2005

A probabilistic multidimensional data model and its applications in business management

Bhaskara Reddy Moole

A Probabilistic Multidimensional Data Model and Its Applications in Business Management

by

Bhaskara Reddy Moole

MS (Computer Science), University South Carolina 1997
M.Tech. (Artificial Intelligence), University of Hyderabad 1991
BE (Computer Science), University of Mysore 1988

Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
Applied Management and Decision Sciences

Walden University
August 2005

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Walden University

SCHOOL OF MANAGEMENT

This is to certify that the doctoral dissertation by

Bhaskara Reddy Moole

has been found to be complete and satisfactory in all respects,
and that any and all revisions required by
the review committee have been made.

Review Committee
Dr. Raghu Korrapati, Committee Chairperson, Management Faculty
Dr. Christopher Kalangi, Committee Member, Management Faculty
Dr. Pamela Wilson, Committee Member, Management Faculty
Dr. Larry Beebe, School Representative, Management Faculty

President and Provost

Paula E. Peinovich, Ph.D.

Walden University
2005

Abstract

This dissertation develops a conceptual data model that can efficiently handle huge volumes of data containing uncertainty and are subject to frequent changes. This model can be used to build Decision Support Systems to improve decision-making process. Business intelligence and decision-making in today's business world require extensive use of huge volumes of data. Real world data contain uncertainty and change over time. Business leaders should have access to Decision Support Systems that can efficiently handle voluminous data, uncertainty, and modifications to uncertain data. Database product vendors provide several extensions and features to support these requirements; however, these extensions lack support of standard conceptual models. Standardization generally creates more competition and leads to lower prices and improved standards of living. Results from this study could become a data model standard in the area of applied decisions sciences.

The conceptual data model developed in this dissertation uses a mathematical concept based on set theory, probability axioms, and the Bayesian framework. Conceptual data model, algebra to manipulate data, a framework and an algorithm to modify the data are presented. The data modification algorithm is analyzed for time and space efficiency. Formal mathematical proof is provided to support identified properties of model, algebra, and the modification framework. Decision-making ability of this model was investigated using sample data. Advantages of this model and improvements in inventory management through its application are described. Comparison and contrast between this model and Bayesian belief networks are presented. Finally, scope and topics for further research are described.

Dedication

Dedicated to my father Moole Chenna Krishna Reddy, whose discipline and moderation in every aspect of life are exemplary even now at his 97 years of age, and to my mother Lakshmi Narayanamma, who showered her love and affection unconditionally.

అంకితం

ఈ పుస్తకం, 97 సంవత్సరాల వయసులో కూడా పరిమిత ఆహార విహారాదులలో, క్రమశిక్షణలో అందరికీ ఆదర్శంగా ఉన్న నా తండ్రి మూలె చెన్నక్రిష్ణా రెడ్డికి, ఫలితం ఆశించకుండా ప్రేమాభిమానాలు చూపించే నా తల్లి లక్ష్మి నారాయణమ్మకు అంకితం.

ACKNOWLEDGEMENTS

In this significant achievement of completion of dissertation research, many individuals and organizations played important roles.

Special thanks to dissertation committee chair Dr. Raghu Babu Korrapati, a long time good friend and advisor whom I consult in many matters, for taking extra interest in this dissertation. I am grateful for his help in administrative matters with the University and in publishing research results.

Thanks to my dissertation committee members Dr. Christopher J. Kalangi and Dr. Pamela Wilson for providing valuable input and school representative Dr. Larry Beebe for ensuring this dissertation complies with school mission and vision. I thank Walden University and School of Management staff and faculty.

Thanks to my wife Padmalatha and sons Ravitheja and Sumanth for putting up with the many inconveniences I put them through during this Ph.D.

Thanks to Wonder Technologies Corporation for providing financial support.

TABLE OF CONTENTS

List of Tables

Table of Figures

CHAPTER 1

INTRODUCTION

Decision Science focuses on the study of decision theory and its applications in areas such as management, economics, social science, and behavioral science. Decision theory is divided into two areas: normative or prescriptive theory and descriptive or positive theory. Studying the applicability of decision theory includes development of techniques, systems, and decision analysis. Decision Support Systems are an important area of application of decision theory. Decision Support Systems assist in decision-making process. There are five categories of Decision Support Systems: data-driven, communications-driven, document-driven, knowledge-driven, and model-driven. Many Decision Support Systems in use today are a combination of these types. Data-driven Decision Support Systems assist decision makers in analysis of large volumes of data (Turban & Aronson, 2001).

Business intelligence and decision-making in today's business world require extensive use of huge volumes of real-world data, which contain uncertainty and change over time. Many enterprises use Decision Support Systems to enhance managerial decisions. These systems should be able to handle efficiently large amounts of data, as well as uncertainty, and modifications to uncertain data. Relational database product vendors have provided several extensions and features to support these requirements, but these extensions lack support of conceptual models, which impedes growth of the software products market. Limited availability of Decision Support Systems to business could result in inconsistent and sub-optimal decisions.

Data Warehouse (DW) and On-Line Analytical Processing (OLAP) are two emerging technologies that enable business enterprises to handle extremely large amounts of data efficiently (Chaudhuri & Dayal, 1997; Chen, Hsu, & Dayal, 2000; Codd, Codd, & Sally, 1993). These technologies are used extensively in several industries such as telecommunications, financial services, retail sales, and general business intelligence gathering. Several vendors have developed DW and OLAP products; however, most of these commercial products lack a standard conceptual data model, a defined operational model (Thomas & Datta, 2001), or mechanisms to handle uncertainty (Moole, 2003).

Conceptual models for data provide a mathematical model and associated operations without reference to implementation details (Date, 2003). Standard conceptual models such as relational algebra and relational calculus proposed in 1970s facilitated product development companies contributing to and developing today's Relational Database Management Systems (RDBMS), languages such as SQL (Codd, 1971; Kimball, Reeves, Ross, & Thornthwaite, 1998; Thomas & Datta, 2001), and numerous related tools. The lack of conceptual models for DW and OLAP is impeding growth of the industry (Agarwal, Gupta, & Sarawagi, 1997; Vassiliadis & Sellis, 1999), which is a significant problem because it is preventing a $4 billion market from achieving its potential (Pendse, 2003). This problem needs attention because timely research into conceptual data models may benefit the product development market, leading to an enhanced managerial decision-making process (Codd et al., 1993; Moole, 2003; Thomas & Datta, 2001).

Growth in semi-conductor technology has resulted in cheaper computers, enabling their widespread use in business and the accumulation of huge volumes of data. The

1990s gave rise to requirements for DW and OLAP (Codd et al., 1993; Moole, 2003; Thomas & Datta, 2001). As the ancient Greek philosopher Plato said, “Necessity is the mother of Invention” (as quoted in E. D. Hirsch, Joseph F. Kett, Trefil, & Trefil, 2002). In the late 1990s, researchers in computer science began reporting data models to address the requirements of DW and OLAP. This dissertation research may have been less useful before 1990s due to the state of technology, which had not evolved to the point where data models were needed.

Background

Businesses today are recording volumes of data reaching terabytes in size. Millions of transactions among retail chains, utility companies, banks, and insurance companies take place each day. Representative financial transactions of the International Technology Group (ITG) report indicate that a telecommunications company receives over 80 million transactions a month, or approximately 2.6 million transactions per day (ITG, 2000). It would be humanly impossible to interpret these transactions to find, for example, which class of customers makes more long distance calls. Similarly, a representative retail chain with 63 supermarkets selling 19,000 products can record a staggering 3.6 million transactions per day (SUN-Microsystems, 1999). Even a small percentage of waste or fraud will result in a loss of millions of dollars and, consequently, higher prices to customers. At the same time, manual inspection of these data is not possible, as they are imprecise and change continuously.

Decision Support Systems (DSS) are used to support managerial decisions. Usually DSS involves analyzing many units of data in heuristic fashion (Inmon, 1990). To make optimal decisions using large volumes of data, managers of large enterprises need Decision Support Systems that interpret huge volumes of uncertain data as well as handle data modifications. For example, a manager assigned to maintaining inventory for a given product category finds that demand for products changes continuously based on variables such as popularity, price, discount, advertising, and competition. Many of these variables cannot be measured precisely. In addition, these variables by themselves do not identify required quantities of a product precisely, although historical information gathered weekly may be used to forecast demand. DSS are shown to improve managerial decision-making in such scenarios (Foote & Krishnamurthi, 2001).

Any forecast is subject to uncertainty; however, a database storing weekly forecasts may be generated and compared to actual sales data after the fact. When this process is applied continuously, the result is a dynamic database accumulating uncertain data. Once such a database is generated, managers may use OLAP and data mining techniques to make better decisions.

Role of DSS in Business Management

Many organizations recognize the effects of unsatisfactory forecasting. Forecasting based on uncertain information requires fairly subjective assessments of domain dependencies and relationship strengths, and tends to be inconsistent. In the words of Foote & Krishnamurthi (2001):

> It can be said that even today forecasting process is generally fairly subjective, driven by intuition of so called “experts” who are company executives, sales force, and industry analysts whose prognostications have been far from

> satisfactory. As a result, companies can miss the boat on achieving profitability, reliability, and competitive advantage in their industries (p. 1).

Information Systems research has focused on developing systems that can enhance the decision-making process and profitability, which have been improved with Data Warehouse/DSS at Wal-Mart (Foote & Krishnamurthi, 2001). Improved business management, profitability, and decision-making at Wal-Mart have been attributed to DSS use.

OLAP, Data Warehouses, and Decision Support Systems

Because managers are faced with making decisions under conditions of uncertainty and huge volumes of data they need Decision Support Systems to make optimal decisions (Inmon, 1990). Decision Support Systems that cannot handle large volumes of data containing uncertainty are less useful for decision-making. Data representation and uncertainty representation are crucial parts of these Decision Support Systems (Moole, 2003). Products that enable organizations to represent and manipulate huge volumes of data are referred to as *data warehouse, OLAP, business intelligence,* and *decision support systems* products by various vendors. These terms are used interchangeably in this study. Currently, these products lack a standard conceptual data model for supporting data representation and operations. They also lack a framework to represent uncertainty, modify uncertain data, and perform imprecise queries. Conceptual models and frameworks supported by theories founded in mathematics enable users of product to understand better the claims of product manufacturers (Codd et al., 1993). They also enable researchers to contribute independently to technology (Agarwal et al., 1997; Thomas & Datta, 2001).

The relational data model uses a Table, a two-dimensional data structure, as its primary data structure (Date, 2003). Traditional relational database management systems technology is unsuitable for OLAP because of queries involved (aggregate, summary, grouping, etc.). According to industry reports, OLAP product sales reached $9 billion in 1997 (Pendse, 2003). The market for these technologies is growing rapidly. Current commercial products offering some features of DW and OLAP include *Red Brick* from Red Brick Systems, *EssBase* from Hyperion, *Express* from Oracle, and *IQ* from Sybase. These systems do not have standard conceptual data models; they are ad hoc extensions of RDBMS technology (Thomas & Datta, 2001). Because of this, extension to products and development of tools is limited to each proprietary product, inhibiting growth of this segment of the market, and requiring use of multidimensional data models (Codd et al., 1993; Thomas & Datta, 2001).

Multidimensional Data Models

Researchers have proposed several data models based on multiple dimensions, referred to as multidimensional data models. Most of them are based on a concept called *data cube* (Codd et al., 1993). Data cubes are primary data structures in OLAP and DW products. Thomas and Datta (2001) proposed one such conceptual multidimensional data model. Advantages of this model include its theoretical framework and associated algebra, which is relationally complete, consistent, and closed. Moole (2003) proposed several enhancements to this model. This enhanced model is referred to as *Probabilistic Multidimensional Data Model* (PMDDM), with its most important enhancement being the addition of a framework based on probability theory to handle uncertainty—an important category of OLAP functionality requirements (Moole, 2003). Other enhancements include uncertainty-related algebraic operations. The model did not provide much required data modification framework or any analysis of its efficiencies. This represented an important area for further research on PMDDM (Moole, 2003).

Problem Statement

Many of today's business leaders make decisions by extensive use of huge volumes of real-world data, which contain uncertainty and change over time. Any decision support system utilized by enterprises should be able to handle efficiently large amounts of data, as well as uncertainty and modifications to uncertain data. RDBMS product vendors provide several extensions and features to support these requirements, but these extensions lack the support of conceptual models, impeding growth of software product market and increasing cost of DSS solutions to business. Recently, researchers focused on this problem, and Moole (2003) proposed a probabilistic multidimensional data model; however, this model lacks the framework for probabilistic data modification. Lack of a framework to modify data diminishes importance of data models and their usefulness (Dey & Sarkar, 2000; Moole, 2003). The purpose of this study was to develop a framework for probabilistic data modification to enhance importance and usefulness of probabilistic multidimensional data model.

Research Question

The data modification framework enhances the underlying model (Dey & Sarkar, 2000). In order to provide maximum benefit and acceptance, it should be closed, complete, and consistent with the underlying model (Date, 2000; Dey & Sarkar, 2000; Klir & Wierman, 1998; Pearl, 1988). Therefore, the research question for this study is: Given the probabilistic multidimensional data model, what data modification framework and algorithm can update uncertain data consistent with the model (*consistent*), resulting in valid data (*closed*) and being reliable in all possible update scenarios (*complete*)?

Consistency, completeness, and closure properties of a framework or an algorithm are important. They show that the framework or algorithm can be used for a data model without resulting in unusable data: (a) The consistency property for PMDDM assures that axioms of probability theory are satisfied; (b) the completeness property ensures that the algorithm can be used in all possible modification scenarios; and (c) the closure property means that the algorithm produces only valid objects (as defined in model definition) for this model.

Significance of the Study

The solution sought by the investigator is significant for several reasons:

1. First of all, a $4 billion market is not achieving its potential (Pendse, 2003);
2. The solution will contribute to data models research and the knowledge base and may result in better DSS tools for business;
3. The solution may help standardize multidimensional database products and related tools;
4. Such standardization can facilitate widespread adoption of these products and tools by business as happened in case of relational databases (Date, 2000); and
5. Utilization of multidimensional databases can enhance the decision-making process of managers.

