

Data Mining Based on HL7 Reference Information Model

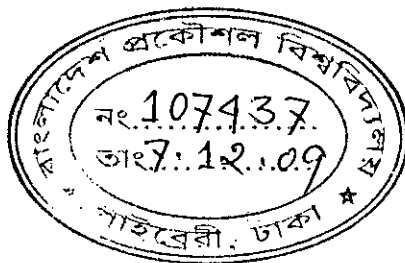
Submitted by

Razan Paul

Student ID: 100605022 P

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of

**MASTER OF SCIENCE IN ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING**



Supervised by

Dr. Abu Syed Md. Latiful Hoque

Associate Professor

Department of Computer Science and Engineering

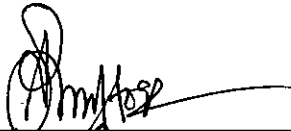
BUET, Dhaka

**Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka**

November, 2009

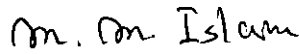
The thesis “Data Mining Based On HL7 Reference Information Model”, submitted by Razan Paul, Roll No. 100605022P, Session: October 2006, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Master of Science in Engineering (Computer Science and Engineering) and approved as to its style and contents for the examination held on November 8, 2009.

Board of Examiners

- 

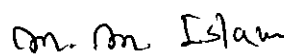
**Chairman
(Supervisor)**

1. Dr. Abu Sayed Md. Latiful Hoque
Associate Professor
Department of CSE
BUET, Dhaka-1000

- 


**Member
(Ex-officio)**

2. Dr. Md. Monirul Islam
Professor and Head
Department of CSE
BUET, Dhaka-1000

- 


Member

3. Dr. Md. Monirul Islam
Professor and Head
Department of CSE
BUET, Dhaka-1000

- 

Member

4. Dr. Mohammad Mahfuzul Islam
Associate Professor
Department of CSE
BUET, Dhaka-1000

- 

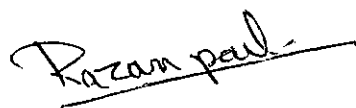
**Member
(External)**

5. Dr. K. M. Azharul Hasan.
Associate Professor
Department of CSE
KUET, Khulna 920300,
Bangladesh

Candidate's Declaration

It is hereby declared that the work presented in this thesis or any part of this thesis has not been submitted elsewhere for the award of any degree or diploma, does not contain any unlawful statements and does not infringe any existing copyright.

Signature

A handwritten signature in cursive script, appearing to read 'Razan Paul', written over a horizontal line.

(Razan Paul)

Table of Contents

Candidate's Declaration	I
Table of Contents	II
List of Figures	V
List of Tables	VII
Acknowledgment.....	VIII
Abstract	IX
 Chapter 1.....	 1
Introduction	1
1.1 Motivation of the Research.....	1
1.2 Aim and Objective of the Thesis	2
1.3 Overview of the Proposed Open Schema Data Models	2
1.4 Overview of the Proposed Data Mining Algorithms for Open Schema Data Models...	3
1.5 Thesis Outlines	4
 CHAPTER 2	 6
Literature Review	6
2.1 Characteristics of Health Care Data	6
2.2 Open Schema Data Model	8
2.2.1 Entity Attribute Value Model.....	9
2.3 Health Level Seven (HL7).....	10
2.4 HL7 Reference Information Model (HL7 RIM).....	10
2.5 Data Warehouse and Healthcare Data.....	12
2.6 Data Mining in Open Schema Data Model.....	13
2.6.1 Approaches of Mining Data in Open Schema Data Models	14

2.6.2 Problems of Converting Open Schema Data Model to Relational Model for Data Mining Purpose.....	15
2.6.3 Association Rule	15
2.6.4 Clustering.....	16
2.7 Comparison with other Works.....	18
Chapter 3.....	20
Search Efficient Physical Data Representations of HL7 RIM.....	20
3.1 Proposed Open Schema Data models.....	20
3.1.1 Optimized Entity Attribute Value (OEAV).....	20
3.1.2 Positional Bitmap Approach (PBA).....	22
3.2 Data Transformation Using Domain Dictionary and Rule Base.....	23
3.3 Physical Representation of HL7 RIM.....	25
3.3.1 Data Type Implementation.....	25
3.3.2 Modeling RIM Classes	26
3.4 Analysis of Different Open Schema Data Models	29
3.4.1 Analysis of Storage Capacity of EAV	29
3.4.2 Space Complexity of Medical domain dictionaries and Rule Base.....	29
3.4.3 Analysis of Storage Capacity of OEAV	30
3.4.4 Analysis of Storage Capacity of PBA.....	30
3.4.5 Selection and Projection	31
3.4.6 CUBE Operation	31
3.4.7 Similarity and Dissimilarity Measures.....	33
3.4.8 Statistical Measures.....	33
3.5 Summary.....	34
Chapter 4.....	35
Data Mining in Open Schema Data Models	35
4.1 Mining Algorithm for Variability Finding.....	35
4.2 Mining Algorithm to Support Medical Research	40
4.3 Constraint k-Means-Mode clustering algorithm.....	45
4.4 Summary	50

Chapter 5.....	51
Results and Discussions.....	51
5.1 Experimental Setup	51
5.2 The Data Sets	51
5.3 Performance Evaluation of Proposed Open Schema Data Models	52
5.3.1 Storage Performance.....	52
5.3.2 Time Comparison of Projection Operations.....	52
5.3.3 Time Comparison of Multiple Predicates Select Queries.....	53
5.3.4 Time Comparison of Aggregate Operations.....	54
5.3.5 Time Comparison of Statistical Operations.....	55
5.3.6 Time Comparison of CUBE Operations.....	55
5.4 Performance Evaluation of Proposed Data Mining Algorithms.....	56
5.4.1 Performance Evaluation of Variability Finding Algorithm.....	56
5.4.2 Performance Evaluation of Medical Research Algorithm.....	57
5.4.3 Performance Evaluation of Constraint K-Means-Mode Clustering Algorithm	58
5.5 Performance Evaluation of Proposed Open Schema Data Models using Oracle DBMS.....	59
5.6 Summary	63
Chapter 6.....	64
Conclusion and Further Research	64
6.1 Summary of the Thesis -----	64
6.2 Fundamental Contributions of the Thesis -----	65
6.3 Future Plan-----	66
References	68
Appendix 1	73

List of Figures

2.1	An ontology representation depicting the class human and the Doctor, Patient, Male, and Female classes as types-of the class human.....	8
2.2	An ontology representation depicting two classes: (1) Human, with the attribute gender, and (2) Role, the relationship has-role indicates the roles a human play: doctor or patient.....	8
2.3	EAV representation with its corresponding relational representation.....	9
2.4	UML class diagram showing the backbone classes of the HL7 Reference Information Model.....	12
3.1	Transformation of EAV model to Optimized EAV (OEAV) model.....	21
3.2	Positional Bitmap Representation.....	22
3.3	Multi level index structure for PBA.....	23
3.4	Data Transformation of medical data.....	24
3.5	Physical representation of Observation class using different data models..	28
3.6	Bottom-up approach of CUBE computation.....	32
4.1	Association mining algorithm for finding variability in Healthcare.....	39
4.2	Association mining algorithm to support medical research.....	43
4.3	Constraint k-Means-Mode clustering algorithm.....	47
5.1	Storage performance.....	52
5.2	Time comparison of projection operations.....	53
5.3	Time comparison of multiple predicates select queries.....	53

5.4 Time comparison of aggregate operations.....	54
5.5 Time comparison of statistical operations.....	55
5.6 Time comparison of CUBE operations.....	56
5.7 Performance of Variability Finding algorithm.....	56
5.8 Performance of Medical Research algorithm	57
5.9 Performance of Constraint K-Means-Mode algorithm.....	58
5.10 Storage performance using Oracle DBMS.....	59
5.11 Time comparison of projection operations using Oracle DBMS.....	60
5.12 Time comparison of multiple predicates select queries using Oracle DBMS.....	60
5.13 Time comparison of aggregate operations using Oracle DBMS.....	61
5.14 Time comparison of statistical operations using Oracle DBMS.....	62

List of Tables

3.1 HL7 data types..... 25

5.1 How many times OEAV and PBA are cheaper and faster compared
to EAV.....62

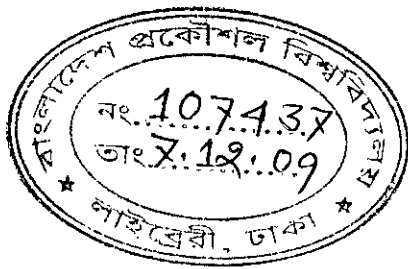
Acknowledgements

First, I would like to express my sincere gratitude to my supervisor, Dr. Abu Sayed Md. Latiful Hoque. Associate professor, Department of Computer Science and Engineering, BUET, for making the thesis as an art of exploiting new ideas and technologies on top of the existing knowledge in the field of the database and the data mining. He provided me moral courage and excellent guideline that made me possible to complete the work. His profound knowledge and expertise in this field and generosity has provided me many opportunities to learn new things for building my carrier. I am glad that he has recovered from a severe illness and returned to work. I wish him a continued good health and long life.

Abstract

The Health Level Seven (HL7) organization has developed a powerful abstract model of patient care called the Reference Information Model (RIM), which is intended to serve as a unified framework for the integration and sharing of information and the usage of data across different healthcare domains. There are a number of exciting research challenges posed by health care data that make them different from data in other industries: data sparseness, high dimensionality, schema change, continuously valued data, complex data modeling features and performance. Entity Attribute Value (EAV) is a widely used solution to handle these above challenges of medical data but EAV is not a search efficient data model for knowledge discovery. The thesis presents two search efficient open schema data models: Optimized Entity Attribute Value (OEAV) and Positional Bitmap Approach (PBA) to handle data sparseness, schema change and high dimensionality of medical data as alternatives of widely used EAV data model. It has been shown in both analytically and experimentally that the proposed open schema data models are dramatically efficient in knowledge discovery operations and occupy less storage space compared to EAV.

We have transformed HL7 RIM healthcare data into EAV, OEAV and PBA data models and applied the proposed data mining algorithms. New data mining algorithms have been proposed to discover knowledge from healthcare data stored in the above models. We have evaluated the performance of the proposed algorithms experimentally by using synthetic datasets. The experimental results show-in all the new developed data mining algorithms, OEAV data model outperforms all the others. Next comes PBA which performs better than EAV and in EAV these algorithms are quite slow.



Chapter 1

Introduction

Due to unique characteristics of medical data like data sparseness, high dimensionality, rapidly changing set of attributes, medical data need open schema data model. Data mining in open schema data model is different from relational model as data semantics in both models are not same. Existing generic data warehouse models are not sufficient to support these unique characteristics of health care data.

1.1 Motivation of the Research

In open schema data model, logical model of data is stored as data rather than as schema, so changes to the logical model can be made without changing the schema. In healthcare, observation data includes lab-test, vital-sign, diagnosis, decision, cost factor and criterion of disease. These data are sparse as doctors perform only a few different clinical lab tests for a patient over his lifetime. This needs schema change to accommodate new laboratory tests. Data is high dimensional (Too many columns) [1]. Entity Attribute Value(EAV) [2] is a widely used solution to handle these above challenges of medical data [3-7] but EAV is not a search efficient data model for knowledge discovery.

Health Level Seven [8] Reference Information Model [9], independent of any implementation technologies, addresses unique challenges of medical data, and broadly covers all aspects of an organization's clinical and administrative information in abstract manner. A RIM-based physical model is suitable as a model of clinical data warehouse. EAV is the widely used solution to handle the challenges of data representation of clinical data warehouse. However, EAV suffers from higher storage requirement and not search efficient.

Health-care organizations have a wealth of RIM based data that can help discovering new treatments and improving patient care. Data mining can drill down into Health care data to discover knowledge in order to improve medical care and to define new guideline. Data mining in open schema data model is different from relational approach as data semantics in both models are not same. In open schema data models, different attribute values are not kept together for a single entity where as relational model does.

Data mining on Clinical information model can determine how much variability occurs in decisions, treatments, costs, care and can find likelihood of disease. Once the variability/ likelihood is discovered, the results of such analyses can be used to define protocols and guidelines to improve the quality of care while simultaneously reducing costs.

1.2 Aim and Objective of the Thesis

The objectives of the thesis are to:

- ❑ develop search and storage efficient open schema data models compared to existing EAV model to convert HL7 RIM abstract information model to physical model,
- ❑ develop data mining algorithm for open schema data model based on HL7 RIM to support medical research, to detect how much variability occurs in decisions, treatments, and cost and to find likelihood of diseases for open schema data models and
- ❑ evaluate the performance of algorithms using the representative dataset

1.3 Overview of the Proposed Open Schema Data Models

In this thesis, we have proposed two search efficient open schema data models: Optimized Entity Attribute Value (OEAV) and Positional Bitmap Approach (PBA) to convert HL7 RIM abstract information model to physical model as alternatives of widely used EAV model. These models are storage efficient compared to existing EAV model. PBA is extra ordinarily storage efficient than OEAV or EAV but slightly less search efficient than OEAV.

OEAV model constructs an attribute dictionary where there is an integer code for each attribute. Attribute name of each fact is mapped to an integer code using the attribute dictionary. All types of values are treated as integer using a data transformation. A compact single integer Attribute Value (AV) is created by concatenating binary representation of attribute code and value. In OEAV, every fact is conceptually stored in a table with two columns: the entity and the AV.

In the PBA model, data is represented in a column wise format. The minimum amount of information that needs to be stored is the position of all non-null elements with their values in a column. Then both the position and the value are converted into a compact single integer PV by concatenating their binary representation. In PBA, every single fact is conceptually stored with a single field: the PV. All types of values are treated as integer using a data transformation.

For the data transformation, medical domain expert have the knowledge of how to map ranges of numerical data for each attribute to a series of items. For example, there are certain conventions to consider a person is young, adult, or elder with respect to age. Data, for which medical domain expert knowledge is not applicable, we have used domain dictionary approach to transform these data to numerical forms.

1.4 Overview of the Proposed Data Mining Algorithms for Open Schema Data Models

Here we have proposed new data mining algorithms to support medical research, to detect how much variability occurs in decisions, treatments, and cost and to find likelihood of diseases for open schema data models.

To detect how much variability occurs in decisions and treatments, we have developed variability finding association rule algorithm. In the mining algorithm for variability finding, we need patterns that are rarely made in Healthcare. Rules discovered by current association mining algorithms are patterns that represent what decisions are routinely made in the Healthcare. We treat all the observation items as being either action, which includes decision, diagnosis and cost factor, or non-action, which includes lab tests, any symptom of patient, and any criterion of disease. In our problem, non-action items appear very frequently in the data, while action items rarely appear with the high frequent non-action items. General intuition of this algorithm is based on the following: if consequent C occurs infrequently with antecedent A and antecedent A occurs frequently, then $A \rightarrow C$ is a rule that is a strong candidate of variability.

Medical Researcher is interested to find relationship among various diseases, lab tests, symptoms, etc. Due to high dimensionality of medical data, conventional association mining algorithm discovers a very high number of rules with many attributes, which are

tedious, redundant to medical researcher and not among his desired set of attributes. Medical researcher may need to find the relationship between rare and high frequent medical items, but conventional mining process for association rules explores interesting relationships between data items that occur frequently together. For these reason, we have proposed mining algorithm to support medical research. The main theme of this algorithm is based on the following two statements: interesting relationships among various medical attributes are concealed in subsets of the attributes, but do not come out on all attributes taken together and all interesting relationships among various medical attributes have not same support and confidence. The algorithm constructs a candidate item sets based on groups constraint and use the corresponding support of each group in candidate selection process to discover all possible desired item sets of that group.

To find likelihood of disease, we have developed constraint k-Means-Mode clustering algorithm. Due to high dimensionality of medical data, if clustering is done based on all the attributes of medical domain, resultant clusters will not be useful because they are medically irrelevant, contain redundant information. Moreover, this property makes likelihood analysis hard and the partitioning process slow. To find the likelihood of a disease clustering has to be done based on anticipated likelihood attributes with core attributes of disease in data point. Attributes of Medical data are both continuous and categorical. The developed algorithm can handle both continuous and discrete data and perform clustering based on anticipated likelihood attributes with core attributes of disease in data point. In this algorithm, the user will set which attributes will be used as data point for a patient and which attributes will participate in clustering process.

1.5 Thesis Outlines

In chapter 2, a review of the research in health care data, open schema data model, and HL7 Reference Information Model have been presented. Association rule mining and clustering have also been discussed.

Chapter 3 describes the overview of EAV, the details organizational structure and analysis of OEAV and PBA. A data transformation is required to adopt the existing data suitable for data warehouse representation for knowledge extraction. The transformation is elaborated. Physical Representation of HL7 RIM has also been elaborated in this chapter. Analytical details of performance of the proposed models are also given.

Chapter 4 describes proposed data mining algorithms in details, which are developed to support medical research, to detect how much variability occurs in decisions, treatments, and cost and to find likelihood of diseases for open schema data models.

Chapter 5 contains details of the experimental work that has been carried out and the discussions on the experiment. The experimental results show the performance of the proposed open schema data models and the proposed data mining algorithms.

Chapter 6 presents the conclusion and findings of the research. It also gives the suggestions for the future research.

Chapter 2

Literature Review

Medical informatics is an important area for the application of computing and database technology. This chapter discusses the topics related to management and modeling of healthcare data and existing data mining approaches to derive knowledge from these data. These include the characteristics of healthcare data, HL7, RIM, EAV, Apriori, and k-means algorithm.

2.1 Characteristics of Healthcare Data

There is a number of exciting research challenges [1] posed by healthcare data. These challenges make them different from data in other industries. Clinical data warehouse introduces several new challenges to data warehouse technology compared to conventional data warehouse for certain properties of healthcare data. These challenges are as follows:

1. Data Sparseness

Only a small subset of the possible attributes associated with a patient are used on any one patient. For example, the Logical Observation Identifiers Names and Codes (LOINC) coding system has over 31,000 codes to represent unique laboratory tests, yet most patients will have only a very small number of different laboratory tests performed on them over their lifetime.

2. Schema Change

New laboratory tests, diseases are being invented every day. We would need frequent altering of the table to accommodate new parts and categories.

3. Too Many Columns (High Dimensional)

The current database systems do not permit a large numbers of columns in a table. This limit is 1000 columns in Oracle 10g and 30,000 columns in Microsoft SQL Server 2008, but there are over 43,000 attribute to represent medical information on a patient.

4. Complex Data Modeling Features

In medical domain, data model must support many-to-many relationships between facts and dimensions. In the multidimensional model [10], facts are in an n-1 relationship to the base elements of the dimensions, which in turn encode strict hierarchies. The relation between patient and diagnosis is most naturally modeled as an n-n relationship, as the same patient may have multiple diagnoses. This is not easily possible using a conventional multidimensional model.

5. Continuously Valued Data

Measurements and lab results are the key facts in the Clinical data warehouse. Unlike typical DW facts, these types of data clearly do not yield any meaning when summed. Other standard aggregation operators, such as MIN, MAX and AVG do apply.

6. Advanced Temporal Support

Important property of clinical data is the importance of temporal aspects. The same test, e.g., the HbA1c% measurement, can be made hundreds of times, so it is important to know both when the data is considered to be valid in the real world, and when it is stored and changed in the database. These temporal aspects of the data, known as valid time and transaction time, must both be supported to provide bi-temporal support.

7. Performance

A query incurs a large performance penalty if the data records are very wide but only a few columns are used in the query [11].

8. Need to Associate Some Metadata

A healthcare data warehouse needs to be able to accommodate how data is captured over time. Procedures for a particular laboratory test can change over time, and different devices can measure the same data differently. An example is the difference between a body's temperature recorded from an oral thermometer in contrast to a rectal thermometer.

9. Knowledge Representation

Let us imagine that in a simplistic clinical domain there are four classes of people: doctors, patients, males, and females. A common way to organize such information would be first to define a human class, with both doctors and patients as types-of human.

Similarly, both males and females are also types-of human (Fig. 2.1). The first problem that arises with the ontology representation in Fig. 2.1 is that doctors can also be patients, and patients can be patients at times, but not all the time. This indicates that doctor and patient are not types of humans, but actually are roles that humans can play. Further, male and female are values that gender as an attribute of humans can take.

