ON THE SELECTION OF REPRESENTATIVE SAMPLES FROM LARGE RELATIONAL TABLES FOR INDUCTIVE INFERENCE

by

Ryszard S. Michalski

Report No. M.D.C. 1.1.9, Department of Engineering, University of Illinois at Chicago Circle, Chicago, Illinois, July 1975.

ON THE SELECTION OF REPRESENTATIVE SAMPLES FROM LARGE RELATIONAL TABLES FOR INDUCTIVE INFERENCE

Ryszard S. Michalski*
Department of Information Engineering
University of Illinois at Chicago Circle
Chicago, Illinois 60680

July 1975

M.D.C 1.1.9

*On leave from Department of Computer Science, University of Illinois, Urbana, Illinois 61801.

ACKNOWLEDGMENTS

The author wishes to express his gratitude to the Department of Information Engineering, University of Illinois at Chicago Circle for providing him with the opportunity to spend a fruitful and, also, pleasant summer there. The author would like to thank specifically Professor Bruce McCormick, Professor S.K. Chang, Professor Steve Vere, Mr. Michael O'Brien and Mr. John Read of the Information Engineering Department for very valuable discussions, comments and exchange of ideas he had with them during his stay. Thanks go also to Mrs. Mary Scott and Mrs. Supriya Chawla for their help in solving various organizational problems and to Miss Bernardine Baran for her excellent typing of this paper.

This work was supported by US PHS MB 00114 from Medical Information Systems Laboratory.

INTRODUCTION

It has been argued [1] that an information system serving as a computer consultant on a specific domain of human knowledge should have not only deductive capabilities but also inductive capabilities. The latter capabilities would enable such a system to infer decision rules from examples and specific facts and to automatically modify the rules in view of new information. To facilitate the implementation of inductive processes the organization of the data base is quite important. The organization employed in a relational data base, as described by Codd [2] and others [3-5], seems to be especially adequate for this purpose. In such a data base facts and examples representing different decision classes are stored in the form of relational tables; i.e., tables whose individual rows are sequences of values of certain multivalued descriptors (variables) and columns correspond to individual descriptors. A relational table represents a relation over a Cartesian product of the domains of the descriptors.

To illustrate an inductive inference problem, let us assume that there is given a large number of such relational tables, and that each table is associated with a certain decision class. Suppose, that for any given sequence of descriptor values ('event'), one has to quickly determine the decision class associated with the relational table which either contains this event or is in some sense 'close' to this event. If the relational tables are large, an efficient way of finding such associations is to use a simple, if possible, and generalized description of each relational table rather than the table itself. To create simple and generalized descriptions of relational tables, one could

use inductive inference programs such as AQVAL/1 programs* [7,8] or others [e.g., 9]. The AQVAL/1 programs can determine an optimal or quasi-optimal (according to a user selected criteria) description of relational tables which is expressed in the form of a DVL_1 formula(s); (i.e., a formula of the variable-valued logic system VL_1 [8]). A DVL_1 formula consists of a sequence of constructs called selectors linked by certain operators. A simple form of a selector would be a test to determine whether the value of a variable is a member of a certain set. Operators might be: 'and', 'or', 'not', 'min', 'max', 'exception', 'separation'. The DVL_1 formulas are very easy to interpret, evaluate, and modify. For this reason the AQVAL/1 programs are potentially useful for a variety of inductive inference problems. However, when the size of relational tables exceeds certain limits (say, a few hundred rows and fifty or so columns) then the computational time required by the programs for inferring table descriptions may become excessively long (say, more than ten minutes† on IBM 360/75). The problems with so large learning data sets can occur, for example, in medical decision making, in the diagnosis of plant diseases, in

*There are four basic AQVAL/1 programs at the present time:

AQ-7 --- which infers from the given event sets an optimized description of one decision class in relation to other classes. The program permits the user to define different optimization functionals, define modes of program operation and some other parameters [5].

AQ-8 --- (Uni-class) which determines an optimized description of each decision class separately, under the constraint that the 'degree of generalization' of the description will not exceed a certain threshold value.

