Decision Trees for Mining Data Streams Based on the Gaussian Approximation

Leszek Rutkowski, *Fellow, IEEE,* Maciej Jaworski, Lena Pietruczuk, Piotr Duda

Abstract—Since the Hoeffding Tree algorithm was proposed in literature, decision trees became one of the most popular tools for mining data streams. The key point of constructing the decision tree is to determine the best attribute to split the considered node. Several methods to solve this problem were presented so far. However, they are either wrongly mathematically justified (e.g. in the Hoeffding Tree algorithm) or time consuming (e.g. in the McDiarmid Tree algorithm). In this paper we propose a new method which significantly outperforms the McDiarmid Tree algorithm and has a solid mathematical basis. Our method ensures, with a high probability set by the user, that the best attribute chosen in the considered node using a finite data sample is the same as it would be in the case of the whole data stream.

Index Terms—Data steam, decision trees, information gain, Gaussian approximation

1 Introduction

DATA stream mining [1], [4], [8], [10] - [12], [23] became recently a very challenging task in the data mining community. Unlike the static dataset, data stream is of infinite size. Data elements arrive to the system continuously, often with very high rates. Moreover, the concept of data can evolve in time, what is known as the concept drift [2], [3], [9], [16], [20], [29]. For these reasons, commonly known data mining algorithms cannot be directly applied to the data streams. In this paper we focus on a classification task [5], [13], [19], [26], [31] as one of the data mining techniques. The classification process is composed of two steps: learning from the training dataset and labeling the unclassified data. The training dataset $\mathbf{S}$ consists of n elements $s_j, j = 1, \ldots, n$, characterized by D attributes $a^1, \ldots, a^D$. Additionally, one of the K classes is assigned to each data element. Each attribute $a^i,\ i \in \{1, \ldots, D\}$ takes values from the corresponding set A^i. Hence, the training data element s_j can be expressed in the form

$$s_j = \left([v_j^1, \ldots, v_j^D], k_j\right),\ \ v_j^i \in A^i,\ k_j \in \{1, \ldots, K\}, \quad (1)$$

- L. Rutkowski is with the Department of Computer Engineering, Czestochowa University of Technology, ul. Armii Krajowej 36, 42-200 Czestochowa, Poland, and also with Information Technology Institute, Academy of Managment, 90-113 Łódź, Poland (e-mail: lrutko@kik.pcz.pl)
- L. Pietruczuk, P. Duda and M. Jaworski are with the Department of Computer Engineering, Czestochowa University of Technology, ul. Armii Krajowej 36, 42-200 Czestochowa, Poland, (e-mail: lena.pietruczuk@kik.pcz.pl, pduda@kik.pcz.pl, maciej.jaworski@kik.pcz.pl

This paper was prepared under project operated within the Foundation for Polish Science Team Programme co-financed by the EU European Regional Development Fund, Operational Program Innovative Economy 2007-2013, and also supported by National Science Center NCN.

where v_j^i is the value of attribute a^i for data element s_j. The training dataset is used to construct the classifier, which labels the unclassified data elements.

In literature a multitude of classification methods for static data were proposed, e.g. k-nearest neighbors [6], [22], neural networks [14], [27] or decision trees [5], [24], [25]. Classifiers based on decision trees, considered in this paper, seem to be one of the most effective. The decision tree is composed of nodes, branches and leaves. Every node, which is not terminal, is accompanied by some attribute a^i. The tree can be either binary or non-binary. If the tree is binary, the node is split into two children nodes (or leaves). In the second case, the node has as many children as the number of elements of set A^i. Children are connected with their parent nodes by branches. To each branch a value of attribute a^i (in the non-binary case) or some subset of A^i (in the binary case) is assigned. It is obvious, that the non-binary trees make sense only if the attributes take nominal values. Depending on attribute values assigned to the branches, the training set is partitioned into subsets, which are sent down to the corresponding children nodes to continue the process of tree growth. The terminal nodes of the tree, called leaves, are used to assign a class to the unclassified data elements. The key point in constructing the decision tree is choosing the best attribute to split the considered node. In the majority of proposed algorithms, the choice is based on some impurity measure of the dataset. For all the possible partitions of the node, the impurity of the data set before the split and the weighted impurity of the resulting subsets are calculated. The difference of these values is a split measure function. The attribute which gives the highest value of this function is called the best attribute and is assigned to the considered node. In the ID3 algorithm [24], the information entropy is taken as the impurity measure.

The corresponding split measure function is called the information gain or the entropy reduction. The main disadvantage of the ID3 algorithm, which produce non-binary trees, is that it favors the attributes with large domain of possible values. To cope with this problem, the C4.5 algorithm [25] was proposed. The main idea is to introduce the split information function, which penalizes the attributes with large domains. The ratio of the information gain and the split information is proposed as the split measure function in the C4.5 algorithm. The Gini index is another impurity measure worth the consideration. It is used in the CART algorithm [5], which is intended to develop binary trees. Therefore it can be applied to the numerical data as well as to the data with nominal attribute values.

The algorithms mentioned above are designed for static datasets and cannot be directly applied to the data streams. Significant modifications are needed. The dominant problem is to establish the best attribute in each node, since the stream is of infinite size. Given the dataset of n elements in considered node, we want to know if the best attribute computed from this dataset is also the best attribute for the whole data stream, with some fixed probability $1-\delta$. Referring to the only two papers which constitute the 'state of the art' in this subject (for details see section 2), the main and original result of this paper can be summarized as follows,

i) In [28] the authors have recently proposed a method based on the McDiarmid's inequality [21]. However, the choice of the best attribute requires very large number of data elements n in the considered node. The method proposed in this paper, supported by Theorem 1, allows to reduce the value of n dramatically, comparing with [28], with the same probability $1-\delta$. In Example 2 in section 2 the result obtained for this method is almost 4000 times better than result obtained using the McDiarmid's bound.

ii) Another method, based on the Multivariate Delta Method, was proposed in [17]. Although the idea was promising, the result was incorrect and not applicable to the problem of constructing decision trees for data streams. Following partly the authors' idea [17], in our paper we propose the statistical method for determining the best attribute in a node, which ensures the highest value of split measure function with significantly high probability. In our method we use the Taylor's Theorem and properties of the normal distribution [18], [30].

Since the ID3 algorithm provides the background of our method, it will be now briefly presented. The ID3 algorithm is initially intended to produce non-binary trees, however it can be easily transformed to the binary mode. For the needs of this paper we focus only on the binary case, however all the presented methods can be adapted to non-binary trees as well. The algorithm starts with a single node L_0 - the root. During the learning process, in each created node L_q a particular subset $\mathbf{S}_q$ of the training dataset $\mathbf{S}$ is processed (for the root $\mathbf{S}_0 \equiv \mathbf{S}$). If all elements of set $\mathbf{S}_q$ are of the same class, the node is tagged as a leaf and the split is not made. Otherwise, according to the split measure function, the best attribute to split is chosen among the available attributes in considered node. For each available attribute a^i, the set of attribute values A^i is partitioned into two disjoint subsets A^i_L and A^i_R ($A^i = A^i_L \cup A^i_R$). The choice of A^i_L automatically determines the complementary subset A^i_R, therefore the partition is represented further only by A^i_L. The set of all possible partitions of set A^i is denoted by $\mathbf{V}_i$. Subsets A^i_L and A^i_R divide the dataset $\mathbf{S}_q$ into two disjoint subsets: left $\mathbf{L}_q(A^i_L)$ and right $\mathbf{R}_q(A^i_L)$.

