BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 63, No. 4, 2015
DOI: 10.1515/bpasts-2015-0105

Discretization of data using Boolean transformations and information theory based evaluation criteria

C. JANKOWSKI[1]*, D. REDA[1], M. MAŃKOWSKI[2], and G. BOROWIK[1]

[1] Institute of Telecommunications, Warsaw University of Technology, 15/19 Nowowiejska St., 00-665 Warszawa, Poland
[2] Institute of Radioelectronics and Multimedia Technology, Warsaw University of Technology, 15/19 Nowowiejska St., 00-665 Warszawa, Poland

Abstract. Discretization is one of the most important parts of decision table preprocessing. Transforming continuous values of attributes into discrete intervals influences further analysis using data mining methods. In particular, the accuracy of generated predictions is highly dependent on the quality of discretization. The paper contains a description of three new heuristic algorithms for discretization of numeric data, based on Boolean reasoning. Additionally, an entropy-based evaluation of discretization is introduced to compare the results of the proposed algorithms with the results of leading university software for data analysis. Considering the discretization as a data compression method, the average compression ratio achieved for databases examined in the paper is 8.02 while maintaining the consistency of databases at 100%.

Key words: machine learning, discretization, discernibility function, logic minimization, information theory, entropy.

1. Introduction

Today, due to the progress in information and communication technologies, the volume of produced data is growing rapidly. However, the primary goal is not to acquire the data itself, but to extract the knowledge from it, e.g., patterns or rules that would allow better systematization and explanation of the observed phenomena. The answer to these needs is knowledge discovery in databases (KDD), described in more detail in [1], as well as many examples of knowledge retrieval methods that applied in processed data can help in decision making [2–5].

Knowledge discovery is a complex process consisting of data selection, preprocessing, transformation, mining, and interpretation. The quality of knowledge discovered is affected by each of these stages.

One of the most common applications of data mining algorithms is processing of multimedia data. This is particularly evident in biomedical engineering where data collected for hundreds of variables describe medical conditions of patients [6]. For example, in [7] the classification of patients with Alzheimer's disease is described.

Real databases often contain numeric data, which is defined by the values of attributes that are in a continuous domain [8]. However, the most common databases contain mixed data, including, among others, real values, integers, symbolic values, and even sets or ranges of values [9]. Characteristic examples of such real data are: the telecommunications database for e-mail classification [10] and medical databases that support diagnosis, i.e., Breast Cancer Wisconsin Database and Pima Indians Diabetes Database [11,12]. Most of the parameters of these databases are continuous, so for proper analysis of these databases it is necessary to discretize the data.

Discretization of data is usually one of the steps in data preprocessing [13]. The transformation of continuous values of attributes to their discrete equivalents – disjoint intervals in the set of real numbers – enables further analysis using data mining with the algorithms that require symbolic data. Even if symbolic data is not required, discretization may speed up the process of mining and improve the accuracy of predictions [14, 15]. In particular, discretization of data determines the quality of the decision-making systems.

A number of methods for discretization have already been proposed in the literature [8, 16–25].

It should be noted that recent research results clearly indicate the advantages of methods based on Boolean algebra (Boolean reasoning) as a tool for knowledge discovery in databases. The methods used in logic synthesis have successfully been employed in data mining for telecommunications and medicine. One of their advantages is, among other things, the ability to systematically examine the solution space in a relatively short time, contrary to classical methods based on heuristics that may provide suboptimal solutions [26].

This paper extends the idea of Boolean reasoning by providing efficient heuristics for data discretization and proposes evaluation criteria for discretization quality based on information theory. The results obtained from the algorithms are compared with the results of the leading software tools for data analysis. The paper begins with a description of basic concepts: information system, decision system, discernibility, cut. Then, a taxonomy of discretization methods is given. Subsequently, in Sec. 3, systematic and heuristic algorithms that utilize Boolean transformations for data discretization are shown. The processes of determining candidate cuts, calculating a discernibility matrix, and obtaining a minimal set of

*e-mail: C.Jankowski@stud.elka.pw.edu.pl

cuts are described. In Sec. 4, discretization quality evaluation criteria are proposed. Section 5 presents the results of experiments intended to assess the quality of discretization using the proposed criteria.

2. Basic concepts

2.1. General information. *Information system* is an ordered quadruple $IS = \langle U, A, V, f\rangle$, where U is a non-empty, finite set of objects (instances) called the universe, A is a non-empty, finite set of attributes, $V = \bigcup_{a\in A} V_a$, where V_a is the domain of attribute $a \in A$, $f : U \times A \to V$ is an information function such that $f(x, a) \in V_a$ for every $a \in A$ and $x \in U$.

The set of attributes in an information system may include one or more distinguished attributes called decisions. Then, the information system is called *decision system* or *decision table*. Decision table is an information system of the form of ordered quintuple $DT = \langle U, C, D, V, f\rangle$, where $C \cup D = A$ and $C \cap D = \emptyset$.

The definition of a decision system is a formal modification of the concept of an information system, obtained by splitting the set of attributes A into disjoint sets C and D. Elements C are called *conditional attributes*, elements D are called *decision attributes*.

Decision tables are a frequently used data model and – most importantly – are directly adaptable to the key data mining algorithms, such as decision rule induction or feature extraction. It should be noted that the application of some specific algorithms is only possible for discrete databases.

A pair of objects $x, y \in U$ is *discernible* by B if there exists an attribute $a \in B$, $B \subset C$, such that $f(x, a) \neq f(y, a)$. This means that a pair of objects is discernible by a set of conditional attributes for which these objects have different values of decision attributes (decisions) [8, 27]. *Inconsistency* occurs when a pair of objects $x, y \in U$ is not discernible by a set B, but has different decisions: $f(x, dec) \neq f(y, dec)$. Consistent (deterministic) system is a system in which all pairs of objects with different decisions are discernible by a set of conditional attributes C.

Cut is a pair (attribute, cut value), denoted as $(a, cutVal)$, where $a \in C$, $cutVal \in \Re$. A pair of objects $x, y \in U$ is discernible by a cut if $f(x, a) < cutVal < f(y, a)$ or $f(y, a) < cutVal < f(x, a)$.

