математическая кибернетика" / В. И. Донской. — М., 1982. — 16 с.
36. Журавлев Ю. И. Теоретико-множественные методы в алгебре логики / Юрий Иванович Журавлев // Проблемы кибернетики. — 1962. — Вып. 2. — С. 5–44.
37. Журавлев Ю. И. Об отделимости подмножеств вершин $n$-мерного куба / Юрий Иванович Журавлев // Науч. Труды Матем. ин-та им. В. А. Стеклова. — 1958. — Т. 1. — С. 143–157.
38. Колмогоров А. Н. Теория информации и теория алгоритмов // А. Н. Колмогоров. — М.: Наука, 1987. — 304 с.
39. Орлов В. А. Применение граф-схемного метода распознавания образов: автореф. дисс. на соиск. уч. степени канд. техн. наук: спец. 05.13.01 "Техническая кибернетика и теория информации" / В. А. Орлов. — Владивосток, 1974. — 23 с.

***Статья поступила в редакцию 26.05.2013***