AQ-9 --- which optimizes a given set of DVL_1 formulas according to a certain optimality functional.

SYM-1 -- which determines symmetry (with regard to a set of variables) in variable-valued functions and creates DVL_1 formulas with symmetric selectors.

In determining a description of a given class(s) of objects, the AQVAL/1 programs are used as modules that can be applied step-by-step to transform the original data into the desired DVL_1 formula(s).

†AQVAL/1 programs are currently implemented in PL/1.

learning the value of a position in chess and other games, etc.)

One way to combat this problem efficiently, is to reduce the original relational tables into smaller tables consisting of the 'most representative' examples. The purpose of this paper is to discuss different methods for making such a reduction and to propose a specific reduction algorithm.

2. NOTATION AND BASIC CONCEPTS

Let $\mathbf{E}(d_1, d_2, \ldots, d_n)$ or, briefly, $\mathbf{E}$, denote a set of all n-tuples $(\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n)$, $\dot{x}_i \in D_i$, $i = 1, 2, \ldots, n$ where D_i are certain sets and d_i is the cardinality of D_i. Thus:

$$\mathbf{E}(d_1, \ldots, d_n) = D_1 \times D_2 \times \ldots \times D_n \qquad (1)$$

$\mathbf{E}$ is called the universe of events and its elements are called events. $\dot{x}_i$ and D_i denote a value and the domain of the descriptor x_i, respectively. Descriptors* are certain direct or derived measurements or characteristics of objects or situations. Depending on the nature of a descriptor, its domain may have a different structure; e.g., it can be a linearly ordered set, a partially ordered set, or an unordered set.

In this paper we will distinguish between three categories of descriptors, depending on the structure of their domains:

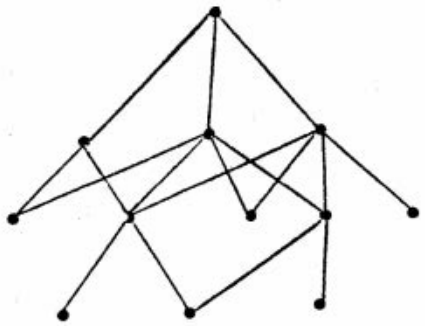
I Interval or ordered descriptors, whose domains are any linearly ordered sets. Thus, this category includes ordinal, interval, ratio and absolute variables as defined in the theory of measurements.

*A descriptor, as described here, is equivalent to a variable. In a more general sense, not considered here, a descriptor can also be an n/ary relation or function.

II Structural descriptors, whose domains are partially ordered sets $< S, \leq >$ that are neither linearly ordered nor totally unordered. In this paper we will restrict ourselves to the case of partially ordered sets having the property that for any two elements $a, b \in S$, there exists at least one element c such that

$$a \leq c \quad \text{and} \quad b \leq c$$

Sets with such structures will be called generalization structures or g-structures. Figure 1 presents a Hasse diagram of a g-structure.



An example of g-structure

Figure 1

In the diagram, a relation $a \geq b$ is represented by placing node a above node b and linking the nodes by an arc.

III Nominal or cartesian descriptors whose domains are sets that have no order.

For example, the height or weight of a person is an interval descriptor, the position of the person in an hierarchy of an institution is a structural descriptor and blood type is a nominal descriptor.

Suppose, without loss of generality, that we are given just two relational tables, E^1 and E^2 (where $E^1, E^2 \subseteq \mathbf{E}$), each associated with a certain decision or

action k (k = 1 and 2, respectively). These sets define a set of functions

$$f: \mathbf{E} \rightarrow D \tag{2}$$

such that

$$\left\{e \;/\; f(e) = k\right\} \supseteq E^k, \qquad k = 1, 2 \tag{3}$$

where

$e \in \mathbf{E}$ and $D = \left\{0, 1, 2\right\}$. '0' in D means 'no decision'.

A problem of inductive inference would be to determine an expression V (of a function f), which is optimal, with respect to some criterion, among all the expressions of all the functions (2). Such an expression will usually also assign values 1 or 2 to events not included in E^k; i.e., the expression will be a certain generalization of the sets E^k. Namely, the initial sets E^k will be transferred into sets $E^k(V) \supseteq E^k$, where

$$E^k(V) = \left\{e \;/\; V(e) = k\right\}, \quad k = 1, 2$$

V(e) - the value of the expression V for the event e.