Lets consider a deterministic decision table S and a set of cuts P. P is *consistent* with S if for each pair of objects $x, y \in U$ of different decisions $f(x, dec) \neq f(y, dec)$ there exists a cut $(a, cutVal) \in P$ that discerns this pair.

2.2. Classification of discretization methods. To classify methods of discretization that are based on different algorithms, the classification criteria must be specified [28]:

- Local or global discretization. In the case of global discretization the entire space of the problem is examined. Local discretization solves only a selected subproblem at a time – the partitioning is made on the basis of limited information. An example of local discretization might be a branch of the decision tree in the C4.5 method [29].
- Supervised or unsupervised discretization. During supervised discretization the class of the object (its decision) is taken into consideration. The basic principle of a supervised method is the separation of instances with different decisions. If the method does not use the information about classes, it is called unsupervised. The advantage of unsupervised methods is their applicability to discretization of databases that do not contain a decision attribute.
- Static or dynamic. In a static method the interdependency of attributes is not taken into account. Dynamic methods simultaneously examine many features which enables use of high level dependencies.

The proposed method that relies on Boolean transformations is global, dynamic and supervised.

3. Discretization using Boolean transformations

The main goal of discretization is to construct a new discrete decision system that is consistent with the original decision system. This can be done by constructing a set of cuts sufficient to discern each pair of objects of different decisions. It is desirable to calculate a set of cuts with a minimal number of elements.

There are three basic steps in the discretization process, i.e., determining the candidate (proposed) cuts, calculating a discernibility matrix, and solving the column cover problem for this matrix.

3.1. Candidate cuts. For each attribute, one should determine its set of values. A set of candidate cuts can be determined, for example, by considering the arithmetic mean for each successive pair of sorted unique attribute values. It means that, for attribute $a \in C$, candidate cuts can be represented as pairs:

$$(a, cutVal)_i = \left(a, \frac{f(\dot{x}_i, a) + f(\dot{x}_{i+1}, a)}{2}\right), \qquad (1)$$

where $i = 1, \ldots, |V_a| - 1$, and $\dot{x}_i$, $\dot{x}_{i+1}$ denotes a pair of objects for consecutively ordered unique values of attribute a.

Such an approach has a major drawback: candidate cuts are initiated between the attribute values held by instances with the same value of decision attribute. To obtain a consistent discretization, all pairs of objects with different decisions only must be separated.

Example 1. Consider a decision system S in Table 1.

Table 1
Example of decision system

S	$\cdots$	a	$\cdots$	dec
x_1	$\cdots$	0.4	$\cdots$	0
x_2	$\cdots$	0.6	$\cdots$	0
x_3	$\cdots$	1.0	$\cdots$	1
x_4	$\cdots$	1.2	$\cdots$	2

In this system, cut $(a, 0.5)$ separates two pairs of objects only, i.e., (x_1, x_3) and (x_1, x_4). However, cut $(a, 0.8)$ separates four pairs of objects, i.e., (x_1, x_3), (x_1, x_4), (x_2, x_3),

and (x_2, x_4). Note that the latter set of pairs includes the former set of pairs.

Let X_i^a denote the set of all instances for unique value t_i of selected attribute a, i.e., $\forall x \in X_i^a \ f(x,a) = t_i$, where $t_i \in V_a$, $i = 1, \ldots, |V_a|$. Let $f(X_i^a, dec)$ denote the set of unique decision values for $x \in X_i^a$. Assuming that t_i are sorted values of V_a, and $|\cdot|$ denotes cardinality, we represent a candidate cut j as:

$$(a, cutVal)_j = \left(a, \frac{t_i + t_{i+1}}{2}\right), \tag{2}$$

when one of the following conditions occurs:

- $|f(X_i^a, dec)| = |f(X_{i+1}^a, dec)| = 1$ and $f(X_i^a, dec) \neq f(X_{i+1}^a, dec)$,
- $|f(X_i^a, dec)| > 1$ or $|f(X_{i+1}^a, dec)| > 1$.

The presented procedure makes it possible to reduce both time and memory complexity for the calculation and subsequent transformations of the discernibility matrix.

3.2. Discernibility matrix. The next step uses the candidate cuts for calculating the discernibility matrix.

In a discernibility matrix $M = [m_{pj}]$, each row is calculated for a pair of objects $(x, y)_p$ of different decisions where each column of the matrix represents one of the possible cuts. Then, for a given decision table, if the p-th pair of objects is discernible by the j-th cut, then $m_{pj} = 1$, otherwise $m_{pj} = 0$. Formally,

$$m_{pj} = \begin{cases} 1, & \text{if } f(x,a) < (a, cutVal)_j < f(y,a) \\ & \text{or } f(y,a) < (a, cutVal)_j < f(x,a), \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $x, y \in U$ and $f(x, dec) \neq f(y, dec)$.

To retain the possibility of classification (to maintain database consistency) after discretization, each pair of objects of different decision value, represented by a row in the discernibility matrix, must be discerned. It means that for each row of the matrix there must be at least one cut selected (column with value 1 in this row). This task corresponds to the problem of obtaining column cover for the discernibility matrix and it is performed in order to obtain a minimal set of cuts. Then, the minimal column cover corresponds to the minimal set of cuts.

It should be noted that there are databases containing symbolic attributes that are character strings or single letters. The simplest approach would be to form the discernibility matrix for cuts of numeric attributes only. However, such a solution can lead to the generation of an excessive number of cuts. A better solution is to add columns in the discernibility matrix corresponding to the symbolic attributes. If the symbolic values are the same, the value in the additional column is 0, otherwise it is 1.

3.3. Column cover algorithm. Calculating the discernibility matrix is an important step towards obtaining the minimal set of cuts. However, searching for the minimal column cover of the discernibility matrix is not trivial.

Column cover of a matrix M is a set of columns L such that, for each row i of M there exists a column $l \in L$ for which $l_i = 1$. If no subset of the column cover L is a column cover, we say that L is a *minimal column cover*.

It has been proved that the minimal column cover calculation can be reduced to the conversion of conjunctive normal form (CNF) to disjunctive normal form (DNF) by prior transformation of matrix M into CNF. This problem is, however, one of the NP-hard problems [8, 30, 31].