$$\mathbf{L}_q(A^i_L) = \{s_j \in \mathbf{S}_q | v^i_j \in A^i_L\}, \tag{2}$$

$$\mathbf{R}_q(A^i_L) = \{s_j \in \mathbf{S}_q | v^i_j \in A^i_R\}. \tag{3}$$

Sets $\mathbf{L}_q(A^i_L)$ and $\mathbf{R}_q(A^i_L)$ depend on a chosen attribute and partition of its values. Let $p_{L,q}(A^i_L)$ ($p_{R,q}(A^i_L)$) denote the fraction of data elements from $\mathbf{S}_q$, which belong to the subset $\mathbf{L}_q(A^i_L)$ ($\mathbf{R}_q(A^i_L)$). Since the fractions $p_{L,q}(A^i_L)$ and $p_{R,q}(A^i_L)$ are dependent, i.e.

$$p_{R,q}(A^i_L) = 1 - p_{L,q}(A^i_L), \tag{4}$$

only one of these parameters is needed to be considered, e.g. $p_{L,q}(A^i_L)$.The fraction of elements from $\mathbf{L}_q(A^i_L)$ ($\mathbf{R}_q(A^i_L)$) from class k, is denoted by $p_{kL,q}(A^i_L)$ ($p_{kR,q}(A^i_L)$). The fraction of all data elements $\mathbf{S}_q$ in considered node L_q, from class k, is denoted by $p_{k,q}$. Note that $p_{k,q}$ are not dependent on chosen attribute a^i and partition A^i_L. As it was mentioned previously, the impurity measure used in the ID3 algorithm is the information entropy. For any subset $\mathbf{S}_q$ of training dataset the entropy is given by

$$Ent(\mathbf{S}_q) = -\sum_{k=1}^{K} p_{k,q} \log_2 p_{k,q}, \tag{5}$$

with condition that if $p_{k,q} = 0$, then $p_{k,q} \log_2 p_{k,q} = 0$. Furthermore, the weighted entropy of subset $\mathbf{S}_q$, resulting from the choice of partition A^i_L, is defined as follows

$$\begin{aligned} wEnt(\mathbf{S}_q, A^i_L) = {} & p_{L,q}(A^i_L) Ent(\mathbf{L}_q(A^i_L)) \\ & + (1 - p_{L,q}(A^i_L) Ent(\mathbf{R}_q(A^i_L)), \end{aligned} \tag{6}$$

where entropies of sets $\mathbf{L}_q(A^i_L)$ and $\mathbf{R}_q(A^i_L)$ are given analogously as in (5), i.e.

$$Ent(\mathbf{L}_q(A^i_L)) = -\sum_{k=1}^{K} p_{kL,q}(A^i_L)) \log_2 p_{kL,q}(A^i_L)), \tag{7}$$

$$Ent(\mathbf{R}_q(A_L^i)) = -\sum_{k=1}^{K} p_{kR,q}(A_L^i)) \log_2 p_{kR,q}(A_L^i)). \quad (8)$$

The information gain, which is used as a split measure function in the ID3 algorithm, is defined as a difference between entropy (5) and the weighted entropy (6). It is dependent on the chosen partition A_L^i of attribute a^i

$$g(\mathbf{S}_q, A_L^i) = Ent(\mathbf{S}_q) - wEnt(\mathbf{S}_q, A_L^i). \quad (9)$$

Among all the possible partitions A_L^i of set A^i, the one which maximizes the value of information gain is chosen

$$\tilde{A}_{L,q}^i = \arg \max_{A_L^i \in \mathbf{V}_i} \{g(\mathbf{S}_q, A_L^i)\}. \quad (10)$$

The partition $\tilde{A}_{L,q}^i$ is called the optimal partition of set A^i for the subset $\mathbf{S}_q$ of training dataset. This optimal partition generates subsets $\mathbf{L}_q^i \equiv \mathbf{L}_q^i(\tilde{A}_{L,q}^i)$ and $\mathbf{R}_q^i \equiv \mathbf{R}_q^i(\tilde{A}_{L,q}^i)$. The value of $g_{i,q} = g(\mathbf{S}_q, \tilde{A}_{L,q}^i)$ is called the information gain of subset $\mathbf{S}_q$ for attribute a^i. Among all the available attributes in the node L_q, the one with the highest value of information gain is chosen. The node L_q is split into two children nodes L_{last+1} and L_{last+2}, where $last$ is the index of the node created recently in the whole tree. Let us assume that the highest value of information gain is obtained for attribute a^x, $x \in \{1, \ldots, D\}$. Then all the calculations described above are performed in node L_{last+1}, using the subset $\mathbf{S}_{last+1} = \mathbf{L}_q^x$, and in node L_{last+2}, using the subset $\mathbf{S}_{last+2} = \mathbf{R}_q^x$. The list of available attributes in nodes L_{last+1} and L_{last+2} is taken from the node L_q, with the exception of the attribute a^x. The considered node L_q is not split if one of the following two conditions occurs:

- the list of available attributes in the node contains only one element,
- all the elements from the subset $\mathbf{S}_q$ are from the same class.

The rest of the paper is organized as follows. The related works on considered subject are reviewed in section 2. In section 3 the main result of this paper is presented and some examples are shown. Section 4 presents the Gaussian Decision Tree (GDT) algorithm, based on the idea proposed in [7]. Experimental results are shown in section 5 and conclusions are described in section 6.

2 Related works

The adaptation of the ID3 algorithm (and any other algorithm based on decision trees) to data streams is a very difficult task. The main problem is to estimate the values of split measure function in each node, since the corresponding subsets of training dataset grow continuously. In the case of data streams, the values of information gain should be calculated, theoretically on a basis of infinite training dataset. Obviously it is not possible, therefore these values have to be estimated from the data sample available in the considered node. As a result, one can decide which attribute is the best only with some probability. In literature few attempts to solve this problem were presented so far.

a) The main result of the work of P. Domingos and G. Hulten was the commonly known algorithm called 'Hoeffdings trees' [7] for mining data streams. It was derived from the Hoeffding's bound [15] which states that with probability $1-\delta$ the true mean of a random variable of range R does not differ from the estimated mean, after n observations, by more than

$$\epsilon_H = \sqrt{\frac{R^2 \ln 1/\delta}{2n}}. \quad (11)$$

However the Hoeffding's bound in not an adequate tool to solve the problem of choosing the attribute according to which the split should be made. In [28] the authors have noticed that it is proper tool only for numerical data ,which does not necessarily have to be met. The second problem is the form of the split measures like information gain and Gini index. Both measures can not be expressed as a sum of elements and they are using only the frequency of elements.

b) In work [17] another method of finding the best attribute was proposed. For convenience, in the following text the situation in one particular node will be considered. Hence the node index q will be omitted in all notations introduced before. Let $\overline{g_x}$ and $\overline{g_y}$ be the values of information gain for attributes a^x and a^y, computed using a data sample in a considered node. Such quantities are random variables, whereas g_x and g_y are their expected values, respectively. The authors of the discussed paper [17] noticed that the value $\overline{g_x}$ can be approximated by a normal distribution

$$\overline{g_x} \longrightarrow N\left(g_x, \frac{\tau_x^2}{n}\right), \quad (12)$$

where n is the number of elements in the sample and τ_x^2/n is a variance of this distribution (discussed later). Therefore, the difference $\overline{g_x} - \overline{g_y}$ can be approximated by the following normal distribution

$$\overline{g_x} - \overline{g_y} \longrightarrow N\left(g_x - g_y, \frac{\tau_x^2 + \tau_y^2}{n}\right). \quad (13)$$

To decide, if the attribute a^x gives higher value of information gain than attribute a^y, the authors proposed an appropriate statistical test, based on distribution (13). To justify approximation (12), the authors made use of the Multivariate Delta

Method.