Social Change

This research may help standardize the OLAP/Data Warehousing software products. Standardized products are easier to understand than custom-built and proprietary products. Product standardization also leads to cheaper products, which leads to higher utilization of the products (Thomas & Datta, 2001). DSS products developed as a result of this research may reduce overall cost of ownership of DSS products to business. Utilization of products based on this research may lead to efficiency of operations due to enhanced decision-making, leading to cheaper products and services to the customers. Reduced prices of goods and services for consumers improve the standard of living and enhance the quality of life (Gairdner, 2000).

Purpose of the Research

The current probabilistic multidimensional data model lacks a framework to update uncertain data (Moole, 2003). Modification of data should be *consistent* (satisfies probability axioms and new beliefs), *complete* (covers all possible modification situations), and *closed* (use of update mechanism results in valid cubes). The purpose of this research was to develop an uncertain data modification framework that is provably consistent, complete, and closed to modify existing probabilistic multidimensional data.

Proposed Research

This study enhances the probabilistic multidimensional data model (Moole, 2003), providing all required algebraic operations as well as a framework for updating probabilistic data. The investigator developed algorithms to update data, analyzed time and space complexity of update algorithms, described an application of this conceptual data model in managerial decision-making, and compared and contrasted it with Bayesian belief networks. These research activities have not previously been performed, to the best of the investigator's knowledge. This research contributes to both Decision Support Systems research and research conducted in uncertainty in artificial intelligence (UAI), which focuses mostly on Bayesian belief networks. The solution is significant in that it removes a major impediment to developing products (Thomas & Datta, 2001; Vassiliadis & Sellis, 1999).

Scope of Research and Delimitations

The following fall within the scope of this research:

1. Integrating the multidimensional data model (MDD) and probabilistic data model and providing all required algebraic operations for the probabilistic MDD model.
2. Providing a comprehensive (*complete*, *consistent*, and *closed*) framework to update probabilistic data.
3. Developing algorithms to update data.
4. Analyzing time and space complexity of updated algorithms.
5. Identifying an application of this conceptual data model in managerial decision-making.
6. Comparing and contrasting this conceptual data model with Bayesian belief networks.

This study will not address the physical data model, implementation of any part of the model, implementation issues, comparison with, or discussion of, proprietary non-published products, or any other aspects not listed in this section.

Assumptions

The most important general assumption is that the probability stamp *pS* is the joint probability of a set of mutually independent variables. Without this assumption, applicability of some formulas would be questionable. Semantics of probability are assumed to conform to Bayesian probability theory. Data modification framework is devised with the assumption that Jeffrey's rule of probability kinematics is applicable to probability distributions in the domains of application. Any additional assumptions specific to a formula will be stated with that formula.

Barriers

The need for multidimensional data models arose when businesses began to accumulate terabytes of data (Kimball et al., 1998). Researchers started addressing functionality requirements demanded by business, looking for a solution that would satisfy four categories of functionality requirements (Moole, 2003). This research may not have been useful a decade ago due to the state of technology at that time, which could not store huge volumes of data economically (Inmon, 1990). Current market conditions show demand for OLAP and DW products (ITG, 2000). This research presents few apparent barriers for researchers with extensive training and experience in the DW and OLAP fields. This research does not require collection of data, as in quantitative studies, nor does it require interviews with other people, as in qualitative studies. There were no apparent barriers for the investigator to complete the research as described in the scope section of this paper.

Limitations

This study is analytical in nature and hence limitations are generally due to interpretations, logical errors, and semantics. Substantiating claims by being thorough in developing formulas and by adhering to well-established conceptual frameworks can minimize the impact of these limitations on results. The limitations on applicability of the research results to industry will be due mainly to lack of availability of the results to the public. This may be overcome by publishing the results in peer reviewed journals and at conferences related to the area of research.

Definitions

For the purpose of this study the following operational definitions will be used. These definitions are used throughout this dissertation. Terms such as technology, products, and tools are used in their general sense as they relate to the software industry. Chapter 2 presents definitions of additional terms used in that chapter.

Algorithm: A sequence of instructions in solving a problem or achieving a goal.

Closure: Use of an algorithm that produces valid results.

Completeness: An algorithm that handles all possible update scenarios correctly.

Consistency: An algorithm that satisfies all probability axioms.

Efficiency: Algorithms measured by their time complexity and space complexity.

Framework: An abstract solution to a number of related problems, which specifies abstract boundary conditions consisting of concepts, assumptions, values, and practices, within which all solutions lie.

Methodology: "A set or system of methods, principles, and rules used in a given discipline" (Steinmetz, 1997, p. 203).

Model: A set of mathematical equations describing domains, constraints, and axioms.

Time complexity: A measure of efficiency that specifies how many units of computational cycles are required to execute steps in algorithm.

Space complexity: A measure of efficiency that specifies how many units of storage are required to execute steps in algorithm.

Summary

In Chapter 1, data warehousing, OLAP, and data models were discussed. The problems faced by business today, due to the large amounts of data generated, were also discussed. The research problem, scope of research, barriers, and limitations were presented.

In Chapter 2, the research work done in this area and a brief description of the basis for current research will be presented. The investigator reviews prior research on uncertainty, data models, multidimensional data models, and probabilistic multidimensional data models, connects prior research to the problem statement, and briefly describes the proposed solution.

In Chapter 3, the research methodology used in prior research, a framework for selection of research methodology for this study, and justification for its selection will be presented. Finally, advantages and disadvantages of selected research methodology and ways to mitigate impact of its disadvantages will be discussed.

In Chapter 4, modifications to probabilistic multidimensional data model definition, additional required algebraic operations, a data modification framework based on Bayesian framework, a data modification algorithm, and time and space complexity analysis are presented. A solution using probabilistic multidimensional data model for a business management problem is also described.

In Chapter 5, research activities performed by the investigator are summarized. Summarization of analytical methodology and mitigation of its disadvantages, summarization of research results, and areas for further research are also presented in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

The purpose of this study was to develop a framework for probabilistic data modification to enhance importance and usefulness of the probabilistic multidimensional data model of Moole (2003). The investigator reviewed literature addressing uncertainty, multidimensional data models, and probabilistic multidimensional data models. A data modification framework based on earlier research is also presented in this chapter. Also included is research in uncertainty and a review of multidimensional data and probabilistic multidimensional conceptual data models. The review of literature is mainly focused on peer-reviewed journals (for example IEEE, ACM, and INFORMS Database), peer reviewed conference proceedings (such as IEEE Conferences and ACM SIG Conferences), University of Maryland Digital Dissertations Database, and, to some extent, business journals and Internet web sites.

Uncertainty Representation

Uncertainty is pervasive. An effective probabilistic multidimensional data model must take into account modification of uncertain data. In this section, a review literature on uncertainty will be presented.

There are three types of basic methods representing uncertainty (Turban & Aronson, 2001):

1. Numeric methods represent uncertainty using a scale with two extreme numbers. For example, complete certainty could be represented as 100

while complete uncertainty is represented as 0. Probability is another example of numeric methods.

2. Graphical methods represent uncertainty as a continuum on a scale.
3. Symbolic methods generally represent uncertainty as a rank, fuzzy logic being a special method of symbolic logic combined with numbers.

There are various frameworks to represent uncertainty (Klir & Wierman, 1998). They are: *Classical set theory, fuzzy set theory, evidence theory, possibility theory,* and *probability theory*. In the probabilistic multidimensional data model probability measures are used to represent uncertainty.

Classic Set Theory

In classical set theory sets are basic building blocks. There is no precise definition of set, but Klir & Wierman's (1998) definition: a "set is a collection of objects chosen from Universe" generally suffices (p. 14). Examples of sets are natural numbers, integers, carnivores, and empty sets. Uncertainty is expressed by specifying membership in a set. Each set is inherently non-specific. Specificity decreases as number of members in set increases. Only when set contains one alternative is full specificity achieved. Classical set theory imposes several limitations compared to probability theory, discussed in detail by Klir & Wierman.

Fuzzy Set Theory

Sets in classical set theory are also called *crisp sets*. An element is either a member of a set or it is not, although the fuzzy set theory developed by Zadeh (1965) specifies degree of membership, as opposed to being a member or non-member. Later developments of fuzzy set theory resulted in fuzzy logic, which formalized information representation (Zadeh, 1965). Representing the degree of membership provides a richer mechanism to represent uncertainty. Fuzzy logic is a relatively recent development compared to probability theory.

Evidence Theory

Originally published in 1976 by Glenn Shafer, evidence theory, popularly known as *Dempster-Shafer theory (DST)*, uses belief functions to represent uncertainty (Shafer, 1990). In this theory, which is sometimes also referred to as *mathematical theory of evidence (MTE)*, probability is a belief function. This theory is a generalization of Bayesian theory. DST is compatible with probability theory (Dezert, 2002).

Possibility Theory

Possibility theory is related to fuzzy set theory and probability theory. A possibility distribution is a fuzzy set. This theory can also represent nonspecificity as a measure, and is similar to entropy in probability theory (Dubois & Prade, 1988). According to this theory, degree of possibility is independent of human beliefs, and exists in physical world. On the other hand, degrees of belief are subjective-personal opinions and result from the limitation of the human knowledge.

Probability Theory

Probability theory is of particular interest, as it is the most frequently used framework to represent, model, and manipulate uncertainty arising in day-to-day business decision-making. Probabilistic analysis of data to derive expected results is employed in several Decision Support Systems. This kind of probabilistic analysis is better than ignoring or avoiding uncertainty (Turban & Aronson, 2001).

Among all frameworks developed to handle uncertainty, probability theory is most developed, well understood, and the most accepted theory (Rao, 1973). In fact, it is the oldest theory to represent uncertainty and is a part of daily conversations. Probability theory uses numbers between 0 and 1 to represent uncertainty. Complete certainty is represented by 1 and complete uncertainty by 0 (Rao, 1973). Bayesian probabilistic theory extends theoretical underpinnings of probability. Causality and decision-making processes described by Pearl (2001) are based on this theory. Due to its strengths in terms of conceptual clarity and acceptance, the investigator used probability theory to handle uncertainty.

Multidimensional Data Models

Data representation frameworks have evolved from two-dimensional structures based on relational data model (Codd, 1971) to multidimensional data models of today. This study is concerned with handling huge volumes of data, for which multidimensional data models are well suited. The earliest attempt at providing a conceptual model for multidimensional data was made by Li and Wang (1996). In 1997, Agarwal, Gupta, and Sarawagi (1997) provided one model for multidimensional data and Gyssens and Lakshmanan (1997) provided another. All of these models placed several restrictions on dimensions, attributes, or types of queries. Thomas and Datta (2001) eliminated most of these restrictions and made their model more generic; however, this model lacked the capacity to represent, manipulate, and update uncertainty. The probabilistic multidimensional data model proposed by Moole (2003) is an enhancement to the Thomas and Datta (2001) model.

Probabilistic Multidimensional Data Model

Probabilistic multidimensional data models handle uncertainty in addition to large volumes of data. They are developed to meet the functionality requirements of DW/OLAP. The functionality requirements of DW/OLAP models are divided into the following four categories (Moole, 2003):

Data Cube Operations

In this category, slice, dice, roll-up, drill-down, and pivot operations are the most important.

1. Slicing is operation of selecting dimensions used to view the cube. It is analogous to selection operation in relational algebra.
2. Dicing is operation of selecting actual positions or values on a dimension. It is analogous to projection operation in relational algebra.
3. Roll-up is operation of increasing granularity along one or more dimensions. For example, an analyst with access to sales in a city may want to see sales for an entire state or region to view the city in proper perspective. That is, the roll-up operation allows analysis across a hierarchy of dimension.
4. Drill-down is converse operation, decreasing granularity. An analyst with access to regional sales data may want to see more detailed data for a state and then for a city. It is traversing dimensional hierarchy in decreasing level of granularity.

5. Pivot refers to aggregation of two or more dimensions to produce a new multidimensional view having an attribute for each grouping dimension and additional attributes for aggregated measure.

Aggregation

The second category of functionality requirements is *aggregation*. In addition to standard SQL aggregate operators (e.g. MIN, MAX, SUM, AVG, COUNT), an OLAP system needs to support operators such as ranking, percentiles, comparisons of aggregates, attribute-based grouping, trends, and time-dimension based aggregate comparisons.

Transformations

The third category of functional requirements is *transformations*. Force operator converts a dimension to a measure and extract operator converts a measure into a dimension.

Uncertain Data

The fourth category is related to handling of *uncertain data*. This category includes a well-defined mechanism for representing, modifying, and transforming uncertain data consistently within the model as well as in associated operations. Without supporting uncertainty, usefulness of OLAP systems will be limited to the point of being unacceptable for many real world business tasks.

Moole (2003) proposed also a probabilistic multidimensional data model which captured uncertainty using probability measures. This model was based on set theory, and had a solid theoretical basis for representing uncertainty: Its algebra was closed, it was at least as expressive as the relational model, and it was relationally complete. This model did not provide any mechanism for updating uncertain data. The following discussion provides a summary of the probabilistic multidimensional data model followed by discussion of enhancements proposed by this dissertation research.

In the probabilistic multidimensional data model, each cell in the cube is stamped with a probability measure, as shown in Figure 1. This probability stamp *pS* represents strength of belief that there exists a real world object with given cell values. It can represent also probability derived from forecasting methods or empirical experimentation or it can be considered belief strength for conceptual clarity and wider applicability. Since the investigator is using probability as the measure of strength of belief, its domain is [0,1]. When *pS* is 0, it is certain that the real world object does not exist and when it is 1, it is certain that it does exist. When it is 0, the cell values are not represented for that object. When it is 1, the investigator will not write *pS* explicitly in cube cell. Content for this section is adopted verbatim from Moole (2003), with only essential parts reproduced. Moole (2003) uses the s*ales* cube shown in Figure 1 as a running example throughout.