It should be noted that for datasets with a limited number of instances and attributes it is possible – using Boolean function complementation based algorithm (Complement algorithm) – to systematically calculate a minimal set of cuts ensuring consistent discretization [9]. This approach is based on the following theorem:

Theorem. (Brayton, 1984) Each row i of C, the binary matrix complement of M, corresponds to a column cover L of M, where $j \in L$, iff $C_{ij} = 1$.

The Complement algorithm has already been used in [26] to calculate reducts of decision tables. Calculation time has been significantly reduced which has made it possible to achieve results that cannot be obtained using earlier published methods and systems.

The key strength of the algorithm lies in transformation (4), i.e., double complementation of a monotone Boolean function f in CNF that represents matrix M

$$f = \prod_k \sum_l x_{kl} = \overline{\overline{\prod_k \sum_l x_{kl}}} = \overline{\sum_k \prod_l \overline{x_{kl}}} \tag{4}$$

and a procedure for Shannon expansion of $\overline{f}$

$$\overline{f} = \overline{x}_j \overline{f}_{\overline{x}_j} + \overline{f}_{x_j}. \tag{5}$$

Due to computational complexity, the systematic algorithm is, however, no longer applicable to large datasets.

Example 2. In the first step of the algorithm we propose a set of cuts determined by arithmetic means of sorted attribute values.

For the decision system in Table 2 we have the following set of cuts:

$$c_{a1} = (a, 1.7), \quad c_{a2} = (a, 1.9), \quad c_{a3} = (a, 2.3),$$
$$c_{a4} = (a, 2.7), \quad c_{a5} = (a, 3.0),$$
$$c_{b1} = (b, 0.325), \quad c_{b2} = (b, 0.45),$$
$$c_{b3} = (b, 0.75), \quad c_{b4} = (b, 1.25).$$

Table 2
Example decision system

S	a	b	dec
x_1	2.6	1.5	0
x_2	2.0	0.25	0
x_3	1.6	1.0	1
x_4	2.8	0.5	1
x_5	2.8	1.0	0
x_6	3.2	1.5	1
x_7	1.8	0.4	0
x_8	2.6	0.5	1

According to (2), the set of cuts is reduced to: c_{a1}, c_{a3}, c_{a4}, c_{a5}, c_{b2}, c_{b3}, c_{b4}. Geometrical interpretations of the objects and the proposed cuts are shown in Fig. 1.

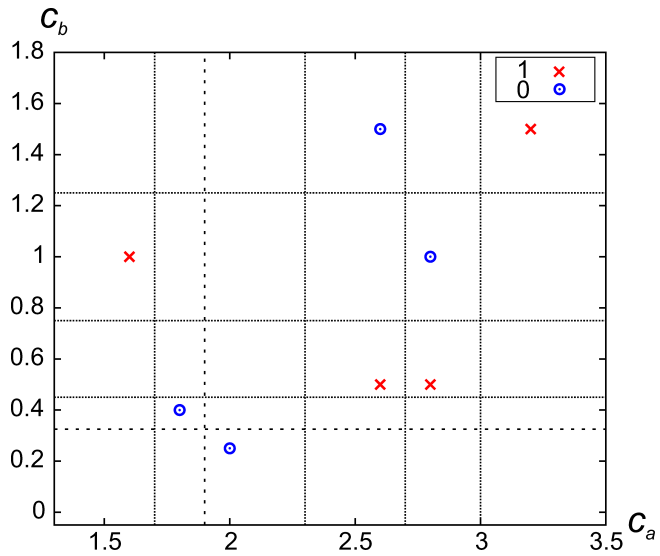

Fig. 1. Geometrical representation of data and the proposed cuts

The proposed method yields a discernibility matrix that is in fact a monotonic CNF. The simplification of the function is carried out by converting CNF to DNF or calculating the column cover of the matrix (Fig. 2).

	c_{a1}	c_{a3}	c_{a4}	c_{a5}	c_{b2}	c_{b3}	c_{b4}
(x_1,x_3)	1	1	0	0	0	0	1
(x_1,x_4)	0	0	1	0	0	1	1
(x_1,x_6)	0	0	1	1	0	0	0
(x_1,x_8)	0	0	0	0	0	1	1
(x_2,x_3)	1	0	0	0	1	1	0
$\vdots$				$\vdots$			
(x_7,x_8)	0	1	0	0	1	0	0

$\downarrow$

c_cover$_1$	0	1	0	1	0	1	0
c_cover$_2$	1	0	0	1	1	1	0
c_cover$_3$	0	1	1	0	0	1	1
c_cover$_4$	0	0	1	0	1	1	1

Fig. 2. Discernibility matrix and its column cover

As a result, we obtain:

$$c_{a3}c_{a5}c_{b3} \vee c_{a1}c_{a5}c_{b2}c_{b3} \vee c_{a3}c_{a4}c_{b3}c_{b4} \vee c_{a4}c_{b2}c_{b3}c_{b4}.$$

Finally, taking as an example the first set of cuts, i.e., the cuts belonging to the first product $\{c_{a3}, c_{a5}, c_{b3}\}$ and encoding corresponding subintervals:

$$P_a = \{[1; 2.3), [2.3; 3.0), [3.0; 4]\} = \{0, 1, 2\},$$
$$P_b = \{[0; 0.75), [0.75; 2]\} = \{0, 1\},$$

we obtain a discrete decision system shown in Table 3. By removing redundant rows we obtain the system shown in Table 4. Geometrical interpretation of this result is shown in Fig. 3.

Table 3
Decision system of Table 2 after discretization

$\mathcal{A}$	a	b	d
x_1	1	1	0
x_2	0	0	0
x_3	0	1	1
x_4	1	0	1
x_5	1	1	0
x_6	2	1	1
x_7	0	0	0
x_8	1	0	1

Table 4
Decision system of Table 3 after removing redundant rows

$\mathcal{A}$	a	b	d
$\{x_1, x_5\}$	1	1	0
$\{x_2, x_7\}$	0	0	0
x_3	0	1	1
$\{x_4, x_8\}$	1	0	1
x_6	2	1	1

