Fundamenta Informaticae 48 (2001) 61–81
IOS Press

On Efficient Handling of Continuous Attributes in Large Data Bases

Nguyen Hung Son*
Institute of Mathematics
Warsaw University
ul. Banacha 2, 02-097, Warsaw, Poland

Abstract. Some data mining techniques, like discretization of continuous attributes or decision tree induction, are based on searching for an optimal partition of data with respect to some optimization criteria. We investigate the problem of searching for optimal binary partition of continuous attribute domain in case of large data sets stored in *relational data bases* (RDB). The critical for time complexity of algorithms solving this problem is the number of I/O database operations necessary to construct such partitions. In our approach the basic operators are defined by queries on the number of objects characterized by means of real value intervals of continuous attributes. We assume the answer time for such queries does not depend on the interval length. The straightforward approach to the optimal partition selection (with respect to a given measure) requires $O(N)$ basic queries, where N is the number of preassumed partition parts in the searching space. We show properties of the basic optimization measures making possible to reduce the size of searching space. Moreover, we prove that using only $O(\log N)$ simple queries, one can construct a partition very close to optimal.

1. Introduction

Searching algorithms for optimal partitions of real value attributes (features) problem, defined by so called cuts, has been studied by many authors (see e.g., [1, 2, 3, 4, 18, 10]). The main goal of such algorithms is to discover *cuts* which can be used to synthesize decision trees or decision rules of high quality with respect to some quality measures (e.g., quality of classification of new unseen objects, quality defined by the decision tree height, support and confidence of decision rules). In general, all those problems are hard from computational point of view (e.g., it has been shown in [10] that the searching problem for minimal and consistent set of cuts is NP-hard). Hence, numerous heuristics have been developed for approximate solutions of these problems. These heuristics are based on some approximate measures estimating the quality of extracted cuts. In Section 2.1 we present examples of such measures.

*Adress for correspondence: Institute of Mathematics, Warsaw University, ul. Banacha 2, 02-097, Warsaw, Poland

Our approach is based on *discernibility measures* relevant for rough set approach [17]. All those methods are very efficient if data sets are stored in operational memory because (after sorting of data) the number of steps to check distribution of objects in intervals defined by consecutive cuts is of order $O(N)$. We consider a searching problem for optimal partition of real value attributes assuming that the large data table is represented in a relational data base. In such case even the linear complexity is not acceptable because of the time for one step. The critical factor for time complexity of algorithms solving the discussed problem is the number of simple SQL queries of the form

```
SELECT COUNT
FROM aTable
WHERE (anAttribute BETWEEN value1 AND value2)
    AND (additional condition)
```

(related to some interval of attribute values) necessary to construct such partitions. We assume the answer time for such queries does not depend on the interval length (this assumption is satisfied in some existing data base servers). Using straightforward approach to optimal partition selection (with respect to a given measure), the number of necessary queries is of order $O(N)$, where N is the number of preassumed parts of the searching space partition. We show some properties of considered optimization measures allowing to reduce the size of searching space. Moreover, we prove that using only $O(\log N)$ simple queries, one can construct a partition very close to optimal. We also extend the searching algorithm for the best cut presented in [14] by adding the global searching strategy.

2. Basic Notions

An *information system* [16] is a pair $\mathbb{A} = (U, A)$, where U is a non-empty, finite set called the *universe* and A is a non-empty finite set of *attributes (or features)*, i.e., $a : U \to V_a$ for $a \in A$, where V_a is called *the value set* (or domain) of a. Elements of U are called *objects or records*. Two objects $x, y \in U$ are said to be discernible by attributes from A if there exists an attribute $a \in A$ such that $a(x) \neq a(y)$.

In this paper we consider information systems of the form $\mathbb{A} = (U, A \cup \{dec\})$ where $dec \notin A$ is a distinguished attribute called *decision attribute* (or decision for short). Usually decision attributes are used to formulate classification problems. Such information systems are called *decision tables*. Without loss of generality we assume $V_{dec} = \{1, \ldots, d\}$. Then the set $DEC_k = \{x \in U : dec(x) = k\}$ will be called the k^{th} *decision class of* $\mathbb{A}$ for $1 \leq k \leq d$.

Any real value attribute a and any real number c define a partition of universe U into two disjoint subsets U_L and U_R, where

$$U_L = \{x \in U : a(x) \leq c\} \qquad U_R = \{x \in U : a(x) > c\}$$

If both U_L and U_R are not empty, then c is called *"a cut on attribute a"*. In general, the cut c on attribute a will be denoted by (a, c) (or, in short, c if a is uniquely determined by the context). We say that *"the cut c on a discerns a pair of objects x, y"* if either $a(x) < c \leq a(y)$ or $a(y) < c \leq a(x)$.

Definition 2.1. The set of cuts $\mathbf{P} = \{(a_{i_1}, c_1), (a_{i_2}, c_2), ..., (a_{i_k}, c_k)\}$ is $\mathbb{A}$*–consistent* iff for any pair of objects $x, y \in U$ if $dec(x) \neq dec(y)$ and x, y are discernible by attributes from A then there exists $(a_{i_j}, c_j) \in \mathbf{P}$ discerning x and y.

Definition 2.2. An $\mathbb{A}$-consistent set of cuts $\mathbf{P}$ is $\mathbb{A}$–*optimal* iff $card(\mathbf{P}) \leq card(\mathbf{P}')$ for any $\mathbb{A}$-consistent set of cuts $\mathbf{P}'$.

The discretization problem can be defined as a problem of searching for consistent, and optimal with respect to some criteria, set of cuts. In [10] it has been shown that the searching problem for discretization using minimal number of cuts is computationally hard.

Theorem 2.1. (see [10])
The problem of searching for the optimal set of cuts $\mathbf{P}$ in a given decision table $\mathbb{A}$ is NP-hard.

Since the searching problem for optimal set of cuts is NP-hard (i.e., there is no algorithm solving this problem in polynomial time, unless $P = NP$), we can only find a semi-optimal solution using some approximate algorithms. In the next section we shortly describe some methods often used in machine learning and data mining.

2.1. The Quality Measures

Developing decision tree induction methods (see [4, 18]) and some supervised discretization methods (see [1, 3, 10, 9, 11]), we should often solve the following problem:

> FOR A GIVEN REAL VALUE ATTRIBUTE a AND SET OF CANDIDATE CUTS $\{c_1, ..., c_N\}$, FIND A CUT c_i THAT MOST PROBABLY BELONGS TO THE SET OF OPTIMAL CUTS.

Usually, we use some *measure (or quality functions)* $\mathcal{F} : \{c_1, ..., c_N\} \to \mathbb{R}$ to estimate the quality of cuts. For a given measure $\mathcal{F}$, the *straightforward searching algorithm* for the best cut should compute the values of $\mathcal{F}$ for all cuts: $\mathcal{F}(c_1), .., \mathcal{F}(c_N)$. The cut c_{Best} optimizing (i.e., maximizing or minimizing) the value of function $\mathcal{F}$ is selected as the result of searching process.