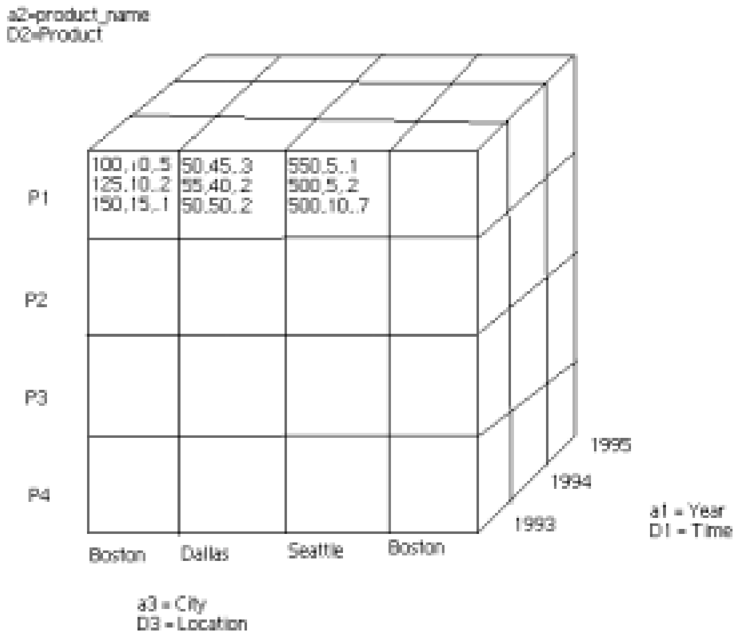

Figure 1. Cube example.

The following text is verbatim from the paper:

Definition 1 Cube: A cube is a logical structure comprising of a six-tuple ⟨*C, A, f, d, O, L*⟩ where:

- *C* is a set of *m* characteristics {c_1, c_2, …, c_m} where each c_i is a characteristic having domain (dom) C, one of which may be BELIEF. If BELIEF is not a characteristic, then the cube is deterministic.

- *A* is a set of *t* attributes {a_1, a_2, …, a_m} where each a_i is an attribute name having domain dom A, one of which may be a probability stamp *pS*. dom(pS) is (0,1]. We assume that there exists an arbitrary total order on A, $\leq_A$. Thus, the attributes in A (and any subset of A) can be listed according to $\leq_A$. Moreover we say that each $a_i \in$ A is *recognizable* to the cube C.

- f is a one-to-one mapping, f: C → 2^A, which maps a set of attributes to each characteristic. The mapping is such that
  - $\forall$i , j, i $\neq$ j, $f(c_i) \cap f(c_j) = \varnothing$ i.e. pairwise disjoint attribute sets
  - $\forall$x, x $\in$ A, $\exists$c, c $\in$ C, x $\in$ f(c) i.e. all attributes are mapped
  - f(BELIEF) → {pS}, *iff* BELIEF $\in$ C i.e. BELIEF is always mapped to pS

Hence, f partitions the set of attributes among the characteristics. f(c) is referred to as the *schema* of c.

- d is a boolean-valued function that partitions C into a set of dimensions D and a set of measures M. Thus, C = D ∪ M where D ∩ M = ∅. The function d is defined as:

$$\forall x \in C , d(x) = \begin{cases} 1 & \textit{if} \quad x \in D , \\ 0 & \textit{otherwise} \end{cases} .$$

- O is a set of partial orders such that each $o_i \in$ O is a partial order defined on $f(c_i)$ and |O| = |C|. In other words, the schema for each characteristic c_i, has a partial order o_i associated with it.

- L is a set of cube cells. A cube cell is represented as an ⟨address, content⟩ pair.
  - The address in this pair is an *n*-tuple, $\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$, where *n* is the number of dimensional attributes in the cube, i.e. n = $|A_d|$, where A_d

represents set of all dimensional attributes; i.e., $A_d = \bigcup_{d_i \in D} f(d_i)$. Each address component, α_i, represents a position along the "axis" of a dimensional attribute in A based on $\leq_A$ (e.g., the third component of the address, α_3, corresponds to the third dimensional attribute in A in $\leq_A$-order). One of these components may be pS.

- The content of a cube cell is a *k*-tuple, $\langle \chi_1, \chi_2, \ldots, \chi_k \rangle$, where *k* is the number of metric attributes in the cube, i.e., $k = |A_m|$, where A_m represents the set of all metric attributes; i.e. $A_m = \bigcup_{m_i \in M} f(m_i)$. Each content component, χ_i, represents the element of the content that corresponds to a particular metric attribute. χ_i corresponds to the *i*th metric attribute in A in $\leq_A$-order. One of these components may be pS.

- The total probability of all the cells having the same address component must be no more than one. i.e.

$$\forall l \in L, \sum_{\substack{y \in L \\ l.AC = y.AC}} y.CC(pS) \leq 1.$$

- Two cells *i* and *j* are said to be ***value-equivalent*** iff the address component of i is identical to the address component of j and the content component of i without pS is identical to the content component of j without pS, when d(BELIEF)=0. Value-equivalence is denoted by $\cong$. That is (see below for the notation),

$$i \cong j \Leftrightarrow \begin{cases} i \in L, \\ j \in L, \\ \textit{if } BELIEF \in C \textit{ then } d(BELIEF) = 0, \\ i.AC = j.AC \wedge i.CC - \{pS\} = j.CC - \{pS\} \end{cases}$$

Value-equivalent cells are not allowed and must be coalesced using the coalescence operations defined in the next section.

The following notations are used:

- g: A → C, such that g(a) = c iff a ∈ f(c)
- structural address component of L is denoted as L.AC
- structural address component of cell *l* is denoted as *l*.AC
- *i*th address component of cell l is denoted as *l*.AC[*i*]
- address component of a cell corresponding to an attribute name *aname* is denoted as l.AC(*aname*)
- structural content component of L is denoted as L.CC
- structural content component of cell *l* is denoted as *l*.CC
- *i*th content value component of cell *l* is denoted as *l*.CC[*i*]
- content component of a cell corresponding to an attribute name *aname* is denoted as l.CC(*aname*)
- *class of L* is defined as $\{A_d \cup A_m\}$
- *object* is a set of domain values corresponding to the set of attributes $\{A_d \cup A_m - pS\}$ (all the attribute values of a cell without their joint probability). A subset of these attributes is called as *partial object*. An object is an instance of a class of L.

Example for the Probabilistic Multidimensional Data Model

To clarify the above definition of the probabilistic multidimensional data model, Moole used the data cube example shown in Figure 1. Note that multiple content components are written into a single box for convenience and separating them each into their individual cells does not affect as long as their address components are properly represented. This **Sales** cube represents the data collected by our fictitious Koke company for the sales of fictitious competitor Bepsi's products. Since it is not possible to get the exact sales information, Koke's agents are allowed to report a guess based on empty cans being recycled and attach a probability measure to each report.

The sales cube has:

- The characteristic set C = {TIME, PRODUCT, LOCATION, SALES, BELIEF}, (m = 5)
- The attribute set A = {day, week, month, year, product_name, size, store_name, city, state, region, amount, quantity, pS}, (t = 14)
- schema of C:
    - f(TIME) = {day, week, month, year}
    - f(PRODUCT) = {product_name, size}
    - f(LOCATION) = {store_name, city, state, region}
    - f(SALES) = {amount, quantity}
    - f(BELIEF) = {pS}
- dimension function d:
    - d (TIME) = 1 i.e., TIME is a dimension

  - d (PRODUCT) = 1 i.e., PRODUCT is a dimension
  - d (LOCATION) = 1 i.e., LOCATION is a dimension
  - d (SALES) = 0 i.e., SALES is a measure
  - d (BELIEF) = 0 i.e., BELIEF is a measure

- A sample partial order on the Sales cube is as follows:
  - O_{TIME} = {⟨day, week⟩, ⟨day, month⟩, ⟨day, year⟩, ⟨week, month⟩, ⟨month, year⟩}
  - $O_{PRODUCT}$ = {⟨product_name, size>}
  - $O_{LOCATION}$ = {⟨store_name, city⟩, ⟨city, state⟩, ⟨state, region⟩}
  - O_{SALES} = {}
  - O_{BELIEF} = {}
- An example of L is as follows:
  - Let us assume the following domains for the attributes
    - A = {year, product_name, city, amount, quantity, pS}
    - dom year = {1993, 1994, 1995}
    - dom product_name = {P1, P2, P3}
    - dom city = {Boston, Dallas, Seattle, Chicago}
    - dom amount = {0, 1, 2, …}
    - dom quantity = {0, 1, 2, …}

- dom pS = {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

- With the above assumptions, an example set of cells is shown below.

l = ⟨l.AC, l.CC⟩

l.AC = ⟨1993, P1, Boston⟩ corresponding to ⟨year, product_name, city⟩

l.CC = ⟨100, 10, 0.5⟩ corresponding to ⟨amount, quantity, pS⟩

Therefore,

l = ⟨⟨1993, P1, Boston⟩, ⟨100, 10, 0.5⟩⟩

This cell represents the Probability of "Sales of P1 by Bepsi in Boston for 1993 are 10 shipments with a revenue of 100k" is 0.5.

Similarly, the values shown in box one of the Figure 1 can be written as:

{ ⟨⟨1993, P1, Boston⟩, ⟨100, 10, 0.5⟩⟩,

⟨⟨1993, P1, Boston⟩, ⟨125, 10, 0.2⟩⟩,

⟨⟨1993, P1, Boston⟩, ⟨150, 15, 0.1⟩⟩ }

In the next section, an operational model is provided for the above data model.

Algebra for Probabilistic Multidimensional Data Model

The content of this section is adopted verbatim from the paper (Moole, 2003). As was mentioned in the Cube definition in the previous section, value-equivalent cells must be coalesced. There are two types of coalescence operations defined on the value-equivalent cube cells. Both these coalescence operators can be applied recursively on any number of value-equivalent cells.

coalescence-PLUS (⊕): This operator is used in the definition of the projection operation and is defined on two value-equivalent cells x and y as:

$z = x \oplus y \Leftrightarrow (x \cong y) \wedge (z \cong x) \wedge (z.CC(pS) = \min\{1, x.CC(pS) + y.CC(pS)\})$

Intuitively, when two value-equivalent cells are combined using projection operation (i.e. we believe in both of the cells) their individual probability is summed together. If the result is greater than 1, then existence of the object is certain and is assigned a probability of 1. Recursive application is denoted as:

$$\bigoplus_{i=1}^{m} x_i = (...((x_1 \oplus x_2) \oplus x_3) \oplus ... \oplus x_{m-1}) \oplus x_m$$

coalescence-MAX (⊙): This operator is used in the definition of the union operation and is defined on two value-equivalent cells x and y as:

$z = x \odot y \Leftrightarrow (x \cong y) \wedge (z \cong x) \wedge (z.CC(pS) = \max\{x.CC(pS), y.CC(pS)\})$

Intuitively, when two value-equivalent cells are combined using union operation (i.e., we believe in only one of the cells, the one with higher strength of belief), maximum probability of these cells is the probability for the result cell. Recursive application is denoted as:

$$\bigodot_{i=1}^{m} x_i = (((\ldots(x_1 \odot x_2) \odot x_3)\ldots\odot x_{m-1}) \odot x_m$$

To denote coalescence performed on cells with value-equivalence defined over an attribute subset S, we write "⊕ over S" or "⊙ over S".

We will proceed to define our algebra operators after providing two more definitions.

Predicate P: A predicate is a well-formed formula in first-order predicate logic.

- An *atomic predicate* is a restriction on the domain of a single attribute or characteristic. e.g. (year = 1994)
- A *compound predicate* is a logical expression of atomic predicates. The logical operators are ∧ (and), ∨ (or), ¬ (not), → (implies), and ↔ (equivalent to). It is of the form: P = p_1 ⟨op⟩ p_2 ⟨op⟩ … ⟨op⟩ p_n. e.g. (year = 1994) ∧ ((quantity < 15) ∨ (amount > 100))

***l* satisfies P**: *l*, an instance of L, with the structure <address, content> satisfies predicate P if and only if:

Case 1: if an element of *l* is a dimension, then *l*.AC satisfies P, otherwise *l*.CC satisfies P, if P is atomic and the truth-value is TRUE.

$$P(l) = TRUE \begin{cases} a\ in\ P, a \in f(d_i), d_i \in D, P(l.AC[a]) = TRUE \\ OR \\ a\ in\ P, a \in f(m_i), m_i \in M, P(l.CC[a]) = TRUE \end{cases}$$

e.g. Upper left most corner cell in the cube of Figure 1 satisfies P=(year=1993)

Case 2: if P is a compound predicate, *l* satisfies P, when all the truth-values evaluated together with the connecting operators results in TRUE.

$$\forall p_i \in P, a\ in\ p_i, p_i(l) = Q_i \quad P(l) = Q_1 \langle op \rangle Q_2 \langle op \rangle ... \langle op \rangle Q_n$$

e.g. Upper left most corner cell in the cube of Figure 1 satisfies

P=(year=1993)∧(city=”Boston”)∧(product_name=”P1”)

Fuzzy membership functions: We can also define a fuzzy membership function for BELIEF characteristic, which maps between natural language sentences and probability. For example, concepts like probably, likely, most likely, certainly, etc. can be mapped to the probability numbers between 0 and 1 using the fuzzy membership functions. These functions can also be used to map probability to all natural language words describing the uncertainty. The process of fuzzification and defuzzification is performed to convert human terminology of uncertainty into a crisp probability measure and vice-versa. Such a function is strictly a helper function and is not part of the model or algebra, as our model for uncertainty is based on probabilities. This function is generally applied for restricting the cells with a desired strength of belief to appear in the output. As an example, let us assume that the query is “Select most likely maximum sales from Sales cube”. Let us also assume a fuzzy membership function defined for this cube maps fuzzy sets “certain”, “most likely”, “very likely”, “likely”, “unlikely”, and “very unlikely” to crisp sets 1.00, 0.99-0.70, 0.75-0.55, 0.60-0.40, 0.45-0.25, and 0.30-0.00. Of course, these are graded memberships, so a formal definition of these fuzzy sets will elaborate the membership gradation very clearly using alpha-cuts, height, plimth and other properties for fuzzy sets. Using this mapping we determine that “most likely” is described with a strength of belief between 0.99 – 0.75. Therefore, our selection predicate can be formed to include “pS >= 0.75 and pS < 1.00”. This selects cells with probability greater than 0.75. Among the resultant cells, we can select the cell with maximum quantity.

Similarly, we can combine the probability distributions for each object with probability distributions of other objects when calculating aggregate values and assign a probability distribution to the result. By doing this, instead of answering a query like “what is the mean sales for Chicago?” with a pointed answer like “25”, we can answer using a confidence-interval statement like “The mean sales for Chicago is 95% certain to be between 23 and 27”. A comprehensive treatment of the probabilistic data can be found in Klir and Wierman (1998) and Pearl (1988). These are some simple examples to demonstrate the power of probabilistic multi-dimensional data.

Now, we define algebra operations for the probabilistic data cube. Each operator is presented in the format used by Thomas and Datta, as follows: the operator name, symbol, a textual description, input, output, mathematical notation, and a simple example of the operator. All the examples use the **Sales** cube shown in Figure 1.