As mentioned before, AQVAL/1 programs can be used to solve the problem if the expression V is restricted to the class of DVL_1 expressions and the size of the sets E^k does not exceed certain limits. If, however, sets E^k are very large, (e.g., more than 500 rows and more than 50 columns), then the computational time required by the programs may be too long. The problem arises as to whether sets E^k could not be reduced to more manageable sizes and still provide sufficient information about decision classes from the viewpoint of inductive inference. The reduction can be made by reducing the number of columns or number of rows, or both.

The reduction of the number of columns means a reduction of descriptors, which is a known problem of "feature selection" in pattern recognition. Most methods developed for this purpose select those descriptors (from the original set of descriptors) whose 'information content' is large in a global sense; i.e., those descriptors that are important with regard to classification of events into all the decision classes. That means, e.g., that if a descriptor is perfect for distinguishing between two particular classes and not very useful for distinguishing between any other classes, it will have low 'information content' and, consequently, will not be selected. These methods also undesirably separate the problem of 'feature selection' from the development of decision rules. It should be the goal of an inductive processor to find out, in the process of the determination of decision rules, which descriptors are important and in which sense they are important for describing individual classes.

Taking the above into consideration, we will be interested here only in the reduction of rows ('events') in a relational table.

If a precise measure of a "degree of representativeness" of each event $e \in E^k$ were available, then an event reduction process could be performed simply by selecting events whose 'degree of representativeness' is above a certain threshold. E.g., the frequency of occurrence of an object with the description e in the class k could serve as an estimate of such a measure. This estimate, however, in many practical problems is either not available or is not adequate. Consequently, some other means must be developed for selecting the most 'representative' events.

In this paper we will outline three different methods for solving this problem, and then describe a specific algorithm implementing one of those methods.

3. REDUCTION METHODS OR, NM and CC

3.1. An outline of methods

The reduction methods to be discussed here require the introduction of a certain distance function between events (rows in a relational table (RT)). This is not an easy task in view of the different structures which the domains of the variables may have. Section 3.2 proposes two distance measures which can be used in the methods described.

Method OR ('Outstanding Representatives')

In this method, the original event set (represented by a relational table) is reduced to a set consisting of events which are most 'distant' from each other. An important feature of this method is that the resulting set will include events which delineate the 'outside boarder' of the events in the original set. For example, if the 'true' but unknown decision class is a circle and its interior, and the original event set consists of a number of randomly selected points from this class, then the reduced set will be a set of nodes lying on or close to the perimeter of the circle and spanning a polygon of approximately equal sides.

The method is, however, very sensitive to events which differ significantly from the rest of the events in the original set. If such events happened to be

errors, then these errors would have a strong effect on the result. Figure 2 illustrates this method.

Method NM ('Near Miss')

This method selects events ('Near Misses') which lie close to 'critical lines' separating events belonging to different decision classes. For example, if there were only two decision classes, each consisting of two circles with their interiors, then the selected events would be as shown in Figure 3. One difficulty with this method is that the determination of such 'near misses' can be quite costly computationally.

Method CC ('Cluster centroids')

In this method, each original event set is partitioned, using some clustering technique, into a number of clusters. Centroids of the clusters are selected as representatives of the original relational table. In addition to centroids, one could also select some events which are of $i\delta$ distance from each centroid (where δ is the standard deviation for the given centroid and i some integer, e.g., i = 2) and are at maximum possible distances from each other.