For example, the algorithm for decision tree induction can be described as follows:

1. *For a given set of objects U, select an attribute a and a cut c_{Best} on a of highest quality among all possible cuts and all attributes;*
2. *Induce a partition U_L, U_R of U by a and c_{Best};*
3. *Recursively apply Step 1 to both sets U_L, U_R of objects until some stopping condition is satisfied.*

In the following sections we outline the most frequently used measures for decision tree induction and discretization like *"χ^2 Test", "Entropy Function" and "Discernibitity Measure"*, respectively. First we fix some notations. Let us consider an attribute a and a set of candidate cuts $\mathbf{C}_a = \{c_1, ..., c_N\}$ on a.

Definition 2.3. *A class distribution* of the set of objects $X \subset U$ is a tuple of integers $(x_1, .., x_d)$ where $x_k = card(X \cap DEC_k)$ for $k \in \{1, ..., d\}$. If the set of objects X is defined by $X = \{u \in U : p \leq a(u) < q\}$ for some $p, q \in \mathbb{R}$ then the class distribution of X is called *the class distribution of a in $[p; q)$*.

Any cut $c \in \mathbf{C}_a$ splits the domain $V_a = (l_a, r_a)$ of the attribute a into two intervals: $I_L = (l_a, c)$; $I_R = [c, r_a)$. For a fixed cut c on a we use the following notation:

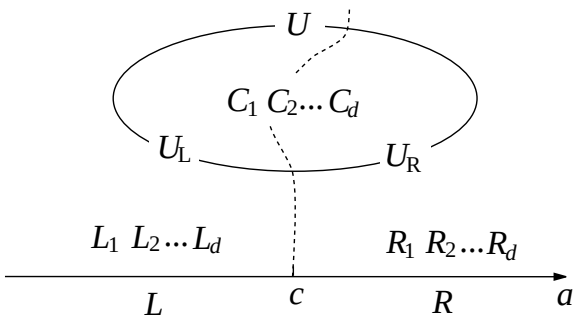

Figure 1. The partition of the set of objects U defined by a cut c on attribute a

- d – the number of decision classes;
- U_{L_j}, U_{R_j} – the sets of objects from j^{th} class in I_L and I_R;
- $U_L = U_{L_1} \cup ... \cup U_{L_d}$ and $U_R = U_{R_1} \cup ... \cup U_{R_d}$;
- $(L_1, .., L_d)$ and $(R_1, .., R_d)$ – the class distributions in U_L and U_R (where $L_j = |U_{L_j}|$ and $R_j = |U_{R_j}|$);
- $L = \sum_{j=1}^{d} L_j$ and $R = \sum_{j=1}^{d} R_j$
- $C_j = L_j + R_j$ – the number of objects in the j^{th} class;
- $n = \sum_{i=1}^{d} C_j = L + R$ – the total number of objects;
- $E(U_{L_j}) = \frac{L \times C_j}{n}$ – the expected frequency of U_{L_j};
- $E(U_{R_j}) = \frac{R \times C_j}{n}$ – the expected frequency of U_{R_j};

where $j \in \{1, ..., d\}$.

2.2. Statistical test methods

Statistical tests allow to check the probabilistic independence between the object partition defined by the decision attribute and by the cut c. The independence degree is estimated by the χ^2 test given by

$$\chi^2(c) = \sum_{j=1}^{d} \frac{\left(L_j - E(U_{L_j})\right)^2}{E(U_{L_j})} + \sum_{j=1}^{d} \frac{\left(R_j - E(U_{R_j})\right)^2}{E(U_{R_j})}$$

Intuitively, if the partition defined by c does not depend on the partition defined by the decision attribute dec then we have $\chi^2(c) = 0$. In opposite case if there exists a cut c which properly separates objects from different decision classes the value of χ^2 test for c is becoming high.

Discretization methods based on χ^2 test are choosing only cuts with large value of this test (and delete the cuts with small value of χ^2 test). There are different versions of this method (see e.g., ChiMerge [8] and Chi2 [9]).

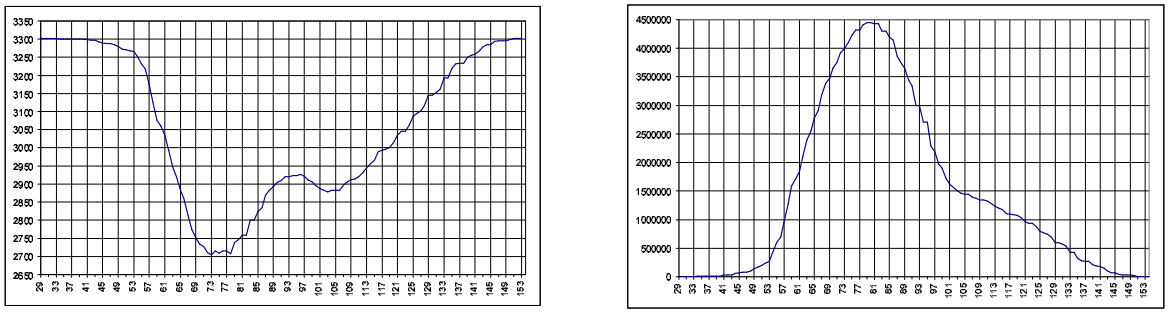

Figure 2. The illustration of entropy measure (left) discernibility measure(right) for the same data. Horizontal axes are labelled by indexes of consequent cuts; vertical axes are labelled by values of those measures.

2.3. Entropy methods

Many methods based on entropy measure have been developed in the domain of decision tree induction and discretization. They use class-entropy as a criterion to evaluate the list of the best cuts which together with the attribute domain induce the desired intervals. The class information entropy of the set of k objects X with class distribution $(x_1, ..., x_d)$, where $x_1 + ... + x_d = k$, is defined by

$$Ent(X) = -\sum_{j=1}^{d} \frac{x_j}{k} \log \frac{x_j}{k}$$

Let $U = U_L \cup U_R$ be a partition of U defined by the cut c on a. *Information Gain* over the set of objects U received by the cut c on a is defined by

$$Gain(a, c; U) = Ent(U) - \left(\frac{|U_L|}{|U|} Ent\,(U_L) + \frac{|U_R|}{|U|} Ent\,(U_R) \right)$$

For a given feature a, the cut c_{Best} that maximizes the *information gain* $Gain(a, c; U)$ over all possible cuts is selected. The cut maximizing information gain $Gain(a, c; U)$ also minimizes the *Entropy induced by this cut* defined by

$$E\,(a, c; U) = \frac{|U_L|}{n} Ent\,(U_L) + \frac{|U_R|}{n} Ent\,(U_R)$$

Many methods based on information entropy theory are reported in, e.g., [1, 5, 2, 18].