Restriction ($\sum$): The *restriction* operator restricts the values on one or more attributes based on specified conditions, where a given condition is in the form of a predicate. This is similar to the selection operator in relational algebra. Only cells that satisfy the predicate are retrieved into the result cube. If there are no cells that satisfy P, the result is an empty cube. Note that pS may also be restricted in the predicate, thus selecting cells representing only real world objects that have satisfied the belief constraints. This operator can be applied multiple times. The order of application is not significant.

The algebra of restriction operator is defined as follows:

Input: A cube $C_I = \langle C, A, f, d, O, L\rangle$ and a predicate P

Output: A cube $C_O = \langle C,A,f ,d, O, L_O\rangle$ where $L_O \subseteq L$ and $L_O=\{l \mid (l\in L)\wedge(l$ satisfies P$)\}$.

Mathematical Notation: $\sum_P(C_I)=C_O$

A Simple Example: If we want to know the sales for P1 in Boston during the year 1993, then we use $\sum_{(\text{year}=1993 \wedge \text{product\_name}=\text{'P1'} \wedge \text{city}=\text{'Boston'})}(\text{Sales}) = C_{\text{Restrict}} =$

$$\{ \quad \langle\langle 1993, \text{P1}, \text{Boston}\rangle, \langle 100, 10, 0.5\rangle\rangle,$$
$$\langle\langle 1993, \text{P1}, \text{Boston}\rangle, \langle 125, 10, 0.2\rangle\rangle,$$
$$\langle\langle 1993, \text{P1}, \text{Boston}\rangle, \langle 150, 15, 0.1\rangle\rangle \qquad \}$$

Metric Projection ($\prod^{\mathbf{M}}$): The *metric projection* operator restricts the output of a cube to include only a subset of the original set of measures. This is similar to the projection operator in the relational algebra. Let S be a set of project metric attributes such that $S \subseteq A_m$. Then the output of the resulting cube includes only those measures in S. Since our cell represents a joint distribution of the attributes and this operation results in a subset of the original attributes, we need to marginalize the probabilities. We use the coalescence-PLUS (⊕) operator for this. Note that the value-equivalence is over the set of attributes S and projecting out the pS itself (i.e. $pS \notin S$) may yield meaningless result.

The algebra of metric projection is defined as follows:

Input: A cube $C_I = \langle C, A, f, d, O, L\rangle$ and a set of projection attributes S

Output: A cube $C_O = \langle C, A_O, f_O, d, O, L_O\rangle$ where,

$A_O = S \cup A_d$,

$f_O: C \rightarrow 2^{A_O} \mid f_O(c) = f(c) \cap A_O$,

$L_O = \{l_O \mid \quad \exists\, l \in L$,

$l_O.AC = l.AC$,

$l_O.CC = \langle l.CC[s_1], l.CC[s_2], \ldots, l.CC[s_3]\rangle$

where $\{s_1, s_2, \ldots, s_3\}$=S and

⊕ over S

l∈L

Mathematical Notation: $\prod_{S}^{M}(C_I) = C_O$

A Simple Example: If we are interested in selecting only the quantity from the previous Restriction operator output above, we use $\prod_{quantity}^{M}(C_{\mathrm{Re}\,strict}) = C_{\mathrm{Pr}\,oject}$ =

{ ⟨⟨1993, P1, Boston⟩, ⟨10, 0.7⟩⟩,

⟨⟨1993, P1, Boston⟩, ⟨15, 0.1⟩⟩ }

Note that the result contains only one cell with coalesced pS for the quantity=10 because there are two cells for that and they become value-equivalent when amount is projected out. Also, note that eliminating pS through this operation for this example would result in cells with identical address components, but different quantities (with belief strength implicitly 1), which is meaningless data.

Rename (*Λ*): The rename operator renames a set of elements. It is similar to the rename operator in relational algebra. Let S_I be some set of elements $\{s_{11}, s_{12}, \ldots, s_{In}\}$. Then, $\mathbf{\Lambda}_S(S_I) = \{S.s_{11}, S.s_{12}, \ldots, S.s_{In}\}$. This operator can be used to eliminate duplicate names in the results of binary operations. For example, Renaming the attributes corresponding to the TIME dimension of the Sales cube can be expressed as follows: $\mathit{\Lambda}_{\mathrm{Sales}}(\mathrm{A})$ = {Sales.day, Sales.week, Sales.month, Sales.year}

Cubic Product (⊗): The *Cubic Product* operator is a binary operator. It is used to relate two cubes. This operator joins the attributes of two cubes together. This operator is similar to the Cartesian Product operator in the relational algebra. The probability of the result is obtained by multiplying the probabilities of joined cells. By noting that the probability of each cell is the joint probability for that set of attributes, we can see that the result set is union of both tuples and the result cell's probability must be a joint probability of all the attributes together. We can also see that the resulting probability is meaningful only when all the attributes are mutually independent. The Cubic Product is defined as follows:

Input: A cube $C_1 = \langle C_1, A_1, f_1, d_1, O_1, L_1 \rangle$ and A cube $C_2 = \langle C_2, A_2, f_2, d_2, O_2, L_2 \rangle$.

Output: A cube $C_O = \langle C_O, A_O, F_O, d_O, O_O, L_O \rangle$, where (• denotes concatenation)

$C_O = \Lambda_{C1}(C_1) \cup \Lambda_{C2}(C_2)$,

$A_O = \Lambda_{A1}(A_1) \cup \Lambda_{A2}(A_2)$,

$L_O = \{l_O \mid \exists\, l_1, \exists\, l_2, l_1 \in L_1, l_2 \in L_2, l_O.AC = l_1.AC \,.\, l_2.AC,$

$l_O.CC = l_1.CC\text{-}\{l_1.CC(pS)\} \bullet l_2.CC\text{-}\{l_2.CC(pS)\},$

$l_O.CC(pS) = \{l_1.CC(pS) * l_2.CC(pS)\}$

In addition,

$\forall_{ci} \in (C_1 \cup C_2)$,

$f_O = f_1$ when applied to $c_i \in C_1.c_i$, f_2 when applied to $c_j \in C_2.c_j$,

$\forall_{ci} \in (C_1 \cup C_2)$,

$d_O = d_1$ when applied to $c_i \in C_1.c_i$, d_2 when applied to $c_j \in C_2.c_j$

$\forall_{ai} \in (f(C_1) \cup f(C_2))$,

$O_O = O_1$ when applied to $a_i \in f(C_1)$, O_2 when applied to $a_j \in f(C_2)$

Mathematical Notation: $C_1 \otimes C_2 = C_O$

A Simple Example: Suppose we have another cube, Discount, containing discount amounts for various combinations of product and city. The definition of Discount cube is characteristics C = {PRODUCT, LOCATION, DISCOUNT, BELIEF}, attributes A = {product_name, city_ID, amount, pS}, dimensions D = {PRODUCT, LOCATION}, measures M = {DISCOUNT, BELIEF}, f(PRODUCT)={product_name}, f(LOCATION)={city_ID}, f(DISCOUNT)={amount}, and f(BELIEF)={pS}. If we want to assess how knowing Discount amounts will change the probability of Sales amounts, we can first use Cubic Product operation to the Sales and Discount cubes as follows: Sales ⊗ Discount = C_{result}. This will result in the superset of the desired information. By using the Restriction and Metric Projection, we can extract the required answers. The Cubic Product operation does not place any restrictions on the domains of the attributes.

Join (Θ_P): The *join* operator relates two cubes having one or more dimensions in common, and having identical mappings from common dimensions to the respective attribute sets of these dimensions. This operation can be expressed using Cubic Product operation. Therefore, this is not a basic operator in our algebra. The description of this operator is as follows: two cubes $C_1 = \langle C_1, A_2, f_1, d_1, O_1, L_1\rangle$ and $C_2 = \langle C_2, A_2, f_2, d_2, O_2, L_2\rangle$ are *join-compatible* if $D_1 \cap D_2 \neq \varnothing$, and $\forall_{ci} \in D_1 \cup D_2$, $f_1(c_i) = f_2(c_i)$. Furthermore, let $cd = D_1 \cap D_2 = \{cd_1, cd_2, \ldots, cd_m\}$ and $A_{cd} = \{a_{cd1}, a_{cd2}, \ldots, a_{cdm}\}$ denote the set of dimensions and corresponding dimensional attributes respectively. Hence, $A_{cd} = \cup_{\forall cd_i \in cd} f(cd_i)$ and $A_{cd} \subseteq A_d$. The algebra of join can be represented in terms of Cubic Product as:

$C_1\ \Theta_P\ C_2 = \sum_P(C_1 \otimes C_2)$ where P is a predicate of the form $[(C_1.a_{cd1} = C_2.a_{cd1}) \wedge (C_1.a_{cd2} = C_2.a_{cd2}) \wedge \ldots \wedge (C_1.a_{cdm} = C_2.a_{cdm})]$.

A Simple Example: Consider the query in the Cubic Product example. It can be answered by joining the **Sales** and **Discount** cubes as follows: **Sales Θ_P Discount = C_{Result}.**

Union-Compatible Cubes: Two cubes are *union-compatible* if they have the same structure. i.e. $C_1 = \langle C_1, A_1, f_1, d_1, O_1, L_1\rangle$ and $C_2 = \langle C_2, A_2, f_2, d_2, O_2, L_2\rangle$ are union-compatible if $C_1=C_2$, $A_1=A_2$, $f_1=f_2$, $d_1=d_2$, and $O_1=O_2$.

Union (∪): The *union* operator is a binary operator that finds the union of two union-compatible cubes. When union of two cubes is performed, value-equivalent cells must be coalesced using the coalescence-MAX (⊙) operator. This is because when we have two statements with varying degrees of belief, we pick the one with higher degree of belief (or more certain about). The algebra of the union operator is defined as follows:

Input: A cube $C_1 = \langle C_1, A_1, f_1, d_1, O_1, L_1\rangle$ and another cube $C_2 = \langle C_2, A_2, f_2, d_2, O_2, L_2\rangle$ which is union-compatible with C1.

Output: A cube $C_O = \langle C_O, A_O, F_O, d_O, O_O, L_O\rangle$, where $C_O = C_1 = C_2$; $A_O = A_1 = A_2$; $f_O = f_1 = f_2$; $d_O = d_1 = d_2$; $O_O = O_1 = O_2$;

$l \in L_O \Leftrightarrow \{ ((l \in L_1 \vee l \in L_2) \wedge ((\forall_k \in L_1 - \{l\}, \neg(k \cong l)) \wedge (\forall_j \in L_2 - \{l\}, \neg(j \cong l))$

$\vee ((j \in L_1) \wedge (k \in L_2) \wedge (l \cong j \cong k) \wedge (l = j \odot k)) \}$

Mathematical Notation: $C_1 \cup C_2 = C_O$

A Simple Example: Consider two cubes, Sales_South and Sales_North, both having the same cube structure. Suppose that Sales_South represents the sales in the southern region and Sales_North represents the sales in the northern region. We want to know the sales for the entire north-south region. Then we can accomplish this by using the union operator as follows: Sales_South ∪ Sales_North = Sales_North_South. The value-equivalent cells (in this example, those belonging to both the regions reported to have different probability measures with all the other attributes being identical) being coalesced.

Belief Difference (θ): The *belief difference* operator is a binary operator that finds the difference of belief measures for two union-compatible cubes. This operator can be used to find how a reporter of information differs with another in terms of belief strengths for the same object. This operator is non-commutative. Suppose we have two cubes: Cube1 and Cube2. It is only possible to find how much more confidence is represented by Cube1 compared to Cube2. By reversing the operands it is possible to find how much more confidence is represented by Cube2 compared to Cube1. Repetitive application of this operator will result in finding the objects for which both cubes have the same confidence as well. When belief difference operation is performed, the probability of value-equivalent cells in the result is calculated to reflect the difference in strengths of belief. If the difference between the probabilities of two value-equivalent cells is positive, then the cell assumes the new probability in the result. If not, it is not included in the result. The algebra of the belief difference operator is defined as follows:

Input: A cube $C_1 = \langle C_1, A_1, f_1, d_1, O_1, L_1\rangle$ and another cube $C_2 = \langle C_2, A_2, f_2, d_2, O_2, L_2\rangle$ which is union-compatible with C1.

Output: A cube $C_O = \langle C_O, A_O, F_O, d_O, O_O, L_O\rangle$, where $C_O = C_1 = C_2$; $A_O = A_1 = A_2$; $f_O = f_1 = f_2$; $d_O = d_1 = d_2$; $O_O = O_1 = O_2$;

$l \in L_O \Leftrightarrow ((j \in L_1)$

$\wedge (k \in L_2)$

$\wedge (l \cong j \cong k)$

$\wedge (j.CC(pS) > k.CC(pS))$

$\wedge (l.CC(pS) = j.CC(pS) - k.CC(pS)))$

Mathematical Notation: $C_1\ \theta\ C_2 = C_O$

A Simple Example: Consider two cubes, Sales_Report_By_John and Sales_Report_By_Jill, both having the same cube structure, representing the competitor's sales as reported by John and Jill respectively. We want to know how they differ in their beliefs for the competitor's sales. We can accomplish this by using the belief difference operator as follows: Sales_Report_By_John θ Sales_Report_By_Jill = Difference_Btwn_John_Jill. We also note that we can find the difference of belief strengths only when there are value-equivalent cells.

Cubic Difference (−): The *cubic difference* operator is a binary operator that finds the difference of two union-compatible cubes ignoring the probability measures. When cubic difference operation is performed, the value-equivalent cells with second cube are eliminated from the first cube. The algebra of the cubic difference operator is defined as follows:

Input: A cube $C_1 = \langle C_1, A_1, f_1, d_1, O_1, L_1 \rangle$ and another cube $C_2 = \langle C_2, A_2, f_2, d_2, O_2, L_2 \rangle$ which is union-compatible with C1.