Multivariate Delta Method

Let $X_1, \ldots, X_n$ be a random sample. Let $X_i = X_{1i}, \ldots, X_{pi}$. Further, let $E(X_{ij}) = \mu_i$ and $Cov(X_{ik}, X_{jk}) = \sigma_{ij}$. Let $\overline{X}_i$ be the mean of $X_{i1}, X_{i2}, \ldots, X_{in}$ and let $\vec{\mu} = (\mu_1, \ldots, \mu_p)$. For a given function h with continuous first partial derivatives, we have

$$h(\overline{X}_1, \ldots, \overline{X}_p) - h(\vec{\mu}) \to N(0, \tau^2/n), \tag{14}$$

where

$$\tau^2 = \sum\sum \sigma_{ij} \frac{\partial h(\vec{\mu})}{\partial \mu_i} \frac{\partial h(\vec{\mu})}{\partial \mu_j}. \tag{15}$$

Particularly, the information gain function g is an example of function h. The authors considered only two-class problem for binary trees. In this case, for a chosen attribute, $p = 3$:

(i) $X_{1i} = 1$ if the i-th element passes through the left branch and $X_{1i} = 0$ otherwise,
(ii) $X_{2i} = 1$ if $X_{1i} = 1$ and the i-th element is from the first class and $X_{2i} = 0$ if $X_{1i} = 1$ and the i-th element is from the second class
(iii) $X_{3i} = 1$ if $X_{1i} = 0$ and the i-th element is from the first class, $X_{3i} = 0$ if $X_{1i} = 0$ and the i-th element is from the second class.

Therefore, g is a function of three variables:

(i) p_L - fraction of elements passing through the left branch,
(ii) p_{1L} - fraction of elements passing through the left branch, from the first class,
(iii) p_{1R} - fraction of elements passing through the right branch, from the first class.

The estimated values of p_L, p_{1L} and p_{1R}, calculated from the sample, are denoted by $\overline{p_L}$, $\overline{p_{1L}}$ and $\overline{p_{1R}}$, respectively. To prove the approximation (12), the authors in [17] proposed the following lemma

Lemma 1: Let n be the sample size and N be the normal distribution. Then, for the entropy function g, we have[1]

$$\overline{g_x} = g(\overline{p_L}, \overline{p_{1L}}, \overline{p_{1R}}) \to N\left(g_x, \tau_x^2/n\right), \tag{16}$$

where

$$\tau_x^2 = \left(\frac{\partial g}{\partial p_L}\right)^2 p_L(1-p_L) + \left(\frac{\partial g}{\partial p_{1L}}\right)^2 p_{1L}(1-p_{1L}) + \left(\frac{\partial g}{\partial p_{1R}}\right)^2 p_{1R}(1-p_{1R}). \tag{17}$$

However, the use of the Multivariate Delta Method to prove the above lemma (equivalent to approximation (12)) requires that $\overline{p_L} = \overline{X}_1$, $\overline{p_{1L}} = \overline{X}_2$ and $\overline{p_{1R}} = \overline{X}_3$. Unfortunately, authors do not say clearly how to calculate $\overline{p_{1L}}$ and $\overline{p_{1R}}$ from the sample. There are two possibilities:

- $\overline{p_{1L}}$ is calculated only from n_L elements, which pass through the left branch ($n_L < n$), and $\overline{p_{1R}}$ is calculated from the remaining $n - n_L$ elements. In this case, approximation (16), with τ_x^2 given by (17), is incorrect, because the assumption of the Multivariate Delta Method is not satisfied ($\overline{p_{1L}} \neq \overline{X}_2$ and $\overline{p_{1R}} \neq \overline{X}_3$),
- $\overline{p_{1L}}$ and $\overline{p_{1R}}$ are calculated from the whole sample of n elements. But, in this case, we do not know what are the values of X_{2i} when $X_{1i} = 0$ (the i-th element passes through the right branch) and X_{3i} when $X_{1i} = 1$ (the i-th element passes through the left branch). Moreover, no matter what value they would take, $\overline{p_{1L}} = \overline{X}_2$ and $\overline{p_{1R}} = \overline{X}_3$ are not estimators of p_{1L} and p_{1R}. Then $g(\overline{p_L}, \overline{p_{1L}}, \overline{p_{1R}})$ can not be a split measure function.

In the reasoning of the authors there is one more weak point. They omitted the issue of computing τ_x^2. Formula (17) requires the knowledge of p_L, p_{1L} and p_{1R}. In general case these values are unknown. One way to solve this problem is to estimate τ_x^2 from the sample, but the authors do not even propose such an estimator. Our solution to this problem is to find an upper bound of τ_x^2.

c) In [28] the authors have worked on correcting the mathematical foundations of the Hoeffding's trees. They propose the McDiarmid's inequality to solve the problem of choosing the best attribute to make a split in the node. Authors define the function $f(\mathbf{S})$ as

$$f(\mathbf{S}) = Gain_{a^x}(\mathbf{S}) - Gain_{a^y}(\mathbf{S}) > 0. \tag{18}$$

The main result of their paper is the *Theorem 1* which states, that for any fixed δ and any attributes a^x and a^y, if

$$\epsilon_M = C_{Gain}(K, n)\sqrt{\frac{ln(1/\delta)}{2n}}, \tag{19}$$

then

$$Pr(f(\mathbf{S}) - E[f(\mathbf{S})] > \epsilon_M) \leq \delta, \tag{20}$$

where K is the number of classes and

$$C_{Gain}(K, n) = 6(K \log_2 eN + \log_2 2N) + 2\log_2 K. \tag{21}$$

It means that for any fixed δ, if $f(\mathbf{S}) > \epsilon_M$, then attribute a^x is better to split than attribute a^y with probability $1 - \delta$.

3 MAIN RESULT

In this paper we partly follow the idea presented in [17]. In this section we describe a method for

1. The authors faultily named the function g as the entropy function. Actually in this case g is the information gain function.

determining the upper bound of τ_x, given by (17), for each attribute a^x. This allows to propose a statistical test, used to determine which attribute is best to split the node, with some probability given by the user. For convenience we limit ourselves to the case of two classes, i.e. $K = 2$. This simplification can significantly facilitate the calculations. In the two-class case, analogously to condition (4), the following constraints are true as well

$$p_{2L}(A_L^i) = 1 - p_{1L}(A_L^i), \tag{22}$$

$$p_{2R}(A_L^i) = 1 - p_{1R}(A_L^i), \tag{23}$$

$$p_2 = 1 - p_1. \tag{24}$$

Therefore, only three of these six parameters are considered further, i.e. $p_{1L}(A_L^i)$, $p_{1R}(A_L^i)$ and p_1. Note that p_1 does not depend on chosen attribute a^i and partition A_L^i, however it can be expressed as a function of $p_L(A_L^i)$, $p_{1L}(A_L^i)$ and $p_{1R}(A_L^i)$ as follows

$$\begin{aligned} p_1(p_L, p_{1L}, p_{1R}) = & \, p_L(A_L^i)p_{1L}(A_L^i) \\ & + (1 - p_L(A_L^i))p_{1R}(A_L^i). \end{aligned} \tag{25}$$