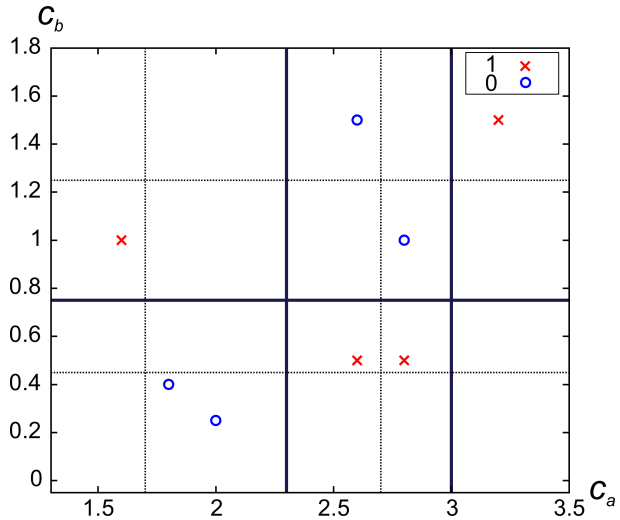

Fig. 3. Geometrical representation of the result shown in Table 4

3.4. Heuristics. It should be emphasized that the column cover problem occurs not only during the transformation of the discernibility matrix. In related fields, an analogous problem is encountered, for example, when inducing decision rules. Thus, aforementioned reasoning suggests that one should look for effective heuristic column cover algorithms.

Fortunately, in case of discretization, only a single minimal set of cuts is needed (while in some applications obtaining not one, but all minimal sets is necessary). It means that one local minimum should be reached. It is also essential to make sure that the finally calculated minimal set of cuts is as small as possible – it would be best if the number of its elements

was the same as in the global minimum (see *exact minimal cover* [32, 33]).

To solve the problem of column cover, we have proposed and implemented three heuristic algorithms, named *MaxCol*, *MinRow*, and *Omega*.

Algorithm 1 *MaxCol*

Step 1. Count 1's in every column.

Step 2. Save the column with most 1's – in case of a draw, arbitrarily choose the column with the lowest index.

Step 3. Remove all rows that contain 1 in the chosen column.

Step 4. If the matrix is not empty, go to **Step 1**.

Step 5. Return saved columns.

Algorithm 2 *MinRow*

Step 1. Count 1's in every row.

Step 2. Choose rows with the lowest number of 1's.

Step 3. Count 1's in columns for which any row chosen in **Step 2** contains a 1.

Step 4. Save the column with most 1's among those chosen in **Step 3** – in case of a draw, arbitrarily choose the column with the lowest index.

Step 5. Remove all rows that contain a 1 in the chosen column.

Step 6. If the matrix is not empty, go to **Step 1**.

Step 7. Return saved columns.

Algorithm 3 *Omega*

Step 1. Count number of 1's in every row. Let l be the ratio of the number of 1's to the number of all elements in a row.

Step 2. For each row and each column containing a 1 on this row increase the metric of that column by $1 + lw$.

Step 3. Save the column with the highest metric – in case of a draw, arbitrarily choose the column with the lowest index.

Step 4. Remove all rows that contain a 1 in the chosen column.

Step 5. If the matrix is not empty, go to **Step 1**.

Step 6. Return saved columns.

The *Omega* algorithm requires an additional parameter w, specified by the user, reflecting the "importance" of the number of 1's in a row, where $w \in (-1.0, 1.0)$. To make the algorithm lean towards columns separating pairs of objects which are the most difficult to discern (i.e., those that can be discerned by the smallest number of cuts), the value of parameter w should be negative. It can be shown that algorithm *Omega* with $w = 0$ is equivalent to *MaxCol* algorithm.

Using the calculated set of cuts, the dataset is discretized. The number of attribute values for the resulting dataset equals

$$|V'| = \sum_{a \in C} \left(\left| \bigcup_j (a, cutVal)_j \right| + 1 \right), \tag{6}$$

where $\left| \bigcup_j (a, cutVal)_j \right|$ denotes the number of calculated cuts for attribute $a \in C$.

4. Discretization quality evaluation criteria

Due to the multitude of existing discretization methods, there is a need for discretization quality evaluation criteria. These were presented by [28]:

- Number of intervals. The fewer intervals, the simpler the resulting discrete decision table. It may be noted that the problem of minimizing the number of intervals is tantamount to minimizing the number of cuts.
- Number of inconsistencies. In the best case the discretization process should not introduce additional inconsistencies over those contained in the input database. Otherwise, relevant information may be lost.
- Accuracy of predictions. This criterion determines improvement of predictions after discretization. It should be noted that its value depends on the classification method.

We must stress that only the first two criteria are directly measurable. The accuracy of prediction is a function of both the discretization and the classification algorithm.

In this paper, we propose evaluation of the quality of discretization based on entropy, known from information theory [34]. One can look at the process of discretization as a process of compressing the database. On the one hand, we want to be able to save the data using the least amount of memory. On the other hand, we should not introduce additional inconsistencies to the output data.

The proposed criterion considers each attribute separately. Then, we can determine the entropy according to formula:

$$H(V_a) = E(I(v_i)) = \sum_{i=1}^{|V_a|} p(v_i) I(v_i). \tag{7a}$$

Thus,

$$H(V_a) = \sum_{i=1}^{|V_a|} p(v_i) log_2 \frac{1}{p(v_i)} \ [bit], \tag{7b}$$

where $|V_a|$ denotes the number of values for attribute a, v_i is the i-th value of a, and $p(v_i)$ denotes the likelihood of value v_i. Due to the nature of the problem, the likelihood is calculated as the number of instances for which the value of a equals v_i divided by the total number of instances, that is $p(v_i) = \frac{|U_{f(x,a)=v_i}|}{|U|}$. The $H(V_a)$ unit is a bit, due to base 2 of the logarithm. Equation (7b) can also be represented as:

$$H(V_a) = -\sum_{i=1}^{|V_a|} p(v_i) log_2(p(v_i)) \ [bit]. \tag{7c}$$

The proposed criterion is based on the calculation of the sum of all attributes that occur in the database:

$$H(V) = \sum_{a \in A} H(V_a) \ [bit]. \tag{8}$$

A smaller value of the proposed metric corresponds to less redundancy in data. Note that $H(V)$ is a real value that approximates the minimum number of bits sufficient to store the database.

It should also be noted that after the process of discretization, many instances in the resulting database are described with the same vector of discrete attribute values. Then, it is possible to reduce repetitive instances. Therefore, the following modification of the metric can be proposed:

$$H(V)_{scaled} = |U'| \, H(V) \ [bit], \tag{9}$$

where $|U'|$ denotes the number of unique instances in the database after the discretization process.

