A LOGIC-BASED APPROACH TO CONCEPTUAL DATABASE ANAYLSIS

by

R. S. Michalski
A. B. Baskin
K. A. Spackman

6th Annual Symposium on Computer Applications in Medical Care (SCAMC-6). George Washington University Medical Center, Washington, DC, November 1-2, 1982.

Reprinted from PROCEEDINGS OF THE SIXTH ANNUAL SYMPOSIUM ON
COMPUTER APPLICATIONS IN MEDICAL CARE, October-November 1982

A LOGIC-BASED APPROACH TO CONCEPTUAL DATABASE ANALYSIS*

R. S. Michalski, Ph.D., A. B. Baskin, Ph.D.
and K. A. Spackman, M.D.

Department of Computer Science and School of Clinical Medicine
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

This paper briefly outlines a logic-based approach to the conceptual analysis of a database. The approach used is a combination of relational database management and machine induction. The merger of a relational database model and inference techniques produces a "database analysis system" rather than the more traditional database management system. Relational commands with Variable-valued Logic qualifiers are used for database management functions. Induction techniques are used to group data records according to conceptual (rather than numeric) critera and infer diagnostic rules. The potential application of this methodology to the analysis of clinical databases is illustrated with a series of examples.

Introduction

Databases are constructed for two major reasons:

1) to keep track of information (database management)

and

2) to learn more about the phenomena which produce the data (database analysis).

Simple database storage and retrieval operations extend human memory and make possible the management of large sets of specific facts. Database analysis techniques extend our ability to interpret and generalize trends shown in the data. Typical database analysis techniques include statistical tests, discriminant function analysis, and probabilistic techniques.

Most database analysis techniques operate on numeric data and require that such arithmetic operations as addition and multiplication be defined over individual data items. In addition, many techniques require that a complete set of values be supplied for each data element. The restriction to complete sets of numeric data items is not in keeping with the growing need to store and manipulate incomplete, imprecise, non-numerical data in the life sciences. In clinical medicine, the restrictions can only be met for small subsets of the possible data and where collection is rigidly monitored.

In contrast, natural language processing techniques have been developed to relax both of the restrictions above (1). Although useful for retrieval from unstructured data bases, these techniques are not generally able to take full advantage of known numerical or logical relationships which may exist in the data.

Our logic-based approach relaxes the restrictions by using logical operations on non-numeric data (nominal data) and by using only numeric operations where numeric relationships are known. Combined with the use of incompletely or imprecisely specified data, our approach relaxes the restrictions without requiring the sophistication of a natural language processor.

Database analysis

In the biological sciences, databases are being constructed in order to formulate hypotheses or to validate existing hypotheses. Indeed, clinical databases, whether a retrospective study of past cases or a prospective study designed to answer a specific question, are being used to monitor and modify patient care.

Database analysis techniques are used:

1) to extract a summary of important features of a database (descriptive analysis) or
2) to derive a classification rule for new or existing data (predictive analysis)

Descriptive database analysis is a process in which normative information is extracted from the database and used to identify structure in the data. The grouping of similar patients into syndromes and the identification of the important characteristics of each syndrome is an example of descriptive analysis.

Predictive database analysis is a process in which rules for classifying data elements into known categories are derived from the database. The development of minimal diagnostic rules which predict the disease a patient has based on the data record is an example of predictive analysis.

In general, descriptive analysis techinques (whether done by human expert or machine) are used to produce a classification of records in a database before predictive analysis techniques can be applied. The analysis system described in this paper has induction operators for both descriptive and predictive analysis of data.

*This research is supported by the Office of Naval Research, NOOO 14-82-K-0186, and by the National Library of Medicine, PHS NLM 5 T15 LM07011.

0195-4210/82/0000/0792$00.75 © 1982 IEEE

QUery and INference (QUIN)

The QUIN system being developed at the University of Illinois (2) is a prototype data management and analysis system which uses a relational database management scheme to store/retrieve data. Database analysis is performed by two separate induction programs which implement conceptual clustering of the data and the induction of classification rules from the data.