3.2. A measure of distance between two events

Let e_1 and e_2 denote two given events:

$$e_1 = (x'_1, x'_2, \ldots, x'_{n1}, x'_{n1+1}, x'_{n1+2}, \ldots, x'_{n2}, x'_{n2+1}, x'_{n2+2}, \ldots, x'_n)$$

$$e_2 = (x''_1, x''_2, \ldots, x''_{n1}, x''_{n1+1}, x''_{n1+2}, \ldots, x''_{n2}, x''_{n2+1}, x''_{n2+2}, \ldots, x''_n)$$

interval descriptors | structural descriptors | cartesian descriptors

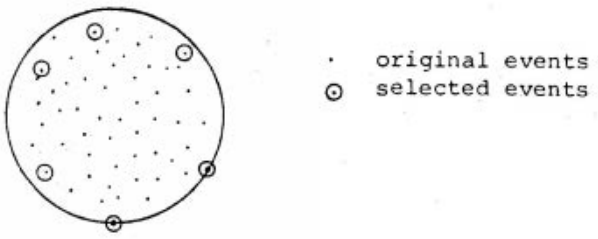


FIGURE 2

An illustration of OR method

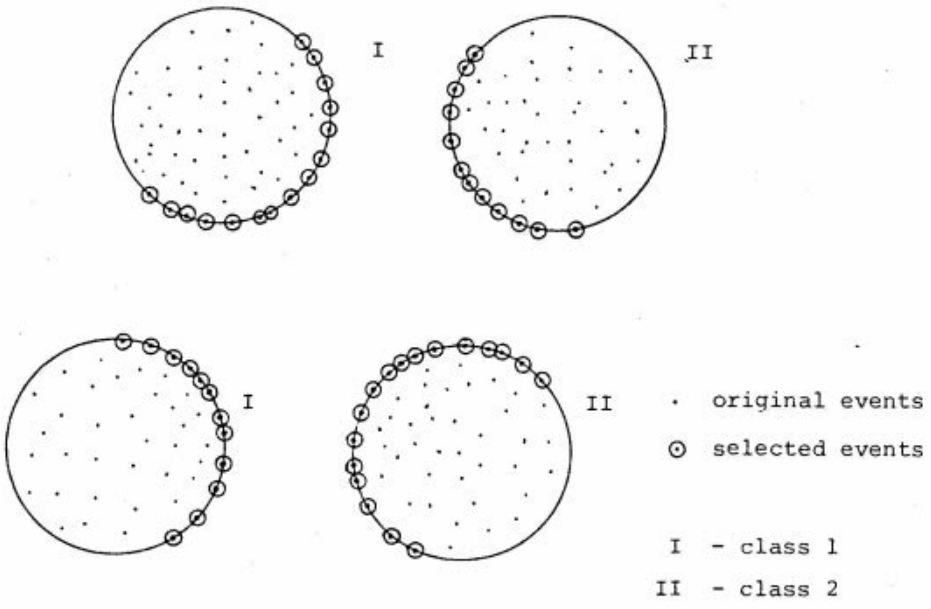


FIGURE 3

An illustration of NM method

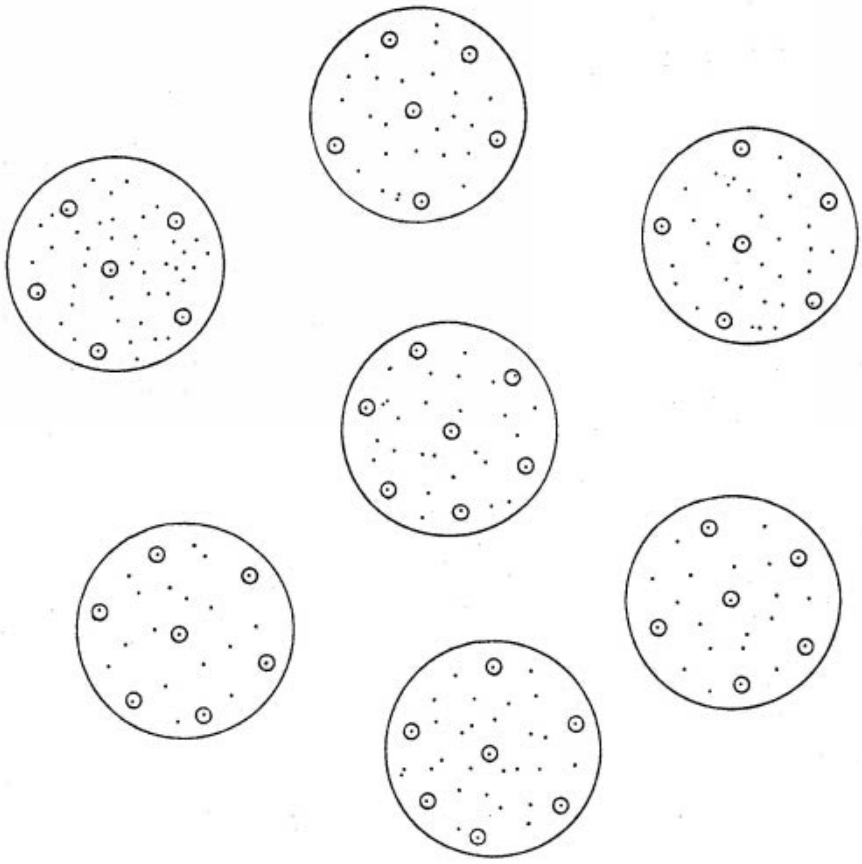


FIGURE 4

An illustration of CC method

where x_i' and x_i'' denote values of descriptors x_i in e_1 and e_2, respectively. Assume, without loss of generality, that the first n1 descriptors in the events above are interval descriptors, the following n2 descriptors are structural descriptors, and those remaining are cartesian descriptors.*