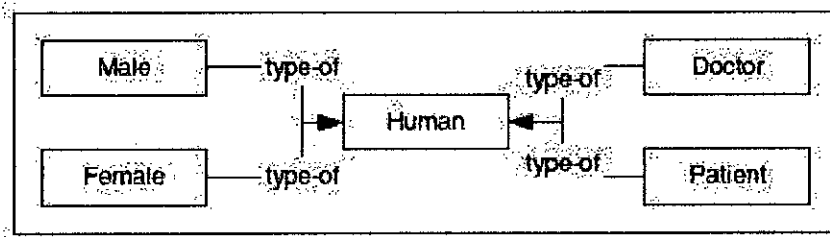


Figure 2.1: An ontology representation depicting the class human and the Doctor, Patient, Male, and Female classes as types-of the class Human

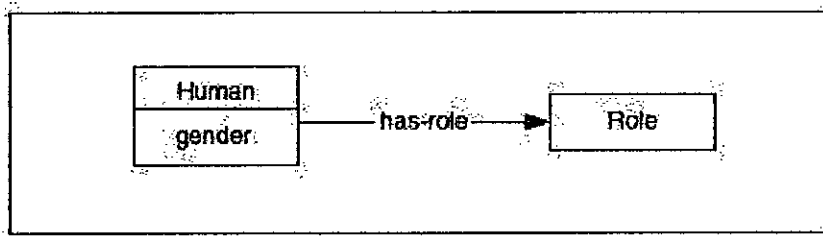


Figure 2.2: An ontology representation depicting two classes: (1) Human, with the attribute gender, and (2) Role, the relationship has-role indicates the roles a human can play: doctor or patient

A more appropriate ontological representation for our example is depicted in Fig. 2.2, with two classes: human and role. Gender is an attribute of the human class. The role class can take one of two possible values: doctor or patient. This is has-role relationship, which connects Human with Role.

2.2 Open Schema Data model

In open schema data model, schema is kept as data to support schema change, sparse data, and high dimensional data. The big advantage of this type model is that much of the logical model is stored as data rather than as schema. So changes to the logical model can be made without changing the schema. The problem is that relational database engines

are just not designed to work with this model, so instead of having the data model stored in the physical table structure, all knowledge about the data ends up being stored in the application.

2.2.1 Entity Attribute Value Model

Entity Attribute Value (EAV) model is suitable for heterogeneous and sparse data. It is an open schema data model where schema is saved as data. EAV model is a conceptual model. It is suitable in circumstances where the number of attributes (properties, parameters) that can be used to describe a thing (an "entity" or "object") is potentially very vast, but the number that will actually apply to a given entity is relatively modest. EAV models are a popular method of storing what are essentially key-value pairs.

ID	A1	A2	A3	A4
1	11	null	51	34
2	null	null	12	32
3	null	null	null	12

Relational representation

Entity	Attribute	Value
1	A1	11
1	A3	51
1	A4	34
2	A3	12
2	A4	32
3	A4	12

EAV representation

Figure 2.3: EAV representation with its corresponding relational representation

In EAV design, every fact is conceptually stored in a table, with three sets of columns: the entity, an attribute, and a value for that attribute. In this design, one row actually stores a single fact. In a traditional table, that has one column per attribute and one row stores a set of facts. EAV gives extreme flexibility, but at the expense of extremely poor performance. It works only when there are a very low load on the system for reporting or data change. Scalability is practically non-existent, and it is a bugger to support. The EAV schema has the advantage of remaining stable when the number of parameters increases that is adding new attributes to an entity does not require changes to the database design. It is efficient at representing sparse data.

It stores data in a way that is not optimized for a relational database. Therefore, user has to code and maintain aspects of the data normally handled automatically by the database. Querying is also not as straightforward, Actually EAV model is difficult to query

and the speed of inserts and updates will be slower as what would normally be one database record now consists of an arbitrary number of records. Most analytical programs require the data in the one column per parameter format. A common exploratory task in epidemiology is identifying associations between clinical parameters. Remember that a single EAV table stores highly heterogeneous facts- apples, oranges, mango, and banana – all in the same column. It would obviously be impossible to do basic and multivariate analysis with the data arranged this way. To be analyzable, this data must be first rearranged into the format that analytical programs expect. A similar argument applies to the association analysis (Data mining). EAV gives us extreme flexibility but it is not search efficient as it keeps attribute name as data in attribute column and has no tracking of how data are stored.

2.3 Health Level Seven (HL7)

One of the aims of standards in software is to allow different applications on different machines to work together, to become interoperable. The International Standards Organization (ISO) specifies a set of levels for Open Systems Interconnection (OSI), and these levels provide a measure of the degree of integration that a particular standard seeks to enable. The highest is level 7: the application level. Level 7 addresses such issues as definition of the data to be exchanged, security checks, participant identification, and data exchange structuring. HL7 is an international community of healthcare subject matter experts and information scientists collaborating to create standards for the exchange, management, and integration of electronic healthcare information. HL7 promotes the use of such standards within and among healthcare organizations to increase the effectiveness and efficiency of healthcare delivery for the benefit of all. At the heart of the HL7 methodology is a model of healthcare information known as the Reference Information Model (RIM) [9]. This is an attempt to describe the people and processes involved in healthcare at a level of abstraction.

2.4 HL7 Reference Information Model (HL7 RIM)

The Health Level Seven(HL7) organization has developed a powerful abstract model of patient care called the Reference Information Model (RIM) [2], which is intended to serve as a unified framework for the integration and sharing of information and the usage

of data across different healthcare domains. Moreover, the RIM can be viewed as ontology insofar as it has developed for a representation of the healthcare domain.

The HL7 RIM is set out using the unified modeling language (UML). The RIM is presented, on the HL7 website, as a class diagram. Classes in UML are abstractions, just as they are in Web Ontology Language (OWL). A class is defined as a name, a set of attributes, and a set of operations. The attributes of a class are the properties that can be used to describe instances of the class. The operations of a class are the functions it performs. The RIM is a static model and does not define operations for classes, which means the information it does represent is similar to that which would be expressed in an OWL ontology. In UML, classes are connected by various kinds of links. The generalization link connects subsets to supersets, the aggregation and composition links identity to other forms of decomposition while association links are used to indicate other relationships between classes. The RIM defines six ‘back-bone’ classes:

- Act – the actions that are executed and must be documented,
- Participation – the context for an act: who performed it, where and for whom,
- Entity – the physical things and beings that are of interest to, and take part in, healthcare,
- Role – the roles entities play in healthcare acts,
- ActRelationship – the relationship between acts and
- RoleLink – the relationships between roles.

A class diagram for the six is shown in Figure 2.4

The RIM’s act-centered view [12] of healthcare is based on the assumption that any profession or business, including healthcare, consists primarily of a series of intentional actions on the part of responsible actors. The varieties of such actions include: actions of clinical observation; actions of assessment of health conditions such as the taking of diagnoses; actions of providing treatment services such as surgery and physical therapy; actions of assisting, monitoring, attending and training; actions of administering education services to patients and their next of kin; actions of providing notary services such as the preparation of an advanced directive or a living will; actions of editing and maintaining documents, and so forth.

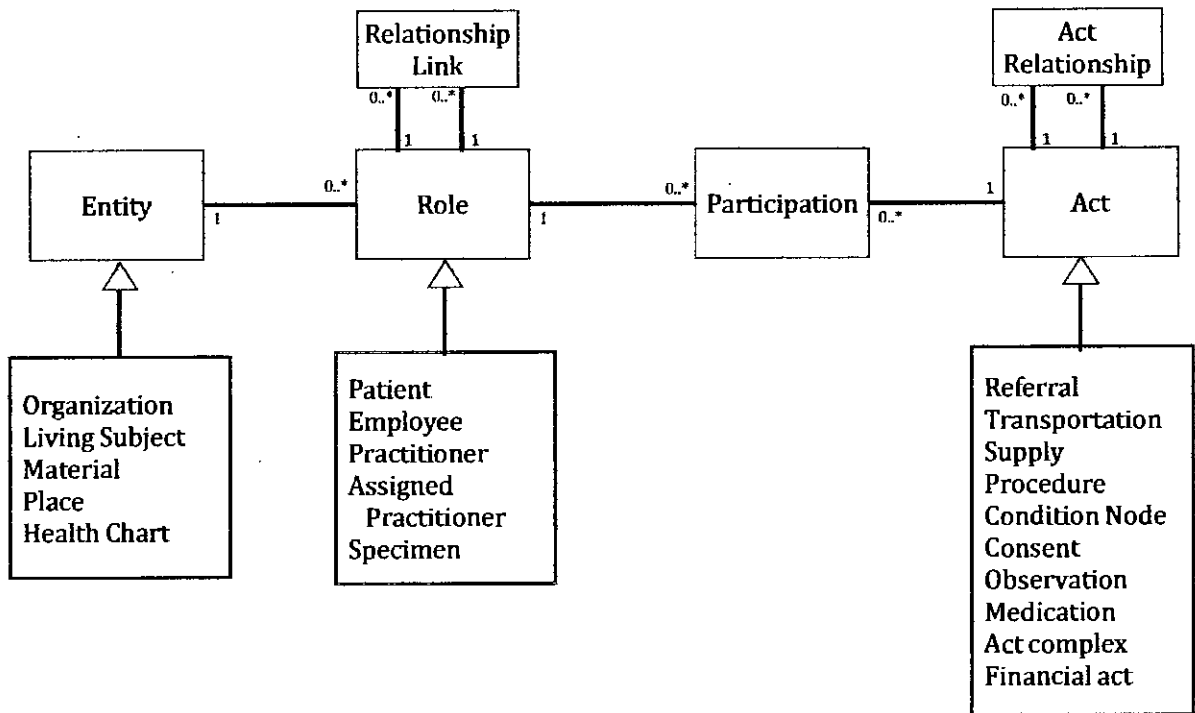


Figure 2.4: UML class diagram showing the backbone classes of the HL7 Reference Information Model

However, of course many features of healthcare go beyond the category of action. These include the participants of the actions themselves, both agent and patient; they include the roles these participants play in actions, their authority to perform a given action, and they include the sorts of entities to which these actions give rise such as obligations and claims. They also include physical objects such as buildings, and healthcare contexts/institutions such as wards and laboratories.

An observation is defined in the HL7 documentation as an act of recognizing and noting information about the subject, whose immediate and primary outcome is new data about a subject. Observations may simply be statements recording a clinician's assessment of findings or of the diagnosis but may equally well be a measurement or test result. The format allows for name value-pairs but they will often have more complex structures, where the Observation includes a report of component observations.

2.5 Data Warehouse and Healthcare Data

According to Bill Inmon, the father of the data warehouse concept [13], the data warehouse is a basis for informational processing. It is defined as being: subject oriented,

integrated, nonvolatile, time variant and a collection of data in support of management's decision. A data warehouse contains integrated granular historical data. Whereas in other industries, generic data warehouse models have proven successful, the characteristics of healthcare data complicate the design of the physical data model.

Healthcare data has many unique characteristics that differentiate it from other industries. These include data sparseness, schema change, a very large number of dimensions, non-additive facts, a constantly changing set of attributes and the need for near real-time data. The data model must support advanced constructs such as many-to-many relationships between facts and dimensions.

To support these features, first of all we have to keep the schema as data so that we can make further changes. The problem is existing generic data warehouse models are not designed to support these medical unique characteristics. But for other industries data, the existing schema is successful. The clinical domain requires more powerful data model constructs than conventional multidimensional approaches. The HL7 RIM can be used as a clinical data warehouse model as HL7 RIM is designed by HL7 to handle unique challenges of medical data.

2.6 Data Mining in Open Schema Data Model

Data Mining [14] is closely related to Knowledge Discovery in Databases (KDD) and quite often, these two processes are considered equivalent. Widely accepted definitions for KDD and DM have been provided by Fayyad, Piatetsky-Shapiro, and Smyth: Knowledge Discovery in Databases is the process of extracting interesting, non-trivial, implicit, previously unknown and potentially useful information or patterns from data in large databases [15]. By the term "pattern" we define a model that is extracted from the data and assigns a number of common characteristics to them, whereas by the term "process" we stress the fact that KDD comprises many steps, such as data preprocessing, patterns query, and result validation. Data Mining is the most important step in the KDD process and involves the application of data analysis and discovery algorithms that, under acceptable computational efficiency limitations. Data Mining can be applied to the vast majority of data organization schemes, most popular of which are the relational databases and data warehouses, as well as various transactional databases. In general, two are the main reasons

for performing DM on a dataset: a) Validation of a hypothesis and, b) Discovery of new patterns.

In open schema data model, schema is kept as data. Here different attribute values are not kept together for a single entity where as relational model does. In open schema data model a single row represent a single attribute value where a row in relational model represents various attribute values for that entity. If a relational table contains m attributes and n row, then its equivalent open schema table contains $m*n$ row to represent the same data. Therefore, there is a difference in performance between scanning a relational table and scanning its equivalent open schema table.

In relational model, a single scan is made in the relational table to determine frequent items sets among the candidates. Here counting is done for each candidate while it is scanning. However, this way is not possible in open schema as each attribute value is kept separately in open schema data model. In relational model, similarity and dissimilarity measure between two entities is straightforward as attribute values for a single entity is kept together. But in open schema model, attribute values for a single entity is not kept together, Hence to measure similarity and dissimilarity, all attributes values have to be retrieve first.

2.6.1 Approaches of Mining Data in Open Schema Data Models

There are two ways we can go to mine data of open schema data models. First one is converting open schema data model to relational model and then performing the data mining using existing data mining algorithm. For that, we need to write a mapping tool, which will map data form open schema data model to relational approach. Here all the open schema data models are row-modeled data as each row of open schema data models represent a single fact. On the other hand, in column modeled data model each column of a row represents a single fact. Another way, mine the data keeping it in open schema data model. As the data semantics of open schema data model are not same as relational model, so new technique need to incorporate for data mining.

2.6.2 Problems of Converting Open Schema Data Model to Relational Model for Data Mining Purpose

First problem is, if the data is high dimensional then we cannot convert this data to relational approach because existing DBMS only support a limited number of columns. Another problem is, when we will convert data to relational approach, a large number of fields will be null which will take extra space and put some challenges in relational based data mining. Another thing is, for data mining we are making another representation of data. So we have to synchronize two representations for insert and update operations. Moreover, new representation is redundant as we have the same data in another representation.

2.6.3 Association Rule

Association rule [14] mining finds interesting associations and/or correlation relationships among large set of data items. Association rules show attributes value conditions that occur frequently together in a given dataset. A typical and widely used example of association rule mining is Market Basket Analysis.

Association rules provide information in the form of "if-then" statements. These rules are computed from the data and, unlike the if-then rules of logic, association rules are probabilistic in nature. In addition to the antecedent (the "if" part) and the consequent (the "then" part), an association rule has two numbers that express the degree of uncertainty about the rule. In association analysis, the antecedent and consequent are sets of items (called itemsets) that are disjoint (do not have any items in common). The first number is called the support for the rule. The support is simply the number of transactions that include all items in the antecedent and consequent parts of the rule. The support is sometimes expressed as a percentage of the total number of records in the database. The other number is known as the confidence of the rule. Confidence is the ratio of the number of transactions that include all items in the consequent as well as the antecedent (namely, the support) to the number of transactions that include all items in the antecedent. For an association rule $s \rightarrow t$ the support and confidence are defined as: $\text{Support} = P(s, t)$, $\text{Confidence} = \frac{P(s, t)}{P(s)}$. The confidence is the conditional probability of t given s , $P(t|s)$. This conditional probability is equal to the unconditional probability $P(t)$ if and only if s and t are independent. The

problem of discovering the association rules can be decomposed into two sub-problems [1, 2]:

1. Find all sets of items (itemsets) that have transaction support above minimum support.
2. Use the large itemsets to generate the desired rules.

Clearly, the sub-problem 2 is quite straightforward once sub-problem 1 is resolved.

2.6.3.1 Apriori

Apriori [16] is a classic algorithm for finding frequent itemsets using candidate generation to mine boolean association rule. The principle of Apriori is as follows. Let L_k denotes the large itemset with k items (also called large k -itemset). The first pass of the algorithm scans the database and counts item occurrences to determine the large 1-itemsets. A subsequent pass, say pass k , consists of two phases. First, the large itemsets L_{k-1} found in the $(k-1)$ th pass are used to generate the candidate itemset C_k . Then the database is scanned and the support of candidates in C_k is counted. This repeats until the generated large itemsets become empty set.

2.6.3.2 Rule Generation

Here conf and minconf denote confidence and minimum confidence respectively. A general way to do this is examining all large itemsets, say $ABCD$ and AB , and determine if the rule $AB \Rightarrow CD$ holds by computing the ratio $\text{conf} = \text{support}(ABCD)/\text{support}(AB)$. If $\text{conf} \geq \text{minconf}$, then the rule holds.

2.6.4 Clustering

Data clustering [14] is considered an interesting approach for finding similarities in data and putting similar data into groups. Clustering partitions a data set into several groups such that the similarity within a group is larger than that among groups [14]. The idea of data grouping or clustering is simple in its nature and is close to the human way of thinking: whenever we are presented with a large amount of data, we usually tend to summarize this huge number of data into a small number of groups or categories in order to further facilitate its analysis.

2.6.4.1 K-Means Algorithm

K-Means clustering algorithm was developed by J. MacQueen (1967) [16]. Simply speaking, k-means clustering is an algorithm to classify or to group objects based on attributes/features into K number of group. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. The sum of squares of distances criterion to partition n data points into K disjoint subsets is represented by the following equation:

$$J = \sum_{j=1}^k \sum_{x_j \in S_j} |x_j - \mu_j|^2$$

Here n data points are (x_1, x_2, \dots, x_n) , where each datapoint is a d-dimensional real vector. K-means clustering aims to partition the n data points into k sets ($k < n$) $S = \{s_1, s_2, \dots, s_k\}$ to minimize the within-cluster sum of squares. The K-Means algorithm is an evolutionary algorithm that gains its name from its method of operation. The algorithm clusters dataset into k groups, where K is provided as an input parameter. It then assigns each observation to clusters based upon the observation's proximity to the mean of the cluster. The cluster's mean is then recomputed and the process begins again. Here's how the algorithm works:

1. The algorithm arbitrarily selects k points as the initial cluster centers ("means").
2. Each point in the dataset is assigned to the closest cluster, based upon the Euclidean distance between each point and each cluster center.
3. Each cluster center is recomputed as the average of the points in that cluster.
4. Steps 2 and 3 are repeated until the clusters converge. Convergence may be defined differently depending upon the implementation, but it normally means that either no observations change clusters when steps 2 and 3 are repeated or that the changes do not make a material difference in the definition of the clusters.

As it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial clusters. As the algorithm is usually very fast, it is common to run it multiple times with different starting conditions.

2.7 Comparison with other Works

The RIM's act-centered view [12] of healthcare is based on the assumption that any profession or business including healthcare, consists primarily of a series of intentional actions. The Observation class of RIM captures most of the clinical related act including vital signs, lab test results, allergies, and diagnoses. Data captured by Observation class are sparse, high dimensional and need frequent schema change. For these characteristics of observation data, in [7] authors use EAV [2] to make physical implementation of HL7 RIM Observation class. Moreover, in non-standard medical database designs, EAV is a widely used solution to handle the challenges of observation data, but EAV is not a search efficient data model for knowledge discovery. Moreover, using different data tables for different data types in observation table, instead of using data transformation to make a unified view of data for several data types, they complicates the further knowledge discovery operations. The EAV model for phenotype data management has been given in [5]. Use of EAV for medical observation data is also found in [3] [4] [6]. None of these works is based on HL7 RIM.

The suitability of the developing HL7 RIM for representing concepts in a data warehouse is described in [18]. In [19], authors have built a centralized index of heterogeneous clinical data according to the act-centered view of healthcare from HL7 RIM. In [20] [21] authors propose several techniques to bridge the impedance mismatch between the object paradigm and the relational paradigm. Generic data warehousing approaches are discussed in [13] [10] [22]. The Entity-Relationship Model is proposed in [23]. Agarwal et al. [24] propose several methods for the efficient computation of multidimensional aggregates. In [25] authors propose data cube as a relational aggregation operator generalizing group-by, crosstab, and subtotals.

Most existing works on finding association rules among all items in a large database are focused to find patterns that occur frequently together [16] [26] [27] [28] [29] [30] [31]. Rare item problem is presented in [14]. According to this problem if minimum support is set too high, the algorithm produces rules eliminating all infrequent item sets. On the other hand, if we set minimum support too low, the algorithm produces far too many rules that are meaningless. In order to deal with this problem many algorithms has been proposed to mine rare associations [32] [33] [34] [35]. However, to find variability in

healthcare data we need rules consist of antecedent which items are high frequent and consequent which items are rare only with these antecedent items:

In [31] the authors propose few algorithms that allow a user to specify Boolean expressions over the presence or absence of items in association rule or to specify a certain hierarchy [30] of items in association rule. These approaches are not enough to mine desired rules for medical researchers. K-means clustering [17] is widely used technique to partition large data sets with numerical attributes. In [36] [37] the authors extends k-means algorithm to partition large data sets with categorical objects. To find likelihood of disease we need a clustering algorithm, which can partition objects consist of both numerical and categorical attributes and can set constraint on presence or absence of items in clustering process and on datapoint. Moreover, no data mining research work is found on open schema data models.