For any dataset $\mathbf{X}$, consisting of elements belonging to one of two classes, its entropy can be calculated using the following formula

$$Ent(P_1) = -P_1 \log_2 P_1 - (1 - P_1)\log_2(1 - P_1), \tag{26}$$

where P_1 denotes the fraction of elements from $\mathbf{X}$ of the first class. Therefore entropies (5), (7) and (8) became the functions of one variable, i.e. $Ent(p_1)$ for (5), $Ent(p_{1L}(A_L^i))$ for (7) and $Ent(p_{1R}(A_L^i))$ for (8). The information gain for attribute a^i and for subset A_L^i can be expressed as a function of the three parameters mentioned above

$$\begin{aligned} & g(A_L^i) = g(p_L(A_L^i), p_{1L}(A_L^i), p_{1R}(A_L^i)) \\ & = Ent(p_1(p_L(A_L^i), p_{1L}(A_L^i), p_{1R}(A_L^i))) \\ & - p_L(A_L^i)Ent(p_{1L}(A_L^i)) \\ & - (1 - p_L(A_L^i))Ent(p_{1R}(A_L^i)), \end{aligned} \tag{27}$$

as we omit $\mathbf{S}_q$ in equation (9).
The optimal partition of set A^i is established analogously as in (10)

$$\tilde{A}_L^i = \arg\max_{A_L^i}\{g((p_L(A_L^i), p_{1L}(A_L^i), p_{1R}(A_L^i)\}. \tag{28}$$

The following notation for fractions associated with the optimal partition is introduced:

$$\left[p_L^i, p_{1L}^i, p_{1R}^i\right] = \left[p_L(\tilde{A}_L^i), p_{1L}(\tilde{A}_L^i), p_{1R}(\tilde{A}_L^i)\right] \tag{29}$$

The value of $g_i = g(p_L^i, p_{1L}^i, p_{1R}^i)$ is called the information gain for attribute a^i.

Parameters $\overline{p_L^i}$, $\overline{p_{1L}^i}$ and $\overline{p_{1R}^i}$ are estimators of p_L^i, p_{1L}^i and p_{1R}^i respectively. They can be treated as appropriate arithmetic means of some independent random variables from binomial distributions. Let us consider the data elements s_j from data set $\mathbf{S}$, $j \in \{1, ..., n\}$. We define the random variable $\zeta_L^{i,j}$, which is equal to 1 if $s_j \in \mathbf{L}^i$ and 0 otherwise. Variable $\zeta_L^{i,j}$ is from binomial distribution with mean $\mu_L^i = p_L^i$ and variance $\left(\sigma_L^i\right)^2 = p_L^i(1 - p_L^i)$. Similarly we define random variables $\zeta_{1L}^{i,j}$ (for elements l^j from set $\mathbf{L}^i$, $j \in \{1, ..., n_L^i\}$) and $\zeta_{1R}^{i,j}$ (for elements r^j from set $\mathbf{R}^i$, $j \in \{1, ..., n_R^i\}$), from binomial distributions with means $\mu_{1L}^i = p_{1L}^i$, $\mu_{1R}^i = p_{1R}^i$ and variances $\left(\sigma_{1L}^i\right)^2 = p_{1L}^i(1 - p_{1L}^i)$, $\left(\sigma_{1R}^i\right)^2 = p_{1R}^i(1 - p_{1R}^i)$, respectively. Variable $\zeta_{1L}^{i,j}$ is equal to 1 if l^j is from the first class and $\zeta_{1R}^{i,j}$ equals 1 if r^j is from the first class.

Lemma 2: If function $g(p_L, p_{1L}, p_{1R})$ is an information gain, given by formula (27), then the value $\overline{g_i} = g(\overline{p_L^i}, \overline{p_{1L}^i}, \overline{p_{1R}^i})$ can be approximated by normal distribution

$$\overline{g_i} \longrightarrow N\left(g_i, \frac{\left(\tau_L^i\right)^2}{n} + \frac{\left(\tau_{1L}^i\right)^2}{n_L^i} + \frac{\left(\tau_{1R}^i\right)^2}{n_R^i}\right), \tag{30}$$

where

$$\left(\tau_L^i\right)^2 = \left(\frac{\partial g}{\partial p_L}\right)^2 \left(\sigma_L^i\right)^2, \tag{31}$$

$$\left(\tau_{1L}^i\right)^2 = \left(\frac{\partial g}{\partial p_{1L}}\right)^2 \left(\sigma_{1L}^i\right)^2, \tag{32}$$

$$\left(\tau_{1R}^i\right)^2 = \left(\frac{\partial g}{\partial p_{1R}}\right)^2 \left(\sigma_{1R}^i\right)^2. \tag{33}$$

Proof: Parameters p_L^i, p_{1L}^i and p_{1R}^i are estimated from the data sample in considered leaf node as follows

$$\overline{p_L^i} = \frac{\sum_{j=1}^{n} \zeta_L^{i,j}}{n}, \tag{34}$$

$$\overline{p_{1L}^i} = \frac{\sum_{j=1}^{n_L^i} \zeta_{1L}^{i,j}}{n_L^i}, \tag{35}$$

$$\overline{p_{1R}^i} = \frac{\sum_{j=1}^{n_R^i} \zeta_{1R}^{i,j}}{n_R^i}, \tag{36}$$

For big values of n, n_L^i and n_R^i distributions of $\overline{p_L^i}$, $\overline{p_{1L}^i}$ and $\overline{p_{1R}^i}$, according to the Central Limit Theorem, can be approximated by appropriate normal distributions

$$\overline{p_L^i} \longrightarrow N\left(\mu_L^i, \frac{\left(\sigma_L^i\right)^2}{n}\right), \tag{37}$$

$$\overline{p_{1L}^i} \longrightarrow N\left(\mu_{1L}^i, \frac{\left(\sigma_{1L}^i\right)^2}{n_L^i}\right), \tag{38}$$

$$\overline{p_{1R}^i} \longrightarrow N\left(\mu_{1R}^i, \frac{\left(\sigma_{1R}^i\right)^2}{n_R^i}\right). \tag{39}$$

For big values of n, n_L^i and n_R^i one can assume, that the values of $\overline{p_L^i}$, $\overline{p_{1L}^i}$ and $\overline{p_{1R}^i}$ are very close to their expected values μ_L^i, μ_{1L}^i and μ_{1R}^i. Therefore one can apply the Taylor Theorem for information gain in neighborhood of point $(\overline{p_L}, \overline{p_{1L}}, \overline{p_{1R}}) = (\mu_L^i, \mu_{1L}^i, \mu_{1R}^i)$ as follows

$$\begin{aligned} g(\overline{p_L^i}, \overline{p_{1L}^i}, \overline{p_{1R}^i}) \approx\ & g(\mu_L^i, \mu_{1L}^i, \mu_{1R}^i) \\ &+ \frac{\partial g(\mu_L^i, \mu_{1L}^i, \mu_{1R}^i)}{\partial p_L}(\overline{p_L^i} - \mu_L^i) \\ &+ \frac{\partial g(\mu_L^i, \mu_{1L}^i, \mu_{1R}^i)}{\partial p_{1L}}(\overline{p_{1L}^i} - \mu_{1L}^i) \\ &+ \frac{\partial g(\mu_L^i, \mu_{1L}^i, \mu_{1R}^i)}{\partial p_{1R}}(\overline{p_{1R}^i} - \mu_{1R}^i). \end{aligned} \tag{40}$$

Therefore, according to (37), (38), (39) and (40), the distribution of random variable $g(\overline{p_L^i}, \overline{p_{1L}^i}, \overline{p_{1R}^i})$ can be approximated by normal distribution as in (30). This completes the proof. □