First, we will define a measure of the distance $d(x_i', x_i'')$ between the values of a descriptor depending on the type of the descriptor.

I For interval descriptors:

$$d(x_i', x_i'') = \frac{|x_i' - x_i''|}{\gamma_i} \quad , \ 1 \leq i \leq n1 \tag{4}$$

It is assumed here that the domain of each interval descriptor is represented by the set $\{0, 1, 2, \ldots, \gamma_i\}$.

II For structural descriptors:

$$d(x_i', x_i'') = \frac{NB}{MNB} \tag{5}$$

$n1 < i \leq n2$ (see Figure 5)

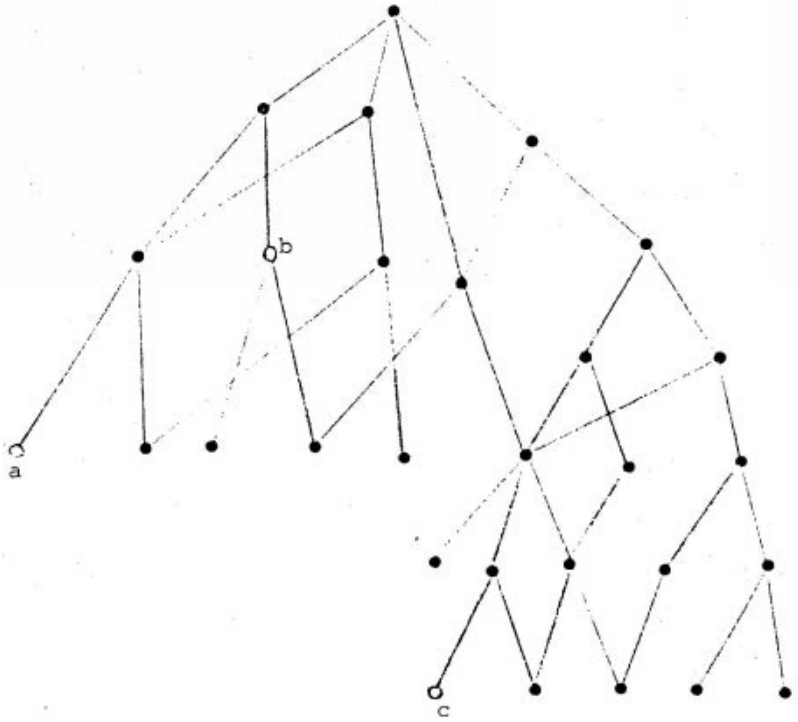
where NB is the length (number of branches) of the shortest path linking x_i' with x_i'' in the Hasse diagram representing the domain of x_i and MNB is the length of the longest of all the shortest paths linking any two nodes of the diagram.