Output: A cube $C_O = \langle C_O, A_O, F_O, d_O, O_O, L_O \rangle$, where $C_O = C_1 = C_2$; $A_O = A_1 = A_2$; $f_O = f_1 = f_2$; $d_O = d_1 = d_2$; $O_O = O_1 = O_2$; $l \in L_O \Leftrightarrow \{ ((l \in L_1) \wedge (\forall_j \in L_2, \neg(j \cong l))) \}$

Mathematical Notation: $C_1 - C_2 = C_O$

A Simple Example: Consider two cubes, Sales_South and Sales_Dallas, both having the same cube structure. Suppose that Sales_South represents the sales in the southern region and Sales_Dallas represents the sales in the Dallas city. We want to know the sales for the entire southern region except the Dallas city. Then we can accomplish this by using the cubic difference operator as follows: Sales_South – Sales_Dallas = Sales_South_without_Dallas.

The cubic intersection operator can be defined using the cubic difference operator. It is expressed as $C_1 - (C_1 - C_2) = C_{O.}$ Intersection is not a fundamental operator since it can be expressed in terms of other operators.

The cubic difference and belief difference operators can be used to find several interesting features of the data such as regions where different agents reported different belief strengths. They also can be used to sanitize the data where conflicting reports are not allowed. By judiciously applying Cubic Difference and Belief Difference operators in combination with other cubic operators defined earlier, it is possible to find the difference in strengths of beliefs for cubes that are union-compatible within a subset of their characteristics.

Aggregation (*Γ*): The *aggregation* operator performs aggregation (MIN, MAX, SUM, AVG, COUNT, RANK, PERCENTILE, etc.) on one or more dimensional attributes. This operator in combination with fuzzy membership functions defined for *pS* can be used to answer queries such as "What is the most likely average sales?", "What is confidence level for average sales to be 25?", etc. The queries that do not contain uncertainty in their formulation may have an answer with uncertainty in it. For example, "What are the maximum sales?" can be answered by selecting the cell with maximum sales reported which may have strength of belief of 0.01 (or very unlikely).

Let μ be a metric attribute to aggregate where $\mu \in A_m$ and G be a set of grouping attributes such that $G \in A_d$. Let F be an aggregate function having the mapping

$F : 2^{\Pi_{\forall gi} \in G^{dom\, gi}}$ → dom_{agg} , where *agg* represents a user-specified attribute name given to the result, which is extracted from the domain **dom agg**. F is assumed to be a first-order definable function including the standard arithmetic functions + (addition), − (subtraction), ∗ (multiplication), and / (division), the standard SQL aggregate functions, and a RANK function. The RANK function takes a group of cells as input and returns an attribute agg corresponding to the ordinal number of the cell. The aggregation operator is defined as follows:

Input: A cube $C_I = \langle C, A, f, d, O, L \rangle$, a set of grouping attributes G, a metric attribute μ, and an aggregate function F.

Output: A cube $C_O = \langle C_O, A_O, F_O, d_O, O_O, L_O \rangle$,

WHERE

$c_o = \{ c \mid c \in C, \exists x, x \in f(c) \cup \{AGG\}$ and $\{AGG\}$ is a new characteristic name defined specifically for the aggregated metrics,

$A_O = G \cup \{agg\}$ and $\{agg\}$ represents the computed aggregate attribute,

$L_O = \{l \mid \exists l \in L, l_o.AC = \langle l.AC[g_1], l.AC[g_2], \ldots, l.AC[g_n]\rangle$, $l_o.CC = \langle l.CC[agg]\rangle\}$,

$$f_O = \begin{array}{l} \{\langle x, y\rangle \mid x \in Co, (\exists\langle x, z\rangle \in f, x \neq \{AGG\}, y = \{a \mid a \in (z \cap Ao)\}) \\ \vee (\exists\langle x, z\rangle \in f, x = \{AGG\}, y = \{a \mid a \in (z \cap Ao) \cup \{agg\}\})\} \ if\ \exists\langle\{AGG\}, z\rangle \in f, \\ \{\langle x, y\rangle \mid x \in Co, (\exists\langle x, z\rangle \in f, y = \{a \mid a \in (z \cap Ao)\})\} \cup \{\langle\{AGG\}, \{agg\}\rangle\} \ otherwise \end{array}$$

$$\forall x \in C, do(x) = \begin{cases} d(x) & if \{AGG\} \in C \\ d(x) \cup \langle AGG, 0\rangle & otherwise \end{cases}$$

Mathematical Notation: $\Gamma_{F,G,\mu}(C_I) = C_O$.

A Simple Example: Consider the Sales cube. Suppose the user wants to see the total annual sales for each product. Then, F = SUM, G = {product_name, year}, and μ = amount. Therefore, the query to get the total annual sales for each product is written as

$\Gamma_{[SUM, \{product\_name, year\}, amount]}$ **(Sales) = $\mathbf{C_{Result}}$**.

Note that prior to applying the aggregation operator, the Sales cube can be operated upon by various operations depending on what strengths of probabilities need to remain. If we want annual sales corresponding to the highest confidence level objects, we first apply MAX(pS) aggregation operator to the Sales cube and then apply aggregation operator for SUM. Applying aggregation operators without first applying a meaningful transformation on the pS may result in meaningless data. In this example, if we apply

SUM on all the objects, then the result contains sum of several amounts for the same PRODUCT, TIME, and LOCATION, which is not a meaningful total. Instead, we could have applied a transformation that picks object with maximum pS or a more complex operation that combines all objects with differing pS but have the address component to obtain a probability distribution for that object and then selects an attribute value with say 95% confidence level. We can even ask for object attribute value that lies between σ and $-\sigma$, thus utilizing all the concepts of statistical distributions.

Furthermore, Force and Extract operations are defined for this probabilistic multidimensional data model. However, applying these operations on pS results in pS losing its special meaning and BELIEF becoming a regular characteristic.

Force (ψ): The force operator converts dimensions to measures. Let a_t be a dimensional attribute to transform such that $g(a_t) \in D$. Let c_t be the corresponding characteristic name for a_t such that $c_t \notin D$ and either $c_t \in M$ or c_t is a new characteristic name. The force operator is defined as follows:

Input: A cube $C_I = \langle C, A, f, d, O, L\rangle$, a dimensional attribute to transform a_t, and a corresponding characteristic name c_t.

Output: A cube $C_O = \langle C_O, A_O, F_O, d_O, O_O, L_O\rangle$

Where

$C_O = C \cup \{c_t\}$,

$f_O = f - f(g(a_t)) + [g(a_t) \rightarrow f(g(a_t) - a_t)] + [c_t \rightarrow a_t]$,

$O_O = O_{prev} \cup O_{new}$ where O_{prev} is obtained by removing ordered pairs containing a_t from O and O_{new} represents a user specified set of ordering relations between a_t and the elements of $f(c_t)$ if $c_t \in M$,

$L_O = \{l_O \mid \exists l \in L, l_O.AC = l.AC - \langle l.AC[a_t]\rangle, l_O.CC = l.CC \bullet \langle l.AC[a_t]\rangle\}$,

$$\forall c_i \in C, d_o(c_i) = \begin{cases} d(c_i) & \text{if } c_i \neq c_t, \\ 0 & \text{otherwise} \end{cases}$$

Mathematical Notation: $\psi_{at,ct}(C_I) = C_O$

A Simple Example: Converting *store_name* from dimension to a measure in the Sales cube. This can be expressed as: $\psi_{store\_name,\ sales}(\mathbf{Sales}) = C_{Result}$

Extract (Φ): The extract operator converts measures to dimensions. Since we assigned a special meaning for the BELIEF characteristic and made it a measure, it cannot be extracted to a dimension without losing its special meaning. Even forcing it back to a measure after extracting *pS* may not restore its meaning after certain operations. Let a_t be a metric attribute to transform such that $g(a_t) \in M$. Let c_t be the corresponding characteristic name for at such that $c_t \notin M$ and either $c_t \in D$ or c_t is a new characteristic name. The extract operator is defined as follows:

Input: A cube $C_I = \langle C, A, f, d, O, L\rangle$, a metric attribute to transform a_t, and a corresponding characteristic name c_t.

Output: A cube $C_O = \langle C_O, A_O, F_O, d_O, O_O, L_O\rangle$

Where

$C_O = C \cup \{c_t\}$,

$f_O = f - f(g(a_t)) + [g(a_t) \rightarrow f(g(a_t) - a_t)] + [c_t \rightarrow a_t]$,

$O_O = O_{prev} \cup O_{new}$ where O_{prev} is obtained by removing ordered pairs containing a_t from O and O_{new} represents a user specified set of ordering relations between a_t and the elements of $f(c_t)$ if $c_t \in D$,

$L_O = \{l_O \mid \exists l \in L, l_O.AC = l.AC \bullet \langle l.AC[a_t]\rangle, l_O.CC = l.CC - \langle l.AC[a_t]\rangle\}$,

$$\forall c_i \in C, d_o(c_i) = \begin{cases} d(c_i) & if\ c_i \neq c_t, \\ 0 & otherwise \end{cases}$$

Mathematical Notation: $\mathbf{\Phi}_{at,ct}(C_I) = C_O$

A Simple Example: Converting store_name from a measure to a dimension in the Sales cube. This can be expressed as: $\mathbf{\Phi}_{store\_name,\ sales}(\mathbf{Sales}) = C_{Result}$

Properties of the Model

When BELIEF is not a characteristic of the cube, this model is reduced to the deterministic model. In this section, a proof is presented that the algebra defined for the model is closed, at least as expressive as relational model, and relationally complete. The following proof starts by showing that the data model is reducible to relational model and content-wise equivalent using the following definitions on the way. The following content is adopted from (Moole, 2003).

Data Equivalence ($\cong_{\mathfrak{R}}$): A relation instance (a row in a table) is a set of n-tuples. Each tuple can correspond to an individual cell in the cube (in fact, our example of cells in the Sales cube are shown as tuples of dimensional attributes and metric attributes). When there are no tuples in the relational instance it is equivalent to an empty cube. A formal definition of Data Equivalence between relational instance r and a cube C is as follows:

An instance r of a relation R and cube C are data equivalent, denoted r $\cong_{\Re}$ c, iff C = ⟨C, A, *f*, d, O, L⟩ such that

C = {M}, where M is an arbitrary characteristic

A = R, i.e., the relation and the cube have the same set of attributes

f = {{M, A}}, i.e., f maps all attributes to the arbitrary characteristic M

d = {{M, 0}}, i.e., characteristic M is a measure

O = ∅, i.e., no partial ordering is present

L = {l | ∃*t* ∈ r, *l*.CC = *t*, *l*.AC = ∅}, i.e., for every tuple *t* in r, there exists a single cell *l* in C which has the tuple as its content component and no address component.

THEOREM 1: *Our algebra is closed.*

To prove this theorem, we must show that all the basic operations defined in our algebra result in Cube as defined in the model. The Cube must satisfy the following three criteria: (1) the values of cells must come from an appropriate domain, (2) no two cells in result cube are value-equivalent, and (3) The result cube is finite collection of cells. Considering the definitions of operators, every basic operator produces a result Cube. The domain of cells other than *pS* are defined to be the same as those in the original cube. The *pS* has the domain of (0, 1]. To prove that domain of *pS* will be (0, 1] after the application of basic operators, we examine each operator except Cubic Difference and find they all result in pS greater than zero. The coalescence operators, which are part of Union, Metric Projection, etc., also explicitly prevent the value of pS to be greater than 1. Therefore, we conclude that (1) is satisfied. Noting that coalescence operators always coalesce value-equivalent cells and produce a single cell trivially proves second criterion. The coalescence operators always produce the same number cells as there are in the input or less. We can also see that the number of cells in the result can be at most $|C_1|$ x $|C_2|$ for Cubic Product operation $C_1 \otimes C_2$. All the other operators result in less number of cells than Cubic Product. Since an empty cube also satisfies the definition, the operators resulting in empty cube are still closed.

THEOREM 2: *Our algebra is at least as expressive as the relational algebra.*

We show that all five basic relational algebra operators (Restriction, Projection, Union, Difference, Product) can be expressed in our algebra. Consequently, it follows that other derived operators can be expressed as well.

Restriction: Given a relation instance r and a cube C such that $r \cong_{\Re} C$. Suppose we have a selection predicate P. Since $\sigma_P(r)$ returns relation instance r′ containing tuples of r that satisfy P and $\sum_P(C)$ returns cube C′ containing cells of C that satisfy P, we conclude that $\sigma_P(r) \cong_{\Re} \sum_P(C)$.

This line of argument can also be used to substantiate the claims made below.

Metric Projection: Given relation instance r and a cube C such that $r \cong_{\Re} C$,

$\pi_S(r) \cong_{\Re} \prod_S^M (C)$

Union: Given relation instances r_1 and r_2 and cubes C_1 and C_2 such that $r_1 \cong_{\Re} C_1$ and $r_2 \cong_{\Re} C_2$, $r_1 \cup r_2 \cong_{\Re} C_1 \cup C_2$.

Difference: Given relation instances r_1 and r_2 and cubes C_1 and C_2 such that $r_1 \cong_{\Re} C_1$ and $r_2 \cong_{\Re} C_2$, $r_1 - r_2 \cong_{\Re} C_1 - C_2$.

Cubic Product: Given relation instances r_1 and r_2 and cubes C_1 and C_2 such that $r_1 \cong_{\Re} C_1$ and $r_2 \cong_{\Re} C_2$, $r_1 \text{ X } r_2 \cong_{\Re} C_1 \otimes C_2$. This holds since (a) $r_1 \text{ X } r_2$ returns a relation r having |r1| x |r2| number of tuples representing all possible combinations of both relational instances and (b) $C_1 \otimes C_2$ returns a cube C having all characteristics and attributes of both C1 and C2 and cells representing all possible combinations of the cells of both cubes.

By showing that every relational algebra operator can be expressed in our algebra, we conclude that our algebra is at least as expressive as relational algebra and possibly more expressive since we can perform several additional operations in our algebra. Intuitively this algebra is relationally complete. The preceding text was reproduced from Moole (2003).

Modification of Uncertain Data

Since *pS* is represented as joint probability of a set of mutually independent variables, Bayesian methods can be applied to update this probability when new information is obtained. Similar applications have been reported earlier for various types of uncertain data (Dey & Sarkar, 2000). Real world data change over time and need corresponding modifications to affected objects. Change can be a result of adding new information, deleting existing information, or modifying existing information. In each of these instances underlying beliefs need to be revised. Bayesian Framework provides a solid basis for revision of belief. *Bayes Conditionalization Formula* (Pearl, 1988) to calculate new probability is given as:

$$\text{prob}(A|e) = \sum_{i=1}^{n} \text{prob}(A|Bi,\ e)\ \text{prob}(Bi|e)$$

A simplified version of this formula (Jeffrey, 1983), known as *Jeffrey's Rule of Probability Kinematics,* calculates new probability using:

$$\text{PROB}(A) = \sum_{i=1}^{n} \text{prob}(A|B_i)\ \text{PROB}(B_i)$$

In data modification frameworks, this latter formula provides computational advantages and hence it is used.