It should be pointed out that if the attribute is numeric (continuous) its entropy tends to infinity. One could use the differential entropy in this case. In our considerations, to have the possibility to compare the results, the value of entropy of continuous attributes is counted using the set of values for this attribute in the training set.

Entropy-based criteria do not clearly indicate which one of examined methods is the most appropriate. Depending on the selected database and the expected results, the weight of each criterion may vary. Moreover, there is no method of discretization that is superior with regard to all the considered criteria.

5. Experiments

For our experiments we used databases contained in [12] repository described in Table 5. For example, the *Spambase Dataset* contains 4601 instances – email messages, characterized by 57 numeric conditional attributes and two decision classes – *spam* and *not spam*.

To compare the quality of discretization performed by our prototype software (implementing three proposed algorithms: *MaxCol*, *MinRow*, and *Omega*) we have chosen the discretization algorithms used in the well-recognized university systems: *Rough Set Exploration System 2* [8] and *Weka 3* [35]. The criterion of the quality of discretization was the metric proposed in this paper, based on information theory (in basic and scaled versions) and the number of introduced inconsistencies. Experiments were conducted in two variants: with reduction and without reduction of duplicate instances. Results are presented in Table 6.

In Table 6 columns "a" refer to databases without reduction of duplicate rows. It means that the number of instances in datasets after discretization is the same as it is in the original datasets. Columns "b" refer to databases with reduction of duplicate rows. In this case, multiple instances with the same attributes' values are reduced to a single object.

The results in Table 6 are explained for benchmark *Australian*.

There are 690 instances in the original *Australian* dataset. After discretization using *Omega* algorithm and reduction of duplicate rows there are 645 unique instances. The entropy-based metric (8) for *Omega* algorithm has the value of 14.95 bits without reduction and 15.01 bits with reduction of duplicate rows. The scaled entropy-based metric without reduction has the value of 10319 bits, and with reduction the value of 9682 bits. There is no inconsistency in this discrete database which implies value 0 in the last four columns.

To recall, consistency means that there is no pair of objects that have the same conditional attribute value and different decision attribute value. It is worth noting how the number of inconsistencies is calculated. Assume that in the dataset which consists of 10 instances there are five objects that have the same conditional attribute values, i.e., four of them have the decision attribute value 0 and the last one has the decision attribute value 1. Suppose that the other part of the dataset is consistent. In this case the number of inconsistencies is 5 which is 50% of the number of instances in this dataset.

The proposed discretization method has proved to be very effective for every column cover algorithm examined. It has not introduced any inconsistencies – their number after discretization remains unchanged. At the same time it was possible to significantly reduce the number of bits required to store the data.

To determine the level of compression, assume that attributes are independent random variables. The Shannon's source coding theorem [34] implies that $\overline{L} \geq H(X)$, where $\overline{L}$ is the average length of a code word and $H(X)$ is the source entropy. It follows from the theorem that the lower bound on the memory size required to save the original, numeric Spambase dataset with duplicate rows reduced is 570933 bits, whereas after discretizing it by applying the *MinRow* algorithm it is only 76244 bits. Therefore the achieved level of compression is $CR = 7.5$. Noticeable compression were achieved for the other examined databases.

For the databases examined in the paper, the average compression ratio for our methods is 8.02 while maintaining the consistency of databases at 100%. This is better than for RSES2 [8] where compression ratio is 7.26 while maintaining the consistency of databases at 100%, and better than for Weka 3 [35], where compression ratio is 32.60 for Fayyad & Irani MDL method and 21.60 for Kononenko MDL method while maintaining the consistency of databases at 62.79% and 64.71% respectively (Table 7).

Table 5
Selected benchmarks information [12]

Dataset (full name)	Short name	No. instances	No. conditional attributes	No. classes
Statlog (Australian Credit Approval) Dataset	Australian	690	14	2
Statlog (German Credit Data) Dataset – numeric	German	1000	24	2
Glass Identification Dataset	Glass	214	10	7
Iris Dataset	Iris	150	4	3
Spambase Dataset	Spambase	4601	57	2
Yeast Dataset	Yeast	1484	8	10

Table 6
Experimental results

Method a – without reduction b – with reduction	No. instances		Entropy-based metric [bit]		Scaled entropy-based metric [bit]		Introduced inconsistencies no.		Introduced inconsistencies no.	
	a	b	a	b	a	b	a	b	a	b
Australian										
[none]	690	690	46.07	46.07	31787	31787	–	–	–	–
Omega	690	645	14.95	15.01	10319	9682	0	0	0	0
MaxCol	690	639	14.62	14.66	10090	9368	0	0	0	0
MinRow	690	636	14.71	14.77	10151	9395	0	0	0	0
RSES2	690	638	14.64	14.68	10100	9366	0	0	0	0
Weka 3 Fayyad & Irani	690	412	12.26	12.76	8458	5258	114	60	16.52	14.56
Weka 3 Kononenko	690	412	12.26	12.76	8458	5258	114	60	16.52	14.56
German										
[none]	1000	1000	37.83	37.83	37827	37827	–	–	–	–
Omega	1000	967	16.58	16.62	16576	16070	0	0	0	0
MaxCol	1000	965	16.95	16.99	16953	16394	0	0	0	0
MinRow	1000	977	17.6	17.63	17597	17223	0	0	0	0
RSES2	1000	965	16.95	16.99	16953	16394	0	0	0	0
Weka 3 Fayyad & Irani	1000	79	5.79	6.53	5793	516	840	68	84.00	86.08
Weka 3 Kononenko	1000	79	5.79	6.53	5793	516	840	68	84.00	86.08
Glass										
[none]	214	213	51.94	51.94	11112	11062	–	–	–	–
Omega	214	155	12.90	12.83	2760	1989	0	0	0	0
MaxCol	214	145	12.56	12.67	2688	1838	0	0	0	0
MinRow	214	166	13.49	13.50	2887	2241	0	0	0	0
RSES2	214	156	12.94	12.87	2770	2008	0	0	0	0
Weka 3 Fayyad & Irani	214	74	10.28	11.19	2200	133	32	68	62.15	43.24
Weka 3 Kononenko	214	89	11.39	12.82	2437	516	113	30	52.80	33.71
Iris										
[none]	150	149	19.51	19.51	2927	2908	–	–	–	–
Omega	150	17	5.84	6.20	875	105	0	0	0	0
MaxCol	150	19	6.27	6.43	941	122	0	0	0	0
MinRow	150	17	6.08	6.28	912	107	0	0	0	0
RSES2	150	19	6.27	6.43	941	122	0	0	0	0
Weka 3 Fayyad & Irani	150	24	7.86	7.36	1179	177	24	10	16.00	41.67
Weka 3 Kononenko	150	24	7.86	7.36	1179	177	24	10	16.00	41.67
Spambase										
[none]	4601	4210	132.10	135.61	607804	570933	–	–	–	–
Omega	4601	3289	23.28	23.26	107116	76509	0	0	0	0
MaxCol	4601	3311	23.76	23.71	109331	78508	0	0	0	0
MinRow	4601	3326	23.05	22.92	106049	76244	0	0	0	0
RSES2	4601	3381	24.90	24.85	114555	84027	0	0	0	0
Weka 3 Fayyad & Irani	4601	3594	44.45	44.79	204535	160977	56	18	1.22	0.50
Weka 3 Kononenko	4601	3604	44.98	45.34	206938	163397	52	16	1.13	0.44
Yeast										
[none]	1484	1453	32.12	32.07	47671	46598	–	–	–	–
Omega	1484	1422	18.5	18.54	27447	26369	0	0	0	0
MaxCol	1484	1425	18.75	18.79	27820	26772	0	0	0	0
MinRow	1484	1413	18.33	18.36	27200	25939	0	0	0	0
RSES2	1484	1424	18.85	18.89	27971	26893	0	0	0	0
Weka 3 Fayyad & Irani	1484	331	8.79	10.73	13040	3552	1285	243	86.59	73.41
Weka 3 Kononenko	1484	440	9.63	11.88	14285	5228	1185	267	79.85	60.68