Chapter 3

Search Efficient Physical Data Representations of HL7 RIM

This chapter describes search efficient physical data representations for clinical data warehouse based on HL7 RIM. Entity Attribute Value (EAV) is the widely used solution to handle the challenges of medical data, but EAV is not search efficient for knowledge extraction. We have proposed two search efficient data models: Optimized Entity Attribute Value (OEAV) and Positional Bitmap Approach (PBA) for physical representation of medical data as alternatives of widely used EAV model.

3.1 Proposed Open Schema Data Models

We require an open schema data model for HL7 RIM observation class to support dynamic schema change, sparse data, and high dimensionality of observation data. In open schema data models, logical model of data is stored as data rather than as schema, so changes to the logical model can be made without changing the schema.

3.1.1 Optimized Entity Attribute Value (OEAV)

EAV model has been described in the previous chapter. To remove the search inefficiency problem of EAV whilst preserving its efficiency of representing medical data, we have developed a search efficient open schema data model OEAV. This model keeps data in a search efficient way. This approach is a read-optimized representation whereas the EAV approach is write-optimized. Most of the data warehouse systems write once and read many times, so the proposed approach can serve the practical requirement of healthcare data warehouse.

Figure 3.1 shows the step by step approach of transformation of an EAV data representation to an equivalent OEAV data representation. In step 1, this model constructs an attribute dictionary where there is an integer code for each attribute. Attribute name of each fact is mapped to an integer code using the attribute dictionary. All types of values are

treated as integer using a data transformation as discussed in the following section. In step2, a compact single integer Attribute Value (AV) is created by concatenating binary representation of attribute code and value. In OEAV, every fact is conceptually stored in a table with two columns: the entity and the AV. It maps attribute code and value to p bit and q bit integer and concatenate them to n bit integer AV. For example, an attribute value pair (A3, 51), the code of attribute A3 is 3, will be converted in the following ways: (A3, 51) \rightarrow (3, 51) \rightarrow (0000000000000011, 0000000000110011) \rightarrow 00000000000000110000000000110011 = 196659.

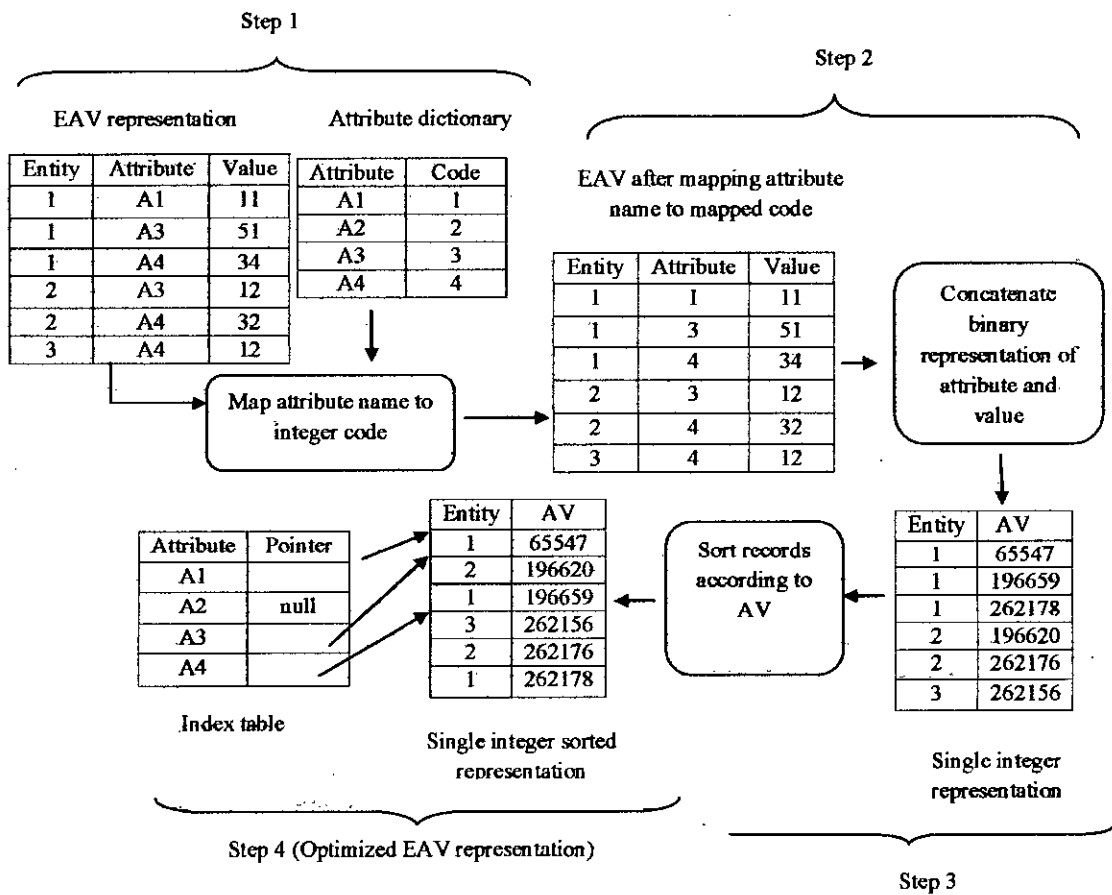


Figure 3.1: Transformation of EAV model to Optimized EAV (OEAV) model

In step 3, the records of OEAV are stored as sorted order of AV field. As data are stored in sorted order of AV and the first p bits of AV are for attribute code, the values of an attribute in OEAV table remains consecutively. In step 4, an index structure, which is a part of OEAV representation, is created to contain the starting record number (Pointer) of each

attribute in OEAV table. This makes the data partitioned attribute wise, which is expected by most analytical program. In sorted AV field, the values of an attribute also remain in sorted order and binary search can be applied on it. This model constructs a modified B+ tree index on entity field of OEAV to make entity wise search efficient. Here each leaf of the tree keeps the block address of attribute values for each entity. These search efficiencies of OEAV are absent in conventional EAV representation.

3.1.2 Positional Bitmap Approach (PBA)

The basic idea of PBA is discussed in [38]. However, our model is significantly different from the basic approach. This model is also a search efficient and read-optimized open schema data model as well as it has efficiency of representing medical data like EAV. Figure 3.2 shows the transformation of sparse relational data representation to an equivalent PBA representation. In this model, data is represented in a column wise format. The minimum amount of information that needs to be stored is the position of all non-null elements with their values in a column. Then both the position and the value are converted into a compact single integer PV by concatenating their binary representation. In PBA, every single fact is conceptually stored with a single field: the PV.

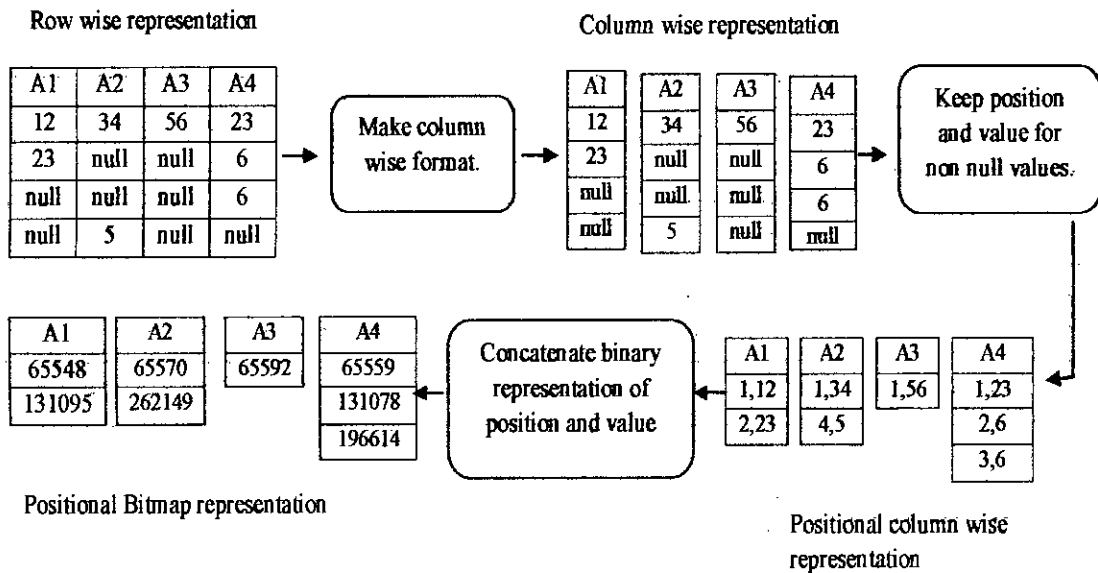


Figure 3.2: Positional Bitmap representation

This model maps position and value of non-null elements to p-bit and q-bit integer respectively and concatenate them to form a single compact n-bit integer PV. For medical

domain, position represents patient ID and value represents various attribute value of patients. This model stores every fact in sorted order of PV and the first p-bit of PV is entity. Hence, facts in this model are stored in sorted order of entity and we can perform binary search on it based on entity.

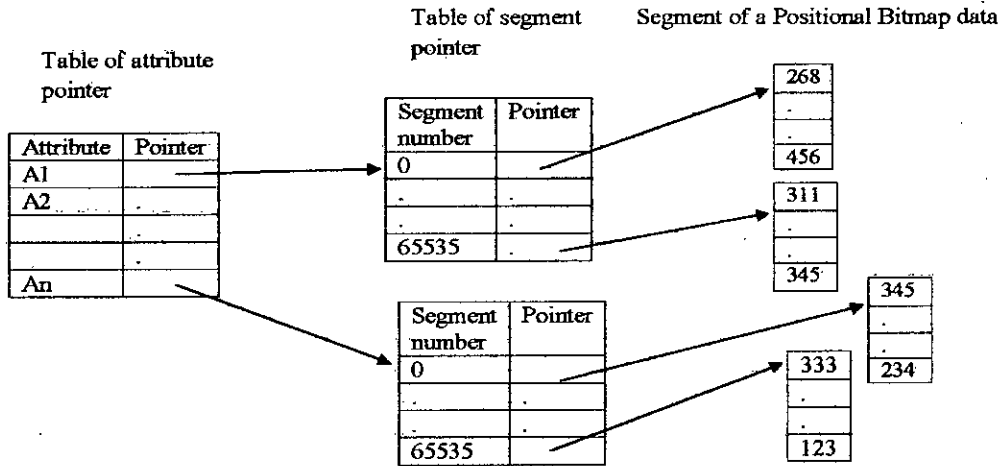


Figure 3.3: Multi level index structure for PBA

We have considered both the position and value is 16 bit. 16 bit is good enough to represent medical information, as cardinality of medical data is not high. However, it is not good enough for position, as the number of patients can be large. To handle this problem, we have developed a multilevel index structure for each attribute as shown in Figure 3.3. Each segment pointer table contains pointer of up to 2^{16} segments of positional data. Each segment can hold up to 2^{16} positional bitmaps. In this way, we can store facts of $2^{32} = 4294967296$ patients. To store value of an attribute of a particular patient ID, this model select segment pointer table of that attribute, and find $\text{segment number} = \text{patient ID} / 65536$. Then it finds the segment and stores the positional bitmap in that segment in sorted order of PV.

3.2 Data Transformation Using Domain Dictionary and Rule Base

For knowledge discovery, the medical data have to be transformed into a suitable transaction format to discover knowledge. We have addressed the problem of mapping complex medical data to items using domain dictionary and rule base as shown in Figure

3.4. The medical data are types of categorical, continuous numerical data, Boolean, interval, percentage, fraction and ratio. Medical domain expert have the knowledge of how to map ranges of numerical data for each attribute to a series of items. For example, there are certain conventions to consider a person is young, adult, or elder with respect to age. A set of rules is created for each continuous numerical attribute using the knowledge of medical domain experts. A rule engine is used to map continuous numerical data to items using these developed rules.

We have used domain dictionary approach to transform the data, for which medical domain expert knowledge is not applicable, to numerical form. As cardinality of attributes except continuous numeric data are not high in medical domain, these attribute values are mapped integer values using medical domain dictionaries. So the mapping process is divided in two phases. Phase 1: a rule base is constructed based on the knowledge of medical domain experts and dictionaries are constructed for attributes where domain expert knowledge is not applicable, Phase 2: attribute values are mapped to integer values using the corresponding rule base and the dictionaries.

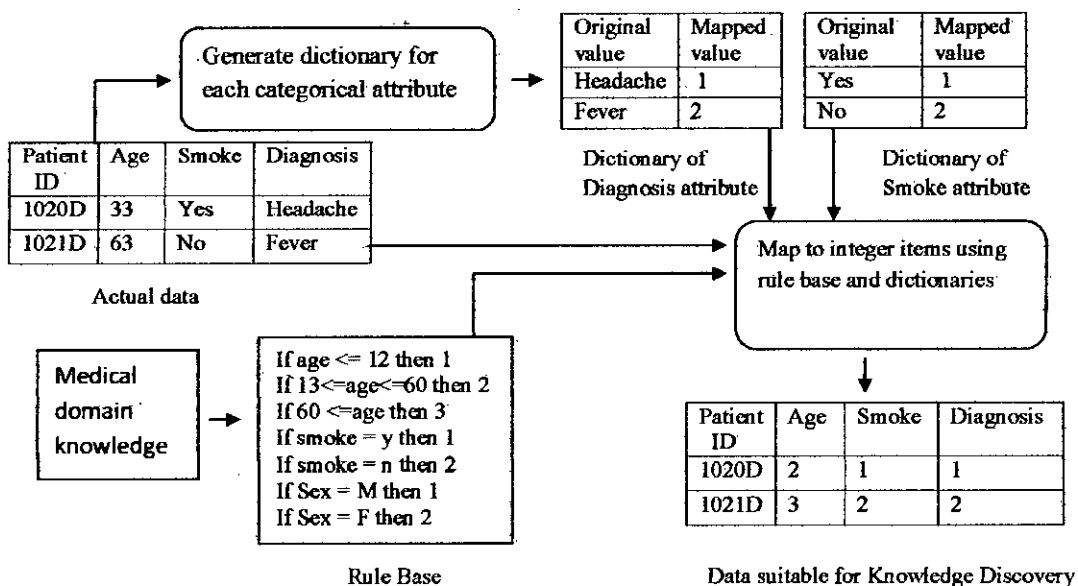


Figure 3.4: Data transformation of medical data

3.3 Physical Representation of HL7 RIM

All latest HL7 RIM specifications are in a form independent of specific representation and implementation technologies. To make physical database model of HL7 RIM from its abstract model, we need to make the physical implementation of following things: data types, classes, aggregations, inheritance structures, and associations.

3.3.1 Data Type Implementation

HL7 has abstract semantic data type specification for its openness towards representation and implementation technologies [39]. HL7 RIM has the following categories of data type, which need to be implemented in physical database context.

We have mapped primitive data types to standard database type. For example, HL7 type ST is converted into standard database type VARCHAR. To implement coded data types [40], we have made a separate table for each coded type and have kept a foreign key named codeID in class table, which refer to the primary key of coded type table. For example, to implement CD data type, a separate table is created with the following column: (codeID BIGINT, code ST, codeSystem UUID, displayName ST). Code system table holds all the information regarding coding system. To implement each compound data type, we have kept multiple columns in the class table, where each column is for each part of compound data type. These columns are used as part of the class table. For example, Encapsulated Data (ED) is implemented with two columns: Media Type CS, DATA CLOB.

Table 3.1: HL7 data types

Category of HL7 Data type	HL7 data types of this category	Description
Primitive	BL, BN, ST, II, TN, INT, REAL, and TS	HL7 specification throws away all of the standard variable types and invents its own (Gives new name)
Coded	CS, CV, CE, CO and CD	They are the Vocabulary Domain [40] types. Vocabulary Domain represents different states in a variety of domain areas by a variety of codes.
Compound	ED, SC, TEL, RTO, PQ, MO, UVP, and PPD	Compound data types allow a collection of data of different types to be grouped into a single object.

Complex	EN, AD, PN, ON, and NPPD	A composite and derivative of other existing data types
Collection	BAG, LIST, SET, IVL, and HIST	Collection data type is a multi values data type. It allows a collection of data of the same types to be grouped into a single object.
Special	PIVL, EIVL, and GTS	These types required special processing logic.

Each Complex type has been implemented according to each one's structure. For example, entity name type specializes LIST<ENXP>. Entity Name is implemented as separate table with following columns: (ID, Valid-Time, and Formatted). For entity name parts, LIST<ENXP> is implemented as a separate table where each row contains name part info and a foreign key refers to the row, for which this name part info belongs to, of Entity Name table. Name Part info includes the following columns: ReferenceID INT, Namepart ST, PartType CS, Qualifier Comma separated string.

To implement collection data type, we have created a separate table per RIM class for each collection data type where each row contains a value from multi values and a foreign key to the row, for which this value belongs to, of parent table. For example, IVL<int> in class ACT for attribute Repeat Number is implemented in the following way: we have created a new table named ACT-IVL-Repeat-Number for class ACT where each row contains an integer value and a foreign key, which refers to the corresponding row of table ACT.

3.3.2 Modeling RIM Classes

As data captured by Observation class is sparse, high dimensional and need schema change, we have modeled Observation class using EAV, OEAV, and PBA data models individually to see what performs better. All the remaining RIM classes have been implemented using relational model. We have implemented each RIM base class using a relation that includes all the attributes of the base class.

3.3.2.1 Implementing HL7 RIM Inheritance

The physical implementation of RIM inheritance can be made in one of the following ways: (i) a single table per entire class hierarchy or (ii) map each concrete class to its own table that contains inherited attributes or (iii) map each class to its own table that

does not contain inherited attributes. We have used the third strategy to model RIM inheritance structure. In this strategy, one table is created per class with one column per class attribute and necessary identification information.

3.3.2.2 Implementing Associations and Aggregation

We have mapped RIM associations using foreign keys. To implement a one-to-many association, we have implemented a foreign key from the “one table” to the “many table”. One-to-one relationship is implemented in the same way. To implement many-to-many association, we have created an associative table, which includes the combination of the primary keys of the tables that it associates. In RIM, only aggregation is Act Relationship. It has been implemented as a separate table that includes primary key consists of two fields: ID of Inbound act and ID of outbound act with the other Act Relationship object attributes in the table.

3.3.2.3 Physical Representation of Observation Class

Observation statements describe what was done, what was actually observed, and assertive statements. The data type of value attribute of Observation class is ANY, which means it can be any valid HL7 data type. To make observation data suitable for knowledge discovery, we have transformed value attribute using domain dictionaries and rule engine to make a unified view of data, integer representation, for several data types. The Observations are name-value-pairs, where the Observation.code (inherited from Act) is the name and the Observation.value is the value of the property.

3.3.2.3.1 Observation Class using EAV Approach

We have implemented observation class using “Map each concrete class to its own table that does not contain inherited attribute” approach with one exception that is Code attributes of Act class is implemented in observation table. To implement in this way, we have kept the following attributes: value, Code, Interpretation Code, method code, TargetSiteCode and a key to keep the reference with the act entity in the observation table using EAV data model. Here, every type of code in HL7 RIM is varchar (64). Here ID, Value is the entity, value of EAV model and Code represents attribute of EAV. ID refers to ID of ACT class of HL7 RIM.

3.3.2.3.2 Observation Class using OEAV Approach

Here observation class has been implemented in the same way as EAV approach. The difference is that Value and Code attributes have been implemented as AV. Therefore, we have kept the attributes: AV, InterpretationCode, MethodCode, TargetSiteCode, and a key to keep the reference with the act entity in the observation table using OEAV data model. Here AV field is the combination of Observation Code and the Observation Value. A modified B+ tree has been created based on patient ID where leaf node of the tree contains the block address of each attribute value of the particular patient. In OEAV, InterpretationCode, MethodCode, and TargetSiteCode are converted into 16-bit integer using domain dictionaries. Each Code represent an attribute and is transformed into 16-bit integer using attribute dictionary. Value is transformed into 16-bit integer using domain dictionaries and Rule base. The compact single integer Attribute Value (AV) is created by concatenating binary representation of 16-bit attribute code and 16-bit value. Here AV is 32-bit integer and ID refers to ID of ACT class of HL7 RIM.

