Efficient SQL-Querying Method for Data Mining in Large Data Bases

Nguyen Hung Son
Institute of Mathematics
Warsaw University
Banacha 2, 02095, Warsaw, Poland

Abstract

Data mining can be understood as a process of extraction of knowledge hidden in very large data sets. Often data mining techniques (e.g. discretization or decision tree) are based on searching for an optimal partition of data with respect to some optimization criterion. In this paper, we investigate the problem of optimal binary partition of continuous attribute domain for large data sets stored in *relational data bases* (RDB). The critical for time complexity of algorithms solving this problem is the number of simple SQL queries like *SELECT COUNT FROM ... WHERE attribute BETWEEN ...* (related to some interval of attribute values) necessary to construct such partitions. We assume that the answer time for such queries does not depend on the interval length. 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 partitions of the searching space. We show some properties of considered optimization measures, that allow to reduce the size of searching space. Moreover, we prove that using only O(logiV) simple queries, one can construct the partition very close to optimal.

1 Introduction

The problem of searching for optimal partitions of real value attributes (features), defined by so called cuts, has been studied by many authors (see e.g. {Catlett, 1991; Chmielewski and Grzymala-Busse, 1995; Dougherty *at* of., 1995; Fayyad and Irani, 1992; Liu and Setiono, 1995; Quinlan, 1993; Nguyen and Skowron, 1995]). The main goal is to discover knowledge in the form of *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 [Nguyen and Skowron, 1995] that the searching problem for minimal and consistent set of cuts is NP-hard). Hence numerous heuristics have been developed searching 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 a short overview of some of these measures. In our approach we use so called *discernibUity measures* used extensively in rough set approach [Polkowski and Skowron,1998]. All those methods are very efficient if data sets are stored in executive 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 the problem of searching for optimal partition of real value attributes assuming that the large data table is represented in relational data base. In this 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 like *SELECT COUNT FROM ... WHERE attribute BETWEEN...* (related to some interval of attribute values) necessary to construct such partitions. We assume that 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(iV), where *N* is the number of preassumed partitions of the searching space. We show some properties of considered optimization measures allowing to reduce the size of searching space. Moreover, we prove that using only O(logiV) simple queries, one can construct the partition very close to optimal.

2 Basic Notions

An *information system* [Pawlak, 1991] is a pair 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 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 $a(x) \neq a(y)$. Any information system of the form

$\mathbf{A} = (U, A \cup \{dec\})$ is called *decision table* where $dec \notin A$ is called *decision attribute* (or decision for short). Without loss of generality we assume that $V_{dec} = \{1, \ldots, d\}$. Then the set $DEC_k = \{x \in U : dec(x) = k\}$ will be called the k^{th} *decision class of* $\mathbf{A}$ for $1 \leq k \leq d$.

Any pair (a, c), where a is an attribute and c is a real value, is called a *cut* (if the attribute $a \in A$ has been uniquely specified, then cut can be denoted simply by any $c \in R$ We say that *"the cut* (a, c) *discerns a pair of objects x, y "* if either $a(x) < c \leq a(y)$ or $a(y) < c \leq a(x)$

Definition 1 *The set of cuts* P *is* A-consistent *iff for any pair of objects* $x,y \in U$ *such that* $dec(x) \neq dec(y)$ *if x and y are discernible by attributes from A then there exists a cut* $(a, c) \in \mathbf{P}$ *discerning x and y.*

Definition 2 *The k-consistent set of cuts* P *is* A-optimal *iff card* (P) $\leq$ card(P') *for any A-consistent set of cuts* P'.

The discretization problem can be defined as a problem of searching for consistent set of cuts which is optimal with respect to some criteria. Nguyen and Skowron have shown that the problem of searching for discretization using minimal number of cuts is hard (see [Nguyen and Skowron, 1995]).

Theorem 1 *The problem of searching for the optimal set of cuts* P *in a given decision table* A *is NP-hard.*

Since the problem of searching for optimal set of cuts is NP-hard (i.e. there is not 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 some decision tree induction methods (see [Fayyad and Irani, 1992; Quinlan, 1993]) and some supervised discretization methods (see [Catlett, 1991; Dougherty *at a*/., 1995; Nguyen and Skowron, 1995; Liu and Setiono, 1995; Nguyen, 1998]), we should often solve the following problem:

> FOR A GIVEN REAL VALUE ATTRIBUTE *a* AND SET OF CANDIDATE CUTS $\{c_1, \ldots, c_N\}$, FIND A CUT (a, c_i) BELONGING TO THE SET OF OPTIMAL CUTS WITH HIGH PROBABILITY.