Knowing that $n_L^i = n\overline{p_L}$ and $n_R^i = n(1 - \overline{p_L})$, the variance of normal distribution (30) can be further simplified as follows

$$\begin{aligned} \frac{\left(\tau_L^i\right)^2}{n} + \frac{\left(\tau_{1L}^i\right)^2}{n_L^i} + \frac{\left(\tau_{1R}^i\right)^2}{n_R^i} &= \frac{\left(\tau_L^i\right)^2 + \frac{\left(\tau_{1L}^i\right)^2}{\overline{p_L^i}} + \frac{\left(\tau_{1R}^i\right)^2}{1-\overline{p_L^i}}}{n} \\ &= \frac{\left(\tau_i\right)^2}{n}. \end{aligned} \tag{41}$$

Convergence (30) can be rewritten as follows

$$\overline{g_i} \longrightarrow N\left(g_i, \frac{\left(\tau_i\right)^2}{n}\right). \tag{42}$$

Let us define a threshold th as a fraction of elements of the first class (p_1), below which we assume that there is no need for splitting the node. Therefore, all the calculations are performed only if the following inequality is satisfied

$$\frac{1 - p_1}{p_1} \leq \frac{1 - th}{th} = C. \tag{43}$$

For example, for threshold value $th = 0,05$, C is equal to 19. To simplify the form of the following theorem, we introduce $Q(C)$ as

$$Q(C) = \frac{\log_2^2 e^2}{Ce^2} + \frac{\log_2^2 e^2}{e^2} + \frac{\log_2 e}{e} + \frac{\log_2^2(2C)}{4}. \tag{44}$$

Theorem 1: Let us consider two attributes a^x and a^y, for which we have calculated the values of information gain $\overline{g_x} = g(\overline{p_L^x}, \overline{p_{1L}^x}, \overline{p_{1R}^x})$ and $\overline{g_y} = g(\overline{p_L^y}, \overline{p_{1L}^y}, \overline{p_{1R}^y})$. If the difference of these values satisfies the following condition

$$\overline{g_x} - \overline{g_y} > \epsilon, \tag{45}$$

where

$$\epsilon = z_{(1-\delta)}\frac{\sqrt{2Q(C)}}{\sqrt{n}} \tag{46}$$

and $z_{(1-\delta)}$ is the $(1-\delta)$-th quantile of the standard normal distribution $N(0,1)$, then g_x is greater than g_y with probability $1-\delta$. If a^x and a^y are attributes with the highest values of information gain, then a^x can be chosen to split the considered leaf node, with the level of confidence $1-\delta$.

Proof: The difference of values $\overline{g_x}$ and $\overline{g_y}$ is from the normal distribution

$$\overline{g_x} - \overline{g_y} \longrightarrow N\left(g_x - g_y, \frac{\left(\tau_x\right)^2 + \left(\tau_y\right)^2}{n}\right).$$

We do not know the true value of mean $g_x - g_y$. According to properties of the normal distribution [30], the following inequality is satisfied with probability $1-\delta$

$$\overline{g_x} - \overline{g_y} < \epsilon_{(1-\delta)}^{x,y} + (g_x - g_y), \tag{47}$$

where

$$\epsilon_{(1-\delta)}^{x,y} = z_{(1-\delta)}\frac{\sqrt{\left(\tau_x\right)^2 + \left(\tau_y\right)^2}}{\sqrt{n}}. \tag{48}$$

Obviously inequality (47) can be expressed in the following form

$$g_x - g_y > (\overline{g_x} - \overline{g_y}) - \epsilon_{(1-\delta)}^{x,y} \tag{49}$$

This leads to the conclusion: If

$$\epsilon_{(1-\delta)}^{x,y} < \overline{g_x} - \overline{g_y}, \tag{50}$$

then

$$g_x - g_y > 0 \tag{51}$$

with probability $1-\delta$. We will show that ϵ given by (46) is the upper bound of $\epsilon_{(1-\delta)}^{x,y}$.

$$\epsilon_{(1-\delta)}^{x,y} \leq \epsilon \tag{52}$$

In consequence, if

$$\overline{g_x} - \overline{g_y} > \epsilon \tag{53}$$

then inequality (51) is still satisfied with probability $1-\delta$. Therefore, one can say that if the inequality (53) is true, then $g_x > g_y$ with probability $1-\delta$. Now we will prove inequality (52).

From (27) we have

$$\begin{aligned}\frac{\partial g}{\partial p_L} =& -(p_{1L}-p_{1R})\log_2 p_1 \\ &-(p_{1R}-p_{1L})\log_2 e(1-p_1)+p_{1L}\log_2 p_{1L} \\ &-(1-p_{1L})\log_2(1-p_{1L}) \\ &-p_{1R}\log_2 p_{1R}+(1-p_{1R})\log_2(1-p_{1R}).\end{aligned} \tag{54}$$

If the fraction of elements of the first class (p_1) is lower than a certain threshold th (or equivalently higher than $1-th$), then we assume that there is no need for splitting the tree node. Therefore, all the calculations are performed only if the following inequality is satisfied

$$th < p_1 < 1-th. \tag{55}$$

According to inequality (43), we can bound the absolute value of partial derivative in (54) as follows

$$\left|\frac{\partial g}{\partial p_L}\right| \le \left|(p_{1R}-p_{1L})\log_2\frac{p_1}{1-p_1}\right. \tag{56}$$

$$\left. - Ent(p_{1L}) + Ent(p_{1R})\right| \tag{57}$$

$$\begin{aligned}&\le |(p_{1L}-p_{1R})|\log_2 C \\ &\quad + |Ent(p_{1L})-Ent(p_{1R})|\end{aligned} \tag{58}$$

$$\le \log_2 C + 1 \tag{59}$$

Therefore

$$\left(\frac{\partial g}{\partial p_L}\right)^2 p_L(1-p_L) \le (\log_2 C+1)^2\frac{1}{4} = \frac{(\log_2 2C)^2}{4} \tag{60}$$

This inequality is satisfied for every attribute. Therefore, for any chosen attribute a^i, we obtain a bound for the value of τ_L^i (see equation (31))

$$(\tau_L^i)^2 \le \frac{(\log_2 2C)^2}{4}. \tag{61}$$

According to (26), we have

$$\begin{aligned}\frac{\partial Ent(p_1)}{\partial p_{1L}} &= -\frac{\partial p_1}{\partial p_{1L}}\log_2 ep_1 - \frac{\partial p_2}{\partial p_{1L}}\log_2 ep_2 \\ &= -p_L\log_2 ep_1 + p_L\log_2 ep_2 \\ &= p_L\log\frac{1-p_1}{p_1}.\end{aligned} \tag{62}$$

Therefore

$$\begin{aligned}\frac{\partial g}{\partial p_{1L}} &= p_L\log_2\frac{1-p_1}{p_1} + p_L\Bigg[\log_2 p_{1L}+\log_2 e \\ &\quad -\frac{\log_2 e}{1-p_{1L}} - \log_2(1-p_{1L}) + \frac{p_{1L}\log_2 e}{1-p_{1L}}\Bigg] \\ &= p_L\log_2\frac{1-p_1}{p_1} + p_L\Bigg[\log_2 p_{1L}+\log_2 e \\ &\quad -\frac{(1-p_{1L})\log_2 e}{1-p_{1L}} - \log_2(1-p_{1L})\Bigg] \\ &= p_L\log_2\frac{1-p_1}{p_1} + p_L\log_2\frac{p_{1L}}{1-p_{1L}} \\ &= p_L\log_2\left[\frac{1-p_1}{p_1}\cdot\frac{p_{1L}}{1-p_{1L}}\right]\end{aligned} \tag{63}$$