Obsevation class using EAV	Obsevation class using OEAV	Obsevation class using PBA For each code
<ul style="list-style-type: none"> • ID BIGINT • Code Varchar(64) • Value Varchar(64) • InterpretationCode Varchar(64) • MethodCode VarChar(64) • TargetSiteCode Varchar(64) 	<ul style="list-style-type: none"> • ID BIGINT • AV 32bit INT • InterpretationCode 32bit INT • MethodCode 32bit INT • TargetSiteCode 32bit INT 	<ul style="list-style-type: none"> • PV 32bit INT • InterpretationCode 32bit INT • MethodCode 32bit INT • TargetSiteCode 32bit INT

Figure 3.5: Physical representation of Observation class using different data models

3.3.2.3.3 Observation Class using Positional Bitmap Approach

Here observation class is implemented in the same way as the previous two approaches. For each Act.Code, a positional bitmap table has been made. Each positional bitmap table contains attributes PV, Interpretation Code, method code, Target Site Code and

a key to keep the reference with the act entity. In PBA, each code is mapped to a column and each one is saved only once in metadata of the corresponding column and does not store attribute name as data. Value is transformed into 16 bit integer using domain dictionaries and Rule base. ID, which refers to the ID of ACT class of HL7 RIM, is mapped to 16-bit position using the multilevel index structure of PBA. Both the 16-bit position and the 16-bit value are converted into a compact single integer PV (32 bit) by concatenating their binary representation.

3.4 Analysis of Different Open Schema Data Models

Let b be the total number of blocks of observation table and k is the total number of attributes of observation table. Analysis of storage capacity and basic operations of knowledge discovery in open schema data models is discussed in the following:

3.4.1 Analysis of Storage Capacity of EAV

Let n = total number of facts, q = Average length of attribute names, g = Average length of values. In EAV, 32 bits (4 bytes) is required to represent entity. Size of each fact in EAV is $(4 + q + g)$ bytes. Hence, the total size to hold all facts is $S = n \times (4 + q + g)$ bytes.

3.4.2 Space Complexity of Medical Domain Dictionaries and Rule Base

Let C_i = cardinality of i^{th} categorical medical attribute, L_i = average length of i^{th} attribute value, u = number of categorical attributes. Integer codes of categorical attribute values, shown in data transformation, are not stored explicitly and the index of attribute is the code. Domain Dictionary Storage of i^{th} attribute is $C_i \times L_i$ bytes. Total domain dictionaries storage (S_D) is $\sum_{i=1}^u (C_i \times L_i)$ bytes. If the size of rule base storage is R , the dictionary and rule base storage (S_{DR}) is $\sum_{i=1}^u (C_i \times L_i) + R$ bytes.

3.4.3 Analysis of Storage Capacity of OEAV

Let p = number of medical attributes, q = average length of attribute names. Total storage of medical attribute dictionary is $p \times q$ bytes. Let S = size of each block address in byte. Total storage of Index table is $p \times q + p \times S$ bytes. In OEAV, 32 bits is required to represent entity and 16 bits are required for attribute and value individually. 64 bits = 8 bytes = Size of each fact in OEAV. Let n = total number of facts, m = total number of facts in a block, w = word size (bytes). Total number of blocks is $\lceil n/m \rceil$. The number of words per fact is $\lceil 64/w \rceil$. For block i where $1 \leq i \leq \lceil n/m \rceil$, the number of words per block is $\lceil (m \times \lceil 64/w \rceil) \rceil$ and the size of the block is $w \lceil (m \times \lceil 64/w \rceil) \rceil$. Hence the size to hold all facts, $S = \lceil n/m \rceil \times w \times \lceil (m \times \lceil 64/w \rceil) \rceil$. In OEAV, total size to hold all facts = storage for facts + storage for domain dictionaries and rule base + storage for attribute dictionary + storage for index table + storage for modified B+ tree

$$= \lceil n/m \rceil \times w \times \lceil (m \times \lceil 64/w \rceil) \rceil + \sum_{i=1}^u (C_i \times L_i) + R + (p \times q) + (p \times q + p \times S) + B$$

bytes.

3.4.4 Analysis of Storage Capacity of PBA

In PBA, 16 bits is required to represent entity and 16 bits is required to represent value individually. Size of each fact in PBA is 32 bits = 4 bytes. Let, n = total number of facts, m = total number of facts in a block, w = word size (bytes). Total number of blocks is $\lceil n/m \rceil$. If word size is less than or equal to size of each fact, the number of words per fact is $\lceil 32/w \rceil$. The number of words per block is $\lceil (m \times \lceil 32/w \rceil) \rceil$, the size of each block is $w \lceil (m \times \lceil 32/w \rceil) \rceil$ and hence the total size to hold all facts is $S = \lceil n/m \rceil \times w \times \lceil (m \times \lceil 32/w \rceil) \rceil$. If word size is greater than size of each fact, the number of facts per word is $\lceil w/32 \rceil$, the number of words per block is $\lceil (m / \lceil w/32 \rceil) \rceil$, the size of each block is $w \lceil (m / \lceil w/32 \rceil) \rceil$ and hence the size to hold all facts is $\lceil n/m \rceil \times w \times \lceil (m / \lceil w/32 \rceil) \rceil$. In PBA, total size to hold all facts = storage for facts + storage for domain dictionaries and rule base

$$= \lceil n/m \rceil \times w \times (\lceil (m \times \lceil 32/w \rceil) \rceil) + \sum_{i=1}^u (C_i \times L_i) + R \text{ (If } w \leq \text{size of fact)} \text{ or}$$

$$= \lceil n/m \rceil \times w \times (\lceil (m / \lceil w/32 \rceil) \rceil) + \sum_{i=1}^u (C_i \times L_i) + R \text{ (If } w > \text{size of fact)}.$$

3.4.5 Selection and Projection

EAV approach requires a full scan of entire table of all attributes i.e. b blocks for a single selection operation. However, OEAV does not require reading all attributes because data are clustered attribute wise. Moreover, data are kept in sorted order of attribute value, so binary search is applicable on attribute value. Index table contains starting and ending block address of each attribute partition. In this approach, $\log_2(\frac{b}{k})$ blocks need to be examined for a single selection. In PBA, data for each attribute are kept separate and data are kept in sorted order of entity rather than in sorted order of attribute. Data is clustered attribute wise but a linear search is required, so the number of blocks that need to be examined is b/k for a single selection operation.

Let m be the number of projected attributes. In EAV, all the records need to be scanned to get a single attribute value, so number of blocks need to be examined is b whatever is the number of projected attributes. OEAV uses a modified B+ tree for indexing entity. Number of blocks need to be examined is $m \times (\frac{b}{k})$ when m attributes are projected for all entities. For a single entity, B+ tree takes $O(\log_q n)$ time to find the address of m blocks which hold the values of m attributes of that entity where n is the number of entity and q is the capacity of nodes. In PBA, to project m attributes for a single entity, no of blocks need to be examined is $m \log_{b_2} b$. No of blocks need to be examined is $m \times (\frac{b}{k})$ when all the entities record need to be retrieved.

3.4.6 CUBE Operation

Group by is basis of all OLAP operation. Multiple Group-by operations can form a CUBE operation and each group-by is called a cuboid. Each cube operation requires aggregation of measures at some level. For aggregate operation, we consider only max, min, average, and count. In EAV, for each aggregate operation, it has to scan all the blocks because it keeps attribute name as data in attribute column. In OEAV, a count operation can

be computed on an attribute from index table without any block access. For max and min operation on a attribute it has to scan only 1 block because it keeps each attribute data separate and in sorted order. It has to scan b/k blocks to compute average. In PBA, for each aggregate operation, it has to scan b/k blocks because it keeps each attribute data separate but not in sorted order.

As medical data is sparse, widely used top down approach of CUBE computation [25], is not a good choice. Because top down approach does not consider null data in cube computation, so computing each cuboid from a number of other parent cuboids does not yield the correct result all the time. Considering null data in cube computation can solve this problem but it will add huge complexity in cube computation with open schema data models. It is because that finding out for which entities a particular attribute value is null is time consuming and hard in these open schema data models.

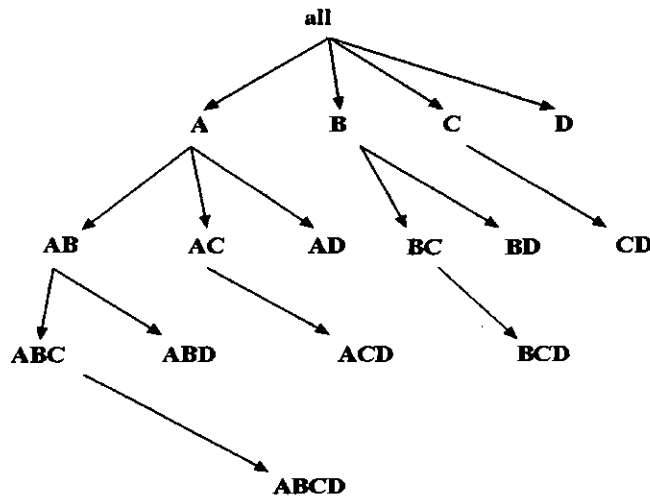


Figure 3.6: Bottom-up approach of CUBE computation

Here we build the CUBE using bottom-up algorithm as in figure 3.6. It starts building CUBE by computing cuboids on a single attribute, then cuboids on a pair of attributes, then cuboids on three attributes, and so on. Candidate of cuboids are generated from child cuboids. This is the opposite of widely used top down approach of CUBE computation. Here pruning is bases on data representation of open schema data models as open schema data models store sparse data, so the algorithm does not need to perform cube computation on sparse data. The algorithm reduces disk read costs by computing all same-length cuboids in one scan of DB. The algorithm uses hash-based method [41] to compute a cuboid. In EAV, facts are not clustered attribute wise and have no entity index, so every

search becomes full scan of all blocks. In OEAV, facts are clustered attribute wise and have entity index, it requires scanning only necessary attribute partitions and values. In PBA, facts are clustered attribute wise and are sorted order of entity. It requires scanning only necessary attribute partitions but some unused values.

3.4.7 Similarity and Dissimilarity Measures

Dissimilarity can be computed between two patients using the following two measures: hamming distance for categorical data objects and Euclidean distance for continuous data. To measure the dissimilarity between two entities all the attributes values have to be retrieved for each entity. To retrieve all the attributes values for each entity we have to use linear search over all the blocks in EAV. In OEAV, by finding corresponding leaf of modified B+ tree for each entity, we get the block addresses of attribute values of that entity and attribute values are retrieved from the block addresses for that entity. To retrieve the block addresses of attribute values for each entity it takes $O(\log_q n)$ time where n is the number of entity and q is the capacity of nodes. In PBA, because of storing data in sorted order of entity wise, it takes $O(k \log_2 v)$ to retrieve attribute values for a single entity where v is the average number of records in each column and total number of blocks need to be read is $\log_2 b$.

3.4.8 Statistical Measures

To compute mean, mode, median, kth-percentile on an attribute, the number of block EAV scans is b as it keeps attribute name as data in attribute column. To compute standard deviation, EAV scans b blocks once to compute mean of the attribute and scans b blocks again to perform standard deviation calculation. To compute mean, mode, median, kth-percentile, Standard deviation on an attribute, OEAV scans b/k , b/k , 1, 1, $2b/k$ blocks respectively because of keeping data separately and in sorted order based on attribute. To compute mean, mode, median, kth-percentile, Standard deviation on an attribute, PBA scans b/k , b/k , b/k , b/k , $2b/k$ blocks respectively as it keeps data separately based on attribute but do not keep data in sorted order based on attribute.

3.5 Summary

Integrating Large-scale medical data is important for knowledge discovery. Once this knowledge is discovered, the results of such analysis can be used to define new guidelines for improving medical care and treatment. We have proposed the conversion of HL7 RIM abstract information model to physical model, which includes HL7 RIM abstract data type conversion, modeling RIM classes, inheritance, association and aggregation. The solution to address the problem of mapping complex medical data to items has also been proposed here.

Observation class represents most of the clinical related act like diagnoses, laboratory results, allergies, vital signs etc. So most of the knowledge discovery operations will be performed on observation data. Observation class is sparse, high dimensional and need frequent schema change and EAV is a widely used solution to handle these challenges. However, EAV is not a search efficient data model for knowledge discovery. We have proposed two search efficient open schema data models OEAV and PBA to handle these challenges as alternatives of EAV for medical data warehouse.

3.5 Summary

Integrating Large-scale medical data is important for knowledge discovery. Once this knowledge is discovered, the results of such analysis can be used to define new guidelines for improving medical care and treatment. We have proposed the conversion of HL7 RIM abstract information model to physical model, which includes HL7 RIM abstract data type conversion, modeling RIM classes, inheritance, association and aggregation. The solution to address the problem of mapping complex medical data to items has also been proposed here.

Observation class represents most of the clinical related act like diagnoses, laboratory results, allergies, vital signs etc. So most of the knowledge discovery operations will be performed on observation data. Observation class is sparse, high dimensional and need frequent schema change and EAV is a widely used solution to handle these challenges. However, EAV is not a search efficient data model for knowledge discovery. We have proposed two search efficient open schema data models OEAV and PBA to handle these challenges as alternatives of EAV for medical data warehouse.

Chapter 4

Data Mining in Open Schema Data Models

Data mining can drill down into Health care data to discover knowledge in order to improve medical care and to define new guideline. Data mining in open schema data model is different from relational approach as data semantics in both models are not same. In open schema data models, different attribute values are not kept together for a single entity where as relational model does. Converting data from various open schema data models to relational model is a way to perform data mining on open schema data models as all the existing data mining algorithm are based on relational data model. However, there are following problems to do so for medical data: high dimensional medical data cannot be converted to relational model because of existing DBMS only support a limited number of columns, sparseness of medical data will put some challenges in relational based data mining, synchronization of two representation whenever insert, update is occurred and new representation is redundant. Here we have proposed new data mining algorithms to support medical research, to detect how much variability occurs in decisions, treatments, and cost and to find likelihood of diseases for open schema data models.

4.1 Mining Algorithm for Variability Finding

Rules discovered by current association mining algorithms [16, 26-31] are patterns that represent what decisions are routinely made in the Healthcare. In this problem, we need patterns that are rarely made in Healthcare. We have proposed a novel mining algorithm that can efficiently discover the association rules from open schema data models that are strong candidates of variability. The algorithm uses a different candidate itemset selection process, a modified candidate generation process, an open schema based support calculation process, and a different mechanism of generating rules from desire itemsets compared to Apriori. The algorithm treats all the observation items as being either actions that include decision, diagnosis and cost or non-actions that include lab tests, any symptom of patient and any criterion of disease. In our problem, non-action items appear very

frequently in the data, while action items rarely appear with the high frequent non-action items.

General intuition of this algorithm is as follows: based on a set of lab tests with same results, if 99% doctors practice patients as disease x and 1 percent doctors practice patients as other diseases, then there is a strong possibility that this 1 percent doctors are doing illegal practice. In other words, if consequent C occurs infrequently with antecedent A and antecedent A occurs frequently, then $A \vdash C$ is a rule that is a strong candidate of variability. The main features of the proposed algorithm are as follows:

- If minimum support is only used like conventional association mining algorithm, desired itemset that involve rarely appeared action items with the high frequent non-action items will not be found. To find rules that involve both frequent antecedent part and rare consequent items with this antecedent part, we have used two supports metrics: minimum antecedent support, maximum consequent support.
- The proposed algorithm uses maximum confidence constraint instead of widely used minimum confidence constraint to form the rules. Moreover, it partitions itemsets into action item and non-action items instead of subset generation to form rules.
- Rules have non-action items in the antecedent and action items in the consequent.
- In candidate generation, it does not check the property "Every subset of a frequent itemset is frequent" on action items of the candidate. It only checks the property on non-action items of the candidate.

Variability Association Rules

Let $D = \{t_1, t_2, \dots, t_n\}$ be a database of n transactions with a set of items $I = \{i_1, i_2, \dots, i_m\}$. Let set of action items of I be $AI = \{ai_1, ai_2, \dots, ai_k\}$ where k is the number of action items. Let set of non-action items of I be $NAI = \{nai_1, nai_2, \dots, nai_{m-k}\}$ where $m - k$ is the number of non-action items. For an itemset $P \subseteq I$ and a transaction t in D , we say that t supports P if t has values for all the attributes in P ; for conciseness, we also write $P \subseteq t$. By D_P we denote the transactions that contain all attributes in P . The support of P is computed as $(P) = \frac{|D_P|}{n}$, i.e. the fraction of transactions containing P . A variability rule is of the form: $P \vdash Q$, with $P \subset AI, Q \subset NAI, P \cap Q = \emptyset$. To hold the rule following condition must meet: $P(P) \text{ or } \text{support}(P) \geq \text{minimum antecedent support}, P(P, Q)$

or $\text{support}(P, Q) \leq \text{maximum consequent support}$ and $P(P, Q)/P(P) \leq \text{maximum confidence}$ where $P(x)$ is the probability of x .

Let AS is minimum antecedent support, CS is maximum consequent support, I_j is the itemsets of size j for next candidate generation, S_m is the desired itemset of size m ; C_k be the sets of candidates of size k . Figure 4.1 shows the association mining algorithm for finding variability in Healthcare. Like algorithm Apriori, our algorithm is also based on level wise search. Each item consists of Attribute Name/code and its value. Retrieving information of a 1-itemset in each open schema data models, we make a new 1-itemset if this 1-itemset is not created already, otherwise update its support. The non-action 1-itemset is selected if it has support greater or equal to minimum antecedent support. The action 1-itemset is selected whatever support it has. By this way, 1-itemsets are explored which has high support for antecedent items and has arbitrary support for consequent items. To retrieve information of each 1-itemset in OEAV, the algorithm reads each OEAV record and extracts value by performing AND operation between $0x0000FFFF$ and AV. It extracts attribute by performing another AND operation between $0xFFFF0000$ and AV and left shift operation by 16 bit respectively. To retrieve information of each 1-itemset in PBA, the algorithm reads each PBA record and extract value by performing bitwise AND operation between $0x0000FFFF$ and positional value. Attribute name is retrieved from initial data blocks of that attribute. To retrieve information of each 1-itemset in EAV, we need to read each EAV record.

Candidate Generation

The intuition behind candidate generation of all level-wise algorithms like Apriori is based on the following simple fact: every subset of a frequent itemset is frequent so that they can reduce the number of itemsets that have to be checked. However, our proposed algorithm in candidate generation phase check this fact if the itemsets only contains non-action items. This idea makes itemsets consist of both rare non-action items and high frequency action items. If the new candidate contains one or more action items and its non-action part hold the fact, it is selected as a valid candidate. If the new candidate contains only non-action items then, it is selected as a valid candidate only if every subset of new candidate is frequent. This way the algorithm keeps all the new candidates that have one or more action items and high frequent non-action items.

Algorithm: Find itemsets which consist of nonaction items with high support and action items with low support based on candidate generation.

Input: Database, minimum antecedent support, maximum consequent support

Output : Itemsets which are strong candidates of variability.