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

1. For a given set of objects *U,* select a cut (a, *cBest)* of high quality among all possible cuts and all attributes;
2. Induce a partition U_1, U_2 **of** U by *(a.cseat)* ;
3. Recursively apply Step 1 to both sets U_1, U_2 of objects until some stopping condition is satisfied.

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

In next sections we recall 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 the attribute a and the set of all relevant cuts $\mathbf{C}_a = \{c_1, \ldots, c_N\}$ on a.

Definition 3 ***The d-tuple of integers*** $(x_1, .., x_d)$ ***is called class distribution of the set of objects*** $X \subset U$ ***iff*** $x_k = card(X \cap DEC_k)$ ***for*** $k \in \{1, \ldots, d\}$. ***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 ***can be called the class distribution 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 (a, c) we use the following notation:

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

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

Statistical test methods

Statistical tests allow to check the probabilistic independence between the object partition defined by 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 very 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 [Kerber, 1992] and Chi2 [Liu and Setiono, 1995])

Entropy methods

A number of methods based on entropy measure formed the strong group of works in the domain of decision tree induction and discretization. This concept uses class-entropy as a criterion to evaluate the list of best cuts

which together with the attribute domain induce the desired intervals. The class information entropy of the set of N objects X with class distribution $\langle N_1, ..., N_d \rangle$, where $N_1 + ... + N_d = N$, is defined by

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

Hence, the class information entropy of the partition induced by a cut point c on attribute a is defined by

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

where $\{U_1, U_2\}$ is a partition of U defined by c.

For a given feature a, the cut Cmin which minimizes the entropy function over all possible cuts is selected. There is a number of methods based on information entropy theory reported in [Catlett, 1991; Fayyad and Irani, 1992; Chmielewski and Grzymala-Busse, 1995; Quinlan, 1993].

Boolean reasoning method

In Boolean reasoning methods, cuts are treated as Boolean variables and the problem of searching for optimal set of cuts can be characterized by a Boolean function f_A (where A is a given decision table). Any set of cuts is Arconsistent if and only if the corresponding evaluation of variables in f_A returns the value *True* (see [Nguyen and Skowron, 1995]). Nguyen and Skowron shown that the quality of cuts can be measured by their *discernibUity properties.* Intuitively, energy of the set of objects $X \subset U$ can be defined by the number of pairs of objects from X to be discerned called *conflict(X).* Let $\langle N_1, ..., N_d \rangle$ be a class distribution of X, then *conflict(X)* can be computed by

$$conflict(X) = \sum_{i<j} N_i N_j$$

The cut c which divides the set of objects U into U_1, and U_2 is evaluated by

$$W(c) = conflict(U) - conflict(U_1) - conflict(U_2)$$

i.e. the more is number of pairs of objects discerned by the cut *(a,c),* the larger is chance that c can be chosen 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 is defined by number of pairs of objects discerned by c i.e.

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

This algorithm is called *the MD-heuristics* [1]

3 Algorithm Acceleration Methods

In this section we present some properties discovered using the Boolean reasoning approach. These properties allow to induce decision trees and make discretization of real value attributes directly from large data base.

[1]Maximal-Discernibility heuristics

3.1 Properties of the Discernibility Measure

First, let us consider two cuts $c_L < c_R$. Let $\langle L_1, ..., L_d \rangle$ be the class distribution in $(-\infty; c_L)$, $\langle M_1, ..., M_d \rangle$ – the class distribution in $[c_L; c_R)$ and $\langle R_1, ..., R_d \rangle$ – the class distribution in $[c_R; \infty)$ (see Figure 1).