Table 7
Experimental results – summary

Method a – without reduction b – with reduction	Compression ratio (average)		Introduced inconsistencies (average) [%]	
	a	b	a	b
Omega	3.36	8.02	0	0
MaxCol	3.32	7.43	0	0
MinRow	3.30	7.83	0	0
RSES2	3.25	7.26	0	0
Weka 3 Fayyad & Irani	4.07	32.60	35.98	37.21
Weka 3 Kononenko	3.93	21.60	34.09	35.29

Table 8
Accuracy of predictions using selected classifiers

Dataset	J48 Decision Tree accuracy [%]	Decision Table accuracy [%]	LRM accuracy [%]	SMO accuracy [%]	Random Forest accuracy [%]
			Australian		
[none]	83.93±4.03	85.07±3.94	86.32±3.67	85.51±3.99	86.29±3.73
Omega	86.83±3.59	85.48±3.89	86.13±3.99	85.51±3.60	86.28±3.82
MaxCol	84.59±4.27	85.42±3.71	84.94±3.71	85.51±3.60	85.57±3.80
MinRow	86.13±3.94	85.33±3.86	86.65±3.70	85.51±3.60	85.72±3.83
RSES2	84.77±3.83	83.83±3.78	86.12±3.60	85.26±3.92	84.03±4.10
Weka 3 Fayyad & Irani	86.57±3.87	85.33±3.86	86.36±3.78	85.46±3.97	85.70±3.86
Weka 3 Kononenko	87.78±3.66	84.94±4.11	86.74±3.77	85.46±3.97	84.39±3.64
			German		
[none]	72.14±4.09	72.25±2.85	76.96±3.98	76.79±3.67	74.93±3.12
Omega	71.73±4.13	73.17±4.10	74.90±3.86	75.17±3.82	72.94±4.24
MaxCol	72.88±3.76	71.61±3.65	73.64±3.63	73.84±3.68	72.87±3.77
MinRow	71.76±3.72	72.80±4.01	74.10±3.22	73.64±3.37	72.47±3.79
RSES2	71.14±3.74	72.12±3.85	73.87±3.82	74.32±3.86	72.15±3.70
Weka 3 Fayyad & Irani	74.94±3.74	73.55±3.32	75.09±3.48	71.41±2.58	74.47±3.89
Weka 3 Kononenko	74.94±3.74	73.55±3.32	75.09±3.48	71.41±2.58	74.47±3.89
			Glass		
[none]	67.90±9.09	66.53±9.52	63.04±8.93	57.44±8.70	76.42±8.59
Omega	70.48±9.41	64.03±8.98	62.16±10.11	67.40±8.36	74.96±8.35
MaxCol	72.34±9.03	60.15±8.40	62.45±8.69	64.87±8.19	74.03±8.30
MinRow	66.25±8.87	60.90±9.16	61.52±8.79	65.12±8.15	71.72±9.49
RSES2	66.77±9.27	66.47±9.52	59.65±10.20	65.73±9.38	70.63±8.20
Weka 3 Fayyad & Irani	74.57±9.59	70.90±9.03	74.29±8.94	77.86±9.17	76.22±8.66
Weka 3 Kononenko	72.51±9.82	70.16±9.45	73.23±9.71	77.86±8.72	76.07±8.58
			Iris		
[none]	94.27±5.53	92.33±5.47	97.27±4.93	96.13±4.46	94.47±5.11
Omega	96.40±4.28	94.20±5.58	95.60±4.36	95.13±5.00	97.40±3.89
MaxCol	96.40±4.28	94.20±5.58	95.60±4.36	95.13±5.00	97.40±3.89
MinRow	96.40±4.28	94.20±5.58	95.60±4.36	95.13±5.00	97.40±3.89
RSES2	96.67±4.08	94.07±5.52	96.47±4.18	96.33±4.58	97.20±3.93
Weka 3 Fayyad & Irani	93.27±5.57	93.53±5.40	93.00±5.64	94.07±5.76	95.33±5.32
Weka 3 Kononenko	93.27±5.57	93.53±5.40	94.13±5.13	93.67±5.71	94.27±5.61
			Spambase		
[none]	92.85±0.95	90.53±1.46	92.68±1.14	90.43±1.27	94.69±0.95
Omega	93.30±1.10	91.45±1.24	93.49±1.14	93.69±1.09	94.28±0.98
MaxCol	92.21±1.25	91.05±1.16	92.85±1.07	92.84±1.08	93.62±1.12
MinRow	92.13±1.18	91.53±1.27	92.64±1.11	92.88±1.11	93.33±1.09
RSES2	93.54±0.92	91.75±1.17	93.76±1.09	93.81±1.15	93.97±1.11
Weka 3 Fayyad & Irani	93.13±1.02	91.27±1.22	94.57±0.94	94.50±0.89	93.94±1.00
Weka 3 Kononenko	93.06±1.02	91.15±1.30	94.57±0.89	94.49±0.87	93.84±1.05
			Yeast		
[none]	56.39±4.12	54.76±3.49	58.98±3.44	56.80±3.36	58.58±3.39
Omega	57.52±3.53	54.17±3.51	57.38±3.69	57.14±3.22	56.40±3.52
MaxCol	55.64±3.44	54.97±3.16	58.50±3.38	58.21±3.48	55.69±3.55
MinRow	56.29±3.51	52.90±3.79	58.29±3.51	57.04±3.18	56.53±3.60
RSES2	52.21±3.48	47.50±3.46	58.38±3.52	57.70±3.73	53.57±3.46
Weka 3 Fayyad & Irani	57.32±3.49	56.50±3.54	58.73±3.26	59.18±3.58	56.99±3.61
Weka 3 Kononenko	56.97±3.37	54.70±3.67	58.48±3.55	58.79±3.68	55.82±3.78