Also, according to (26), we have

$$\begin{aligned}\frac{\partial Ent(p_1)}{\partial p_{1R}} &= -\frac{\partial p_1}{\partial p_{1R}}\log_2 ep_1 - \frac{\partial p_2}{\partial p_{1R}}\log_2 ep_2 \\ &= -(1-p_L)\log_2 ep_1 + (1-p_L)\log_2 ep_2 \\ &= (1-p_L)\log_2\frac{1-p_1}{p_1}.\end{aligned} \tag{64}$$

Therefore

$$\begin{aligned}\frac{\partial g}{\partial p_{1R}} &= (1-p_{1L})\log_2\frac{1-p_1}{p_1} + (1-p_L)\log_2\frac{p_{1R}}{1-p_{1R}} \\ &= (1-p_L)\log_2\left[\frac{1-p_1}{p_1}\cdot\frac{p_{1L}}{1-p_{1L}}\right]\end{aligned} \tag{65}$$

Combining (32) and (63), for any attribute a^i, $i \in \{1,\ldots,D\}$, we get

$$\frac{\left(\tau_{1L}^i\right)^2}{p_L^i} = \frac{(p_L^i)^2\log_2^2\left[\frac{1-p_1^i}{p_1^i}\frac{p_{1L}^i}{1-p_{1L}^i}\right]p_{1L}^i(1-p_{1L}^i)}{p_L^i}. \tag{66}$$

Using C given by (43), the term (66) can be bounded by

$$\frac{\left(\tau_{1L}^i\right)^2}{p_L^i} \le p_L^i\left[\log_2(Cp_{1L}^i) - log_2(1-p_{1L}^i)\right]^2 p_{1L}^i(1-p_{1L}^i). \tag{67}$$

Developing the term in brackets from (67) we obtain

$$\begin{aligned}\frac{\left(\tau_{1L}^i\right)^2}{p_L^i} \le p_L^i\Bigg[&log_2^2(Cp_{1L}^i) - 2\log_2(Cp_{1L}^i)\log_2(1-p_{1L}^i) \\ &+ \log_2^2(1-p_{1L}^i)\Bigg]p_{1L}^i(1-p_{1L}^i).\end{aligned} \tag{68}$$

The term $\log_2^2(Cp_{1L}^i)p_{1L}^i$ in interval $[0;1]$ takes its maximum value for $p_{1L}^i = \frac{1}{Ce^2}$, therefore it can be bounded by

$$\log_2^2(Cp_{1L}^i)p_{1L}^i \le \frac{\log_2^2 e^2}{Ce^2}. \tag{69}$$

The term $\log_2^2(1-p_{1L}^i)(1-p_{1L}^i)$ takes its maximum value for $(1-p_{1L}^i) = 1/e^2$. Hence the following bound is true

$$\log_2^2(1-p_{1L}^i)(1-p_{1L}^i) \le \frac{\log_2^2 e^2}{e^2}. \tag{70}$$

The term $-\log_2(1-p_{1L}^i)(1-p_{1L}^i)$ has maximum for $(1-p_{1L}^i) = 1/e$, therefore we have

$$-\log_2(1-p_{1L}^i)(1-p_{1L}^i) \le \frac{\log_2 e}{e}. \tag{71}$$

Combining bounds (68), (69), (70) and (71) we get the following inequality

$$\frac{\left(\tau_{1L}^i\right)^2}{p_L^i} \leq p_L^i \left[\frac{\log_2^2 e^2}{Ce^2} + \frac{\log_2^2 e^2}{e^2} + \frac{\log_2 e}{e}\right]. \quad (72)$$

In the same way, the similar bound for term (33) can be obtained

$$\frac{\left(\tau_{1R}^i\right)^2}{1-p_L^i} \leq (1-p_L^i) \left[\frac{\log_2^2 e^2}{Ce^2} + \frac{\log_2^2 e^2}{e^2} + \frac{\log_2 e}{e}\right]. \quad (73)$$

Combining bounds (61), (72) and (73) one can obtain the bound for $(\tau_i)^2$

$$\begin{aligned}(\tau_i)^2 \leq & \frac{\log_2^2(2C)}{4} + \left[\frac{\log_2^2 e^2}{Ce^2} + \frac{\log_2^2 e^2}{e^2} + \frac{\log_2 e}{e}\right] \\ = & Q(C).\end{aligned} \quad (74)$$

The inequality (74) holds for any attribute a^i, in particular for $a^i = a^x$ and $a^i = a^y$. Therefore we establish the upper bound for $\epsilon_{(1-\delta)}^{x,y}$, given by (48), as follows

$$\epsilon_{(1-\delta)}^{x,y} \leq z_{(1-\delta)} \frac{\sqrt{2Q(C)}}{\sqrt{n}}. \quad (75)$$

□

EXAMPLE 1

Let us consider a two class case with threshold $th = 0,01$. Suppose that after 10000 observations the difference $\overline{g_x} - \overline{g_y}$ was equal to 0,09371. We want to know if we should make a split with 0,95 level of confidence. From normal distribution tables we get $z_{(0,95)} = 1,64854$. First we need to calculate C

$$C = \frac{0,99}{0,01} = 99.$$

After calculating C we can compute the value of function $Q(C)$

$$\begin{aligned}Q(C) = & \frac{\log_2^2 e^2}{99e^2} + \frac{\log_2^2 e^2}{e^2} + \frac{\log_2 e}{e} + \frac{\log_2^2(2 \cdot 99)}{4} \\ \approx & 16,22062.\end{aligned}$$

Then, according to (45), we obtain

$$\begin{aligned}z_{(1-\delta)} \frac{\sqrt{2Q(C)}}{\sqrt{n}} \approx & 1,64854 \frac{\sqrt{2 \cdot 16,22062}}{\sqrt{10000}} \\ \approx & 0,093894 > \overline{g_x} - \overline{g_y} = 0,09371.\end{aligned}$$

Therefore we should not yet make a split. Assume that after 10100 observations the difference $\overline{g_x} - \overline{g_y}$ was equal to 0,09356. Now we get

$$\begin{aligned}z_{(1-\delta)} \frac{\sqrt{2Q(C)}}{\sqrt{n}} \approx & 16,22062 \frac{\sqrt{2 \cdot 1,22062}}{\sqrt{10100}} \\ \approx & 0,093429 < \overline{g_x} - \overline{g_y}.\end{aligned}$$

This time our result shows, that with 0,95 level of confidence, g_x is greater than g_y, so we can make a split.

EXAMPLE 2

In this example we compare our method with the McDiarmid bound for information gain [28]. We will calculate the number of data elements needed to make a split. Let us consider the following situation. We calculated the difference $Gain_{a^x}(Z) - Gain_{a^y}(Z) = \overline{g_x} - \overline{g_y} = 0,2479$ and we want to know, with probability 95%, if he should make a split. The number of classes is $K = 2$ and the threshold th is 0,02.

a) First we use the McDiarmid bound for information gain. According to inequality (20), n is sufficient number if $f(Z) = Gain_{a^x}(Z) - Gain_{a^y}(Z) > \epsilon$, where ϵ is given by (19). From equation (21) we can not compute analytically the number of data elements. However we can solve this problem numerically. In this case, for $n = 4'345'299$ obtained ϵ was equal to 0,2479000136. This value is greater than $f(Z)$. Therefore the number of data is too small. However for $n = 4'345'300$ the value of ϵ was 0,2478999886. This value is smaller than $f(Z)$ and therefore n is large enough to say that attribute x is better to split than y, with probability 95%.
b) Now we will compute the sufficient number of data elements using method presented in this paper. First we have to calculate C and $Q(C)$ according to (43) and (44), i.e.

$$C = \frac{1-0,2}{0,02} = 49, \quad (76)$$

$$Q(C) \approx 12,61905957. \quad (77)$$

Using inequality (45) we calculate n as follows

$$\overline{g_x} - \overline{g_y} > z_{(1-\delta)} \frac{\sqrt{2Q(C)}}{\sqrt{n}},$$

$$\sqrt{n} > z_{(1-\delta)} \frac{\sqrt{2Q(C)}}{\overline{g_x} - \overline{g_y}},$$

$$n > z_{(1-\delta)}^2 \frac{2Q(C)}{(\overline{g_x} - \overline{g_y})^2}.$$

$$\begin{aligned}z_{(1-\delta)}^2 \frac{2Q(C)}{(\overline{g_x} - \overline{g_y})^2} \approx & (1,64854)^2 \frac{2 \cdot 12,61905957}{(0,2479)^2} \\ \approx & 1116,099494\end{aligned}$$

Therefore if the number of data elements is grater than 1116 then, with probability 95%, a split should be made.