2.4. Boolean reasoning methods

If Boolean reasoning is used then cuts are treated as Boolean variables and the searching problem for the optimal set of cuts can be characterized by a Boolean function $f_{\mathbb{A}}$ (where $\mathbb{A}$ is a given decision table). Any set of cuts is $\mathbb{A}$-consistent if and only if the corresponding evaluation of variables in $f_{\mathbb{A}}$ returns the value $True$ (see [10]). In [10] it was shown that the quality of cuts can be measured by their *discernibility properties*. Intuitively, internal conflict of the set of objects $X \subset U$ can be defined by the number of

pairs of objects from X to be discerned. Let $(x_1, ..., x_d)$ be a class distribution of X, then $conflict(X)$ can be computed by

$$conflict(X) = \sum_{i<j} x_i x_j$$

The cut c which splits the set of objects U into U_L, and U_R is evaluated by

$$W(a, c) = conflict(U) - conflict(U_L) - conflict(U_R)$$

i.e., the more pairs of objects are discerned by the cut c on a, the larger is a chance that c can be included to the optimal set of cut. Hence, in the discretization and decision tree induction algorithms based on Boolean reasoning approach, the quality of a given cut c on a is defined by the number of pairs of objects discerned by c, i.e.,

$$W(a, c) = \sum_{i \neq j}^{d} L_i R_j = \sum_{i=1}^{d} L_i \sum_{i=1}^{d} R_i - \sum_{i=1}^{d} L_i R_i \tag{1}$$

Algorithms based on the discernibility measure are called *the MD-heuristics*[1].

2.5. Complexity of searching for best cuts

Assume a set of candidate cuts $\mathbf{C}_a = \{c_1, ..., c_N\}$ on an attribute a and the quality measure $\mathcal{F} : \mathbf{C}_a \to \mathbb{R}^+$ are given. Any searching algorithm for the best cuts from $\mathbf{C}_a$ with respect to measure $\mathcal{F}$ requires at least $O(N + n)$ steps, where n is the number of objects in decision table. In case of large data tables stored in relational data base, it requires at least $O(Nd)$ simple queries, because for every cut $c_i \in \mathbf{C}_a$ the algorithm is using the class distribution $(L_1, ..., L_d)$ of a in $(-\infty, c_i)$ and the class distribution $(R_1, ..., R_d)$ of a in $[c_i, \infty)$ to compute the value of $\mathcal{F}(c_i)$.

Let us consider the client–server architecture with many data tables containing millions of objects stored in a server. In such architecture, many clients perform at the same time decision tree induction or discretization for different tables. Then for each client, the set $\mathbf{C}_a$ can have millions of candidate cuts. The number of simple queries grows to millions and in consequence the time complexity of algorithm becomes unacceptable. Of course, some simple queries can be wrapped in packages or replaced by complex queries, but the data base still has to transfer millions of class distributions from server to every client. Hence, straightforward methods can not be realized in client–server environment.

The most popular strategy used in data mining is based on sampling technique, i.e., building a model (i.e., either decision tree or discretization) for small, randomly selected subset of data, and next on evaluating the quality of decision tree for the whole data. If the quality of generated model is not sufficient enough, we have to repeat this step for a new sample (see e.g., [7]).

We would like to present an alternative solution to sampling techniques.

3. Algorithm Acceleration Methods

In this section we present some properties of discernibility measure (based on Boolean reasoning approach). These make possible to apply MD heuristics to induce decision trees and perform discretization

[1] Maximal-Discernibility heuristics

of real value attributes directly from large data. We will expand the presented methods in this section for other measures in the next section.

3.1. Properties of the Discernibility Measure

In this section we consider cuts on a fixed attribute a. For any cut c on a we denote the discernibility function of c by $W(c)$ instead of $W(a,c)$.

First, let us consider two cuts $c_L < c_R$. Let $(L_1, ..., L_d)$ be the class distribution in $(-\infty; c_L)$, $(M_1, ..., M_d)$ – the class distribution in $[c_L, c_R)$ and $(R_1, ..., R_d)$ – the class distribution in $[c_R; \infty)$ (see Figure 3).

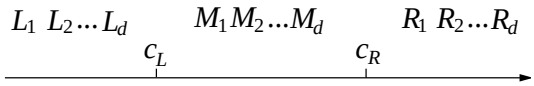

Figure 3. The class distributions defined by cuts c_L, c_R

Now we are ready to show how to compute the difference between the discernibility measures of c_L and c_R using information about class distribution in intervals defined by these cuts. The exact formula is given in the following lemma.

Lemma 3.1. The following equation holds:

$$W(c_R) - W(c_L) = \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} M_j \right] \tag{2}$$

Proof:
We have

$$\begin{aligned} W(c_L) &= \sum_{i=1}^{d} L_i \sum_{i=1}^{d} (M_i + R_i) - \sum_{i=1}^{d} L_i (M_i + R_i) = \\ &= \sum_{i=1}^{d} L_i \sum_{i=1}^{d} M_i + \sum_{i=1}^{d} L_i \sum_{i=1}^{d} R_i - \sum_{i=1}^{d} L_i (M_i + R_i) \end{aligned}$$

Analogously

$$\begin{aligned} W(c_R) &= \sum_{i=1}^{d} (L_i + M_i) \sum_{i=1}^{d} R_i - \sum_{i=1}^{d} (L_i + M_i) R_i = \\ &= \sum_{i=1}^{d} L_i \sum_{i=1}^{d} R_i + \sum_{i=1}^{d} M_i \sum_{i=1}^{d} R_i - \sum_{i=1}^{d} L_i (M_i + R_i) \end{aligned}$$

Hence,

$$\begin{aligned} W(c_R) - W(a, c_L) &= \sum_{i=1}^{d} M_i \left(\sum_{i=1}^{d} R_i - \sum_{i=1}^{d} L_i \right) - \sum_{i=1}^{d} ((L_i + M_i)R_i - L_i(M_i + R_i))) \\ &= \sum_{i=1}^{d} M_i \sum_{i=1}^{d} (R_i - L_i) - \sum_{i=1}^{d} M_i(R_i - L_i) \end{aligned}$$

After simplifying of the last formula we obtain (2). □

Our goal is to find cuts maximizing the function $W(c)$. We define the notion of *boundary cut* and we recall the well known notion in statistics called *median* (using the notation presented in Section 2.1). Let $\mathbf{C}_a = \{c_1, .., c_N\}$ be a set of candidate cuts on an attribute a such that $c_1 < c_2 < ... < c_N$. Then

Definition 3.1. Any cut $c_i \in \mathbf{C}_a$, where $1 < i < N$, is called the *boundary cut* if $a(u_1) \in [c_{i-1}, c_i)$, $a(u_2) \in [c_i, c_{i+1})$ and $dec(u_1) \neq dec(u_2)$ for some objects $u_1, u_2 \in U$.

Definition 3.2. By a median of the k^{th} decision class we mean a cut $c \in \mathbf{C}_a$ minimizing the value $|L_k - R_k|$. The median of the k^{th} decision class will be denoted by $Median(k)$.

We will show that it is enough to restrict the search to the set of boundary cuts.

Theorem 3.1. The cut c_{Best} maximizing the function W(a,c) can be found among boundary cuts.