Table 9
Accuracy of predictions using selected classifiers – summary

Data set	[none] mean	*Omega* mean	*MaxCol* mean	*MinRow* mean	$(b+c+d)/3$	$e-a$	$\max(b,c,d)$	$g-a$	RSES2 mean	Weka 3 Fayyad & Irani mean	Weka 3 Kononenko mean
	a	b	c	d	e	f	g	h	i	j	k
Australian	85.42	86.05	85.21	85.87	85.71	0.28	86.05	0.62	84.80	85.88	85.86
German	74.61	73.58	72.97	72.95	73.17	−1.45	73.58	−1.03	72.72	73.89	73.89
Glass	66.27	67.81	66.77	65.10	66.56	0.29	67.81	1.54	65.85	74.77	73.97
Iris	94.89	95.75	95.75	95.75	95.75	0.85	95.75	0.85	96.15	93.84	93.77
Spambase	92.24	93.24	92.51	92.50	92.75	0.52	93.24	1.01	93.37	93.48	93.42
Yeast	57.10	56.52	56.60	56.21	56.44	−0.66	56.60	−0.50	53.87	57.74	56.95
					mean:	−0.03	mean:	0.41			

The experiments also indicated the need for selection of an individual solution for each database. This can be achieved by extensive adjustment of discretization parameters along with a selection of an algorithm to solve the column covering problem.

We have conducted an additional experiment examining the impact of data discretization on the quality of classification (accuracy of predictions) by algorithms that do not require discrete data, i.e., are capable of working with continuous data. A discretization model has been constructed on a random training set comprising 90% of data, which was then applied to a test set, constituting the remaining 10% of data. In the next stage, we have used Weka 3 to generate five classification models, i.e., J48 decision tree [36], Decision Table [37], Logistic Regression Model [38], Sequential Minimal Optimization [39], and Random Forest [40], which we have then verified on the discretized test set. The criterion chosen to assess the quality of classification is the number of correctly classified test instances. Aforementioned 10-fold cross-validation has been repeated 10 times.

Results of the experiment are presented in Table 8. The table is partitioned into 6 sections corresponding to the arrangement of Table 6. Successive columns correspond to employed classification models. The results of the experiments are the mean and standard deviation of prediction accuracy.

The observations from Table 8 have been summarized in Table 9. In columns b, c, d, the average accuracy for the methods proposed in the paper has been calculated as the arithmetic mean for five considered classifiers. As shown in columns f and h, our methods yield nearly the same quality of classification (or slightly greater quality for at least one of the proposed methods of discretization) as the methods provided in columns i, j, k, irrespective of the use of continuous or discrete data.

Nevertheless, the use of discrete data brings major benefits, including faster building of a classification model or the ability to use classification methods that require symbolic data. Furthermore, the application of proposed discretization methods results in a reduction of memory required to store the data, which has been confirmed by tests presented in Table 6.

6. Conclusions

In this paper, we present results of experiments by comparing the discretization performed by the proposed algorithms with discretization performed by the leading knowledge discovery systems. To achieve this, special criteria describing the quality of discretization have been introduced and used in the experiments. These criteria account for data compression and combined with information about introduced inconsistencies make it possible to assess the quality of discretization.

Experimental results confirmed the usefulness of the proposed algorithms. With the proposed methods, based on Boolean reasoning, we are able to achieve high quality of discretization necessary for knowledge discovery. Our results are better than the results achieved using RSES2 and Weka 3. Considering the discretization as a data compression method, the average compression ratio achieved for databases examined in the paper is 8.02 while maintaining the consistency of databases at 100%.

REFERENCES

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases", *AI Magazine* 17, 37–54 (1996).

[2] C. Moraga, "Design of neural networks", *11th Int. Conf. Knowledge-Based Intelligent Informational and Engineering Systems, Lecture Notes in Computer Science* 4692, 26–33 (2007), DOI: 10.1007/978-3-540-74819-9_4.

[3] A. Raghuvanshi and M.A. Perkowski, "Image processing and machine learning for the diagnosis of melanoma cancer", *BIODEVICES 2011 – Proc. Int. Conf. on Biomedical Electronics and Devices* 1, 405–410 (2011).

[4] S. Hui and S. Żak, "Discrete Fourier transform based pattern classifiers". *Bull. Pol. Ac.: Tech.* 62 (1), 15–22 (2014), DOI: 10.2478/bpasts-2014-0002.

[5] A. Jastriebow and K. Poczęta, "Analysis of multi-step algorithms for cognitive maps learning", *Bull. Pol. Ac.: Tech.* 62 (4), 735–741 (2014), DOI: 10.2478/bpasts-2014-0079.

[6] M.G.M. Hunink, P.P. Glasziou, J.E. Siegel, J.C. Weeks, J.S. Pliskin, A.S. Elstein, and M.C. Weinstein, *Decision Making in Health and Medicine: Integrating Evidence and Values*, University Press, Cambridge, 2001.