As can be seen from the above related research review, prior research is completely analytical in nature. The analytical research method is best suited to an extension of research in this area (Martin, 2004).

The above discussion reviewed most recent and relevant related research concerning probabilistic multidimensional data model, and literature survey indicated that enhancements have not yet been reported. The research conducted so far does not address topics of additional required operators, a data modification framework, data modification algorithm(s), and applicability of the probabilistic multidimensional data model to business management problems.

Summary

In this chapter, the investigator reviewed research on uncertainty and conceptual data models. The importance of conceptual data models in the development of products, history of the relational data model, and the impact of conceptual data models on product development were discussed as well as multidimensional data models and their shortcomings. Research related to enhancement of multidimensional data models and probabilistic relational data models was presented along with required enhancements to the probabilistic multidimensional data model. The relationship of current research to required enhancements was also presented.

In Chapter 3, research methodology used for this study, justification for selecting methodology, and advantages and disadvantages of that methodology will be discussed.

CHAPTER 3

METHODOLOGY

This study is aimed at addressing data modification framework and algorithms. The investigator performed the following tasks: (a) Enhanced the probabilistic multidimensional data model (Moole, 2003), (b) provided all required algebraic operations, (c) provided a framework to update probabilistic data, (d) developed algorithm(s) to update probabilistic data, and (e) analyzed the time and space complexity of update algorithm(s). Additionally, a fictitious business management application was used to help understand the model and appreciate its usefulness.

According Buckley, Buckley, and Chiang, there are multiple methods of conducting scientific research (Martin, 2004); suitable research methods depend on the subject being researched. This chapter includes a framework for selecting research. Justification for the selected method, its advantages, and its disadvantages are presented. A strategy to mitigate disadvantages is discussed.

Analytical methods of research required the researcher's internal logic in analysis, synthesis, and construction of theories. Results are generally reported as mathematical formulas, often accompanied by proofs. This method is suitable for mathematics research.

The research problem was identified by reviewing prior research (Moole, 2003; Thomas & Datta, 2001). The analytical method is best suited to solving this research problem because of the need for an analysis of set theoretical mathematical concepts and probability theory axioms without reference to any empirical data. Mathematical proof of correctness of the new formulas was provided. Other methods (quantitative, qualitative, experimental, case studies, etc.) are deemed unsuitable for this problem because of its mathematical nature (Martin, 2004). Use of deductive logic on set theory and probability axioms was the predominant analytical method, starting with general theories of sets and probability and deriving specific theories applicable to probabilistic multidimensional data. The probabilistic multidimensional data were compared and contrasted to Bayesian belief networks to elicit their relative strengths and weaknesses. Unlike other methods, such as quantitative and qualitative methods, which consist mainly of data collection and interviews, the analytical method uses step-by-step derivation of new formulas from proven set theory and probability axioms. The new formulas derived were mathematically proven to establish correctness. Properties of the model and the modification algorithm were also mathematically proven.

The investigator performed the following steps during this research, applying analytical method:

1. Investigated conceptual models for DW and OLAP;
2. Synthesized a probabilistic multidimensional data model;
3. Developed an uncertain data modification framework;
4. Developed data modification algorithms; and
5. Analyzed time and space complexities of algorithms.

Justification for Selecting the Analytical Method

This research problem is derived from a deductive syllogistic work whereby the investigator used internal logic to perform mathematical analysis of the subject. The analysis presented in chapter 2 is based on mathematical modeling of the probabilistic data. It logically followed that current research to extend that preliminary work be performed using the same method. Figure 2 below, adapted from Martin (2004), presents a framework for selecting the methodology, which can be applied to the underlying model as well as to the current research work.

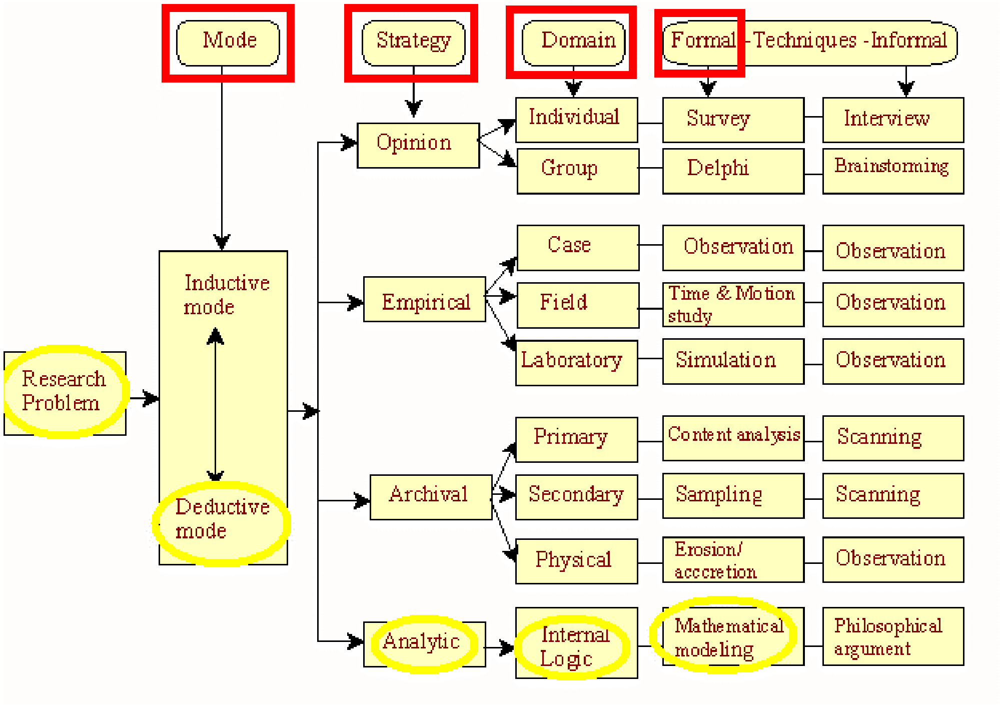

Figure 2. Framework to select a research methodology. Adapted from Martin (2004) with permission from author.

The analytical method based on internal logic of the authors has several advantages and disadvantages. The following excerpt from Martin (2004) describes advantages in the first column and disadvantages in the second.

Table 1.

Advantages and Disadvantages of the Analytical Method.

Advantages	Disadvantages
There is no need to search for additional data and analytic research is not limited by existing data. It provides the broadest scope for imagination and creativity. Best suited for use of logic, philosophy, and operation research techniques.	The most abused strategy and most difficult to criticize. Requires a first rate mental ability that is rare. Can more readily be used to mislead. Often sloppy. It is subject to logical errors, problems of semantics, etc. Temptation to focus on trivial and irrelevant problems.

Note: Adapted from (Martin, 2004) with permission from the author

As Table 1 indicates, creativity and imagination are not limited in the analytical method as may be the case in empirical (qualitative and quantitative) studies, which fix frameworks for data collection and contain limits on collectable data, and in which results may be generalized from inappropriate sample sizes. These all limit scope and impact of results and their benefits.

One might misuse the analytical method. Therefore, the investigator paid close attention to detail and the logical flow of research results. Logical errors are difficult to identify; however, the investigator conducted previous research using the analytical method and found that it is possible not only to identify logical errors published by eminent researchers but also to prove claims which are mathematically beyond doubt or disagreement (Moole & Valtorta, 2003). Problems related to semantics will be avoided by using semantics established by reputable researchers.

The investigator performed mathematical derivation of equations and validated them using mathematical proof of correctness. Algorithms were analyzed using methods established in the field of computer science (Knuth, 1973). The algorithmic analysis focused mainly on space and time complexities.

Additionally, the following fictitious application of a multidimensional data model to a business management problem was described in detail: Consider the situation of a typical category manager in a retail food store. The general process to assess inventory starts with the manager walking through aisles and scanning bar code labels on shelves or products using a handheld device (Symbol, 2004). At the end of the process the manager connects the handheld device to a computer to transmit orders for items scanned. The manager generally forecasts the number of items to order. An inaccurate forecast could result in empty or overstocked shelves. If the shelves are empty customer dissatisfaction leads to business loss. Overstocked items cost in terms of money and shelf space leading to losses. Accuracy of forecast depends on the manager’s experience (PCG, 1998). In this situation, even though theoretically it is possible to forecast demand based on past sales data, due to the volume of sales transactions, it would be unrealistic to do so without using Decision Support Systems that can handle large volumes of uncertain data. Using DSS could result in optimal inventory management. This also helps in collaborative planning with manufacturers and distributors. Decision Support Systems based on the probabilistic multidimensional data model could provide several benefits in this situation, which are described in Chapter 4. This application is fictitious, and is meant to enhance readers' understanding of the formulas and aid them in the interpretation of results.

Summary

In this chapter, reasons for choosing the analytic research method were presented. Advantages and disadvantages of the analytical method were discussed. Elaboration of equations, their validity, proof of correctness, and analysis of the algorithms were included. In Chapter 4, research results and required mathematical proofs for formulas are presented.

CHAPTER 4

RESULTS

In this chapter, all results of this dissertation research are described. Results contain required operators on PMDDM, uncertain data modification framework, uncertain data modification algorithm, proof of correctness of the modification algorithm, and an analysis of time and space complexities. In addition, the investigator discusses a fictitious business management application to illustrate the data model and its application. Finally, the investigator compared and contrasted the PMDDM with the Bayesian belief network framework.

Operators on PMDDM

Following are additional definitions and operators required to perform modification of the data.

Predicate P: A predicate is a well-formed formula in first-order predicate logic.

An *atomic predicate* is a restriction on the domain of a single attribute or characteristic, e.g. (year = 1994).

A *compound predicate* is a logical expression of atomic predicates. Logical operators are $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implies), and $\leftrightarrow$ (equivalent to). It is of form: P = p_1 $\langle$op$\rangle$ p_2 $\langle$op$\rangle$ … $\langle$op$\rangle$ p_n. e.g. (year = 1994) $\wedge$ ((quantity < 15) $\vee$ (amount > 100))

***l* satisfies P**: *l*, an instance of L, with structure <address, content> satisfies predicate P if and only if:

Case 1: If an element of *l* is a dimension, then *l*.AC satisfies P, otherwise *l*.CC satisfies P, if P is atomic and truth-value is TRUE.

$$P(l) = TRUE \quad \begin{cases} a \ in \ P, a \in f(d_i), d_i \in D, \\ P(l.AC[a]) = TRUE \\ OR \\ a \ in \ P, a \in f(m_i), m_i \in M, \\ P(l.CC[a]) = TRUE \end{cases}$$

e.g. Upper left most corner cell in cube of Figure 1 satisfies P=(year=1993)

Case 2: If P is a compound predicate, *l* satisfies P, when all truth-values evaluated together with connecting operators results in TRUE.

$$\forall p_i \in P, a \ in \ p_i, p_i(l) = Q_i$$
$$P(l) = Q_1 \langle op \rangle Q_2 \langle op \rangle \ldots \langle op \rangle Q_n$$

e.g. Upper left most corner cell in cube of figure 1 satisfies

P=(year=1993)∧(city=”Boston”)∧(product_name=”P1”)

Cardinality of Predicate (η_P): Cardinality, denoted by η, of a predicate is defined as number of unique attributes appearing in predicate. An atomic predicate has cardinality of 1. Cardinality of a compound predicate is ≥1 (a compound predicate may be constructed using a single attribute, hence η=1).

Selectivity of Predicate P on a cube C ($\delta_{[P,C]}$): Selectivity of a predicate P, denoted by δ, for a given cube C, is size of subset of cube cells in L of C that satisfy P.

Cardinality and selectivity are useful in ordering and identifying cube cells. The update algorithm uses this ordering capability to ensure correct handling of marginal probability specifications.

Framework for Modification of Uncertain Data

This section includes a description of the modification of probabilistic data. Since the model is representing probability for each object as joint distribution of all attributes of that object, modifying the probability represents change in belief about that object. This change can be a result of adding new information, deleting existing information, or modifying existing information. In each of these instances there is a need to revise beliefs. The framework is similar to Bayesian Framework proposed for probabilistic relational models by Dey and Sarkar (Dey & Sarkar, 2000), but a different algorithm is used in order to handle marginal probability specifications. Bayesian Framework provides a solid basis for belief revision. A summary of Pearl's (Pearl, 1988) discussion of Bayesian belief revision follows.

Bayes conditionalization formula: Let prob(A|e) is belief in proposition A after evidence e is observed. If prob(A|B_i, e) represents conditional probability of A given B_i and e (after evidence e), prob(A|B_i) represents the conditional probability of A given B_i (before evidence e), and prob(B_i|e) represents the conditional probability of B_i given e (after evidence e), then prob(A|e) can be computed from prob(A|B_i,e) and prob(B_i|e), if A and e are conditionally independent given B_i using the following formula:

$$\text{prob}(A|e) = \sum_{i=1}^{n} \text{prob}(A|Bi, e)\ \text{prob}(Bi|e)$$

This formula requires knowledge of conditional probability of proposition A given B changes when e is observed. This is often not possible. Jeffrey (Jeffrey, 1983) proposed a simplification of this formula known as *Jeffrey's Rule of Probability Kinematics*.

Jeffrey's Rule: Let *PROB* denote the new degree of belief and *prob* denote prior belief. If belief in proposition A does not directly depend on new evidence e, but changes degree of belief in B_i. Since A is conditionally dependent on B_i, new evidence e effects a change in degree of belief in A. If one assumes PROB(A|B_i) = prob(A|B_i), i.e. conditional probability of A given B_i does not change when e is observed, then above conditionalization formula reduces to:

$$\text{PROB(A)} = \sum_{i=1}^{n} \text{prob}(A|B_i)\ \text{PROB}(B_i)$$

The advantage with this formula is that one already knows prob($A|B_i$) prior to evidence e, which will not change due to e (since A is not directly dependent on e), and one can assess PROB(B_i) easily after e.

This simplification provides a great computational advantage and practical applicability. A more involved discussion of semantics and philosophical underpinnings of these two formulas with examples can be found in Pearl (1988).