Proof:
Assume that c_a and c_b are consecutive boundary cuts. Then the interval $[c_a, c_b)$ consists of objects from one decision class, say C_i. For arbitrary cuts c_L and c_R such that $c_a \leq c_L < c_R \leq c_b$, we have $M_i \neq 0$ and $\forall_{j \neq i} M_j = 0$. Then the equation 2 has a form

$$W(c_R) - W(c_L) = M_i \sum_{j \neq i} (R_j - L_j)$$

Thus, function $W(c)$ is monotonic in the interval $[c_a, c_b)$ because $\sum_{j \neq i}(R_j - L_j)$ is constant for all sub intervals of $[c_a, c_b)$. □

Theorem 3.1 makes it possible to restrict searching for optimal cuts to boundary cuts only. This property also holds for Entropy measures (see [4]). This property is interesting but can not be used to construct any efficient heuristic for the investigated in the paper problem because of high complexity of the algorithm detecting the boundary cuts (in case of data tables stored in RDB). However, it was possible to find another property allowing to eliminate the large number of cuts. Let $c_1 < c_2 ... < c_N$ be the set of candidate cuts, and let

$$c_{min} = \min_i \{Median(i)\} \text{ and } c_{max} = \max_i \{Median(i)\}$$

Then we have the following theorem:

Theorem 3.2. The quality function $W : \{c_1,..,c_N\} \to \mathbb{N}$ defined over the set of cuts is increasing in $\{c_1, ..., c_{min}\}$ and decreasing in $\{c_{max}, ..., c_N\}$. Hence

$$c_{Best} \in \{c_{min}, ..., c_{max}\}$$

Proof:
Let us consider two cuts $c_L < c_R < c_{min}$. Using Equation 2 we have

$$W(c_R) - W(c_L) = \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} M_j \right]$$

Because $c_L < c_R < c_{min}$, hence $R_i - L_i \geq 0$ for any $i = 1, ...d$. Thus $W(c_R) \geq W(c_L)$. □

This property is interesting because it states that one can use only $O(d \log N)$ queries to determine the medians of decision classes by using Binary Search Algorithm. Hence one can reduce the searching space using $O(d \log N)$ SQL queries. Let us also observe that if all decision classes have similar medians then almost all cuts can be eliminated.

3.2. "Divide and Conquer" Algorithm

The main idea is to apply the *"divide and conquer"* strategy to determine the best cut $c_{Best} \in \{c_1, ..., c_n\}$ with respect to a given quality function.

First we divide the interval containing all possible cuts into k intervals ($k = 2, 3$, etc.). We will use some *approximate discernible measures* to predict the interval which most probably contains the best cut with respect to discernibility measure. This process is repeated until the considered interval consists of one cut. Then the best cut can be chosen between all visited cuts.

The problem arises how to define the measure evaluating the quality of the interval $[c_L, c_R]$ having class distributions: $(L_1, ..., L_d)$ in $(-\infty, c_L)$; $(M_1, ..., M_d)$ in $[c_L, c_R)$; and $(R_1, ..., R_d)$ in $[c_R, \infty)$ (see Figure 3). This measure should estimate the quality of the best cut among those belonging to the interval $[c_L, c_R]$.

We consider two specific probabilistic models for distribution of objects in the interval $[c_L, c_R]$.

Let us consider an arbitrary cut c lying between c_L and c_R and let us assume that $(x_1, x_2, ..., x_d)$ is a class distribution of the interval $[c_L, c]$. Let us we assume that $x_1, x_2, ..., x_d$ are independent random variables with uniform distribution over sets $\{0, ..., M_1\}$, ..., $\{0, ..., M_d\}$, respectively. This assumption is called "*full independency assumption*". One can observe that under this assumption

$$E(x_i) = \frac{M_i}{2} \text{ and } D^2(x_i) = \frac{M_i(M_i + 2)}{12}$$

for all $i \in \{1, .., d\}$. We have the following theorem

Theorem 3.3. The mean $E(W(c))$ of quality $W(c)$ for any cut $c \in [c_L, c_R]$ satisfies

$$E(W(c)) = \frac{W(c_L) + W(c_R) + conflict([c_L, c_R])}{2} \tag{3}$$

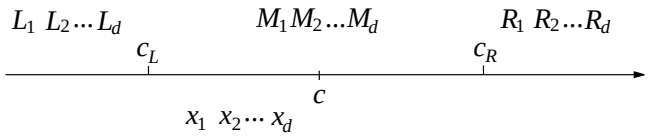

Figure 4. A random cut c and the random class distribution $x_1, ..., x_d$ induced by c

where $conflict([c_L, c_R]) = \sum_{i \neq j} M_i M_j$. For the standard deviation of $W(c)$ we have

$$D^2(W(c)) = \sum_{i=1}^{n} \left[\frac{M_i(M_i+2)}{12} \left(\sum_{j \neq i} (R_j - L_j) \right)^2 \right] \tag{4}$$

Proof:
Let us consider any random cut c lying between c_L and c_R. The situation is shown in the Figure 4.

$$\begin{aligned} W(c) - W(c_L) &= \sum_{i=1}^{d} \left[(R_i + M_i - x_i - L_i) \sum_{j \neq i} x_j \right] \\ &= \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} x_j + (M_i - x_i) \sum_{j \neq i} x_j \right] \\ W(c) - W(c_R) &= \sum_{i=1}^{d} \left[(L_i + x_i - R_i) \sum_{j \neq i} (M_j - x_j) \right] \\ &= \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} (x_j - M_j) + x_i \sum_{j \neq i} (M_j - x_j) \right] \end{aligned}$$

Thus

$$2W(c) - (W(c_L) + W(c_R)) = 2 \sum_{i \neq j} x_i (M_j - x_j) + \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} (2x_j - M_j) \right]$$

Hence

$$W(c) = \frac{W(c_L) + W(c_R)}{2} + \sum_{i \neq j} x_i (M_j - x_j) + \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} \left(x_j - \frac{M_j}{2} \right) \right] \tag{5}$$

Then we have

$$\begin{aligned} E(W(c)) &= \frac{W(c_L)+W(c_R)}{2} + \sum_{i\neq j} E(x_i)(M_j - E(x_j)) \\ &\quad + \sum_{i=1}^{d}\left[(R_i - L_i)\sum_{j\neq i}\left(E(x_j) - \frac{M_j}{2}\right)\right] \\ &= \frac{W(c_L)+W(c_R)}{2} + \frac{1}{4}\sum_{i\neq j} M_i M_j \\ &= \frac{W(c_L)+W(c_R)+conflict(c_L,c_R)}{2} \end{aligned}$$

In the consequence we have

$$W(c) - E(W(c)) = \sum_{i\neq j}\left(x_i - \frac{M_i}{2}\right)\left[(R_j - L_j) - \left(x_j - \frac{M_j}{2}\right)\right]$$

Thus

$$\begin{aligned} D^2(W(c)) &= E\left([W(c) - E(W(c))]^2\right) \\ &= E\left(\sum_{i\neq j}\left(x_i - \frac{M_i}{2}\right)\left[(R_j - L_j) - \left(x_j - \frac{M_j}{2}\right)\right]\right) \\ &= \sum_{i=1}^{n}\left[\frac{M_i(M_i+2)}{12}\left(\sum_{j\neq i}(R_j - L_j)\right)^2\right] \end{aligned}$$

what ends the proof. □

One can use formulas (3) and (4) to construct a measure estimating the quality of the interval $[c_L, c_R]$

$$Eval\left([c_L, c_R], \alpha\right) = E(W(c)) + \alpha\sqrt{D^2(W(c))} \tag{6}$$

where the real parameter α from $[0, 1]$ can be tuned in learning process. The details of our algorithm can be described as follows:

ALGORITHM: Searching for semi-optimal cut

PARAMETERS: $k \in \mathbb{N}$ and $\alpha \in [0;1]$.