III For cartesian descriptors:

$$d(x_i', x_i'') = \begin{cases} 1, & \text{if } x_i' \text{ is not identical to } x_i'' \\ 0, & \text{otherwise} \end{cases}$$

$(n2 < i \leq n)$

*It is assumed here that if the domain of a structural descriptor is not a g-structure, then the descriptor is treated as a cartesian descriptor.



NB(a,b) = 3
NB(b,c) = 6
NB(a,c) = 7

MNB = 9

d(a,b) = 3/9
d(b,c) = 6/9
d(a,c) = 7/9

FIGURE 5

Illustration of a distance between values of a structural variable.

We will introduce two types of distance measures between events:

I. Quantized measure:

$$d_q(e_1, e_2) = \sum_{i=1}^{n2} q(d(x_i', x_i'')/T_i) + \sum_{i=n2+1}^{n} wd(x_i', x_i'') \tag{6}$$

where $T_i = (t_{i1}, t_{i2}, \ldots, t_{ip})$ is a sequence of thresholds t_{ij} associated with descriptor x_i, $i = 1, 2, \ldots, n2$, $0 \leq t_{ij} \leq 1$

q is a quantization function

$q: d(x_i', x_i'') \rightarrow \{0, 1, 2, \ldots, p\}$

defined as

$$q(d(x_i', x_i'')/T) = \begin{cases} 0, & \text{if } d(x_i', x_i'') \leq t_{i1} \\ 1, & \text{if } t_{i1} < d(x_i', x_i'') \leq t_{i2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ p, & \text{if } d(x_i', x_i'') > t_{ip} \end{cases}$$

w - an integer representing a 'weight' of cartesian descriptors in relation to non-cartesian descriptors.

II. Continuous measure:

$$d_c(e_1, e_2) = \sum_{i=1}^{n} w_i d(x_i', x_i'') \tag{7}$$

where w_i is a weight associated with the descriptor x_i.

The threshold sequence T_i in the quantized measure and the weight w_i in the continuous measure represent two different means to control the effect of a single descriptor on the distance between events.

As we can see, control by the threshold sequence avoids a multiplication operation in computing the distance, as opposed to control by weight, and is thus computationally simpler than the latter. It does, however, require a knowledge of the value of p and the p numbers (thresholds) for each variable, as opposed to the single number (weight), required in control by weight.

3.3. Algorithm OR_1

We will now describe a specific algorithm implementing the OR method. The algorithm is applied in the same way to every set E^k, k = 1, 2, ... Let us then assume that E stands for anyone of these sets. Either of the distance measures introduced in section 3.2 can be used in the algorithm.

1. For each $e \in E$ determine the distance $d(e, e_0)$, where

   $e_0 = (0, 0, 0, \ldots, 0)$.

2. Find events e_{min} and e_{max} such that

   $$d(e_{min}, e_0) = \min_{e \in E} d(e, e_0)$$

   $$d(e_{max}, e_0) = \max_{e \in E} d(e, e_0)$$ (steps 1 and 2 can be done simultaneously)

3. Determine the distance $d(e_{min}, e_{max})$ and divide it into r intervals*, where r is between 0.01 and 0.1 of the size c(E) of the original set E (e.g., if c(E) = 3000 then r is between 30 and 300.)

*The intervals do not have to be equal. The desired situation here is to have intervals which will lead to the subsets E_i (determined in step 4) of approximately the same size.

4. Partition E into r subsets, $E_1, E_2, \ldots, E_r$, such that E_i consists of events whose distance $d(e, e_0)$ lies in the ith interval, i = 1, 2,..., r:

$$a_{i-1} < d(e, e_0) \leq a_i$$

where a_{i-1} and a_i are the endpoints of the ith interval ($a_0 = d(e_{min}, e_0)$ and $a_r = d(e_{max}, e_0)$).

5. From each set E_i, i = 1, 2,..., r, select a subset $E_{is} \subseteq E_i$ consisting of events (where s is such that r•s gives the desired size of the reduced relational table). The selection is made in the following way:

1.) Find e_1 and e_2 in E_i such that

$$d(e_1, e_2) = \max_{e_a, e_b \in E_i} d(e_a, e_b)^*$$

2.) Find e_3 such that

$$d(e_3, e_1) \cdot d(e_3, e_2) = \max_{e \in E_i} \{d(e, e_1) \cdot d(e, e_2)\}$$

.