New information is presented in the form of a cube. This new information can result in modification of existing information in multiple ways. The effects of and method of handling new information can be divided into two categories. The first category is new information containing different schema, different domain for attributes, different partial orders, a dimension as a measure, or a measure as a dimension. The second category is new information containing different objects, but the cube structure and definition remain identical with existing information.

New information in the first category can be merged with existing information as follows:

1. Schema change case (different C, A, or f) can be handled by applying the cubic product operator. If there are additional attributes, they expand attributes of joint probability distribution.
2. The case of different domains for attributes can be handled by applying a combination of algebraic operations, under the assumption that existing

data hold true with new probability distributions. Only dimensional attributes with different domains need special handling. Measure attributes with different domains can be handled by considering them as belonging to the second category. Each dimensional attribute with a different domain compared to the existing dimensional attribute domain can be handled independently. The resulting domain will be the union of new domain and existing domain. Then objects in both new information and existing information can be considered as partial objects in the result, and handled by considering them as second category objects.

3. The case of different partial orders can be handled by defining a new partial order for the result. This is because partial orders have semantics associated with them and may not be useful if one disregards semantics.
4. If a dimensional attribute became a measure or a measure changed to a dimension, a decision could be made on what this attribute would become as a result. Then a force or extract operator could be used to convert the dimension to measure or vice-versa. The first category information could be regarded as a structural change of the cube.

New information in the second category will have identical structure as existing information. New information may require the addition of a new object, deletion of an existing object, or modification of degree of belief in an existing object, or it may specify joint probability of a subset of attributes of an existing object. In all these cases, modification of existing data should: (a) be consistent with new information and, (b) result in assigning probabilities to unspecified realizations of stochastic variables in a manner consistent with existing data.

Let one denote the class of L *c* of a Cube C_I as a set of attributes AXY ∪ {pS} or {A, X, Y, pS}, where A is the address component, X and Y are mutually exclusive subsets of the set $\{\{A_d \cup A_m\} - \{A \cup pS\}\}$, one of which may be empty. An object of this class is denoted as {A=a, X=x, Y=y, pS=q}. In this representation, A corresponds to L.AC and X and Y are subsets of remaining attributes in the class of L without including pS. For example, if one has the cube of figure 1, then A = L.AC = {year, product, city}, X may be {amount} and Y may be {quantity}. This can also be represented as the union of all these attributes {year, product, city, amount, quantity, pS} in which pS represents joint probability of the remaining attributes. Let one suppose receipt of new information consisting of objects representing new beliefs. Assume that new information is specified as a cube C_{new} with the same structure as the existing cube C_{old}. In following sections, one says ”the new set of objects matches the existing set of objects” to indicate a selection predicate P constructed on {X ∪ Y} evaluating to true for both sets. When there is no match, there does not exist a selection predicate that satisfies both sets of objects. A special case is when all attributes of an object are unknown, i.e. ⟨A=a, X=*, Y=*, pS=q⟩. In this case, there are an infinite number of predicates that match. This is considered as not matching. The resulting cube after applying the updates described in each of the following cases is denoted by $C_{updated}$.

Case 1: There exists a set of objects ⟨a, x, y, q⟩ ∈ C_{new} that specifies complete joint probability distribution for A=a, i.e. $\Gamma_{SUM,pS,P}\left(\prod_{pS}^{M}\sum_{[A=a]}C_{new}\right) = 1$. In this case, all existing objects must be replaced with the new set of objects. The remaining cases assume the new probability distribution specified is incomplete, It should be noted that partial distributions can be made complete distributions by assigning unspecified probability to unknown values.

Case 2: There exists an object $\langle a, x, y, q\rangle \in C_{new}$ that does not match with any object in existing data. In this case one has to create a new object. When a new object with an address component "a" is created, one has to adjust strength of belief in other objects with the same address component. An extremity is when the new object has $q = 1$, in which case it replaces all existing objects, because the data model restricts the sum of beliefs for an address not to exceed 1 (this is handled by case 1). Cases 3 and 4 handle allocation of residual probability when $q < 1$.

Case 3: There exists an object $\langle a, x, y, q\rangle \in C_{old}$, for some $q \in (0, 1]$, such that $x = x_i$, for some $i \in \{1, 2, \ldots, m\}$. This is a case where an existing object matches an object in the new information on attributes A and X. It is possible for an existing object to match new information based on more than one predicate. In such cases, the predicate selection is made by maximizing η and minimizing δ. The rationale behind this is that when η is maximum, there are a greater number of attributes in a predicate which indicates a more precise match of objects (less marginalization) and a minimal δ indicates less number of objects matched (an exact match will have δ=1). This is essential in order to handle marginal probability specifications.

In this case, the new probability Q for the matching object is calculated, using Jeffrey's Rule, as follows:

$$\begin{aligned} Q &= \text{PROB}[A = a, X = x_i, Y = y] \\ &= \text{prob}[Y = y \mid A = a, X = x_i]*\text{PROB}[A = a, X = x_i] \end{aligned}$$

= (prob[A = a, X = x$_i$, Y = y] / prob[A = a, X = x$_i$]) * PROB[A = a, X = x$_i$]

$$= \frac{q}{\Gamma_{SUM,pS,P}\left(\prod_{pS}^{M}\sum_{[A=a,X=xi]}C_{old}\right)} * p_i \quad \rightarrow \quad \text{Equation } (\mathbf{I})$$

where p_i is PROB[A = a, X = x$_i$] for i = 1, 2, …, m.

Then object ⟨a, x, y, q⟩ should be replaced with ⟨a, x, y, Q⟩.

Case 4: There exists an object ⟨a, x, y, q⟩ ∈ C_{old}, for some q ∈ (0, 1], such that x ≠ xi, for all i ∈ {1, 2, …, m}. In this case, an existing object does not have a matching object in the new information. The object ⟨a, *, *, q⟩ also has no match, therefore it will be handled by this case.

In this case, new probability for objects without a match is calculated by proportionately distributing the difference between old residual probability and new residual probability after resolving the objects of above cases, if any. The old residual probability P_{oldRes} of objects with A = a and X≠x$_i$ before applying cases 1, 2, and 3 is calculated by:

$$P_{oldRes} = \text{prob[A=a, X=x]} = 1 - \Gamma_{SUM,pS,P_{old\,Res}}\left(\prod_{pS}^{M}\sum_{[A=a,X\neq xi]}C_{old}\right)$$

The new residual probability after previous cases P_{newRes} of objects with A=a and X≠x$_i$ is calculated by:

$$P_{newRes} = \text{PROB[A=a, X=x]} = 1 - \Gamma_{SUM,pS,P_{newRes}}\left(\prod_{pS}^{M}\sum_{[A=a,X\neq xi]}C_{updated}\right)$$

One proportionately distributes residual probability $P_{oldRes} - P_{newRes}$ based on old probabilities. This distribution has to be to objects other than x$_i$, i = 1, 2, …, m.

With this, one can now calculate new probability Q associated with ⟨a, x, y, q⟩ as below:

$Q = \text{PROB}[A = a, X = x, Y = y]$

$= \text{prob}[Y = y \mid A = a, X = x] * \text{PROB}[A = a, X = x]$

$= (\text{prob}[A = a, X = x, Y = y] / \text{prob}[A = a, X = x]) * \text{PROB}[A = a, X = x]$

$= (q / P_{\text{oldRes}}) * P_{\text{newRes}}$ → Equation (**II**)

Then object ⟨a, x, y, q⟩ should be replaced with ⟨a, x, y, Q⟩.

The following example illustrates all of the above cases,

{ ⟨⟨1993, P1, Boston⟩, ⟨100, 10, 0.5⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨125, 10, 0.2⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨150, 15, 0.1⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨140, 15, 0.1⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨160, 20, 0.01⟩⟩ }

and new information with three objects

{ ⟨⟨1993, P1, Boston⟩, ⟨170, 25, 0.01⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨160, 20, 0.02⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨*, 15, 0.1⟩⟩ }

This new information specifies a partial distribution (total probability of the new objects is 0.13, hence one assumes a partial distribution. If this were a complete distribution, then it should contain another object ⟨⟨1993, P1, Boston⟩, ⟨*, *, 0.87⟩⟩).

Case 1 is not applicable. The new object ⟨⟨1993, P1, Boston⟩, ⟨170, 25, 0.01⟩⟩ falls under case 2. One creates this new object in the updated cube. The remaining new objects fall under case 3. The object ⟨⟨1993, P1, Boston⟩, ⟨160, 20, 0.02⟩⟩ modifies the existing object's probability. The object ⟨⟨1993, P1, Boston⟩, ⟨*, 15, 0.1⟩⟩ matches on predicate P=(amount=*)∧(quantity=15) with two existing objects ⟨⟨1993, P1, Boston⟩, ⟨150, 15, 0.1⟩⟩ and ⟨⟨1993, P1, Boston⟩, ⟨140, 15, 0.1⟩⟩. Their new probabilities can be calculated using case 3. The remaining existing objects are:

{ ⟨⟨1993, P1, Boston⟩, ⟨100, 10, 0.5⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨125, 10, 0.2⟩⟩}

These two existing objects fall under case 4. In this case, P_{oldRes} = 0.79 and P_{newRes} = 0.87. Using equation II above, one will have following final cube.

{ ⟨⟨1993, P1, Boston⟩, ⟨100, 10, 0.55⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨125, 10, 0.22⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨150, 15, 0.05⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨140, 15, 0.05⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨160, 20, 0.02⟩⟩,
⟨⟨1993, P1, Boston⟩, ⟨170, 25, 0.01⟩⟩ }

These four cases illustrate modification of existing probabilistic data when new information is obtained. The following sections describe an algorithm to revise belief strengths of probabilistic multidimensional data and its proof of correctness.

Algorithm for Modification of Uncertain Data

```
1     Input: $C_{old}$, $C_{new}$
2     Output: $C_{updated}$
3     BEGIN
3.1     for each A=a do
3.2     begin
```

```
3.2.1      $p_{old} := \Gamma_{SUM,\,pS,\,P}\left(\prod_{pS}^{M}\sum_{[A=a]}C_{old}\right)$
3.2.2      $p_{new} := \Gamma_{SUM,\,pS,\,P}\left(\prod_{pS}^{M}\sum_{[A=a]}C_{new}\right)$
3.2.3      $m_{new} := |\sum_{[A=a]}C_{new}|$
3.2.4      if ($p_{new}$ = 1) then                    { Case 1 }
3.2.4.1      begin
3.2.4.1.1      $C_{updated} := C_{updated}$ U $\sum_{[A=a]}C_{new}$
3.2.4.2      end
3.2.5      else
3.2.6      begin
3.2.6.1      for each object $o_k$ of $\sum_{[A=a]}C_{new}$ without a match do   { Case2 }
3.2.6.2      begin
3.2.6.2.1      $C_{updated} := C_{updated}$ U $o_k$
3.2.6.2.2      $C_{new} := C_{new} - o_k$
3.2.6.3      end
3.2.6.4      for each object $o_j = \langle a, x, y, q\rangle$ of $C_{old}$ with a match do
3.2.6.5      begin
3.2.6.5.1      construct a set of predicates SP from $\{x \cup y\}$
3.2.6.5.2      sort them first by max($\eta$) and then by min($\delta$)
3.2.6.6      end
3.2.6.7      for each object $o_j = \langle a, x, y, q\rangle$ of $C_{old}$
3.2.6.8      with a  match based on the set SP do              { Case 3 }
3.2.6.9      begin
3.2.6.9.1      calculate Q from case 3 [Equation I]
3.2.6.9.2      $C_{updated} := C_{updated} \cup o_j$
3.2.6.10     end
```

```
3.2.6.11   for each object $o_j = \langle a, x, y, q\rangle$ of $C_{old}$ without a match            { Case 4 }
3.2.6.12   begin
3.2.6.12.1   calculate Q from case 4 [Equation II]
3.2.6.12.2   $C_{updated} := C_{updated} \cup o_j$
3.2.6.13   end
3.2.7    end
3.3    end
4    END
```

Proof of Correctness of the Algorithm

Investigator proves the correctness by showing the belief revision algorithm is (1) Complete (it covers all possible modification situations), (2) Consistent (satisfies the axioms of probability theory) and (3) Closed (only valid objects will result). Assume the new information provided is a valid cube.

Completeness: Any object $\langle a, x, y, q\rangle \in C_{new}$ will either have a match in the C_{old} or it does not. This is because a predicate P constructed from $\{X \cup Y\}$ will evaluate to TRUE or FALSE on C_{new} and C_{old} resulting in 'match' or 'no match'. The algorithm adds all the unmatched objects of new information to the result {Case 2}. The remaining objects have a match in C_{old}. All the objects with a match in C_{old} are handled by {Case 3}. The Equation (I) incorporates all the matching objects of C_{new} into the updating of existing probabilities for $X=x_i$, i=1, 2, 3, …,m. This shows that algorithm is complete.

Consistency: This algorithm does not violate the axioms of probability. To prove this one needs to show that (i) The new probabilities are >= 0, (ii) sum of the probabilities assigned to the set of objects with the same address component is <=1, and

(iii) the total probability assigned to two disjoint sets of objects is the sum of the probability masses in those two sets.

To show that first axiom holds, one shows that each case satisfies this axiom. The Case 1 replaces all the existing objects with new objects. The Case 2 adds unmatched new objects to the result. If new information is valid it should satisfy the first axiom, hence the updated information in these two cases. The Case 3 uses Equation (I) to update the probabilities. By noting the three terms in that equation are all nonnegative numbers, one concludes Case 3 satisfies first axiom. The Case 4 uses Equation (II) to update the probabilities. The P_{oldRes} and P_{newRes} should both be nonnegative numbers because sum of probabilities assigned to an address component cannot be > 1. All the terms used in Equation (II) are nonnegative, hence the result of this equation as well. This shows that all the probabilities in the result are nonnegative.