1. $K=1, S = \{\emptyset\};$
2. Read the metadata about which attributes are action type and which are not.
3. $I_k =$ Select 1-itemsets either which consist of a non-action item and has support greater or equal to minimum antecedent support or which consists of an action item for each data representation
4. While($I_k \neq \emptyset$) {
 - 4.1 $K++;$
 - 4.2 $C_k = \text{Candidate_generation}(I_{k-1})$
 - 4.3 For each data representation
 $\text{CalculateCandidatesSupport}(C_k)$
 - 4.4 $I_k = \text{SelectDesiredItemSetFromCandidates}(C_k, S_k, AS, CS);$
 - 4.5 $S = S \cup S_k$
5. return S

Procedure CalculateCandidatesSupport(C_k) for OEAV

1. For all patient,
 - 1.1 Find the corresponding leaf of modified B+ tree.
 - 1.2 Get the block addresses for each attribute value from the leaf.
 - 1.3 Retrieve the attribute values as a transaction t.
 - 1.4 $\text{CalculateSupportFromOneTransactionForCandidates}(C_k, t);$

procedure CalculateCandidatesSupport(C_k) for PBA

1. For each patients p,
 - 1.1 Binary search on each attribute partition based on entity to retrieve attribute values of the entity as a transaction t
 - 1.2 $\text{CalculateSupportFromOneTransactionForCandidates}(C_k, t);$

procedure CalculateCandidatesSupport(C_k) for EAV

1. For each patients p,
 - 1.1 Linear search all blocks to retrieve attribute values of the entity as a transaction t.
 - 1.2 $\text{CalculateSupportFromOneTransactionForCandidates}(C_k, t);$

procedure CalculateSupportFromOneTransactionForCandidates(C_k, t)

1. $C_1 =$ Find the subsets of C_k which are candidate
2. For each candidate $c \in C_1$
 - 2.1 $c.\text{count}++$

Algorithm : Find Association rules for Variability Finding

Input: I (Variability Itemsets), maximumConfidence

Output: R (set of rules)

1. $R = \emptyset$
2. For each $X \in I$
 - 2.1 Antecedent set $AS = (as_1, as_2, \dots, as_n)\{$
 where $as \in X$ and $AC(as) \neq 2\}$
 - 2.2 Consequent set $CS = (cs_1, cs_2, \dots, cs_n)\{$
 where $cs \in X$ and $AC(cs) \neq 1\}$
 - 2.3 if $(\text{support}(AS \cup CS) / \text{Support}(AS)) \leq \text{maximum confidence}$
 - 2.3.1 $AS \vdash CS$ is a valid rule.
 - 2.3.2 $R = R \cup (AS \vdash CS)$

<pre> procedure Candidate_generation(I_{k-1}) 1. For each Itemset $i_1 \in I_{k-1}$ 1.1 For each Itemset $i_2 \in I_{k-1}$ 1.1.1 Newcandidate, NC = Union(i_1, i_2); 1.1.2 If Size of NC is k 1.1.2.1 If NC contains one or more action items 1.1.2.1.1 Add it to C_k if every subset of non-action items is frequent. 1.1.2.2 else 1.1.2.2.1 If every subset of NC is frequent 1.1.2.2.1.1 Add it to C_k otherwise remove it. 2. return C_k; </pre>	<pre> procedure SelectDesiredItemSetFromCandidates (C_k, S_k, AS, CS) 1. For each Itemset $c \in C_k$ 1.1 If c contains only non-action items 1.1.1 If c.support $\geq AS$ 1.1.2 Add it to I 1.2 else if c contains one or more action items with non-action items. 1.2.1 If c.support $\leq CS$ 1.2.2 Add it to I & S_k 1.3 If c contains only action items 1.3.1 Add it to I 2. return I </pre>
--	---

Figure 4.1: Association mining algorithm for finding variability in healthcare

Candidate Selection

We have used two separate supports metrics to filter out candidates. Itemset with only non-action items is compared with minimum antecedent support metric as non-action items can only take part in antecedent part of variability rule, which need to be high frequent. Itemset with one or more action items is compared with maximum consequent support metric to keep rare action items with the high frequent non-action items. The itemset with only non-action items is selected if it has support greater or equal to minimum antecedent support. Itemset with one or more action items is selected if it has support smaller or equal to maximum confidence support. By this way, itemsets are explored which has high support for non-action items and low support for action items with high support non-action items. Here Pruning is based mostly on minimum antecedent support, maximum confidence support and checking the property “every subset of a frequent itemset is frequent” on non-action items.

Generating Association Rule

This problem needs association rules that represent variability relationships between action and non-action items that occur rarely together. For this reason, the proposed algorithm use maximum confidence constraint to form rules as it needs rule that has high support in antecedent portion and has very low support in itemset from which the rule is generated. It selects a rule if its confidence is less or equal to maximum confidence constraint. Moreover, it does use subset generation to itemsets to form rules. Here itemset is

partitioned into action item and non-action items. Action items are for consequent part and non-action items are for antecedent part. Here each itemset is mapped to only one rule.

Calculating Candidate Support in Open Schema Data Models

In healthcare, attribute value contains patient information that is multidimensional. For support calculation, the algorithm performs the count operation by comparing the value of attributes instead of determining presence or absence of value of attributes. To calculate support for candidate itemsets, support of candidate itemsets is counted during the pass over the data. As different attribute values are not kept together for a single entity where as relational model does, calculating candidate support in open schema data models is not straightforward as in relational model. In OEAV, like all other open schema data models attribute values are kept in separate data block for a single entity. In OEAV, modified B+ tree is used to index entity field to make entity wise search efficient. Each leaf of B+ tree keeps block address of each attribute value for an entity. By finding corresponding leaf of modified B+ tree for each patient, we get the block addresses of attribute values of that patient. Then attribute values are retrieved from the block addresses to form the transaction for that patient. For a single entity, B+ tree takes $O(\log_b n)$ time to find the address of m blocks which hold the values of m attributes where n is the number of entity and b is the capacity of nodes. In PBA, data is stored in sorted order of entity so that binary search on each attribute partition based on entity is possible. Attribute values are retrieved from each attribute partition using binary search to form the transaction for a patient. To retrieve attribute values for a single entity it takes $O(k \log_2 n)$ where n is the number of records in each column and k is the number of attributes. EAV approach has no tracking of how data are stored so it reads all attributes. It needs full scan of data using linear search to form the transaction for a patient.

4.2 Mining Algorithm to Support Medical Research

In medical qualitative research, medical researcher analyzes historical patient data to verify known trends and to discover unknown trends and relationships among medical attributes. For instance, a medical researcher can discover relationship between the age and the HbA1c% of a patient. Medical researcher is interested to find relationship among various diseases, lab tests, symptoms, etc. In other words, medical researcher is interested in

finding association rules to see relationship among specified items and to see how a group of items is related with a different group of items. Due to high dimensionality of medical data, conventional association mining algorithm discovers a very high number of rules with many attributes, which are tedious, redundant to medical researcher and not among his desired set of attributes. Medical researcher may need to find the relationship between rare and high frequent medical items, but conventional mining process for association rules explores interesting relationships between data items that occur frequently together [16]. For these reasons, we have proposed an association-mining algorithm, which will find rules among the attributes of researcher interest, so that it can help in decision making of the researcher. This algorithm allows the researchers to define the following constraints: group information of attributes, minimum confidence and support for each group, which item will appear in antecedent and which item will appear in consequent and which attributes will appear in both. One attribute can belong to several groups.

The main theme of this algorithm is based on the following two statements. interesting relationships among various medical attributes are concealed in subsets of the attributes, but do not come out on all attributes taken together. All interesting relationships among various medical attributes have not same support and confidence. The algorithm constructs a candidate itemsets based on groups constraint and use the corresponding support of each group in candidate selection process to discover all possible desired itemsets of that group. The goals of this algorithm are the following: finding desired rules of medical researcher, efficient use of open schema data models and running fast. The features of this proposed algorithm are as follows:

- It allows grouping of attributes to find relationship among medical attributes. This provides control on the search process.
- Minimum confidence and support can vary from one group to another group.
- One item can belong to several groups
- Attributes are constrained to appear on either antecedent or consequent or both side of the rule.
- It does not generate subsets on full desired itemset, but generates subsets for items that can appear in both consequent and antecedent.

- Uninteresting relationships among medical attributes are avoided in the candidate generation phase which reduces number of rules, finds out only interesting relationships and makes the algorithm fast.

Figure 4.2 shows the association-mining algorithm to support medical research. As calculating candidate itemsets support in medical research algorithm is same as in variability finding algorithm, so we will not repeat these information in this section. Like Apriori, our algorithm is also based on level wise search. The major difference in our proposed algorithm is candidate generation process with Apriori. Each item consists of Attribute Name/code and its value. Having retrieved information of a 1-itemset in each open schema data models, we make a new 1-itemset if this 1-itemset is not created already, otherwise update its support. The 1-itemset can belong to zero or more groups. 1-itemset is selected if it has support greater or equal to one of its corresponding group support. The previous section describes the methodology to retrieve information in three different open schema data models. As medical attribute value contains patient information that is multidimensional, the algorithm performs the count operation by comparing the value of attributes instead of determining presence or absence of value of attributes to calculate support.

Candidate Generation and Selection

The intuition behind candidate generation of all level-wise algorithms like Apriori is based on the following simple fact: Every subset of a frequent itemset is frequent so that they can reduce the number of itemsets that have to be checked. However, the idea behind candidate generation of proposed algorithm is every item in the itemset has to be in the same group. This idea makes the new candidates that consist of items in the same group and keeps itemsets consist of both rare items and high frequency items. If all the items in the new candidate set are in the same group, then it is selected as a valid candidate, otherwise the new candidate is not added to valid candidate itemsets. Here for each group there are different support and confidence. Each candidate itemset belongs to a particular group. After finding group id of an candidate itemset, the algorithm uses corresponding support for candidate selection where as Apriori uses a single support threshold for all the candidate itemsets. By this way, itemsets are explored which are desired to medical researchers.

<p>Algorithm: Find itemsets which has high support and are in the same group. Input: Data and index files. Output : Itemsets which are desired to Medical Researchers.</p> <ol style="list-style-type: none"> 1. $K=1$; 2. Read the metadata about which attributes can only appear in the antecedent of a rule, can only appear in the consequent and can appear in either 3. Read Groups Information along with each group support and confidence from configuration file and make dictionary, here key is the attribute number and value is a list of group numbers on which the corresponding attribute belongs to. 4. $I_k = \text{Select } k\text{-itemsets that have support greater or equal to one of its corresponding group support.}$ 5. While($I_k \neq \emptyset$) <ol style="list-style-type: none"> 5.1 $K++$; 5.2 $C_K = \text{Candidate_generation}(I_{k-1})$ 5.3 call the corresponding CalculateCandidatesSupport(C_k) method for each data representation. 5.4 $I_k = \text{SelectDesiredItemSetFromCandidates}(C_K, \text{GroupSupports})$; 5.5 $I = I \cup I_k$ 6. return I <p>procedure Candidate_generation(I_{k-1}: frequent ($k-1$) itemsets) 1. for each Itemset $i_1 \in I_{k-1}$ 1.1 for each Itemset $i_2 \in I_{k-1}$ 1.1.1 newcandidate, $NC = \text{Union}(i_1, i_2)$; 1.1.2 if size of NC is k 1.1.2.1 $\text{isInSameGroup} = \text{TestWhetherAllTheItemsInSameGroup}(NC)$ 1.1.2.2 if ($\text{isInSameGroup} = \text{true}$) 1.1.2.2.1 add NC to C_k otherwise remove it. 2. return C_k;</p>	<p>procedure SelectDesiredItemSetFromCandidates($C_K, \text{GroupSupports}$) 1. for each Itemset $c \in C_k$ 1.1 $j = \text{FindGroupNoWhichHasMinimumSupportIfMultipleGroupsExist}(c)$ 1.2 If $c.\text{support} \geq \text{GroupSupports}[j]$ 1.3 Add it to I 2. return I</p> <p>Algorithm : Find Association rules for decision supportability of medical researcher. Input: I : Itemsets, GroupConfidences Output: R: Set of rules</p> <ol style="list-style-type: none"> 1. $R = \emptyset$ 2. For each $X \in I$ 2.1 $j = \text{FindGroupNoWhichHasMinimumConfidenceIfMultipleGroupsExist}(X)$ 2.2 Both Set $B = (b_1, b_2, \dots, b_n)$ { where $a \in X$ and $AC(a) = 0$ } 2.3 Antecedent set $AS = (as_1, as_2, \dots, as_n)$ { where $a \in X$ and $AC(a) = 1$ } 2.4 Consequent set $CS = (cs_1, cs_2, \dots, cs_n)$ { where $cs \in X$ and $AC(cs) = 2$ } 2.5 For each subset Y of B 2.5.1 $Y_1 = B - Y$; 2.5.2 $AS_1 = AS \cup Y$ 2.5.3 $CS_1 = CS \cup Y_1$ 2.5.4 if ($\text{support}(AS_1 \cup CS_1) / \text{Support}(AS_1) \geq \text{GroupConfidences}[j]$; 2.5.4.1 $AS_1 \Rightarrow CS_1$ is a valid rule. 2.5.4.2 $R = R \cup (AS_1 \Rightarrow CS_1)$ 2.5.5 $AS_2 = AS \cup Y_1$ 2.5.6 $CS_2 = CS \cup Y$ 2.5.7 if ($\text{support}(AS_2 \cup CS_2) / \text{Support}(AS_2) \geq \text{GroupConfidences}[j]$; 2.5.7.1 $AS_2 \Rightarrow CS_2$ is a valid rule. 2.5.7.2 $R = R \cup (AS_2 \Rightarrow CS_2)$
---	---

Figure 4.2: Association mining algorithm to support medical research

Generating Association Rules

Let $AC(\text{item})$ be the function which returns one out of three values: 1 if item is constrained to be in the antecedent of a rule, 2 if it is constrained to be in the consequent and 0 if it can be in either. Using this function, itemset is partitioned into antecedent set, consequent set and both set. Moreover, it does not use subset generation to itemsets to form rules like conventional association mining algorithm; it only uses subset generation to both

set. Each subset of both set is added in antecedent part in one rule and is added in consequent part in another rule. Each itemset belongs to a particular group. In addition to, there is a different confidence for each group whereas Apriori uses a single confidence for all the itemsets. After finding group id of an itemset, the algorithm uses corresponding confidence to form rules. By this way, rules are explored which are desired of medical researchers.

Correlation

For the ranking of medical relationship, a direct measure of association rule between variables is a perfect scheme. For a medical relationship $s \rightarrow t$, s is a group of medical items where each item is constrained to be appear in antecedent or both and t is a group of medical attributes where each item is appear to be in consequent or both. Moreover, $s \cap t = \emptyset$. For this relationship, the support is defined as $\text{support} = P(s, t)$ and the confidence is defined as $= P(s, t)/P(t)$ where P is the probability. The correlation coefficient (also known as the Φ -coefficient) measures the degree of relationship between two random variables by measuring the degree of linear interdependency. It is defined by the covariance between the two variables divided by their standard deviations:

$$\rho_{st} = \frac{\text{Cov}(s, t)}{\sigma_s \sigma_t} \dots \dots \dots (1)$$

Here $\text{Cov}(s, t)$ represents the covariance of the two variables and σ_x and σ_y are stand for standard deviation. The covariance measures how two variables change together:

$$\text{Cov}(s, t) = P(s, t) - P(s)P(t) \dots \dots \dots (2)$$

As we know, standard deviation is the square root of its variance and variance is a special case of covariance when the two variables are identical.

$$\sigma_s = \sqrt{\text{Var}(s)} = \sqrt{\text{Cov}(s, s)} = \sqrt{P(s, s) - P(s)P(s)} = \sqrt{P(s) - P(s)^2}$$

$$\text{Similarly, } \sigma_t = \sqrt{P(t) - P(t)^2} \dots \dots \dots (3)$$

$$\rho_{st} = \frac{P(s, t) - P(s)P(t)}{\sqrt{P(s) - P(s)^2} \sqrt{P(t) - P(t)^2}} \dots \dots \dots (4)$$

Here $P(s, t)$ is the support of itemset consists of both s and t . Let the support of the itemset be S_{st} . Here $p(s)$ and $p(t)$ is the support of antecedent s and antecedent t

respectively. Let the support of antecedent s and consequent t be S_s and S_t . The value of S_{st} , S_s and S_t are computed during the desired itemset generation of our proposed algorithm. Using these values, we can calculate the correlation of every medical relationship rule between a group of medical items to another group of medical items. The correlation value will indicate the medical researcher how strong a medical relationship is in perspective of historical data.

$$\rho_{st} = \frac{S_{st} - S_s S_t}{\sqrt{S_s - S_s^2} \sqrt{S_t - S_t^2}} \dots \dots \dots (5)$$

So putting the value of S_{st} , S_s and S_t in association rule generation phase, we have found the single metric, correlation coefficient, to represent how much antecedent and consequent are medically related with each other. For each medical relationship or rule, this metric has been used to indicate the degree of strong relationship between a group of items to another group of items to support medical qualitative research. The ranges of values for ρ_{st} is between -1 and +1. If two variables are independent then ρ_{st} equals 0. When ρ_{st} equals +1 the variables are considered perfectly positively correlated. A positive correlation is the evidence of a general tendency that when a group of attribute values s for a patient happens, another group of attribute values y for the same patient happens. More positive value means the relationship is more strong. When ρ_{st} equals -1 the variables are considered perfectly negatively correlated.

4.3 Constraint K-Means-Mode Clustering Algorithm

Due to high dimensionality of medical data, if clustering is done based on all the attributes of medical domain, resultant clusters will not be useful because they are medically irrelevant, contain redundant information. Moreover, this property makes likelihood analysis hard and the partitioning process slow. To find the likelihood of a disease clustering has to be done based on anticipated likelihood attributes with core attributes of disease in data point. For example, clustering a large number of patients with selecting age, weight, sex, smoke, HbA1c% as data point and allowing only age, weight, sex, smoke in clustering process, we can find clusters partitioned by age, weight, sex, smoke. This way we get clusters that have similar age, weight, sex, smoke value. Then analyzing each cluster

based on HbA1c% can give likelihood information of diabetes. Attributes of Medical data are both continuous and categorical. K-means and K-modes clustering algorithm are recognized techniques to partition large data sets based on numerical attributes and categorical attributes respectively.

Figure 4.3 shows the proposed hybrid-partitioning algorithm, which can handle both continuous and discrete data and perform clustering based on anticipated likelihood attributes with core attributes of disease in data point. In this algorithm, the user will set which attributes will be used as data point for a patient and which attributes will participate in clustering process. The goals of this algorithm are the following:

- (i) Making clusters to find likelihood. Healthcare data is sparse as doctors perform only few different clinical lab tests for a patient over his lifetime. This is natural many patients have not all anticipated attributes for likelihood. When a patient does not have one or more anticipated attributes for likelihood, keeping this patient in clustering process will make clusters useless to find likelihood. Therefore, we are ignoring that patient in the clustering process.
- (ii) Improve performance.

Retrieving All Patients' Records from Open Schema Data Models

Here constraint k-Means-Mode expects dataset in row wise fashion but in open schema models data are in different format, so we have to make row wise data set from each representation. As user can set which attributes will be used as data point for a patient so each function `RetrieveAllPatientsRecord()` retrieves the attributes, which will be used as data point, set by the user. In OEAV & PBA, we only read the data blocks of attributes that are parts of datapoint as all data of each attribute are stored separately. In EAV, we read the blocks of all attributes as it has no tracking how data are stored. EAV reads unnecessary attributes whereas OEAV and PBA do not.

Algorithm: Partition patients to find likelihood of disease based on MeanMode value of patients.

1. Read the metadata about which attributes will only appear in clustering process.
2. Partition patient data into k cluster in random and assign each partition to each cluster. To retrieve patient data use the corresponding RetrieveAllPatientsRecord() for each data model.
3. Repeat
 - 3.1 Call UpdateMeanModeofClusters(K, M) to update Mean-Mode value of k clusters
 - 3.2 Move patient P_i to the cluster with least distance and find the distance between a patient and a cluster using the function Distance (P, C, m);

Until no patient is moved

Procedure UpdateMeanModeofClusters(K: no of clusters, M: medical attributes)

1. For each cluster $c \in K$
 - 1.1 $i = 0$
 - 1.2 For each attribute $A \in M$ where A can appear in clustering
 - 1.2.1 If A is continuous attribute
 - 1.2.1.1 $\text{MeanMode}_c[i] = \text{Find the mean among the attribute named A values of data points in cluster c.}$
 - 1.2.2 else If A is category attribute
 - 1.2.2.1 $\text{MeanMode}_c[i] = \text{Find the mode among the attribute named A values of data points in cluster c.}$
 - 1.2.3 $i++$;

Procedure Distance (P: Patient, c: Cluster, m: no of attributes)

//Here P_i represent the i^{th} attribute value of Patient P and C_i represents i^{th} MeanMode value of Cluster C

1. for $i = 1$ to m where i^{th} attribute value of Patient can appear in clustering
 - 1.1 If P_i is continuous
 - 1.1.1 Then $D_1 = D_1 + (P_i - C_i)^2$
 - 1.2 Else (categorical)
 - 1.2.1 Then $D_2 = D_2 + \text{NumberOfOnes}(P_i \wedge C_i)$;
 - 1.3 $d = \text{SQRT}(D_1) + D_2$;
2. return d;

Procedure RetrieveAllPatientsRecord() for OEAV

1. Create a matrix P of size $M \times N$, where M is the number of attributes in datapoint and n is the number of patients.
2. For each attribute A set by the uses as part of datapoint.
 - $J = \text{Attribute index of A in P}$
 - Read each OEAV record e of A
 - Attribute value = $e.AV \& 0x0,00000000,FFFF$
 - $P[e.Entity][j] = \text{Attribute value}$
3. return P

Procedure RetrieveAllPatientsRecord() for PBA

1. Create a matrix P of size $M \times N$, where M is the number of attributes in datapoint and n is the number of patients.
2. For each attribute A set by the uses as part of datapoint.
 - 2.1 $J = \text{Attribute index of A in P}$
 - 2.2 Read each PBA record e of A
 - AttributeValue = $e.PositionalValue \& 0x0,00000000,FFFF$
 - Entity = $(e.PositionalValue \& 0xF,FFFF,0000) \gg 16$
 - $P[Entity][j] = \text{AttributeValue}$
3. return P

Procedure RetrieveAllPatientsRecord() for EAV

1. Create a matrix P of size $M \times N$, where M is the number of attributes in datapoint and n is the number of patients.
1. For each EAV record e
 - If e.Attribute is part of datapoint
 - $J = \text{Attribute index of A in P}$
 - $P[e.Entity][j] = e.Value$
2. return P

Figure 4.3: Constraint k-Means-Mode clustering algorithm

Updating Cluster Center

We need to update the k clusters centre dynamically in order to minimize the intra cluster distance of patients. Here K is the number of clusters we would like to make and P_i is the i^{th} patient attribute and C_i is the i^{th} mean-mode value of cluster C . As the patient attributes are both continuous and discrete, each cluster center is an array of both average and mode value where average and mode is computed for continuous and discrete attribute respectively. Mean is computed for each continuous attribute by calculating average of that attribute among the data points in that cluster. Mode is computed for each discrete attribute by calculating maximum frequent value of that attribute among the data points in that cluster.