INPUT: attribute a; the set of candidate cuts $\mathbf{C}_a = \{c_1, .., c_N\}$ on a;

OUTPUT: The optimal cut $c \in \mathbf{C}_a$

begin

$Left \leftarrow \min$; $Right \leftarrow \max$; {see Theorem 3.2}

while $(Left < Right)$

1. Divide $[Left; Right]$ into k intervals with equal length by $(k+1)$ boundary points i.e.,
$$p_i = Left + i * \frac{Right - Left}{k};$$
for $i = 0, .., k$.

2. For $i = 1, .., k$ compute $Eval([c_{p_{i-1}}; c_{p_i}], \alpha)$ using Formula (6). Let $[p_{j-1}; p_j]$ be the interval with maximal value of $Eval(.)$;

3. $Left \leftarrow p_{j-1}$; $Right \leftarrow p_j$;

endwhile;

Return the cut c_{Left};

end

One can see that to determine the value $Eval\,([c_L, c_R], \alpha)$ we need to have the class distributions $(L_1, ..., L_d)$, $(M_1, ..., M_d)$ and $(R_1, ..., R_d)$ of the attribute a in $(-\infty, c_L)$, $[c_L, c_R)$ and $[c_R, \infty)$. This requires only $O(d)$ simple SQL queries of the form:

```
SELECT COUNT
FROM DecTable
WHERE (attribute_a BETWEEN value_1 AND value_2) AND (dec = i)
```

Hence the number of queries required for running our algorithm is of order $O(dk \log_k N)$. In practice we set $k = 3$ because the function $f(k) = dk \log_k N$ over positive integers is taking minimum for $k = 3$. For $k > 2$, instead of choosing the best interval $[p_{i-1}, p_i]$, the algorithm can select the best union $[p_{i-m}, p_i]$ of m consecutive intervals in every step for a predefined parameter $m < k$. The modified algorithm needs more – but still $O(\log N)$ simple queries only.

3.3. Examples

We consider a data table consisting of 12000 records. Objects are classified into 3 decision classes with the distribution $(5000, 5600, 1400)$, respectively. One real value attribute has been selected and $N = 500$ cuts on its domain have generated class distributions as shown in Figure 5.

The medians of classes are c_{166}, c_{414} and c_{189}, respectively. The median of every decision class has been determined by *binary search algorithm* using $\log N = 9$ simple queries. Applying Theorem 3.2 we conclude that it is enough to consider only cuts from $\{c_{166}, ..., c_{414}\}$. In this way 251 cuts have been eliminated by using 27 simple queries only.

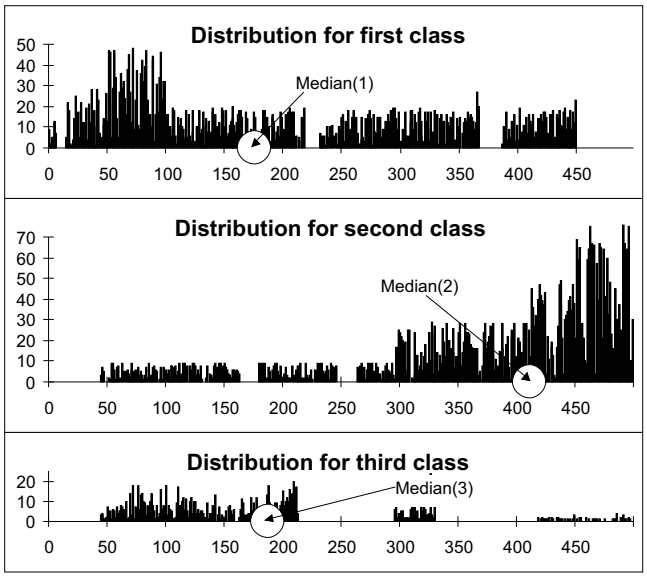

Figure 5. Distributions for decision classes 1, 2, 3.

In Figure 6 we show the graph of $W(c_i)$ for $i \in \{166, ..., 414\}$ and we illustrate the outcome of application of our algorithm to the reduce set of cuts for $k = 2$ and $\alpha = 0$.

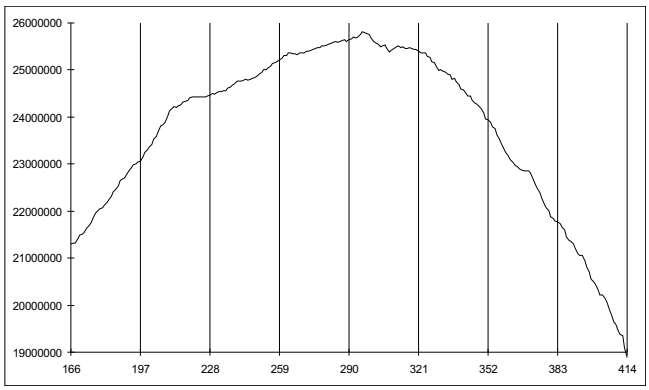

Figure 6. Graph of $W(c_i)$ for $i \in \{166, .., 414\}$.

First the cut c_{290} is chosen and it is necessary to determine to which of the intervals $[c_{166}, c_{290}]$ and $[c_{290}, c_{414}]$ the best cut belongs. The values of function $Eval$ on these intervals are computed: $Eval([c_{166}, c_{290}], 0) = 23927102$, $Eval([c_{290}, c_{414}], 0) = 24374685$. Hence, the best cut is predicted to belong to $[c_{290}, c_{414}]$ and the search process is reduced to the interval $[c_{290}, c_{414}]$. The above procedure is repeated recursively until the selected interval consists of single cut only. For our example, the best cut c_{296} has been successfully selected by our algorithm. In general the cut selected by the algorithm is

not necessarily the best. However numerous experiments on different large data sets have shown that the cut c^* returned by the algorithm is close to the best cut c_{Best} (i.e., $\frac{W(c^*)}{W(c_{Best})} \cdot 100\%$ is about 99.9%).

4. Local and Global Search

The presented above algorithm is called also "*local search strategy*". Using local search algorithm, first we have to discover the best cuts on every attribute separately. Next, we compare all locally best cuts to find out the best globally. This is a typical search strategy for decision tree construction (see e.g., [18]).

The approximate measure makes possible to construct "*global search strategy*" for the best cuts. This strategy becomes helpful if we want to control the computation time, because it performs both attribute selection and cut selection processes at the same time.

The global strategy is searching for the best cut over all attributes. At the beginning, the best cut can belong to every attribute, hence for each attribute we keep the interval in which the best cut can be found (see Theorem 3.2), i.e., we have a collection of all potential intervals

$$\textbf{Interval\_Lists} = \{(a_1, l_1, r_1), (a_2, l_2, r_2), ..., (a_k, l_k, r_k)\}$$

Next we iteratively run the following procedure

- *remove the interval $I = (a, c_L, c_R)$ consisting of the best cut with highest probability (using Formula 6);*
- *divide interval I into smaller ones $I = I_1 \cup I_2 ... \cup I_k$;*
- *insert $I_1, I_2, ..., I_k$ to* **Interval_Lists**.