[7] R. Cuingnet, E. Gerardin, J. Tessieras, G. Auzias, S. Lehericy, M.O. Habert, M. Chupin, H. Benali, and O. Colliot, "Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database", *NeuroImage* 56 (2), 766–781 (2011), DOI: 10.1016/j.neuroimage.2010.06.013.

[8] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, *Rough Sets: A Tutorial*, Springer, Singapore, 1998.

[9] G. Borowik, "Boolean function complementation based algorithm for data discretization", *Computer Aided Systems Theory – EUROCAST 2013, Lecture Notes in Computer Science* 8112, 218–225 (2013), DOI: 10.1007/978-3-642-53862-9_28.

[10] M. Žádník and Z. Michlovský, "Is spam visible in flow-level statistics?", *CESNET National Research and Education Network, Tech. Rep.*, CESNET 1, Prague, 2009.

[11] O.L. Mangasarian and W.H. Wolberg, "Cancer diagnosis via linear programming", *SIAM News* 23 (5), 1–18 (1990).

[12] K. Bache and M. Lichman, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml (2013).

[13] M.R. Chmielewski and J.W. Grzymala-Busse, "Global discretization of continuous attributes as preprocessing for machine learning", *Int. J. Approximate Reasoning* 1, 294–301 (1996).

[14] A. Ekbal, "Improvement of prediction accuracy using discretization and voting classifier", *18th Int. Conf. on Pattern Recognition* 2, 695–698 (2006), DOI: 10.1109/ICPR.2006.698.

[15] E. Frank and I.H. Witten, "Making better use of global discretization", *Proc. Sixteenth Int. Conf. on Machine Learning* 1, 115–123 (1999).

[16] H.-V. Nguyen, E. Müller, J. Vreeken, and K. Böhm, "Unsupervised interaction-preserving discretization of multivariate data", *Data Mining and Knowledge Discovery* 28 (5–6), 1366–1397 (2014), DOI: 10.1007/s10618-014-0350-5.

[17] P. Chaudhari, D.P. Rana, R.G. Mehta, N.J. Mistry, and M.M. Raghuwanshi, "Discretization of temporal data: a sur-

vey", *Int. J. Computer Science and Information Security* 11 (2), 66–69 (2014).

[18] D. Farid and C. Rahman, "Mining complex data streams: discretization, attribute selection and classification", *J. Advances in Information Technology* 4 (3), 2013.

[19] J.W. Grzymala-Busse, "Discretization based on entropy and multiple scanning", *Entropy* 15 (5), 1486–1502 (2013), DOI: 10.3390/e15051486.

[20] L. Zou, D. Yan, H.R. Karimi, and P. Shi, "An algorithm for discretization of real value attributes based on interval similarity", *J. Applied Mathematics*, Article ID 350123 (2013), DOI: 10.1155/2013/350123.

[21] K. Shehzad, "Edisc: a class-tailored discretization technique for rule-based classification", *IEEE Trans. Knowledge and Data Engineering* 24 (8), 1435–1447 (2012), DOI: 10.1109/TKDE.2011.101.

[22] D.M. Maslove, T. Podchiyska, and H.J. Lowe, "Discretization of continuous features in clinical datasets", *J. American Medical Informatics Association* 20 (3), 544–553 (2013), DOI: 10.1136/amiajnl-2012-000929.

[23] M.G. Augasta and T. Kathirvalavakumar, "A new discretization algorithm based on range coefficient of dispersion and skewness for neural networks classifier", *Applied Soft Computing* 12 (2), 619–625 (2012), DOI: 10.1016/j.asoc.2011.11.001.

[24] J.L. Lustgarten, S. Visweswaran, V. Gopalakrishnan, and G.F. Cooper, "Application of an efficient Bayesian discretization method to biomedical data", *BMC Bioinformatics* 12 (1), 2011, DOI: 10.1186/1471-2105-12-309.

[25] T.W. Rondeau and C.W. Bostian, *Artificial Intelligence in Wireless Communications*, Artech House, London, 2009.

[26] G. Borowik and T. Łuba, "Fast algorithm of attribute reduction based on the complementation of Boolean function", *Advanced Methods and Applications in Computational Intelligence* 1, 25–41 (2014), DOI: 10.1007/978-3-319-01436-4_2.

[27] Z. Pawlak, *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, London, 1991, DOI: 10.1007/978-94-011-3534-4.

[28] H. Liu, F. Hussain, C. Tan, and M. Dash, "Discretization: An enabling technique", *Data Mining and Knowledge Discovery* 6 (4), 393–423 (2002), DOI: 10.1023/A:1016304305535.

[29] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, 1993.

[30] C.H. Papadimitriou, *Computational Complexity*, Academic Internet Publ., London, 2007.

[31] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, *Algorithms*, McGraw-Hill, London, 2008.

[32] B. Steinbach and C. Posthoff, "Improvements of the construction of exact minimal covers of Boolean functions", *Computer Aided Systems Theory – EUROCAST 2011, Lecture Notes in Computer Science* 6928, 272–279 (2012), DOI: 10.1007/978-3-642-27579-1_35.

[33] R.K. Brayton, G.D. Hachtel, C.T. McMullen, and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Academic Publishers, Berlin, 1984, DOI: 10.1007/978-1-4613-2821-6.

[34] C.E. Shannon, "A mathematical theory of communication", *Bell System Technical J.* 27 (3), 379–423 (1948), DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[35] G. Holmes, A. Donkin, and I.H. Witten, "Weka: a machine learning workbench", *Proc. 1994 Second Australian and New Zealand Conf. Intelligent Information Systems* 1, 357–361 (1994), DOI: 10.1109/ANZIIS.1994.396988.

[36] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, 1993.

[37] R. Kohavi, "The power of decision tables", *Machine Learning: ECML-95* 912, 174–189 (1995), DOI: 10.1007/3-540-59286-5_57.

[38] L.S. Cessie and J.C. van Houwelingen, "Ridge estimators in logistic regression", *Applied Statistics* 41 (1), 191–201 (1992).

[39] J.C. Platt, "Fast training of support vector machines using sequential minimal optimization", *Advances in Kernel Methods* 1, 185–208 (1999).

[40] L. Breiman, "Random forests," *Machine Learning* 45 (1), 5–32 (2001), DOI: 10.1023/A:1010933404324.