The QUIN program provides a concise and natural human interface to relational database management functions as well as the induction operations. The command language uses Variable-valued Logic (3) as a relational calculus and is similar to Codd's relational data sublanguage called ALPHA (4).

Using the Variable-valued Logic (VL) language interactively, a database can be built, perused, and analyzed. The analysis is usually an iterative process in which portions of the database are analyzed and the analysis is refined based on the results. For instance, in a database with records for several thousand patients each containing 30-50 attributes, only a subset of the patients or attributes might be used at any one time. In this manner, the most important attributes can be identified before the entire set of patient data is used.

Database Management with QUIN

The QUIN program uses relational tables to store data and operations on tables to store and retrieve data. Apart from some normalization requirements which are not important for this discussion, a relational table in QUIN consists of a rectangular array of names/numbers arranged in rows and columns. The topmost row contains labels for the columns and the entire table is given a name. Columns correspond to some attribute and the value in the column may be a name or a number. Rows in the table correspond to the attribute values for a single data base entry such as a patient record.

As an example of a relational table, consider a database of information collected about twin births. Such a database might contain a table showing the identification number of the mother (ID), the number of pregnancies for that mother (GR), the number of previous full term pregnancies (PTP), the number of previous premature deliveries (PPP), and the number of previous abortions (PAB). The QUIN table would be:

Mother

ID	GR	PTP	PPP	PAB
6	3	1	1	0
16	3	1	0	0
165	2	0	1	0
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

The use of non-numeric data attributes is shown in the following table listing the delivery routes used by twin A and twin B:

Route

ID	DRA	DRB
6	MFD	TBE
16	NSD	TBE
165	NSD	TBE
.	.	.
.	.	.
.	.	.

DRA=delivery route twin A
DRB=delivery route twin B
TBE=total breech extraction
MFD=mid forceps delivery
NSD=normal spontaneous delivery

The system provides commands to define new tables, add rows to a table, change rows, and delete rows. Existing tables may be concatenated together (joined) if they share a common attribute such as ID in the tables above. The resulting table has one new row for each ID value where the original rows are concatenated. The join of the two tables above would be of the form:

Mother * Route

ID	GR	PTP	PPP	PAB	DRA	DRB
6	3	1	1	0	MFD	TBE
16	3	1	0	0	NSD	TBE
165	2	0	1	0	NSD	TBE
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

The complete database about twin births can be assembled by multiple joins of the individual tables using the mother's ID number to determine which rows to match. QUIN allows columns to be selected from a table by typing the table name and then the names of the columns desired:

```
Route(ID,DRA).
```

The route table with just columns ID and DRA contains data about the delivery route of the first twin only.

Logical Qualifiers in QUIN

The three examples below illustrate how the logical expressions of the VL language can be used with data management operations. Complex retrievals are easily specified using the command GET with logical restrictions on the retrieval request and can be used for quality control.

In the twin database described above, all data has been joined together in a single table called "twin". The following QUIN command can be used to find any records in which the number of pregnancies is not equal to the number of previous full term pregnancies plus the number of previous premature deliveries plus the number of abortions plus one (for the present pregnancy):

```
get twin(ID,GR,PPP,PTP,PAB) : [GR not = PTP + PPP
                                          + PAB + 1]
```

which produces the following table:

```
|------------------|
|ID |GR|PTP|PPP|PAB|
|---|--|---|---|---|
| 16| 3|  1|  0|  0|
|------------------|
```

showing that only one inconsistent data record exists and should be checked.