This iterative step can be continued until we have one–element interval or the time limit of searching algorithm is exhausted. This strategy can be simply implemented using priority queue to store the set of all intervals, where priority of intervals is defined by Formula 6.

5. Further Results

We presented the approximate discernibility measure with respect to the full independency assumption, i.e., assuming distributions of objects from each decision class in $[c_L, c_R]$ are independent. Under this assumption, the quality of the best cut in interval $[c_L, c_R]$ was evaluated by

$$\frac{W(c_L) + W(c_R) + conflict([c_L, c_R])}{2} + \alpha \sqrt{\sum_{i=1}^{n} \left[\frac{M_i(M_i+2)}{12} \left(\sum_{j \neq i} (R_j - L_j) \right)^2 \right]}$$

for some $\alpha \in [0, 1]$.

In this section we would like to consider the approximate discernibility under "*full dependency assumption*" as well as approximate entropy measure under both independency and dependency assumptions.

The full dependency is based on the assumption that the values x_1,...,x_d are proportional to M_1,...,M_d, i.e.,

$$\frac{x_1}{M_1} \simeq \frac{x_2}{M_2} \simeq ... \simeq \frac{x_d}{M_d}$$

Let $x = x_1 + ... + x_d$ and let $t = \frac{x}{M}$, we have

$$x_1 \simeq M_1 \cdot t; \qquad x_2 \simeq M_2 \cdot t; \qquad ... \qquad x_d \simeq M_d \cdot t \tag{7}$$

where t is a real number from $[0, 1]$.

5.1. Approximation of discernibility measure under full dependency assumption

After replacing the values of $x_1, ..., x_d$ in (7) to Equation 5 we have

$$\begin{aligned} W(c) &= \frac{W(c_L) + W(c_R)}{2} + \sum_{i \neq j} x_i(M_j - x_j) + \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} \left(x_j - \frac{M_j}{2} \right) \right] \\ &= \frac{W(c_L) + W(c_R)}{2} + \sum_{i \neq j} M_i \cdot t \, (M_j - M_j \cdot t) + \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} \left(M_j \cdot t - \frac{M_j}{2} \right) \right] \\ &= At^2 + Bt + C \end{aligned}$$

where

$$\begin{aligned} A &= -\left(\sum_{i \neq j} M_i \cdot M_j \right) = -2 \cdot conflict([c_L, c_R]) \\ B &= \sum_{i \neq j} M_i \cdot M_j + \sum_{i \neq j} M_i \cdot (R_j - L_j) = 2 \cdot conflict([c_L, c_R]) + W(c_R) - W(c_L) \\ C &= \frac{W(c_L) + W(c_R)}{2} - \frac{1}{2} \sum_{i \neq j} M_i \cdot (R_j - L_j) \\ &= \frac{W(c_L) + W(c_R)}{2} - \frac{W(c_R) - W(c_L)}{2} = W(c_L) \end{aligned}$$

We want to find the maximal value of $f(t)$ for $t \in [0, 1]$.

It is easy to check that the function $f(t) = At^2 + Bt + C$ with $A < 0$ reaches his global maximum for

$$t_{max} = -\frac{B}{2A} = \frac{1}{2} + \frac{W(c_R) - W(c_L)}{4 \cdot conflict([c_L, c_R])}.$$

and the maximal value is equal to

$$f(t_{max}) = -\frac{\Delta}{4A} = \frac{W(c_L) + W(c_R) + conflict([c_L, c_R])}{2} + \frac{[W(c_R) - W(c_L)]^2}{8 \cdot conflict([c_L, c_R])}$$

Then we have

$$\max_{t \in [0,1]} f(t) = \begin{cases} f(t_{max}) & \text{if } t_{max} \in [0, 1] \\ \max\{f(0), f(1)\} & \text{otherwise} \end{cases}$$

Thus we have the following

Theorem 5.1. Under full independency assumption, the quality of the interval $[c_R, c_L]$ can be evaluated by $Eval([c_L, c_R])$, where

- if $|W(c_R) - W(c_L)| < 2 \cdot conflict([c_L, c_R])$ then

$$Eval([c_L, c_R]) = \frac{W(c_L) + W(c_R) + conflict([c_L, c_R])}{2} + \frac{[W(c_R) - W(c_L)]^2}{8 \cdot conflict([c_L, c_R])} \qquad (8)$$

- otherwise $Eval([c_L, c_R])$ is evaluated by

$$\max\{W(c_L), W(c_R)\}$$

One can see that for both dependency and independency assumptions, the discernibility measure of the best cut in the interval $[c_R, c_L]$ can be evaluated by the same component

$$\frac{W(c_L) + W(c_R) + conflict([c_L, c_R])}{2}$$

and it is extended by the second component δ, where

$$\begin{aligned} \delta &= \frac{[W(c_R) - W(c_L)]^2}{8 \cdot conflict([c_L, c_R])} \quad \text{(under full dependency assumption)} \\ \delta &= \alpha \cdot \sqrt{D^2(W(c))} \qquad \text{for some } \alpha \in [0,1]; \quad \text{(under full independency assumption)} \end{aligned}$$

Moreover, under full dependency assumption, one can predict the placement of the best cut. This observation is very useful in construction of efficient algorithms.

5.2. Approximate Entropy Measures

In previous sections, the discernibility measure has been successfully approximated. The experimental results show that the decision tree or discretization of real value attributes constructed by means of approximate discernibility measures (using small number of SQL queries) are very close to those which are generated by the exact discernibility measure (but using large number of SQL queries). In this section, we would like present similar results for entropy measure.

Using the standard Entropy-based methods (see e.g., [18]) we need the following notions:

1. *Information measure* of the set of objects *U*

$$\begin{aligned} Ent(U) &= -\sum_{j=1}^{d} \frac{N_j}{N} \log \frac{N_j}{N} = -\sum_{j=1}^{d} \frac{N_j}{N} (\log N_j - \log N) = \log N - \frac{1}{N} \sum_{j=1}^{d} N_j \log N_j \\ &= \frac{1}{N} \left(N \log N - \sum_{j=1}^{d} N_j \log N_j \right) = \frac{1}{N} \left(h(N) - \sum_{j=1}^{d} h(N_j) \right) \end{aligned}$$

where $h(x) = x \log x$.

2. *Information Gain* over the set of objects U received by the cut c is defined by

$$Gain(a,c;U) \quad = \quad Ent(U) - \left(\frac{|U_L|}{|U|} Ent\left(U_L\right) + \frac{|U_R|}{|U|} Ent\left(U_R\right) \right)$$

where $\{U_L, U_R\}$ is a partition of U defined by c. We have to chose such a cut c that maximizes the *information gain* $Gain(a,c;U)$ or minimizes the *Entropy induced by this cut*

$$\begin{aligned} Ent\left(a,c;U\right) &= \frac{|U_L|}{|U|} Ent\left(U_L\right) + \frac{|U_R|}{|U|} Ent\left(U_R\right) \\ &= \frac{L}{N}\left[\frac{1}{L}\left(h(L) - \sum_{j=1}^{d} h(L_j)\right)\right] + \frac{R}{N}\left[\frac{1}{R}\left(h(R) - \sum_{j=1}^{d} h(R_j)\right)\right] \\ &= \frac{1}{N}\left[h(L) - \sum_{j=1}^{d} h(L_j) + h(R) - \sum_{j=1}^{d} h(R_j)\right] \end{aligned}$$

where $(L_1, ..., L_d)$, $(R_1, ..., R_d)$ are class distribution of U_L and U_R, respectively.