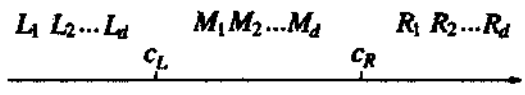

Figure 1: The class distributions defined by cuts c_L, c_R

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 1 *The following equation holds:*

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

Proof: We have

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

$$= \sum_{i=1}^{d} L_i \sum_{i=1}^{d} M_i + \sum_{i=1}^{d} L_i \sum_{i=1}^{d} R_i - \sum_{i=1}^{d} L_i (M_i + R_i)$$

Analogously,

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

$$= \sum_{i=1}^{d} L_i \sum_{i=1}^{d} R_i + \sum_{i=1}^{d} M_i \sum_{i=1}^{d} R_i - \sum_{i=1}^{d} L_i (M_i + R_i)$$

Hence,

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

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

Our goal is to find cuts maximizing the function *W(c).* Let us recall the notion of *boundary cuts* and the well known notion in statistics called median(using the notations presented in Section 2.1):

Definition 4 Any $c_i \in \mathbf{C}_a$ is called the boundary cut if there exist two objects $u_1, u_2 \in U$ such that $a(u_1) \in [c_{i-1}; c_i)$; $a(u_2) \in [c_i; c_{i+1})$ and dec(u\) $\neq dec(u_2)$.

Definition 5 For given set of cuts $\mathbf{C}_a = \{c_1, .., c_N\}$ on a, by median of the k[ih] decision class we mean the cut $c \in \mathbf{C}_a$ which minimizes 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 2 The cut CBest which maximizes the function W(c) can be found among boundary cuts.

Proof: Assume that c_a and Cb are consecutive boundary cuts. Then the interval $[c_a; c_b]$ consists of only one decision class, say C_i. For arbitrary cuts CL and CR 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 2 allows to look for the optimal cuts among boundary cuts only. This property also holds for Entropy measures (see [Fayyad and Irani, 1992]). Although it is interesting but can not be used to construct efficient heuristic for the investigated in the paper problem because of 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

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

This property is formulated in the following theorem.

Theorem 3 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

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

This property is interesting because it states that one can use only 0{dlog N) queries to determine the medians of decision classes by using Binary Search Algorithm. Hence one can reduce the searching space using O(dlogiV) SQL queries. Let us also observe that if all decision classes have similar medians then almost all cuts can be eliminated.

3.2 Fast Algorithm

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

First we divide the set of possible cuts into k intervals (e.g. $k = 2, 3, ..$). Then we choose the interval to which the best cut may belong with the highest probability. We will use some approximating measures to predict the interval which probably contains the best cut with respect to disceraibility 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 [CL;CR] having class distributions: $\langle L_1, ..., L_d \rangle$ in $(-\infty; c_L)$; $\langle M_1, ..., M_d \rangle$ in $[c_L; c_R)$; and $\langle R_1, ..., R_d \rangle$ in $[c_R; \infty)$ (see Figure 1). This measure should estimate the quality of the best cut among those belonging to the interval $[c_L; c_R]$.

Let us consider an arbitrary cut c lying between CL and CR. We have the following theorem (the proof of this theorem is included as an Appendix)

Theorem 4 *The mean $E(W(c))$ of quality $W(c)$ for any cut $c \in [c_L; c_R]$ satisfies*

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

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

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

One can use formulas (3) and (4) to construct the measure estimating quality of the best cut in [CL;CK]

$$Eval([c_L; c_R], \alpha) = E(W(c)) + \alpha \sqrt{D^2(W(c))} \quad (5)$$

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}

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 (5). 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 only 0(d) simple SQL queries of the form: SELECT COUNT FROM ... WHERE attribute BETWEEN .... Hence the number of queries necessary 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 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 of order 0{log N) - simple questions only.

4 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 has generated class distributions as shown in Figure 2.

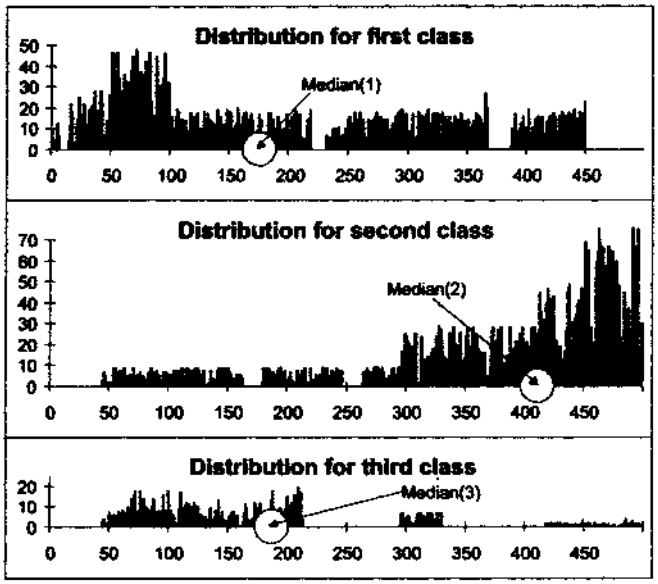

Figure 2: Distributions for decision classes 1, 2, 3.