An improbable finding such as an infant born at less than 28 weeks gestation with a weight of over 2000 grams can be detected with the following retrieval request:

```
get twin(ID,GA,WTA,WTB) : [GA<28]&[WTA>2000] or
                          [GA<28]&[WTB>2000]
```

which will detect an improbable birth weight in either twin and produce the following table containing one implausible data item:

```
|----------------|
|ID |GA|WTA |WTB |  GA=gestational age
|---|--|----|----|  WTA=weight twin A
|  6|28|2214|2221|  WTB=weight twin B
|----------------|
```

Finally, the example below checks for an improbable delivery sequence in which the first twin was delivered by c-section and the second twin was not:

```
get twin(ID,DRA,DRB) : [DRA = CS]&[DRB not = CS]
```

which produces an empty table showing that no such data records exist.

Database Analysis with QUIN

The programs invoked by QUIN to perform descriptive analysis and predictive analysis are called CLUSTER (5) and AQ11 (6) respectively. The CLUSTER program takes a table of data elements (rows) and attempts to partition the data elements into a specified number of groups. The program searches for conceptual rather than statistical groupings (clusters) which can be described with logical statements. The AQ11 program searches for predictive rules which distinguish two or more already identified classes of data elements. Rules induced are optimized according to a generalized cost criterion.

The operation of CLUSTER and AQ11 can best be illustrated by a simple example. Although the data below was taken from a real database, only a subset of the possible descriptors were used. The number of patients and the complexity of the example have been kept small for purposes of illustration.

CLUSTER

The CLUSTER command in QUIN was used for descriptive analysis of a database containing the records of patients with craniosynostosis syndromes. CLUSTER was used to partition the database into subgroups in much the same way a physician might divide the patients into different syndromes. After CLUSTER partitioned the data, the discriminate command in QUIN was used to determine rules which can be used to predict membership in each of the "syndromes" found by CLUSTER. Figure 1 shows the eighteen attributes which were used together with a one letter abbreviation. The attributes may take the values present(+) or absent(-). Figure 2 shows the database of patients where each row specifies a value for each attribute for each patient.

Variables

abbreviation	attribute
A	craniostenosis/craniosynostosis
B	facial asymmetry
C	flat forehead/low-set hairline
D	ears malformed/low-set
E	hearing impairment
F	ptosis
G	proptosis/exophthalmos
H	strabismus
I	excessive tearing/tear duct stenosis
J	cleft palate
K	high arched palate
L	midface or maxillary hypoplasia
M	spinal malformations
N	complete syndactyly-fingers
O	impaired CNS function
P	complete syndactyly-toes
Q	cutaneous syndactyly-fingers (webbing)
R	cutaneous syndactyly-toes (webbing)

Figure 1. The 18 variables in the example study together with one letter abbreviations that will be used in the remainder of the example.

DATA

pt#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	+	+	-	-	-	-	+	-	-	+	-	+	-	+	-	+	-	-
2	+	+	-	-	-	-	-	-	-	-	+	+	-	+	-	+	-	-
3	+	-	-	+	+	-	-	-	-	-	+	+	-	+	-	+	-	-
4	+	-	-	-	-	-	-	+	-	-	+	+	-	+	-	+	-	-
5	+	+	-	-	-	-	+	-	-	+	-	+	-	+	-	+	-	-
6	+	-	-	-	-	-	+	-	-	-	+	+	-	+	-	+	-	-
7	+	-	-	-	-	-	+	-	-	-	+	+	-	-	-	-	-	-
8	+	-	-	-	-	+	+	-	-	-	-	+	-	-	-	-	-	-
9	+	-	-	-	-	-	+	-	-	-	-	+	-	-	-	-	-	-
10	+	-	-	-	-	-	+	-	-	-	-	+	+	-	+	-	-	-
11	+	-	-	+	-	-	+	-	-	-	+	+	+	-	+	-	-	-
12	+	-	-	-	-	-	+	-	-	-	+	+	-	-	-	-	-	-
13	+	-	+	-	-	-	-	-	-	-	-	+	-	-	-	-	+	-
14	+	-	+	-	-	-	+	-	+	-	-	+	-	-	-	-	-	+

Figure 2. The 14 data records with variable names and values abbreviated. The data table used by CLUSTER contains no syndrome classification. For AQ11, the classification returned by CLUSTER was used to separate the cases into machine derived syndromes.