As we can see the obtained number of data elements in the case of the McDiarmid bound is incomparably higher than obtained based on our new method.

4 The Gaussian Decision Tree Algorithm

Theorem 1 allows us to propose an algorithm called *Gaussian Decision Tree (GDT)*. This algorithm is a modification of the *Hoeffding Tree algorithm* proposed in [7]. For clarity of the pseudocode the following notation will be introduced:

- $\overline{g_{i,q}}$ is the information gain computed for the attribute a^i in the leaf L_q.
- $n^k_{i,\lambda,q}$ is the number of elements from the k-th class in the leaf L_q, for which the value of attribute a^i is equal to a^i_λ.
- n^k_q is the number of elements from the k-th class in the leaf L_q.

Table 1: The GDT

Inputs: **S** is a sequence of examples,
$\mathscr{A}$ is a set of discrete attributes,
th is a minimal fraction of elements belonging to the one class,
δ is one minus the desired probability of choosing the correct attribute at any given node.
Output: GDT is a decision tree.

Procedure GDT($\mathbf{S}, \mathscr{A}, th, \delta$)
Let GDT be a tree with a single leaf L_0 (the root).
Let $\mathscr{A}_0 = \mathscr{A}$
Let $C = \frac{1-th}{th}$, $last = 0$
For each attribute $a^i \in \mathscr{A}$
  For each value a^i_λ of attribute a^i
    $n^1_{i,\lambda,0} = 0$, $n^2_{i,\lambda,0} = 0$.
For each example s in **S**
  Sort s into a leaf L_q using GDT.
  For each attribute $a^i \in \mathscr{A}_q$
    For each value a^i_λ of attribute a^i
      For each class $k = 1, 2$
        If value of example s for attribute a^i is equal to a^i_λ and s is from the k-th class then
          Increment $n^k_{i,\lambda,q}$.
  Label L_q with the majority class among the examples seen so far at L_q.
  If $\frac{\max\{n^1_q, n^2_q\}}{\min\{n^1_q, n^2_q\}} < C$ then
    For each attribute $a^i \in \mathscr{A}_l$
      For each partition of the set A^i into A^i_L, A^i_R
        Compute $\overline{g_q}(A^i_L)$ using the counts $n^k_{i,\lambda,q}$, $k = 1, 2$.
      $\overline{g_{i,q}} = \max_{A^i_L \in \mathbf{V}_i} \{\overline{g_q}(A^i_L)\}$
    $a^x = \arg\max_{a^i \in \mathscr{A}_q} \{\overline{g_{i,q}}\}$
    $a^y = \arg\max_{a^i \in \mathscr{A}_q \setminus \{a^y\}} \{\overline{g_{i,q}}\}$
    Compute ϵ using formula (46)
    If $\overline{g_{x,q}} - \overline{g_{y,q}} > \epsilon$, then
      Replace L_q by an internal node that splits on a^x.
      For both branches of the split
        Add a new leaf L_{last+1} and let $\mathscr{A}_{last+1} = \mathscr{A}_q \setminus \{a^x\}$ at L_{last+1}.
        For each attribute $a^i \in \mathscr{A}_{last+1}$
          For each value a^i_λ of a^i
            $n^1_{i,\lambda,last+1} = 0$, $n^2_{i,\lambda,last+1} = 0$.
        $last = last + 1$
Return GDT.

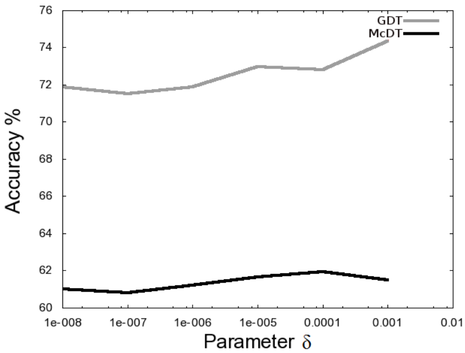

Fig. 1. The dependence between the value of parameter δ and the accuracy of the Gaussian Decision Tree and the McDiarmid Tree algorithms

5 Experimental Results

In this section the performance of the proposed method is examined and compared with the McDiarmid Tree algorithm [28] and the ID3 algorithm [24]. Synthetic data were used, generated on a basis of synthetic decision trees. These synthetic trees were constructed in the same way as described in [7]. At each level of the tree, after the first d_{min} levels, each node is replaced by a leaf with probability ω. To the rest of nodes a splitting attribute is randomly assigned; it has to be an attribute which has not already occurred in the path from the root to the considered node. The maximum depth of the synthetic tree is d_{max} (at this level all nodes are replaced by leaves). After the whole tree is constructed, to each leave a class is randomly assigned. Each synthetic tree represents a different data concept. Data concept is a particular distribution of attributes values and classes. In this work twelve synthetic trees were generated (all of them with $\omega = 0,15$, $d_{min} = 3$ and $d_{max} = 18$) giving twelve different data concepts. The value of parameter th was set to 0,05. In the following simulations, for any set of Gaussian Decision Tree parameters (δ, n), algorithm was run twelve times, once for each synthetic data concept. The final result was obtained as the average over all runs.

First we examine the performance of the Gaussian Decision Tree algorithm depending on the values of parameter δ and compare it with the performance of the McDiarmid Tree for the same values of parameter δ. The experiment was performed on synthetic dataset

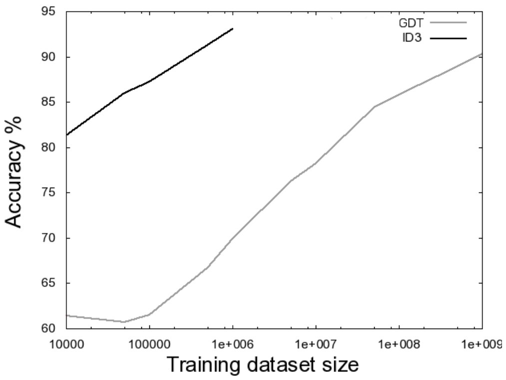

Fig. 2. The dependence between the number of training data and the accuracy of the Gaussian Decision Tree and the ID3 algorithms

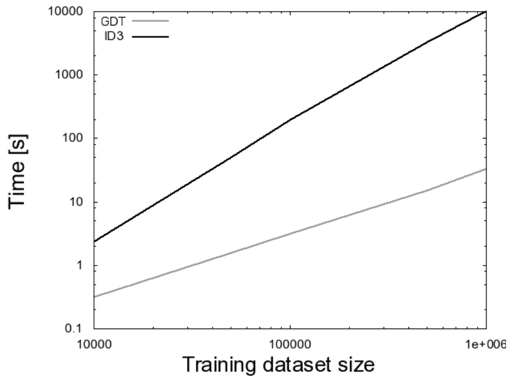

Fig. 3. The dependence between the number of training data and the processing time of Gaussian Decision Tree and the ID3 algorithms

of size $n = 10^8$. In Figure 1 we can see that the accuracy of the Gaussian Decision Tree algorithm is always better than the accuracy of the McDiarmid Tree algorithm. It happens because there are not enough data elements for the McDiarmid algorithm to make a split. Therefore only the root is created. However for $n = 10^8$ and $th = 0,05$ the Gaussian Decision Tree algorithm creates more complex tree. We can also observe that with growing value of δ the accuracy increases (in the considered interval) but the changes are small (about 4%).