.

.

s-1.) Find e_s such that

$$\prod_{j=1}^{s-1} d(e_s, e_j) = \max_{e \in E_i} \prod_{j=1}^{s-1} d(e, e_j)$$

where • and $\prod$ denotes the arithmetic multiplication.+

*A more computationally efficient process, though one which might lead to a less desirable result, is to replace step 1 by two steps:

1a) find e_1 such that

$$d(e_1, e_0) = \min_{e \in E_i} d(e, e_0)$$

1b) find e_2 such that

$$d(e_1, e_2) = \max_{e \in E_i} d(e, e_1).$$

+The reason for using multiplication in steps 2,..., s-1, is to select events which are at similar distances from each other.

6. The union of the sets E_{is}:

$$E_s = \bigcup_{i=1}^{r} E_{is} \tag{8}$$

gives the reduced relational set.

The number of operations required by the algorithm is of the order:

$$N = t_0 + r(C_t^2 + \sum_{j=2}^{s-1} j(t-j),$$

where t_0, t is the cardinality of E and sets E_i, respectively. (E_i are assumed to be all of equal size). An 'operation' may involve computing the distance between two events, the comparison of two distances, the comparison of the distance with a threshold, etc. In the modified form of the algorithm we have:

$$N' = t_0 + r\sum_{j=1}^{s-1} j(t-j) \quad .$$

For example, if $t_0 = 3000$, $t = 100$, $r = 30$, $s = 10$, then $N = 273000$ ($N' = 268000$) and the cardinality of the reduced set would be $c(E_s) = 300$.

4. CONCLUSION

Three methods have been proposed for selecting the 'most representative' events (rows) from large relational tables for the purpose of inductive inference:

OR- which selects events that stand at greatest distances from each other,

NM- which selects events close to the 'critical lines' that separate events from different decision classes,

CC- which selects events that are the centers of clusters detected within the original sets, plus events at some fixed distance from the centers.

Also, an algorithm OR_1 has been described which is a computationally efficient implementation of the OR method. The algorithm is oriented toward relational tables of a size ranging from a few hundred to a few thousand events (rows).

REFERENCES

1. Michalski, R.S., Problems of Designing an Inferential Medical Consulting System, First Illinois Conference on Medical Information Systems, Urbana, Urbana, Illinois, October 17-18, 1974.
2. Codd, E.F., "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, XIII, 6, June, 1970, pp. 377-387. (Introduction to the relational model of information.)
3. Date, C.T., An Introduction to Database Systems, Addison-Wesley Publishing Company, 1975.
4. Chang, S.K., Preliminary Report on the Implementation of a Relational Data Base Management System with Structurally Decomposed Relations, Department of Information Engineering, MDC 1.1.3, July 17, 1975.
5. Manacher, Glenn K., "On the Feasibility of Implementing a Large Relational Data Base with Optimal Performance on a Minicomputer," Department of Information Engineering, MDC 1.1.4, presented at the International Conference on Very Large Data Bases, September 22-24, 1975, Framingham, Massachusetts.
6. Michalski, R.S., AQVAL/1--Computer Implementation of a Variable-Valued Logic System VL_1 and Examples of Its Application to Pattern Recognition, Proceedings of the First International Joint Conference on Pattern Recognition, Washington, D.C., October 30-November 1, 1973, pp. 3-17.
7. Larson, J., Michalski, R.S., AQVAL/1: User's Guide and Program Description, Department of Computer Science, University of Illinois, Urbana, May, 1975.
8. Michalski, R.S., VARIABLE-VALUED LOGIC: System VL_1, Proceedings of the 4th International Symposium on Multiple-Valued Logic, Morgantown, West Virginia, May 29-31, 1974.
9. Vere, Steven A., "Induction of Concepts in the Predicate Calculus," Department of Information Engineering, MDC 1.1.2 , January, 1975.