Analogously to the discernibility measure case, the main goal is to predict the quality of the best cut (in sense of Entropy measure) among those from the interval $[c_L, c_R]$, i.e., $Ent(a,c;U) = \frac{1}{N} f(x_1, ..., x_d)$ where

$$f(x_1, ..., x_d) = h(L + x) - \sum_{j=1}^{d} h(L_j + x_j) + h(R + M - x) - \sum_{j=1}^{d} h(R_j + M_j - x_j)$$

5.2.1. Approximation of entropy measure under full dependency assumption

In this model, the values $x_1, ..., x_j$ can be replaced by

$$x_1 \simeq M_1 \cdot t; \qquad x_2 \simeq M_2 \cdot t; \qquad ... \qquad x_d \simeq M_d \cdot t$$

where $t = \frac{x}{M} \in [0;1]$ (see Section 5.1). Hence, the task is to find the minimum of the function

$$f(t) = h(L + M \cdot t) - \sum_{j=1}^{d} h(L_j + M_j \cdot t) + h(R + M - M \cdot t) - \sum_{j=1}^{d} h(R_j + M_j - M_j \cdot t)$$

where $h(x) = x \log x$ and $h'(x) = \log x + \log e$. Let us consider the derivative of function $f(t)$

$$\begin{aligned} f'(t) &= M \log(L + M \cdot t) - \sum_{j=1}^{d} M_j \log(L_j + M_j \cdot t) \\ &\quad -M \log(R + M - M \cdot t) + \sum_{j=1}^{d} M_j \log(R_j + M_j - M_j \cdot t) \\ &= M \log \frac{L + M \cdot t}{R + M - M \cdot t} - \sum_{j=1}^{d} M_j \log \frac{L_j + M_j \cdot t}{R_j + M_j - M_j \cdot t} \end{aligned}$$

Theorem 5.2. $f'(t)$ is decreasing function.

Proof:
Let us compute the second derivative of $f(t)$:

$$f''(t) = \frac{M^2}{L+M\cdot t} - \sum_{j=1}^{d}\frac{M_j^2}{L_j+M_j\cdot t} + \frac{M^2}{R+M-M\cdot t} - \sum_{j=1}^{d}\frac{M_j^2}{R_j+M_j-M_j\cdot t}$$

One can show that $f''(t)\leq 0$ for any $t\in(0,1)$. Recall the well known Minski inequality:

$$\sum_{i=1}^{n}a_i^2\sum_{i=1}^{n}b_i^2 \geq \left(\sum_{i=1}^{n}a_ib_i\right)^2$$

for any $a_1,...,a_n,b_1,...,b_n\in\mathbb{R}$. Using this inequality we have:

$$(L+M\cdot t)\sum_{j=1}^{d}\frac{M_j^2}{L_j+M_j\cdot t} = \sum_{j=1}^{d}(L_j+M_j\cdot t)\sum_{j=1}^{d}\frac{M_j^2}{L_j+M_j\cdot t} \geq \left(\sum_{j=1}^{d}M_j\right)^2 = M^2$$

Hence

$$\sum_{j=1}^{d}\frac{M_j^2}{L_j+M_j\cdot t} \geq \frac{M^2}{L+M\cdot t}$$

Similarly we can show that

$$\sum_{j=1}^{d}\frac{M_j^2}{R_j+M_j-M_j\cdot t} \geq \frac{M^2}{R+M-M\cdot t}$$

Hence, for any $t\in(0;1)$ we have $f''(t)\leq 0$. This means that $f'(t)$ is decreasing function in the interval $(0,1)$. □

The following example illustrates the properties of $f'(t)$. Let us consider the interval (c_L,c_R) consisting of 600 objects. The class distributions of intervals $(-\infty;c_l)$, (c_L,c_R) and $(c_R;\infty)$ are following:

	Left	Center	Right
Dec = 1	L_1 = 500	M_1 = 100	R_1 =1000
Dec = 2	L_2 = 200	M_2 = 400	R_2 =800
Dec = 3	L_3 = 300	M_3 = 100	R_3 =200
Sum	L = 1000	M = 600	R =2000

For this data the graph of deviative $f'(t)$ is shown in the Figure 7.

The proved fact can be used to find the value t_0, for which $f'(t_0)=0$. If such t_0 exists, the function f has maximum at t_0. Hence, one can estimate the quality of the interval $[c_L,c_R]$ (under assumption about strong dependencies between classes) as follows:

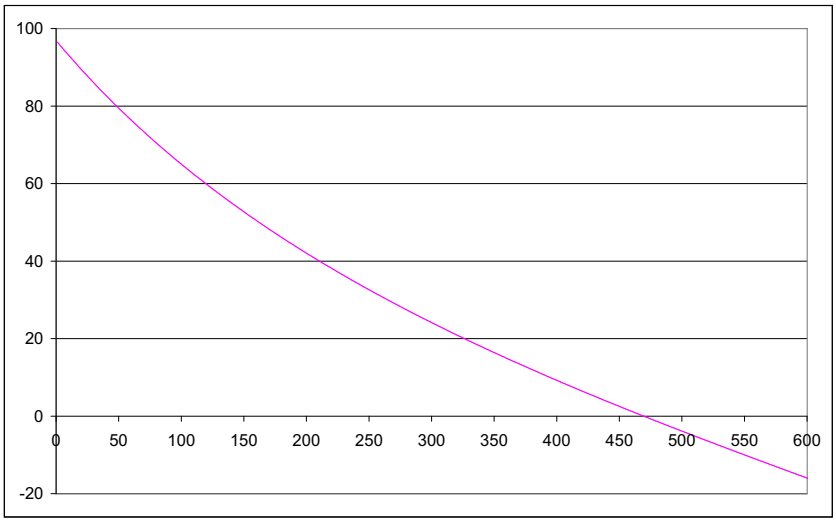

Figure 7. The function $f'(t)$

- If $f'(1) \geq 0$ then $f'(t) > 0$ for any $t \in (0;1)$, i.e., $f(t)$ is increasing function. Hence c_R is a best cut.
- If $f'(0) \leq 0$ then $f'(t) \leq 0$ for any $t \in (0;1)$, i.e., $f(t)$ is decreasing function. Hence c_L is a best cut.
- If $f'(0) < 0 < f'(1)$ then locate the root t_0 of $f'(t)$ using "Binary Search Strategy". Then the best cut in $[c_L, c_R]$ can be estimated by $\frac{1}{N} f(t_0)$

5.2.2. Approximation of entropy measure under full independency assumption

In the independency model, one can try to compute the expected value of the random variable $f(x_1, ..., x_d)$ using assumption that for $i = 1, ..., d$, x_i are random variables with discrete uniform distribution over interval $[0, M_i]$.