Even though we know the syndrome classifications which have been assigned by physicians, this information was not provided to CLUSTER. Indeed, CLUSTER was initially asked to cluster the patient data into two classes even though three syndromes are present in the data (Apert, Crouzon, and Saethre-Chotzen).

The best grouping of patient records found by CLUSTER for two groups is:

Class	Rule	complexity
1	[A=+][B=-][E=-][H=-][J=-][L=+] [N=-][P=-]	1016
2	[A=+][C=-][F=-][I=-][L=+][M=-] [O=-][P=+][Q=-][R=-]	122

which can be paraphrased:

1 present: craniostenosis, maxillary hypoplasia
absent: facial asymmetry, strabismus, cleft palate, complete syndactyly of hands or feet

2 present: cranostenosis, maxillary hypoplasia, complete syndactyly of hands and feet
absent: lowset hairline, ptosis, tear-duct stenosis, spinal malformations, CNS impaired, webbing

From a comparison of the rules above with the simple database, it can be shown that class 1 contains both the Crouzon and Saethre-Chotzen syndromes while class 2 contains only the Apert syndrome. Furthermore, the large complexity measure for class 1 suggests that further clustering of the data might produce a better classification of the data.

The best grouping of patient records found by CLUSTER for three groups can be paraphrased:

1 present: craniostenosis, low-set hairline, maxillary hypoplasia
absent: facial asymmetry, ears malformed/ low-set, hearing impairment, ptosis, strabismus, cleft palate, high arched palate, spinal malformations, impaired CNS function, complete syndactyly-fingers and toes,

2 present: craniosynostosis, maxillary hypoplasia, complete syndactyly-fingers and toes
absent: low-set hairline, ptosis, tear-duct stenosis, spinal malformations, impaired CNS function, webbing

3 present: craniosynostosis, proptosis, maxillary hypoplasia
absent: facial asymmetry, low-set hairline, hearing impairment, strabismus, tear duct stenosis, cleft palate, complete syndactyly fingers and toes, webbing

with complexity measures 14, 122, 26 respectively. Notice that the complexity measures are comparable and much reduced from the case for two groups.

Among the 14 patient records used, the grouping above splits the patients into the same three syndromes as the human experts. The low complexity measures suggest it is time to use AQ11 to derive rules which discriminate between the classes found by CLUSTER.

DISCRIMINATE

The discriminate command in QUIN which invokes AQ11 produced the following discrimination rules when given the three separate classes of records found by CLUSTER:

Apert:	[P = +]
Crouzon:	[G = +]&[N = -]&[R = -]
Saethre-Chotzen:	[C = +]

which can be paraphrased:

Apert: complete syndactyly of toes

Crouzon: proptosis and
no complete syndactyly of fingers
and no webbing of the toes

Saethre-Chotzen: low-set hairline/flat forehead

Because of the small number of patient records used, the discrimination rules given by the program are not general purpose rules which can be used to diagnose all patients with these three syndromes. The rules above do accurately separate all of the patient records in this simple example.

Unlike the CLUSTER command which describes the conceptually important features for defining a class (even when they are common to all classes), the DISCRIMINATE command finds only those attributes which differentiate the members of each class. In this example, the discrimination rules accurately differentiate the three syndromes given that one of the syndromes is present.

Comparison to traditional techniques

It is useful to compare the logical database analysis described here with other kinds of analysis which use statistics, probability and numerical taxonomy. CLUSTER has two advantages over traditional numerical taxonomic techniques: first, it is capable of handling nominal or ordinal values as well as numeric data; second, it not only generates clusters but provides descriptions of the clusters in conceptual or logical form which can be critiqued by the investigator. In addition, the criteria on which the clusters are generated (the "similarity measure") may be based on concepts other than numeric distance, and this has resulted in clusters which tend to match human solutions more closely (6).