The medians of classes are Ci66, C414 and cig9, respectively. The median of every decision class has been determined by binary search algorithm using logiV = 9 simple queries. Applying Theorem 3 we conclude that it is enough to consider only cuts from {c166, —,c414}- In this way 251 cuts have been eliminated by using 27 simple queries only.

In Figure 3 we show the graph of W(ci) for $i \in \{166, ..., 414\}$ and we illustrated the outcome of application of our algorithm to the reduce set of cuts for $k = 2$ and $\alpha = 0$.

First the cut C290 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

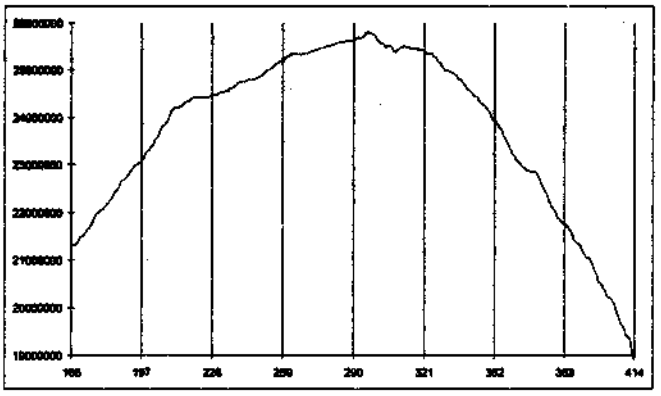

Figure 3: Graph of $W(c_i)$ for $i \in \{166, .., 414\}$.

on these intervals is 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 reduce 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 C296 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 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%).

5 Conclusions

The problem of optimal binary partition of continuous attribute domain for large data sets stored in relational data bases has been investigated. We show that one can reduce the number of simple queries from O(N) to 0(log N) to construct the partition very close to the optimal one. We plan to extern these results for other measures.

References

[Catlett, 1991] 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, 164-178

[Chmielewski and Grzymala-Busse, 1995] 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 294-297

[Dougherty at a/., 1995] Dougherty J., Kohavi R., M.: Supervised and unsupervised discretization of continuous features. In. Proceedings of the Twelfth

International Conference on Machine Learning, Morgan Kaufmann, San FVancisco, CA

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

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

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

[Liu and Setiono, 1995] 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 388-391

[Nguyen and Skowron, 1995] 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, 34-37.

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

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

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

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

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

Appendix A

Proof:(of the Theorem 4)

Let us consider the random cut c lying between CL and CR. The situation is shown in the Figure 4.

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

$$= \sum_{i=1}^{d} \left[(R_i - L_i) \sum_{j \neq i} x_j + (M_i - x_i) \sum_{j \neq i} x_j \right]$$

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

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

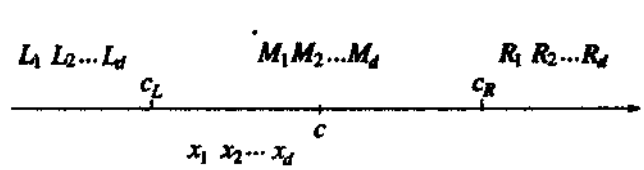

Figure 4: The random cut c and random class distribu tion $x_1, \ldots, x_d$ induced by c

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,

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

Let us assume that $x_1, x_2, \ldots, x_d$ are independent random variables with uniform distribution over sets $\{0, \ldots, M_1\}$, ..., $\{0, \ldots, M_d\}$, respectively. One can observe that

$$\forall_{i \in \{1, \ldots, d\}} E(x_i) = \frac{M_i}{2} \text{ and } D^2(x_i) = \frac{M_i(M_i + 2)}{12}$$

Then we have

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

$$= \frac{W(c_L) + W(c_R)}{2} + \frac{1}{4} \sum_{i \neq j} M_i M_j$$

$$= \frac{W(c_L) + W(c_R) + conflict(c_L; c_R)}{2}$$

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

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

What ends the proof. □