First, we will show some properties of the function $h(x)$. Let x be a random variable with discrete uniform distribution over interval $[0; M]$. If M is sufficiently large integer, the expected value of $h(a + x) = (a + x) \cdot \log_2(a + x)$ can be evaluated by:

$$E(h(a+x)) \simeq \frac{1}{M} \int_0^M (a+x) \log(a+x) dx$$

We have

$$\begin{aligned} E(h(a+x)) &= \frac{1}{M}\int_a^{a+M} x\log x dx = \frac{1}{M}\left(\frac{x^2\log x}{2} - \frac{x^2}{4\ln 2}\right)\Bigg|_a^{a+M} \\ &= \frac{1}{M}\left[\frac{(a+M)^2\log(a+M)}{2} - \frac{(a+M)^2}{4\ln 2} - \frac{a^2\log a}{2} + \frac{a^2}{4\ln 2}\right] \\ &= \frac{1}{M}\left[\frac{(a+M)}{2}h(a+M) - \frac{a}{2}h(a) - \frac{(a+M)^2 - a^2}{4\ln 2}\right] \\ &= \frac{(a+M)h(a+M) - ah(a)}{2M} - \frac{2a+M}{4\ln 2} \end{aligned}$$

Now one can evaluate the average value of $E(a,c;U)$ by

$$\frac{1}{N}E(f(x_1,...,x_d)) = E(h(L+x)) - \sum_{j=1}^{d} E(h(L_j+x_j)) + E(h(R+M-x)) - \sum_{j=1}^{d} E(h(R_j+M_j-x_j))$$

6. Conclusions

The problem of optimal binary partition of continuous attribute domain for large data sets stored in *relational data bases* has been investigated. We have shown that one can reduce the number of simple queries from $O(N)$ to $O(\log N)$ to construct the partition very close to the optimal one. We defined some approximated discernibility measures and approximated entropy measures. The theoretical results showed that it is easier to approximate the discernibility than entropy measures.

Acknowledgement

This paper has been partially supported by Polish State Committee of Research (KBN) grant No 8T11C02519, grant of the Wallenberg Foundation and British Council/KBN grant "Uncertainty Management in Information Systems: Foundations and Applications of Non-Invasive Methods (1999-2001)".

References

[1] Catlett, J.: On changing continuous attributes into ordered discrete attributes. In. Y. Kodratoff (ed.), Machine Learning-EWSL-91, Proc. of the European Working Session on Learning, Porto, Portugal, Lecture Notes in Artificial Intelligence, Springer–Verlag, Berlin, 1991, pp. 164–178.

[2] Chmielewski, M. R., Grzymala-Busse, J. W.: Global discretization of attributes as preprocessing for machine learning. In. T.Y. Lin, A.M. Wildberger (eds.). Soft Computing. Rough Sets, Fuzzy Logic Neural Networks, Uncertainty Management, Knowledge Discovery, Simulation Councils, Inc., San Diego, CA, 1995, pp. 294–297.

[3] Dougherty J., Kohavi R., Sahami M.: Supervised and unsupervised discretization of continuous features. In. Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1995, pp. 194–202.

[4] Fayyad, U. M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. Machine Learning **8**, 1992, pp. 87–102.

[5] Fayyad, U. M., Irani, K.B.: The attribute selection problem in decision tree generation. In. Proc. of AAAI-92, San Jose, CA, MIT Press, 1992, pp. 104–110.

[6] Hong, S. J.: Use of contextual information for feature ranking and discretization. IEEE Transactions on Knowledge and Data Eng., **Vol. 9(5)**, 1997, pp. 718–730,

[7] John, G. H., Langley, P.: Static vs. dynamic sampling for data mining. Proceedings of the Second International Conference of Knowledge Discovery and Data Mining, Portland, AAAI Press 1996, pp. 367–370.

[8] Kerber, R.: Chimerge. Discretization of numeric attributes. In. Proc. of the Tenth National Conference on Artificial Intelligence, MIT Press, 1992, pp. 123–128.

[9] Liu, H., Setiono, R: Chi2. Feature selection and discretization of numeric aAttributes. In. Proc. of The Seventh IEEE International Conference on Tools with Artificial Intelligence (TAI95), Washington DC, 1995, pp. 388–391.

[10] Nguyen, H. Son, Skowron, A.: Quantization of real value attributes. In. P.P. Wang (ed.), Second Annual Joint Conference on Information Sciences (JCIS'95), Wrightsville Beach, NC, 1995, pp. 34-37.

[11] Nguyen, H. Son: Discretization Methods in Data Mining. In L. Polkowski, A. Skowron (Eds.): *Rough Sets in Knowledge Discovery* **1**, Springer Physica-Verlag, Heidelberg, 1998, pp. 451–482.

[12] Nguyen H.Son, Skowron A.: Boolean reasoning for feature extraction problems. In: Z.W. Raś and A.Skowron (Eds.): Proceedings of Tenth International Symposium on Foundation of Intelligent Systems, ISMIS'97, NC, USA, *Foundation of Intelligent Systems* **LNAI 1325**, Springer Verlag, 1997, pp. 117-126.

[13] H.S. Nguyen and S.H. Nguyen: From Optimal Hyperplanes to Optimal Decision Trees, *Fundamenta Informaticae* **34** No 1–2, 1998, pp. 145–174.

[14] Nguyen, H. Son: Efficient SQL-Querying Method for Data Mining in Large Data Bases. Proc. of Sixteenth International Joint Conference on Artificial Intelligence, IJCAI-99, Morgan Kaufmann Publishers, Stockholm, Sweden, 1999, pp. 806-811.

[15] Nguyen, H. Son: On Efficient Construction of Decision tree from Large Databases. Proc. of the Second International Conference on Rough Sets and Current Trends in Computing (RSCTC'2000). Springer-Verlag, pp. 316-323.

[16] Pawlak Z.: *Rough sets: Theoretical aspects of reasoning about data*, Kluwer Dordrecht, 1991.

[17] Polkowski, L., Skowron, A. (Eds.): *Rough Sets in Knowledge Discovery* **Vol. 1,2**, Springer Physica-Verlag, Heidelberg, 1998.

[18] Quinlan, J. R. *C4.5. Programs for machine learning*. Morgan Kaufmann, San Mateo CA, 1993.

[19] Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In. R. Słowiński (ed.). Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, Dordrecht, 1992, pp. 311–362

[20] Komorowski, J., Pawlak, Z., Polkowski, L. and Skowron, A.: Rough sets: A tutorial. In: S.K. Pal and A. Skowron (eds.), Rough - fuzzy hybridization: A new trend in decision making, Springer-Verlag, Singapore, 1999, pp. 3-98.

[21] Ziarko, W.: Rough set as a methodology in Data Mining. In Polkowski, L., Skowron, A. (Eds.): *Rough Sets in Knowledge Discovery* **Vol. 1,2**, Springer Physica-Verlag, Heidelberg, 1998, pp. 554–576.