Dissimilarity Measure

The object dissimilarity measure is derived from both numeric and categorical attributes. For discrete features, the dissimilarity measure between two data point depends on the number of different values in each categorical feature. For continuous features, the dissimilarity measure between two data point depends on Euclidean distance. Here we have used the following two functions to measure dissimilarity: hamming distance function for categorical objects and Euclidean distance function for continuous data. To measure distance between two objects based on several features, for each feature we test whether this feature is discrete or continuous. If the feature is continuous, distance is measured using Euclidean distance and added it to D_1 and if the feature is discrete, the dissimilarity is measured using hamming distance and added it to D_2 . The resultant distance is computed by adding square root of D_1 with D_2 . The computational complexity of the algorithm is $O((I+1)kp)$, where p is the number of patients, k the number of clusters and I is the number of iterations.

Let the anticipated likelihood attributes be $L = \{l_1, l_2, l_3, \dots, l_s\}$. Let the core attributes of disease, $CA = \{ca_1, ca_2, ca_3, \dots, ca_n\}$. In the clustering process, only anticipated likelihood attributes participate. The anticipated likelihood attributes consist of both continuous and categorical attribute. Let first e attributes of L are continuous and the remaining $s - e$ attributes are categorical. Let the anticipated likelihood attributes of two data points are L_i and L_j . Dissimilarity between the anticipated likelihood attributes of two data points is the sum of dissimilarity of continuous attribute and dissimilarity of categorical attribute. Distance is measure using Euclidian distance function for continuous attributes.

Distance between L_i and L_j based on continuous attributes is $\sqrt{\sum_{p=1}^e (l_{ip} - l_{jp})^2}$ where $i, j \in n$ and n is the number of patients. Distance is measured using Hamming distance function for categorical attributes. Distance between L_i and L_j based on categorical attributes is $\sum_{p=e}^s f(l_{ip}, l_{jp})$ where $f(l_{ip}, l_{jp}) = \begin{cases} 0 & \text{if } l_{ip} == l_{jp} \\ 1 & \text{if } l_{ip} \neq l_{jp} \end{cases}$

Likelihood

Likelihood is the probability of a specified outcome. After clustering using constrained K-Means-Mode algorithm we get a set of clusters, $C = \{c_1, c_2, c_3, \dots, c_k\}$. Each cluster contains a set of data points, which consist of anticipated likelihood attributes and core attributes of disease. Data points for cluster c_j is $D_j = \{d_{j1}, d_{j2}, d_{j3}, \dots, d_{ju}\}$. There are a set of boolean functions on core attributes of disease to determine whether a data point has the presence of the disease or not. Let the set of boolean functions be $F = \{f_1, f_2, f_3, \dots, f_v\}$. A data point d_t has presence of the disease if $\bigcap_{i=1}^v f_i(d_t) == \text{true}$ for the data point. In a cluster, the number of data points which has presence of the disease is $\sum_{j=1}^u \bigcap_{i=1}^v f_i(d_j)$. The number of total data points in the cluster is $\sum_{j=1}^u u$. So likelihood of a cluster for the disease is $\frac{\sum_{j=1}^u \bigcap_{i=1}^v f_i(d_j)}{\sum_{j=1}^u u}$ where f_i is the function, which returns either one or zero.

Here each cluster is represented by the mean mode value of that cluster. Now we will find the equation of mean mode value of a cluster c . Mean is calculated among the continuous attributes and mode is calculated among the categorical attributes. Let the mean mode value of a cluster be $MM = \{mm_1, mm_2, mm_3, \dots, mm_z\}$ where z is the number of attributes in the clustering process. Let first y attributes of MM are continuous and remaining $z - y$ are categorical. The continuous part of mean mode value is $\{MM_i\}_{i=1, \dots, y} =$ the mean among i^{th} attribute values of cluster c . The categorical part of mean mode value is $\{MM_j\}_{j=y+1, \dots, z} =$ the mode among j^{th} attribute values of cluster c .

4.4 Summary

Data mining in open schema data model is different from relational approach as data semantics in both models are not same. Although we have used level-wise search for both variability finding and medical research algorithm, each step of our algorithm is different from that of algorithm Apriori, from initialization, candidate itemsets generation, support calculation for valid candidate itemsets, pruning of candidate itemsets. Rules generation from desired itemsets is also different from conventional association mining algorithm. In Constraint k-Means-Mode algorithm, we have made modifications in the k-means algorithm to address the problem of clustering categorical data and to allow user to specify constraint on what attributes will participate in clustering process and what attributes will be selected as data point.

Chapter 5

Results and Discussions

This chapter describes the experimental works to evaluate the performance of the proposed data mining algorithms using the representative dataset and to verify the feasibility and scalability of the design of Optimized EAV and Positional Bitmap Approach to model the observation class of HL7 RIM in a clinical data warehouse. A data warehouse serves as a basis for advanced decision support systems based on statistical, data mining methods. The experimental evaluation has been performed with large synthetic data. The storage, retrieval (query) time, performance of basic operations of knowledge discovery and performance of proposed data mining algorithms in OEAV, PBA data model are compared with widely used EAV approach for the same purpose in medical domain.

5.1 Experimental Setup

The experiments were done using PC with core 2 duo processor with a clock rate of 1.8 GHz and 3GB of main memory. The operating system is Microsoft Vista and implementation language is c#.

5.2 The Data Sets

It is quite a hard task to gather the real data sets for the Health care data. We have designed a data generator that generates the data that resembles the real data sets for health care. We assume an approximate cardinality of the domain values of each attribute. We have considered all categories of healthcare data: ratio, interval, decimal, integer, percentage etc. To create synthetic data, we generate random data for all possible categories of data. All these synthetic data is converted into items (integer representation) using rule generator and data dictionary. This data set is generated with 5000 attributes and (5-10) attributes per transaction on average. We have used highly skewed attributes in all performance evaluations to measure the performance improvement of our proposed open schema data models in worst case. For all performance measurement except storage performance, we have used 1 million transactions.

5.3 Performance Evaluation of Proposed Open Schema Data Models

5.3.1 Storage Performance

Figure 5.1 shows the Storage Performance of observation table for synthetic dataset in three different open schema data models.

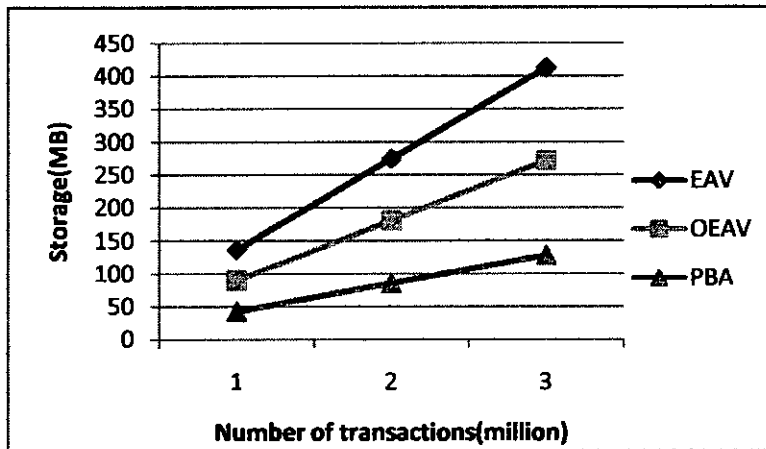
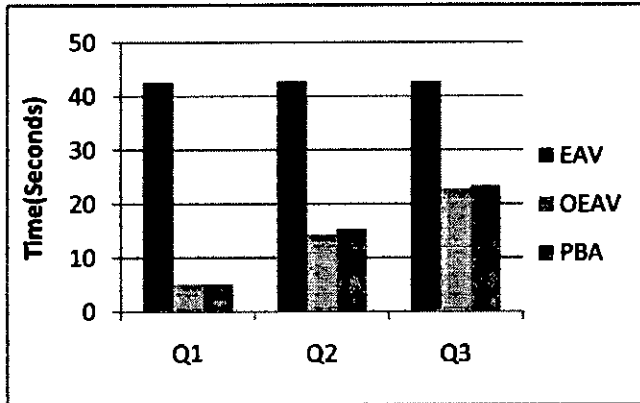


Figure 5.1: Storage performance

The storage space required by various approaches shows that the EAV occupies significantly higher amount of storage than OEAV and PBA. This is due to the data redundancy of EAV models. PBA occupies less storage than EAV and OEAV as PBA does not store attribute name as data rather than it stores attribute name in its metadata only once.

5.3.2 Time Comparison of Projection Operations

Figure 5.2 shows the performance of Projection operations on various combinations of attributes. Almost same time is needed with different number of attributes in EAV, as it has to scan all the blocks whatever the number of attributes. In OEAV and PBA, it can be observed that the time requirement is proportional to the number of attributes projected. This is because that the query needs to scan more number of blocks as the number of attributes increases. The time increase is linear with the number of attributes.



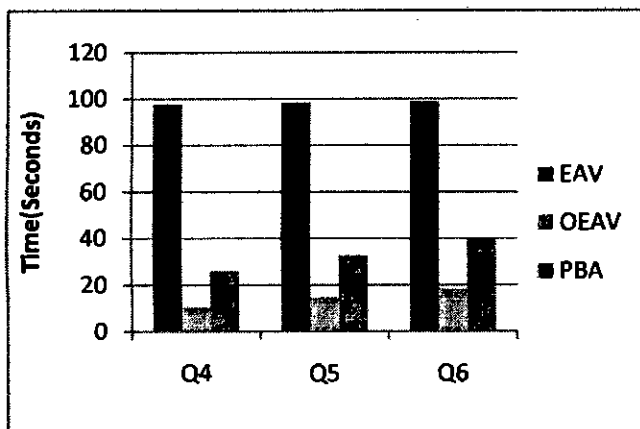
Q1: SELECT A_i
FROM observation;
Q2: SELECT A_i, A_j, A_k
FROM observation.
Q3: SELECT A_i, A_j, A_k, A_l, A_m
FROM observation;

Figure 5.2: Time comparison of projection operations

The other observation is that OEAV and PBA require almost same time, as both do not need to read unnecessary attributes. PBA takes slightly more time than OEAV, as it has to perform 16 bit left shift after each record read to get the entity.

5.3.3 Time Comparison of Multiple Predicates Select Queries

Figure 5.3 shows the performance of multiple predicates select queries on various combinations of attributes in three different open schema data models. In PBA, It can be observed that the number of attributes in select queries leads to the time taken. It is because as number of attributes in select queries increases, it has to scan more number of attribute partitions. Almost same time is taken with different number of attributes in EAV as it has to scans all the blocks twice whatever the number of attributes in predicate.



Q4: SELECT * FROM observation
WHERE $A_i = 'XXX'$;

Q5: SELECT * FROM observation
WHERE $A_i = 'XXX'$ AND $A_j = 'YYY'$;

Q6: SELECT * FROM observation
WHERE $A_i = 'XXX'$ AND $A_j = 'YYY'$
AND $A_k = 'ZZZ'$

Figure 5.3: Time comparison of multiple predicates select queries

This experiment shows that EAV requires much higher time compared to other models. It is because that EAV has no tracking of how data are stored, so it has to scans all

the blocks once to select entities and has to scan all the blocks again to retrieve the attribute values for the selected entities. We can see from this figure OEAV has taken the lowest time as it does not need to read unnecessary attributes to select entities and can retrieve attribute values of these entity without reading any unnecessary attribute value using entity indexing. PBA does not need to read unused attributes to select entities too. Nevertheless, it takes more time than OEAV as attribute values of these entities are retrieved from each attribute partition using binary search and it reads unused attribute values.

5.3.4 Time Comparison of Aggregate Operations

Aggregate operations compute a single value by taking a collection of values as input. Figure 5.4 shows the performance of various Aggregate operations on a single attribute. Time is not varied significantly from one aggregate operation to another as different aggregate operations need same number of data block access for most of the cases.

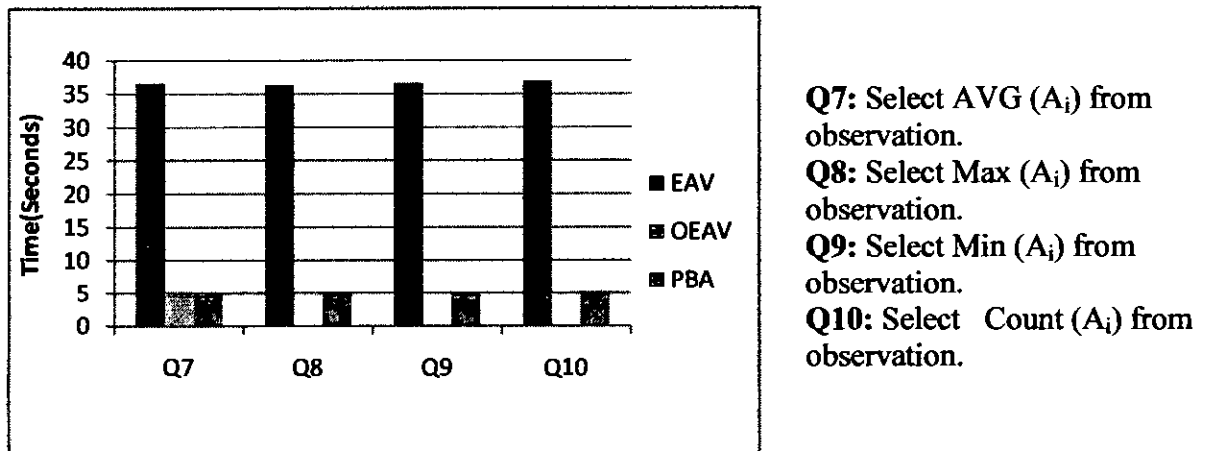


Figure 5.4: Time comparison of aggregate operations

PBA has taken almost same time to find count, min, max, average on a single attribute. This is because for all operations, it needs to scan the blocks of the attribute for which particular operation is executing. EVA has taken much higher time than other data models as it has to scan all the blocks to compute each operation. OEAV has taken negligible time for max, min, count operations on a single attribute as to find max and min it has to scan only 1 block and count result is computed from its index table only. For average operation on an attribute, it has taken considerable time, as it has to scan all the blocks of that attribute.

5.3.5 Time Comparison of Statistical Operations

Figure 5.5 shows the performance of various statistical operations on a single attribute. Time is varied significantly from one statistical operation to another as different statistical operations need different types of processing.

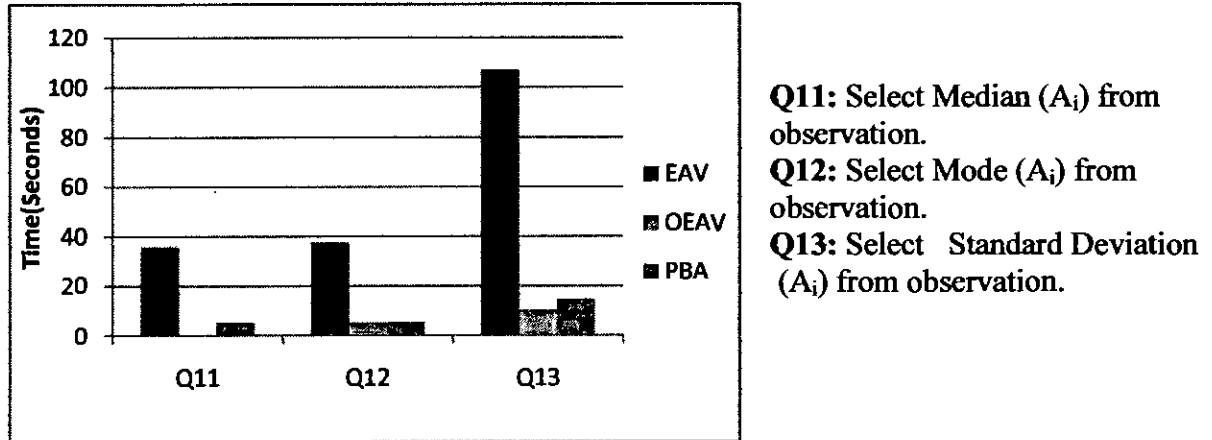


Figure 5.5: Time comparison of statistical operations

This experiment shows that EAV has taken much higher time compared to other models. It is because it has no tracking of how data are stored, so it has to scan all the blocks to compute each operation. Results shows that OEAV has taken negligible time for median operation as it has to scan 1 or 2 blocks for this operation. For mode and standard deviation, it has to scan all data blocks of the attribute for which particular operation is executing once, twice respectively. For median, mode and standard deviation, PBA has to scan the blocks of attribute for which particular operation is running one, one and two times respectively.

5.3.6 Time Comparison of CUBE Operations

The CUBE operation is the n-dimensional generalization of group-by operator. The cube operator unifies several common and popular concepts: aggregates, group by, roll-ups, drill-downs and cross tabs. Here no pre-computation is done for aggregates at various levels and on various combinations of attributes. Figure 5.6 shows the performance of CUBE Operations on various combinations of attributes. It can be observed that the number of attributes in cube operations determines the time taken by CUBE operation as it computes group-bys corresponding to all possible combinations of CUBE attributes.

The experimental results show that EAV has taken much higher time compared to other models as it is not partitioned attribute wise and it has no entity index, so every search becomes full scan of all blocks. Result shows that OEAV has taken lowest time. It is because OEAV Data is partitioned attribute wise and a modified B+ tree is constructed for indexing entity. PBA has taken the second lowest time, as it does not need to read unused attributes but does read unused values during the binary search.