AQ11 generates rules for differentiating classes of events based on logical and numeric information. This means that, unlike most probabilistic techniques, the distinction between two classes of events can be made even with incomplete information. The rules are optimized to be simple and to include the factors considered most important by the investigator (via weighting), and they are easily understood and critiqued. Although QUIN supports primarily logical operations on individual data items, it also is capable of more traditional statistical measures (e.g. mean, variance, chi square).

When using QUIN, it is important to realize that it is difficult to fit certain types of data into the relational formalism. Consider a database which contains descriptions of abnormalities found in patients. Let each patient occupy a row in the relational table, and each abnormality be a column. It is readily apparent that these will be large sparse tables because of the limited number of abnormalities that occur in any one patient. Such large sparse tables are computationally expensive as inputs to the analysis procedures and generally must be collapsed using the relational table operations. Thus, the analysis process is an interactive iterative process involving judgements on each cycle.

Conclusion

One objective of QUIN is to provide the medical researcher with a tool which enhances his ability to discover new syndromes (descriptive analysis) and generate rules for differentiating known syndromes (predictive analysis). It should be emphasized that the intellectual involvement of the clinician is crucial to the success of the induction algorithms, because he must make the initial observations, record them, decide which descriptors (symptoms and signs) are present and how to express them, and finally, decide which patients and descriptors to give to the induction algorithms. He must choose a set of descriptors large enough to adequately characterize the patient data but which does not exceed the computational limits of the algorithms. If his data contain large amounts of irrelevant information, the induction operations will not always be able to detect this and may return irrelevant classifications and rules.

The major benefit of automated induction is the potential elimination of time consuming non-intellectual activities associated with clinical investigations in which large amounts of data are collected. Decisions about the relevance and importance of symptoms, signs, laboratory tests, and automated rules and classifications still must be made with care and thought. Automated database analysis should allow a researcher more time for productive thought and less time spent doing tedious and boring tabulations and assessments which the computer does well with so little complaint.

Acknowledgements

The authors wish to acknowledge the assistance of Dr. Lance Rodewald in providing helpful comments and the examples of quality control of the twins data, and the Center for Craniofacial Anomalies, University of Illinois at the Medical Center, for providing the data from which the craniosynostosis example was derived.

References

(1) Shapiro, A. R., "A System for Conceptual Analysis of Medical Practices," Fourth Annual Symposium on Computer Applications in Medical Care, 1980.

(2) Spackman, K. A., "A Relational Database with Learning Capabilities in an Expert System," University of Illinois Department of Computer Science, Master's Thesis (forthcoming).

(3) Michalski, R. S., "Variable-Valued Logic: System VL_1," 1974 International Symposium on Multiple-valued Logic, West Virginia University, Morgantown, West Virginia, May 29-31, 1974.

(4) Codd, E. F., "A Data Base Sublanguage Founded on the Relational Calculus," Proc. 1971 ACM SIGFIDET Workshop on Data Description, Access and Control.

(5) Stepp, R. E., "Learning from Observations: Experiments in Conceptual Clustering," Workshop on Current Developments in Machine Learning, Carnegie-Mellon University, Pittsburgh, July 16-18, 1980.

(6) Michalski, R. S. and Larson, J. B., "Selection of Most Representative Training Examples and Incremental Generation of VL_1 Hypotheses: the underlying methodology and the description of programs ESEL and AQ11," University of Illinois Department of Computer Science Report No. UIUCDCS-R-78-867, May 1978.

(7) Michalski, R. S. and Stepp, R. E., "Concept-based Clustering versus Numerical Taxonomy," University of Illinois Department of Computer Science Report No. UIUCDCS-R-81-1073, October
1981.