In the second experiment we compare the accuracy of the Gaussian Decision Tree algorithm and the ID3 algorithm. Taking into account the results of the previous experiment we decided to add a new mechanism. The tie breaking parameter θ was added. Then, if $\epsilon < \theta$, the split is made. Idea of this mechanism was introduced in the VFDT system [7]. For this simulation the value of parameter θ was set to $0,05$ and the value of parameter δ was set to 10^{-7}. The experiment was performed on different numbers of training data elements. The Gaussian Decision Tree algorithm was tested on dataset size varying from $n = 10^4$ to $n = 10^9$ but the ID3 algorithm was performed on dataset size varying from $n = 10^4$ to $n = 10^6$, because of the random access memory limits. As we can see in Fig. 2 the accuracy of the ID3 algorithm is better than the Gaussian Decision Tree algorithm. For both algorithms the accuracy is increasing with growing number of training data elements. For $n = 10^9$ the obtained accuracy was greater than 90%.

We also compare the processing time with the growing number of training data elements. The results are presented in Fig. 3. As we can see the ID3 algorithm is much more time consuming than the Gaussian Decision Tree. For $n = 10^6$ ID3 algorithm needed about $9941s$ and the Gaussian Decision Tree algorithm needed only $33s$. The processing time of the ID3 algorithm is growing as fast as a power function of n whereas for the Gaussian Decision Tree algorithm the processing time increases almost linearly. The memory consumption is another advantage because for the Gaussian Decision Tree algorithm, unlike for the ID3 algorithm, it does not depend on the size of training dataset.

6 Conclusions

In this paper we considered the issue of mining data streams with the application of decision trees. The key point in construction of decision tree is the choice of the best attribute to split the considered node. We proposed a new method for deciding if the best attribute determined for the current set of data elements in the node is also the best according to the whole stream. The method is based on the Taylor's Theorem and on the properties of the normal distribution. It is mathematically justified by the theorem presented in the paper. Following the idea presented in [7] we proposed the GDT algorithm. In the example we showed that the GDT algorithm dramatically outperforms the McDiarmid Tree algorithm in the field of time consumption. Numerical simulations proved that the GDT algorithm is able to give satisfactory accuracies in data streams classification problems.

References

[1] C. Aggarwal, *Data Streams. Models and Algorithms*, Springer, LLC, New York, 2007.
[2] A. Bifet, *Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams*, IOS Press BV, Netherlands, Amsterdam, 2010.
[3] A. Bifet, G. Holmes, G. Pfahringer, R. Kirkby and R. Gavalda, "New ensemble methods for evolving data streams", Proc. KDD'09, Paris, France, June/July 2009.
[4] A. Bifet and R. Kirkby, *Data Stream Mining a Practical Approach*, University of WAIKATO, Technical Report, 2009.
[5] L. Breiman, J.H. Friedman, R.A. Olshen and Ch.J. Stone, *Classification and Regression Trees*, Chapman and Hall, New York, 1993.
[6] Cover TM, Hart PE, "Nearest neighbor pattern classification", IEEE Transactions on Information Theory, vol. 13, issue 1, pp. 2127, 1967.

[7] P. Domingos and G. Hulten, "Mining high-speed data streams", Proc. 6th ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining, pp. 71-80, 2000.
[8] W. Fan, Y. Huang, H. Wang, and P.S. Yu, *Active Mining of Data Streams*, Proc. SDM, 2004.
[9] C. Franke, *Adaptivity in Data Stream Mining*, PhD dissertation, University of California, DAVIS, 2009.
[10] M.M Gaber, A. Zaslavsky and S. Krishnaswamy, "Mining data streams: A review", Sigmod Record, vol. 34, no. 2, pp. 18-26, June 2005.
[11] J. Gama, R. Fernandes and R. Rocha, "Decision trees for mining data streams", Intelligent Data Analysis, vol. 10 no. 1, pp. 23-45, March 2006.
[12] Jing Gao, Wei Fan and Jiawei Hang, "On Appropriate Assumptions to Mine Data Streams: Analysis and Practice", 2007 IEEE International Conference on Data Mining (ICDM07), Omaha, NE, Oct 2007.
[13] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, second ed., Elsevier, USA, 2006.
[14] J. Hertz, A. Krogh, R. G. Palmer,*Introduction to the theory of neural computation*, Addison-Wesley Pub. Co., 1991.
[15] W. Hoeffding, "Probability inequalities for sums of bounded random variables", Journal of the American Statistical Association, vol. 58, issue 301, pp. 13-30, March 1963.
[16] G. Hulten, L. Spencer and P. Domingos, "Mining time-changing data streams", Proc. 7th ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining, pp. 97-106, 2001.
[17] R. Jin and G. Agrawal, "Efficient Decision Tree Construction on Streaming Data", Proc. 9th ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining, 2003
[18] O. Kardaun, *Classical Methods of Statistics*, first ed., Springer, 2005.
[19] D.T. Larose *Discovering Knowledge in Data. An Introduction to Data Mining*, Wiley & Sons, Inc, 2005.
[20] J. Liu, X. Li and W. Hong, "Ambiguous decision trees for mining concept-drifting data streams", Pattern Recognition Letters, Elsevier, vol. 30, issue 15, pp. 1347-1355, November 2009.
[21] C. McDiarmid, "On the method of bounded differences, Surveys in Combinatorics", ed. J. Siemons, London, Math. Soc. Lecture, no. 141, pp. 148-188, 1989.
[22] M. Narasimha Murty, V. Susheela Devi,*Pattern Recognition: An Algorithmic Approach*, Springer, 2011.
[23] B. Pfahringer, G. Holmes, R. Kirkby, *New options for Hoeffding trees*, M.A.Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 90-99, 2007.
[24] J.R. Quinlan, *Learning efficient classification procedures and their application to chess end games*, In R. S. Michalski, J. G. Garbonell, and T. M. Mitchell (Eds.), Machine Learning: an Artificial Inteligence Approach, pp. 463-482. Morgan Kaufmann, San Francisco, CA, 1983.
[25] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, CA, 1993
[26] L. Rutkowski, *Computational Intelligence. Methods and Techniques*, Springer-Verlag, 2008.
[27] L. Rutkowski, "Adaptive probabilistic neural-networks for pattern classification in time-varying environment", IEEE Trans. Neural Networks, vol. 15, pp. 811-827, 2004.
[28] L. Rutkowski, L. Pietruczuk, P. Duda, M. Jaworski, "Decision trees for mining data streams based on the McDiarmid's bound", IEEE Transactions on Knowledge and Decision Engineering, vol. 99, no. PrePrints, 2012.
[29] A. Tsymbal, "The problem of concept drift: definitions and related work", TCD-CS-2004-15. Computer Science Department, Trinity College Dublin, Ireland, April 2004.
[30] L. Wasserman, *All of statistics: a concise course in statistical inference*, Springer, New York, 2005
[31] I.H. Witten, E. Frank, G. Holmes, *Data Mining: Practical Machine Learning Tools and Techniques*, Amsterdam, Boston, MA : Morgan Kaufman, 2005.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.