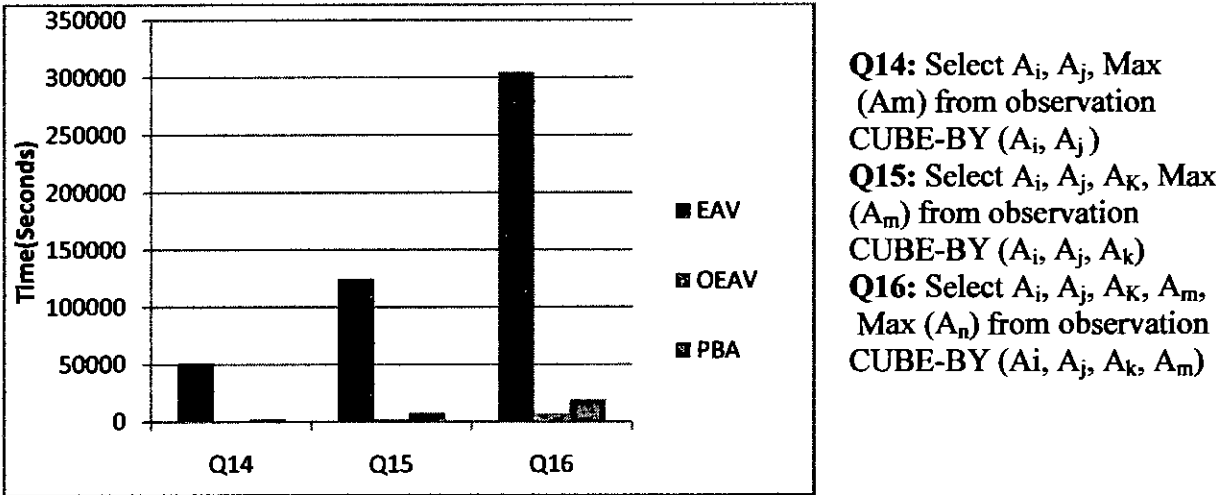


Figure 5.6: Time comparison of CUBE operations

5.4 Performance Evaluation of Proposed Data Mining Algorithms

5.4.1 Performance Evaluation of Variability Finding Algorithm

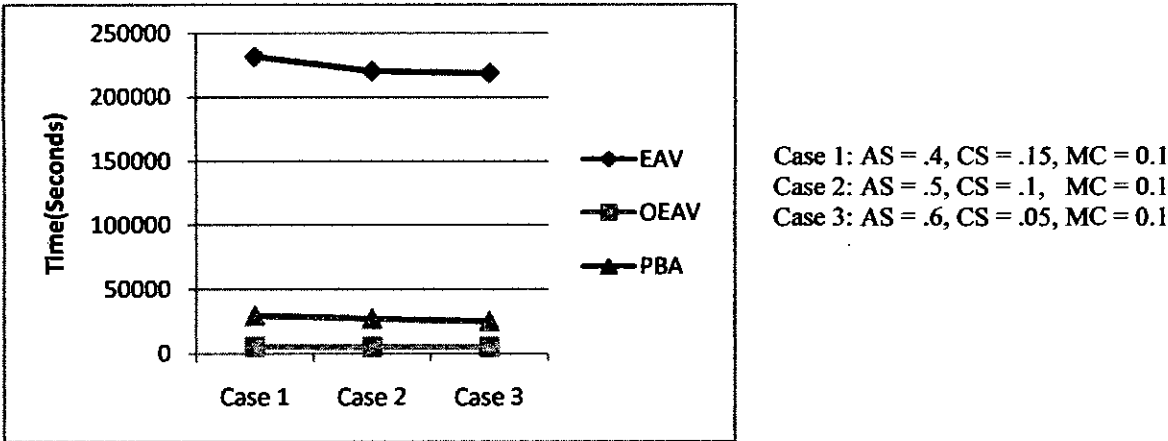


Figure 5.7: Performance of Variability Finding algorithm

Figure 5.7 shows how time varied with different antecedent minimum support (AS) and consequent maximum support (CS) values for Variability Finding algorithm. Here we measured the performance of Variability Finding algorithm in terms of AS and CS keeping Maximum Confidence (MC) constant. Time is not varied significantly because AS and CS have no lead to reduce disk access as synthetic data has all sizes of candidates for these AS and CS values. These parameters have only lead to the number of valid candidate generations and it can save some CPU time. As the parameters have lead to the CPU time, the three different cases for a specific model takes slightly different time. As minimum antecedent support increases, generating of number of valid candidates increases. As maximum consequent support decreases, number of valid candidate generation increases. For this reason, for a particular data model case 2 takes less time than case 1 and case 3 takes less time than case 2. The experiment shows that EAV has taken much higher time compared to other open schema data models. It is because it has no tracking of how data are stored, so it has to scans all the blocks for a patient record. We can see from these figures OEAV has taken the lowest time as it does not need to read unused attributes and unused values, and PBA has taken the second lowest time as it does not need to read unused attributes but does read unused values during the binary search.

5.4.2 Performance Evaluation of Medical Research Algorithm

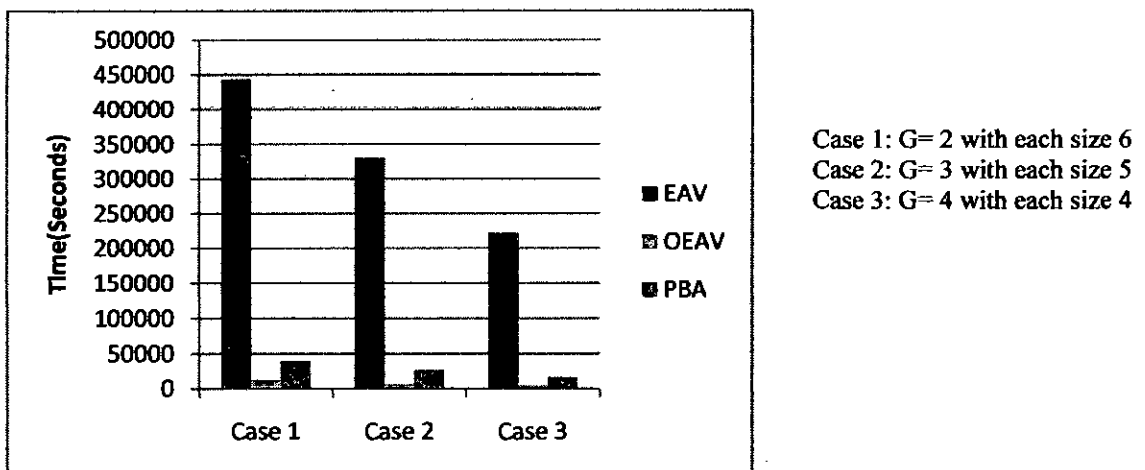


Figure 5.8: Performance of Medical Research algorithm

Figure 5.8 shows how time varied with different number of groups values with different group size for medical research algorithm. Here we measured the performance of

Medical Research algorithm in terms of number of groups and group size keeping support and confidence of each Group, antecedent and consequent constrains on attributes constant. Time is varied as small group size leads to less number of candidate generations as well as less number of support calculations, which needs to pass over the data. Number of Groups has no lead to the number of candidate generations and to the number of support calculations. The experiment shows that EAV has taken much higher time compared to other open schema data models. It is because it has no tracking of how data are stored, so it has to scans all the blocks for a patient record. We can see from these figures OEAV has taken the lowest time as it does not need to read unused attributes and unused values, and PBA has taken the second lowest time as it does not need to read unused attributes but does read unused values during the binary search.

5.4.3 Performance Evaluation of Constraint K-Means-Mode Clustering Algorithm

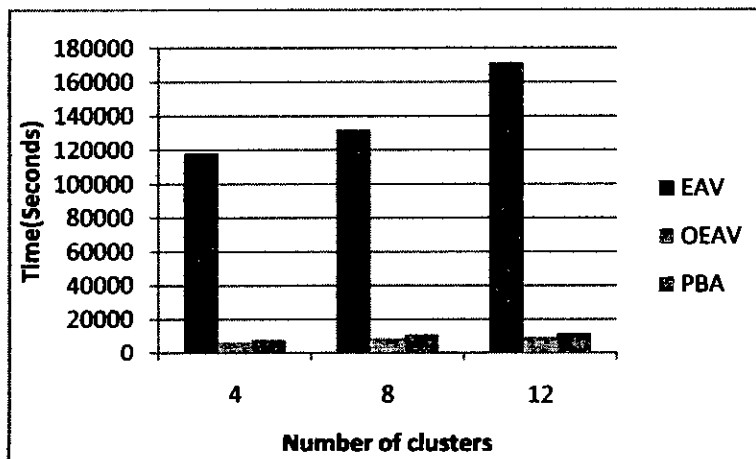


Figure 5.9: Performance of Constraint K-Means-Mode algorithm

Figure 5.9 shows the performance of proposed k-Means-Mode algorithm at various numbers of clusters (K) for synthetic dataset. It can be observed that value of k leads to the number of moves of patients from one cluster to another. For greater value of k, large number of patients move from one cluster to another and need to update more clusters center, so time taken to converge depends on the value of k. The experiment results show that EAV has taken much higher time compared to other open schema data models, as it has no tracking of how data are stored. Figure 5.9 shows OEAV has taken a bit lower time than PBA as PBA has to perform extra left shifting operation after every record read and its columnar implementation.

5.5 Performance Evaluation of Proposed Open Schema Data Models using Oracle DBMS

The objective of this experiment is to analyze the feasibility of the proposed open schema data model in commercial database context. EAV has been implemented using a single table of oracle with three attribute entity, attribute, and value. Oracle non-unique indexing is applied to both entity and attribute field. Non-unique indexes permit duplicates values in the indexed column. These oracle non-unique indexes are type of B-tree indexes. In a B- tree, we can store both keys and data in the internal/leaf nodes. However, in a B+ tree we have to store the data in the leaf nodes only. The primary advantage of B-tree index is that there are early outs when we might have found a match in an internal node. To implement OEAV in oracle database, we have kept each attribute partition in a table, and keep the metadata, which represent what attribute partition in which table, has been kept in index table. To implement PBA, we have also kept separate table for each attribute partition.

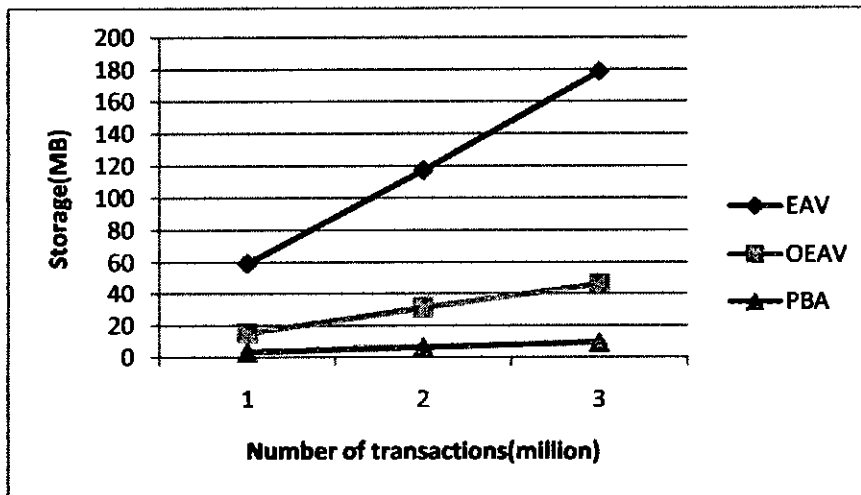


Figure 5.10: Storage performance using Oracle DBMS

Figure 5.10 shows the Storage Performance of three different open schema data models for synthetic dataset in Oracle DBMS. The storage space required by various approaches shows that the EAV occupies significantly higher amount of storage than OEAV and PBA. This is due to the data redundancy of EAV models and non-unique indexing on both attribute and entity field. PBA occupies less storage than EAV and OEAV as PBA does not store attribute name as data rather than it stores attribute name in its

metadata only once. OEAV keeps attribute as integer code whereas EAV keeps attribute name as string.

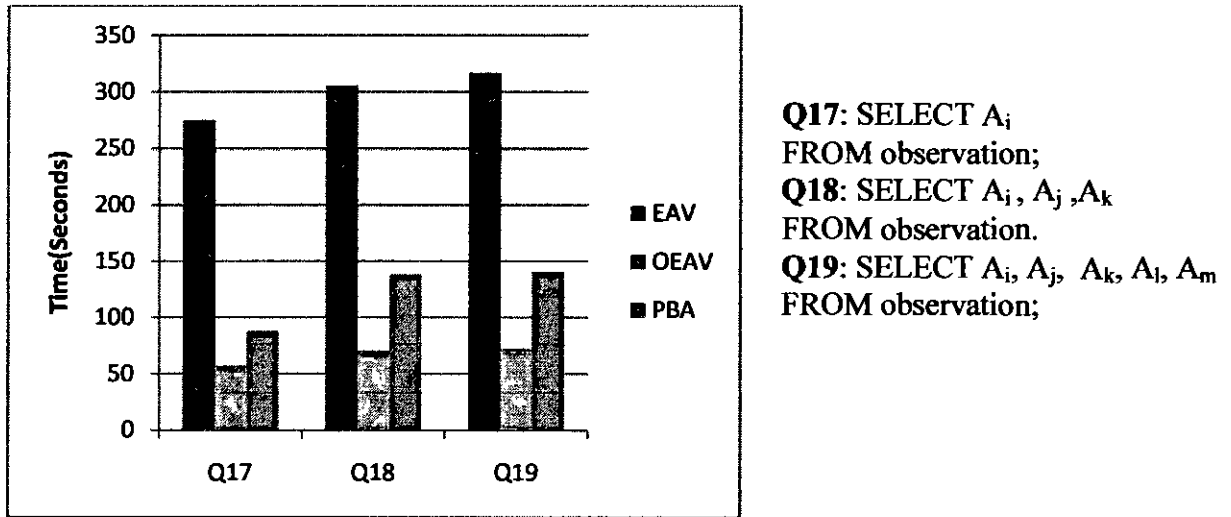


Figure 5.11: Time comparison of projection operations using Oracle DBMS

Figure 5.11 shows the performance of Projection operations on various combinations of attributes. it can be observed that the time requirement of EAV is higher than OEAV and PBA as data are not partitioned attribute wise in EAV. The other observation is that PBA requires more time than OEAV as it has to perform left shift operation after each record read to get the entity and value.

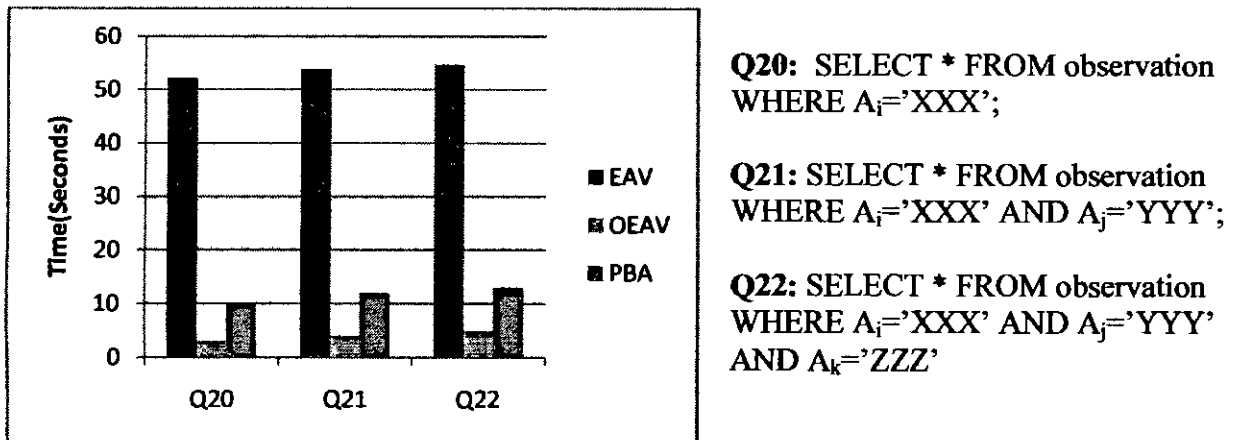


Figure 5.12: Time comparison of multiple predicates select queries using Oracle DBMS

Figure 5.12 shows the performance of multiple predicates select queries on various combinations of attributes in three different open schema data models. This experiment shows that EAV requires much higher time compared to other models. It is because that

EAV has no tracking of how data are stored. We can see from this figure OEAV has taken the lowest time as it does not need to read unnecessary attributes to select entities and can retrieve attribute values of these entity without reading any unnecessary attribute value using entity indexing. PBA does not need to read unused attributes to select entities too. Nevertheless, it takes more time than OEAV as attribute values of these entities are retrieved from each attribute partition using binary search and it reads unused attribute values.

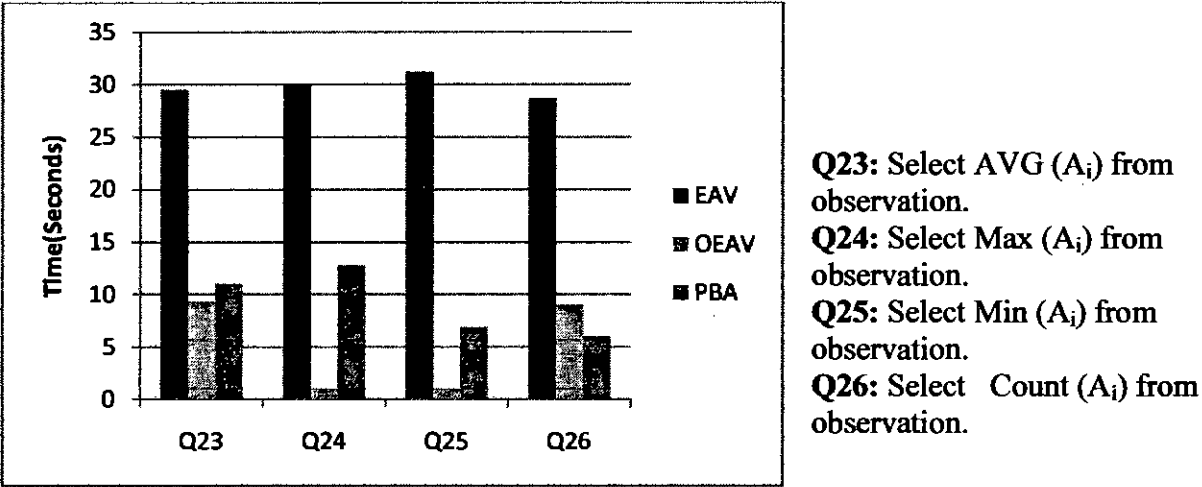


Figure 5.13: Time comparison of aggregate operations using Oracle DBMS

Aggregate operations compute a single value by taking a collection of values as input. Figure 5.13 shows the performance of various Aggregate operations on a single attribute. it can be observed that the time requirement of EAV is higher than OEAV and PBA as data are not partitioned attribute wise in EAV. PBA has taken higher time than OEAV because PBA does not keep data in sorted order of value. OEAV has taken negligible time for max, min, count operations on a single attribute as to find max and min it has to scan only 1 block and count result is computed from its index table only. For average operation on an attribute, it has taken considerable time, as it has to scan all the blocks of that attribute.

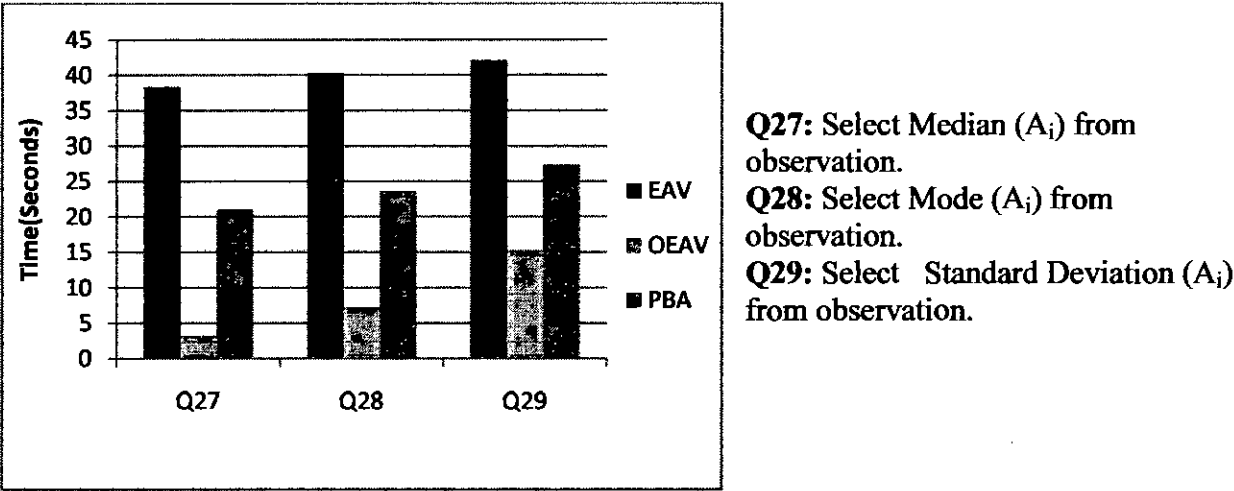


Figure 5.14: Time comparison of statistical operations using Oracle DBMS

Figure 5.14 shows the performance of various statistical operations on a single attribute. Time is varied significantly from one statistical operation to another as different statistical operations need different types of processing. This experiment shows that EAV has taken much higher time compared to other models. It is because it has no tracking of how data are stored. Results show that OEAV has taken negligible time for median operation, as it has to scan one block for this operation. For mode and standard deviation, it has to scan all data blocks of the attribute for which particular operation is executing once, twice respectively. For median, mode, and standard deviation, PBA has to scan the blocks of attribute for which particular operation is running one, two and two times respectively.

Table 5.1: How many times OEAV and PBA are cheaper and faster compared to EAV

Operation	OEAV	PBA
Storage	1.26	6.001
Projection	15.15	14.6
Selection	56.3	15.25
Aggregate	69.54	35.4152
Statistical	50.45	35.23
CUBE	45.95	16.40
Variability finding	41.97	8.20
Medical Relationship finding	41.30	11.95
Constrained K-Means-Mode	14.60	13.46

5.6 Summary

Table 5.1 summarizes how many times OEAV and PBA are cheaper and faster compared to EAV. The experimental results show our proposed open schema data models are dramatically efficient in knowledge discovery operation and occupy less storage compared to EAV. PBA outperforms other data models in terms of storage consumption and OEAV outperforms other data models in terms of search efficiency. Our solution is based on industry standard so these approaches are widely applicable. OEAV data model outperforms all the others in data mining operation. For all mining operations, the performance of EAV is significantly lower than the proposed models.

107437

Chapter 6

Conclusion and Further Research

Integrating Large-scale medical data is important for knowledge discovery. Once this knowledge is discovered, the results of such analysis can be used to define new guidelines for improving medical care and treatment. Observation class represents most of the clinical related act like diagnoses, laboratory results, allergies, vital signs etc. So most of the knowledge discovery operations will be performed on observation data. Observation data is sparse, high dimensional and need frequent schema change and EAV is a widely used solution to handle these challenges. However, EAV is not a search efficient data model for knowledge discovery. We have proposed two search efficient open schema data models OEAV and PBA to handle these challenges as alternatives of EAV for medical data warehouse.

6.1 Summary of the Thesis

The following sections summarize the thesis:

- ❑ Due to unique characteristics of medical data, it needs open schema data model.
- ❑ EAV model, which is widely used to handle these unique challenges, is not search efficient.
- ❑ To handle these unique challenges, we have proposed two search efficient open schema data models OEAV and PBA.
- ❑ OEAV and PBA are search efficient in knowledge discovery operations than EAV. OEAV and PBA also occupy less storage space than EAV.
- ❑ We have done the conversion of HL7 RIM abstract information model to physical model, which includes HL7 RIM abstract data type conversion, modeling RIM classes, inheritance, association and aggregation.
- ❑ The solution to address the problem of mapping complex medical data to items has also been proposed here.

- ❑ The experiment results show our proposed open schema data models are dramatically efficient in knowledge discovery operation and occupy less storage compared to EAV. The experimental results also show PBA outperforms other data models in terms of storage consumption and OEAV outperforms other data models in terms of search efficiency.
- ❑ Data mining in open schema data model is different from relational approach.
- ❑ We have proposed new data mining algorithms to support medical research, to detect how much variability occurs in decisions, treatments, and cost and to find likelihood of diseases for open schema data models.
- ❑ Although we have used level-wise search for both medical research and variability finding algorithm, each step of our algorithm is different from that of algorithm Apriori, from initialization, candidate item sets generation, support calculation for valid candidate item sets, pruning of candidate item sets. Rules generation from desired item sets is also different from conventional association mining algorithm.
- ❑ In constraint k-Means-Mode algorithm, we have made modifications in the k-means algorithm to address the problem of clustering categorical data and to allow user to specify constraint on what attributes will participate in clustering process and what attributes will be selected as data point.
- ❑ The experiment results show-in all the new developed data mining algorithms, OEAV data model outperforms all the others. Next, comes PBA that performs better than EAV and in EAV these algorithms are quite slow.
- ❑ Our solution is based on industry standard so these approaches are widely applicable.

6.2 Fundamental Contributions of the Thesis

- ❑ The most important contribution of this thesis is development of two search efficient open schema data models Optimized Entity Attribute Value (OEAV) and Positional Bitmap Approach (PBA) to handle data sparseness, schema change and high dimensionality of medical data as alternatives of widely used EAV data model.

- ❑ We have proposed the conversion of HL7 RIM abstract information model to physical model, which includes HL7 RIM abstract data type conversion, modeling RIM classes, inheritance, association and aggregation.
- ❑ The solution to address the problem of mapping complex medical data to items has also been proposed here.
- ❑ For open schema data models, new data mining algorithms have been proposed to support medical research, to detect illegal practice, fraud, abuse in healthcare and to find likelihood of diseases based on HL7 RIM so that it becomes widely applicable to discover new pattern and rules which can be used to improve medical care and define new treatment.

6.3 Future Plan

Medical Research algorithm runs sequentially for user-defined groups of attributes. Instead of this, the algorithm can be run in parallel to reduce the execution time and reach more enhancements. To implement so, the database transactions can be divided into a number of fragments k . If minimum support threshold for group1 is $S1$, minimum support threshold for group1 in each partition will be $S1/k$. Then the parallel algorithm will run in each database fragments. After the execution of the algorithm, desired itemsets for each fragment will be available. These desired itemset are local to one fragment and not global to database. Any desired itemset of any group respect to all transactions is also desired itemset in at least one of the fragments. Candidate itemset respect to the global database will be formed by taking the desired itemsets of all fragments. Then a second database scan will be performed to select desired itemset respect to all transactions and group policy. From these selected desired itemset, medical relationship will be found using the existing rule generation algorithm.

Medical observation data can be modeled using graph database. A graph database uses node, edges and properties to store information. To model observation data, entity and attribute will be modeled by node. Attribute value will be stored in the edge between entity node and attribute node. By this way, attribute name do not need to be stored multiple times and data will be stored attribute wise partitioned as all the values of an attribute are stored in its incident edges of that attribute node. This model can handle high dimensionality, sparseness and frequent schema change of medical data. The attributes of an entity, which

do not have any value, will be not stored by not making the edge between the entity nodes and attribute node. Data can be high dimensional because an entity node can link as many attribute nodes as it needs. It supports schema change because we can add new attribute node if we want.

Medical observation data can also be modeled using Resource Description Framework (RDF). The RDF data model is in the form of subject-predicate-object expression. These expressions are known as triples in RDF terminology. Here subject can represent entity or patient, predicate can represent the attribute and object can represent the value of the attribute for the patient. RDF has the similarity with EAV model. As RDF is an abstract model, an efficient physical implementation of this is needed to model medical observation data.

To handle terabyte level storage of health care data, distributed clinical data warehouse architecture is essential. Distributed clinical data warehouse architecture can be designed based on HL7 RIM. Fragmentation can be done based on patient or attributes of patient. Each attribute partition or a group of attribute partitions of observations class can be kept in separate workstation.

Our proposed clinical data warehouse model can be used for different types of knowledge discovery work in Healthcare. Algorithms can be developed to make decision support model (classifier) from the existing wealth of healthcare data. This decision support model can guide in treatments to improve patient care. We can use association rule algorithm to mine existing healthcare data where action type data to be constrained in consequent and non-action type data to be constrained in antecedent. Then taking the common part of antecedents of rules for the same consequent, we can create pathways of decision, treatment, and diagnosis. These pathways can be used to make the model of decision support system for healthcare.

References

- [1] Torben, P. B. and Christian, J. S., "Research Issues in Clinical Data Warehousing," in Proceedings of the 10th International Conference on Scientific and Statistical Database Management , Capri, p. 43–52, 1998.
- [2] Stead, W. W., Hammond, E. W. and Straube, J. M., "A chartless record—Is it adequate?," *Journal of Medical Systems*, vol. 7, no. 2, p. 103-109, 1983.
- [3] Anhøj, J., "Generic design of Web-based clinical databases," *Journal of Medical Internet Research*, vol. 5, no. 4, p. 27, 2003.
- [4] Brandt, C., Deshpande, A., and Lu, C., "TrialDB: A Web-based Clinical Study Data Management System AMIA 2003 Open Source Expo," in Proceedings of the American Medical Informatics Association Annual Symposium, Washington, p. 794, 2003.
- [5] Li, J., Li, M., Deng, H., Duffy, P. and Deng, H., "PhD: a web database application for phenotype data management," *Oxford Bioinformatics*, vol. 21, no. 16, p. 3443-3444, 2005.
- [6] Nadkarni, P. M. et al., "Managing Attribute—Value Clinical Trials Data Using the ACT/DB Client—Server Database System," *The Journal of the American Medical Informatics Association*, vol. 5, no. 2, p. 139–151, 1998.
- [7] Thomas, E. J., Jeffrey, T. W. and Joel, C. D., "A health-care data model based on the HL7 reference information model," *IBM Systems Journal*, vol. 46, no. 1, p. 5 - 18, 2007.
- [8] HL7. [Online]. <http://www.hl7.org/>
- [9] HL7. [Online]. http://www.hl7.org/Library/data-model/RIM/modelpage_mem.htm
- [10] Kimball, R. and Ross, M., *The Data Warehouse Toolkit Second Edition.*: John Wiley and Sons, Inc, 2002.

-
- [11] Rakesh, A., Amit, S. and Yirong, X., "Storage and Querying of E-Commerce Data," in Proceedings of the 27th International Conference on Very Large Data Bases, p. 149 - 158, 2001.
- [12] Lowell, V., Barry, S., and Werner, C., Foundation for the electronic health record: an ontological analysis of the HL7's reference information model. [Online]. http://ontology.buffalo.edu/medo/HL7_2007.pdf
- [13] Inmon, W. H., Building the Data Warehouse, 2nd ed.: Wiley Computer Publishing, 1996.
- [14] Mannila, H., "Database methods for data mining," in The Fourth International Conference on Knowledge Discovery and Data Mining, 1998.
- [15] Piatetsky-Shapiro, G., "Data Mining and Knowledge Discovery - 1996 to 2005: Overcoming the Hype and moving from "University" to "Business" and "Analytics", " Data Mining and Knowledge Discovery journal, 2007.
- [16] Agrawal, R. and Srikant, R., "Fast Algorithms for Mining Association Rules in Large Databases," in Proceedings of the 20th International Conference on Very Large Data Bases, San Francisco, CA, USA, p. 487 - 499, 1994.
- [17] Macqueen, J., "Some methods of classification and analysis of multivariate observations," in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Berkeley, CA, p. 281-297, 1967.
- [18] Jason, L. A. et al., "Mapping From a Clinical Data Warehouse to the HL7 Reference Information Model," in AMIA Annu Symp Proc, p. 920, 2003.
- [19] Jiye, A., Xudong, L., Huilong, D., Haomin, L. and Peipei, J., "An Act Indexing Information Model for Clinical Data Integration," in Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on, p. 1099 - 1102, 2007.

-
- [20] Kyle, B. and Bruce, W. G., "Crossing Chasms: a pattern language for object-RDBMS integration: the static patterns," in Pattern languages of program design 2. Boston: Addison-Wesley Longman Publishing Co, 1996.
- [21] Scott, A. W., Techniques for Successful Evolutionary/Agile Database Development. [Online]. <http://www.agiledata.org/essays/mappingObjects.html>
- [22] Widom, J., "Research Problems in Data Warehousing," in In Proceedings of CIKM, p. 25–30, 1995.
- [23] Peter, C. P., "The entity-relationship model—toward a unified view of data," ACM Transactions on Database Systems, vol. 1, no. 1, p. 9 - 36, 1976.
- [24] Agarwal, S. et al., "On the Computation of Multidimensional Aggregates," in Proceedings of the 22th International Conference on Very Large Data Bases, p. 506-521, 1996.
- [25] Jim, G., Surajit, C. and ADAM, B., "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab and Sub-Totals," Data Mining and Knowledge Discovery, vol. 1, no. 1, p. 29–53, 1997.
- [26] Brin, S., Motwani, R., Ullman, J. D. and Tsur, S., "Dynamic Itemset Counting and Implication Rules for Market Basket Data," in Proceedings of the 1997 ACM SIGMOD international conference on Management of data, Tucson, Arizona, United States, p. 255-264, 1997.
- [27] Park, J. S., Chen, M. and Yu, P. S., "An Effective Hash based Algorithm for mining association rules," in Prof. ACM SIGMOD Conf Management of Data, New York, NY, USA, p. 175 - 186, 1995.
- [28] Agrawal, R., Imieliński, T. and Swami A., "Mining Association Rules between Sets of Items in Very Large Databases," in Proceedings of the 1993 ACM SIGMOD international conference on Management of data, Washington, D.C., p. 207-216, 1993.
- [29] Mannila, H., Toivonen, H. and Verkamo, A. I., "Efficient Algorithms for Discovering

-
- Association Rules," in AAAI Workshop on Knowledge Discovery in Databases, p. 181-192, 1994.
- [30] Srikant, R. and Agrawal, R., "Mining Generalized Association Rules," in In Proc. of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, 1995.
- [31] Srikant, R., Vu, Q. and Agrawal, R., "Mining association rules with item constraints," in In Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, p. 67--73, 1997.
- [32] Liu, B., Hsu, W. and Ma, Y., "Mining association rules with multiple minimum supports," in Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, San Diego, California, United States, p. 337 - 341, 1999.
- [33] Yun, H., Ha, D., Hwang, B. and Ryu, K. H., "Mining Association Rules On Significant Rare Data Using Relative Support," The Journal of Systems and Software, vol. 67, p. 181-191, 2003.
- [34] Hahsler, M., "A Model-Based Frequency Constraint for Mining Associations from Transaction Data," Data Mining and Knowledge Discovery, vol. 13, no. 2, p. 137 - 166, 2006.
- [35] Zhou, L. and Yau, Stephen., "Association Rule and Quantitative Association Rule Mining among Infrequent Items," in International Workshop on Multimedia Data Mining, p. 156-167, 2007.
- [36] Huang, Z., "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values," Data Mining and Knowledge Discovery, vol. 2, no. 2, p. 283 - 304, September 1998.
- [37] San, O. M., Huynh, V. and Nakamori, Y., "An Alternative Extension of the k-Means Algorithm for Clustering Categorical Data," JAMCS, vol. 14, no. 2, p. 241-247, 2004.
- [38] Hoque, A. S. M. L. "Storage and Querying of High Dimensional Sparsely Populated Data in Compressed Representation," Lecture Notes on Computer Science, vol. 2510,

p. 418 - 425, 2002.

[39] Health Level Seven. [Online].

<http://www.hl7.org/v3ballot/html/infrastructure/datatypes/datatypes.htm>

[40] Health Level Seven. [Online].

<https://www.hl7.org/library/data-model/RIM/C30202/vocabulary.htm>

[41] Graefe, G., "Query Evaluation Techniques for Large Databases," ACM Comp. Surveys, vol. 25, no. 2, p. 73-170, June 1993.

Appendix 1

HL7 Data Types

Name	Symbol	Description
Boolean	BL	BL stands for the values of two-valued logic. A BL value can be either true or false, or, as any other value may be NULL.
BooleanNonNull	BN	BN constrains the boolean type so that the value may not be NULL. This type is created for use within the data types specification where it is not appropriate for a null value to be used
Encapsulated Data	ED	Data that is primarily intended for human interpretation or for further machine processing outside the scope of HL7. This includes unformatted or formatted written language, multimedia data, or structured information in as defined by a different standard (e.g., XML-signatures.) Instead of the data itself, an ED may contain only a reference (see TEL.) Note that ST is a specialization of the ED where the mediaType is fixed to text/plain.
Character String	ST	The character string data type stands for text data, primarily intended for machine processing (e.g., sorting, querying, indexing, etc.) Used for names, symbols, and formal expressions.
Concept Descriptor	CD	A CD represents any kind of concept usually by giving a code defined in a code system. A CD can contain the original text or phrase that served as the basis of the coding and one or more translations into different coding systems. A CD can also contain qualifiers to describe, e.g., the concept of a "left foot" as a postcoordinated term built from the primary code "FOOT" and the qualifier "LEFT". In cases of an exceptional value, the CD need not contain a code but only the original text describing that concept.
Coded Simple Value	CS	Coded data in its simplest form, where only the code is not predetermined. The code system and code system version are fixed by the context in which the CS value occurs. CS is used for coded attributes that have a single HL7-defined value set.
Coded Ordinal	CO	Coded data, where the coding system from which the code comes is ordered. CO adds semantics related to ordering so that models that make use of such domains may introduce model elements that involve statements about the order of the terms in a domain.

Name	Symbol	Description
Coded With Equivalents	CE	Coded data that consists of a coded value and, optionally, coded value(s) from other coding systems that identify the same concept. Used when alternative codes may exist.
Character String with Code	SC	A character string that optionally may have a code attached. The text must always be present if a code is present. The code is often a local code.
Instance Identifier	II	An identifier that uniquely identifies a thing or object. Examples are object identifier for HL7 RIM objects, medical record number, order id, service catalog item id, Vehicle Identification Number (VIN), etc. Instance identifiers are defined based on ISO object identifiers.
Telecommunicati on Address	TEL	A telephone number (voice or fax), e-mail address, or other locator for a resource mediated by telecommunication equipment. The address is specified as a Universal Resource Locator (URL) qualified by time specification and use codes that help in deciding which address to use for a given time and purpose.
Postal Address	AD	Mailing and home or office addresses. A sequence of address parts, such as street or post office Box, city, postal code, country, etc.
Entity Name	EN	A name for a person, organization, place or thing. A sequence of name parts, such as given name or family name, prefix, suffix, etc. Examples for entity name values are "Jim Bob Walton, Jr.", "Health Level Seven, Inc.", "Lake Tahoe", etc. An entity name may be as simple as a character string or may consist of several entity name parts, such as, "Jim", "Bob", "Walton", and "Jr.", "Health Level Seven" and "Inc.", "Lake" and "Tahoe".
Trivial Name	TN	A restriction of entity name that is effectively a simple string used for a simple name for things and places.
Person Name	PN	An EN used when the named Entity is a Person. A sequence of name parts, such as given name or family name, prefix, suffix, etc. A name part is a restriction of entity name part that only allows those entity name parts qualifiers applicable to person names. Since the structure of entity name is mostly determined by the requirements of person name, the restriction is very minor.
Organization Name	ON	An EN used when the named Entity is an Organization. A sequence of name parts.
Integer Number	INT	Integer numbers (-1,0,1,2, 100, 3398129, etc.) are precise

Name	Symbol	Description
		numbers that are results of counting and enumerating. Integer numbers are discrete, the set of integers is infinite but countable. No arbitrary limit is imposed on the range of integer numbers. Two NULL flavors are defined for the positive and negative infinity.
Real Number	REAL	Fractional numbers. Typically used whenever quantities are measured, estimated, or computed from other real numbers. The typical representation is decimal, where the number of significant decimal digits is known as the precision.
Ratio	RTO	A quantity constructed as the quotient of a numerator quantity divided by a denominator quantity. Common factors in the numerator and denominator are not automatically cancelled out. The RTO data type supports titers (e.g., "1:128") and other quantities produced by laboratories that truly represent ratios. Ratios are not simply "structured numerics", particularly blood pressure measurements (e.g. "120/60") are not ratios. In many cases the REAL should be used instead of the RTO.
Physical Quantity	PQ	A dimensioned quantity expressing the result of measuring.
Monetary Amount	MO	An MO is a quantity expressing the amount of money in some currency. Currencies are the units in which monetary amounts are denominated in different economic regions. While the monetary amount is a single kind of quantity (money) the exchange rates between the different units are variable. This is the principle difference between PQ and MO, and the reason why currency units are not physical units.
Point in Time	TS	A quantity specifying a point on the axis of natural time. A point in time is most often represented as a calendar expression.
Set	SET	A value that contains other distinct values in no particular order.
Sequence	LIST	A value that contains other discrete (but not necessarily distinct) values in a defined sequence.
Bag	BAG	An unordered collection of values, where each value can be contained more than once in the collection.
Interval	IVL	A set of consecutive values of an ordered base data type.
History	HIST	A set of data values that have a valid-time property and thus conform to the HXIT type. The history information is not limited to the past; expected future values can also appear.

Name	Symbol	Description
Uncertain Value - Probabilistic	UVP	A generic data type extension used to specify a probability expressing the information producer's belief that the given value holds.
Non-Parametric Probability Distribution (NPPD)	NPPD	A set of <u>UVP</u> with probabilities (also known as a histogram.) All the elements in the set are considered alternatives and are rated each with its probability expressing the belief (or frequency) that each given value holds.
Periodic Interval of Time	PIVL	An interval of time that recurs periodically. PIVL has two properties, phase and period. phase specifies the "interval prototype" that is repeated every ..
Event-Related Periodic Interval of Time	EIVL	Specifies a periodic interval of time where the recurrence is based on activities of daily living or other important events that are time-related but not fully determined by time.
General Timing Specification	GTS	A <dt-TS>, specifying the timing of events and actions and the cyclical validity-patterns that may exist for certain kinds of information, such as phone numbers (evening, daytime), addresses (so called "snowbirds," residing closer to the equator during winter and farther from the equator during summer) and office hours.
Parametric Probability Distribution	PPD	A generic data type extension specifying uncertainty of quantitative data using a distribution function and its parameters. Aside from the specific parameters of the distribution, a mean (expected value) and standard deviation is always given to help maintain a minimum layer of interoperability if receiving applications cannot deal with a certain probability distribution.