Let us suppose Q_t is the sum of probabilities of all the new objects. Since C_{new} is valid, $Q_t \geq 0$ and $Q_t \leq 1$. If Q_n is the sum of probabilities for unmatched objects and Q_m for matching objects, then $Q_n + Q_m = Q_t$. One observes that the sum of old probabilities q for [A=a,X=xi] for i=1,2,…,m (matched objects) is equivalent to the denominator in Equation (I). Therefore, summation of all the updated probabilities, ΣQ, will become, $\sum_{i=1}^{m} p_i$ which is equal to Q_m. The cases 2 and 3 assign the entire new probability to the existing objects. The remaining probability is $(1-Q_t)$. One also observes that the sum of old probabilities q for [A=a,X≠xi] for i=1,2,…,m (unmatched objects) is equivalent to the denominator in Equation (II). Therefore, summation of all the updated probabilities, ΣQ, is equal to the P_{newRes}. One notes that $P_{newRes} = (1-Q_t)$. Therefore, the total probability assigned to the existing objects by cases 2, 3, and 4 is equal $Q_t + P_{newRes} = Q_t + (1-Q_t) = 1$

(Note that unmatched existing objects include <a, *, *, q>, which collects all the unassigned residual probability, hence the total probability is 1). This shows the algorithm satisfies the second axiom. The preceding argument also shows us that disjoint sets of objects (for example, as partitioned by Cases 2 and 3) contain the total probability assigned to them (because the total of the numerators will become equal to the denominator and hence evaluating to 1, leaving the remaining term). The Equations (I) and (II) are applicable to any disjoint subsets. This shows the algorithm satisfies the third axiom.

Closure: To show the algorithm results in only valid objects, one has to simply observe that updated objects do not have nonnegative probabilities and the sum of all the probabilities is not more than 1. The proofs for completeness and consistency assure us this. Both existing set of objects and new set of objects do not contain value-equivalent objects. Only objects newly created in the result are the new objects that did not have any match with existing objects. If the result contains value-equivalent objects, then these newly created objects should have a match on at least one predicate. This contradiction proves there are no value-equivalent objects in the result. Therefore, one concludes this algorithm always results in valid objects.

Time and Space Complexities of the Algorithm

An analysis of time and space complexities is presented below. The analysis is made on each step in the algorithm separately (please refer to the line numbers in the algorithm), followed by an analysis of overall algorithm. This analysis follows the conventions and uses the results of Knuth (1973). Only the worst case time and space complexities are shown with Big O notation. Where it is not obvious, an average case analysis is also shown.

1. Let us denote S_o as the number of cells in the input cube C_{old}, and S_n as the number of cells in the input cube C_{new}. The input cubes C_{old} and C_{new} are union compatible. Space complexity is $O(S_o)$.

2. Let us denote S_u as the number of cells in the output cube $C_{updated}$. Space complexity is $O(S_u)$.

3.1. The class of L *c* of input cubes is denoted as AXY U pS. Therefore, the lower limit on |A| is 0 and the upper limit is $\max(S_o, S_n)$. The outer for loop will be executed worst case $\max(S_o, S_n)$ times and on average $\max(S_o, S_n)/2$ times.

3.2.1. The calculation of p_{old} can be performed by adding up the number of operations required to compute the individual segments of the equation. Restriction operation requires S_o comparisons for non-indexed cubes. Metric Projection operation on the results of Restriction operation requires the same number of operations, hence it is S_o operations. The aggregation (SUM) operation on these results also requires the same number of operations, hence it is S_o operations. Therefore, this step requires a total of $3*S_o$ operations. Two temporary cubes result in a space complexity of $O(S_o)$. Time complexity of for this step is $O(S_o)$.

3.2.2. This step is similar to 3.2.1 with space complexity of $O(S_n)$ and time complexity of $O(S_n)$.

3.2.3. This restriction operation requires S_n comparisons for non-indexed cubes, therefore the time complexity is $O(S_n)$. Space complexity is $O(1)$.

3.2.4. This requires one comparison operation and hence the time complexity is O(1).

3.2.4.1.1. The union operation combines two separate cubes into one. Therefore, this step has time complexity of $O(S_u+S_n)$ and space complexity of $O(S_u+S_n)$.

3.2.4.1. This block (3.2.4.1 to 3.2.4.2) has only one step, 3.2.4.1.1, hence the complexities are same for this step.

3.2.4. If the condition is true, then this step will have the same complexity as that of 3.2.4.1.1., therefore, it has time complexity of $O(S_u+S_n)$ and space complexity of $O(S_u+S_n)$.

3.2.6.1. The lower limit for the restriction operator is 0 and the upper limit is S_n. Therefore, this loop will be executed worst case S_n times, and average case $S_n/2$.

3.2.6.2.1. This union operator requires S_u+1 units of space. Its space complexity is $O(S_u)$. It adds one object to the existing cube. Therefore, its time complexity is O(1).

3.2.6.2.2. This step deletes one object from new input cube, therefore its space complexity is $O(S_n)$ and time complexity is O(1).

3.2.6.2. Space complexity for this block is $O(S_u+S_n)$ and time complexity is O(1).

3.2.6.1. This loop gets executed S_n times on a block with $O(S_u+S_n)$, therefore, its time complexity is $O(S_uS_n + S_n^2)$. Space complexity is $O(S_n)$.

3.2.6.4. This for loop can be executed as many as S_o times on worst case and $S_o/2$ on average case.

3.2.6.5.1. Construction of predicates for $\{x \cup y\}$ of C_{old} requires on worst case S_o operations. This operation has time complexity of $O(S_o)$ and space complexity of $O(S_o)$.

3.2.6.5.2. Sort operation using Quick Sort algorithm on a list of predicates of size $O(S_o)$ takes $O(S_o^2)$. Average case time complexity for this step is $O(S_o * \log_2 S_o)$. Space complexity is $O(S_o)$.

3.2.6.5. This block has worst case time complexity of $O(S_o^2)$ and space complexity of $O(S_o)$.

3.2.6.4. This for loop has a time complexity of $O(S_o * S_o^2) = O(S_o^3)$ and a space complexity of $O(S_o)$.

3.2.6.7. This for loop gets executes as many times as the number of objects in C_{old}, which is S_o. It also requires a search for each object within SP. Using binary search requires time complexity of $O(\log_2 S_o)$.

3.2.6.9.1. This operation requires constant time and stores one number, therefore its time complexity is O(1) and space complexity of O(1).

3.2.6.9.2. This union operator requires S_u+1 units of space. Its space complexity is $O(S_u)$. It adds one object to the existing cube. Therefore, its time complexity is O(1).

3.2.6.9. This block requires worst case time complexity of $O(\log_2 S_o)$ and worst case space complexity of $O(S_u)$.

3.2.6.7. The worst case time complexity for this loop is $O(\log_2 S_o * \log_2 S_o) = O((\log_2 S_o)^2)$ and space complexity of $O(S_u)$.

3.2.6.11. This for loop gets executes as many times as the number of objects in C_{old}, which is S_o. It also requires a search for each object within SP. Using binary search requires time complexity of $O(\log_2 S_o)$.

3.2.6.12.1. This operation requires constant time and stores one number, therefore its time complexity is O(1) and space complexity of O(1).

3.2.6.12.2. This union operator requires S_u+1 units of space. Its space complexity is O(S_u). It adds one object to the existing cube. Therefore, its time complexity is O(1).

3.2.6.12. This block requires worst case time complexity of O($\log_2 S_o$) and worst case space complexity of O(S_u).

3.2.6.11. The worst case time complexity for this loop is O($\log_2 S_o * \log_2 S_o$) = O($(\log_2 S_o)^2$) and space complexity of O(S_u).

3.2.6. The for loop of 3.2.6.1 gets executed worst case S_n times, and average case S_n/2. The blocks 3.2.6.2., 3.2.6.4., 3.2.6.7., and 3.2.6.11. have worst case time complexity of O(S_u+S_n), O(S_o^3), O($(\log_2 S_o)^2$), and O($(\log_2 S_o)^2$) respectively. Therefore, overall time complexity for this block is O(S_n * (S_u+ S_n + S_o^3 + $(\log_2 S_o)^2$ + $(\log_2 S_o)^2$)) = O(S_nS_u + S_n^2 + S_n*S_o^3 + S_n*$(\log_2 S_o)^2$). S_n*S_o^3 is greater than S_nS_u, and S_nS_u is greater than S_n^2. Therefore, this reduces to O(S_nS_u + S_n*$(\log_2 S_o)^2$). This step has space complexity of O(So)+O(Su). Since, S_u is greater than or equal to S_o, one can consider this as O(S_u).

3.2. This step is composed of functional steps 3.2.1., 3.2.2., 3.2.3., 3.2.4., and 3.2.6. They have time complexities of O(S_o), O(S_n), O(S_n), O(S_u+S_n), and O(S_nS_u + S_n*$(\log_2 S_o)^2$) respectively. The time complexity for this step is O(S_o) + O(S_n) + O(S_n) + O(S_u+S_n) + O(S_nS_u + S_n*$(\log_2 S_o)^2$), which can be simplified to O(S_nS_u + S_n*$(\log_2 S_o)^2$) by ignoring the smaller terms. The space complexity for this block is same as that of 3.2.6., with O(S_u).

3.1. The worst case time complexity for this step is calculated by multiplying the loop count, max(So,Sn) with worst case time complexity for the body, 3.2. Therefore, it is max(S_o,S_n) * O(S_nS_u + S_n*$(\log_2 S_o)^2$). If one assumes S_o is larger then, it is O($S_oS_nS_u$ + S_oS_n*$(\log_2 S_o)^2$). Otherwise, it is O($S_uS_n^2$ + S_n^2*$(\log_2 S_o)^2$). The worst case space complexity is same as 3.2., which is O(S_u). By recalling that S_u is the number of objects in the updated result cube, intuitively, this algorithm requires enough space to store the updated cube. This step is the only higher level step in the outer most block of the algorithm, therefore same worst case time and space complexities apply to the whole algorithm.

The above complexity analysis is for update algorithm only. This complexity analysis shows that the update of probabilistic multidimensional data can be performed in a finite amount of time using the least amount of space to store results. However, it is possible to rewrite most algorithms to improve either time or space complexity by compensating one with the other.

Application of Model in Business Management

In this section, a fictitious business management application is used to describe the abstract mathematical model better. Even though this is a fictitious application, it is described with detail as close to general market conditions as possible. This application is for a fictitious grocery retail chain, which has several retail stores throughout the nation. General grocery store inventory management is done by category, referred to as *category* management. Category management involves dividing an entire store inventory into different categories such as fresh foods, soft drinks, cereals, and snacks. The rationale behind this is that consumers are more interested in a category of groceries than in a particular brand name product. For example, consider a consumer shopping for a party. Consumer may be more interested in buying soft drinks than in Coke, Pepsi, or any other brand name product in particular. A primary advantage of this method of inventory management is that it does not require forecasting based on individual products within the category, therefore eliminating the requirement to capture and process data for individual products. Categorization of inventory simplifies data collection, storage, and analysis, and demand forecasting (Mantrala & Raman, 1999; Symbol, 2004). However, this simplification also results in loss of forecast accuracy and prevents inclusion of product specific promotion and competition information in forecasting process.

The most common process for inventory management, according to Symbol (2004) is as follows:

> The most common reordering process in use today relies entirely on human estimations. Using a handheld mobile computer with integrated bar code scanner, the department or night crew manager walks the store isles scanning bar code labels on shelves or products. Most systems enter an order for a single case of each item scanned unless a larger order quantity is manually entered. When all needed products have been ordered, the mobile computer is connected to a phone line and the order is sent to the store's distribution center. The entire process can take 3-4 hours if done correctly. Normally, orders completed and sent to the distribution center by nine o'clock in the morning arrive at the store by seven o'clock in the evening so the night crew can restock the shelves (p.1).

This method of reordering heavily relies on the experience and educated guess of a single person. It could lead to inaccurate demand forecasts resulting in either overstocks or out-of-stocks (OOS). Overstocking costs money in terms of capital investment and shelf space. OOSs result in dissatisfied customers, leading them to competitors. Demand is dependent on several parameters such as category, price, promotion, competition, historic sales, new products, and even weather. Different demand forecast methods use different sets of these parameters in forecasting. Researchers found that forecast based on even primitive analysis of historical data (e.g. forecast based on average for previous week), is generally better than a pure guess (Foote & Krishnamurthi, 2001; Lancaster & Lomas, 1986; Symbol, 2004). Thus, ability to store and analyze historical sales data is a critical component of all these forecasting methods.

Consider using a decision support system for inventory management, which can store and analyze large amounts of data. Due to this capability to store large amounts of data, historical sales data for each individual product can be maintained. This facilitates data analysis and demand forecast by product instead of category. The actual category inventory itself simply becomes an aggregation of individual product inventory. Forecasting by individual product also facilitates incorporation of product specific promotion information. For example, if a superbowl promotion advertisement for Diet Coke is running, it is possible to use this information in forecasting sales for Diet Coke. Even though overall soft drinks category sales may not increase, it can be assumed that Diet Coke sales will be higher due to the promotion. This information can be used also to adjust the sales forecast for other products in the soft drinks category (Lancaster & Lomas, 1986).

Forecast method

The forecast method used for this fictitious business application uses historical data, product promotion data, and competition data. For the purpose of this application, *forecast for demand* is considered *forecast for sales*, and these two phrases are used interchangeably. The forecast method utilizes heuristic rules specified below to forecast demand for each product in the category for the next day.

1. Average sales for the previous week (seven days): Demand is directly proportional to average sales M.
2. Promotion data for the previous week: Promotional factor, P, multiplies M. P is derived by incorporating total number of promotions.
3. Competition data (from other retailers in area as reported by agents) for the previous week: Competition, C, decreases sales.
4. Category sales forecast information: Final category demand, T_f, is equal to average category demand within the previous four weeks, T_a. This means, if tentative category demand, T_t (calculated by aggregating individual product demand in that category), is different from T_a, then the individual product demand has to be adjusted to make T_t equal to T_a.

To forecast demand using the above forecast method, the following information is needed:

1. Sales data for the previous week for the category;
2. Product promotion data for the previous week;
3. Competitor sales information for the previous week, as reported by agents; and
4. Daily category sales forecasts for the previous week.

Assume that this fictitious grocery retail store chain has 40 stores selling 19,000 products averaging 87,000 transactions per day in each store. That is approximately 3.5 million transactions per day and 97.5 million transactions every four weeks, throughout the chain. Assume sales data captured contains product, time, location, price, and amount

(These attributes are similar to the example cube schema attributes used in Chapter 1). If 64 bytes are required to store each of these attributes, the total amount of space required to store the data is 215 MB per day or 6 GB for four weeks. Researchers and practitioners recommend storing data for at least 65 weeks (Foote & Krishnamurthi, 2001), which would require a capacity of about 100 GB to store sales data. Using similar assumptions and requirements for promotion data and competition data, this fictitious retail chain would require about 605 MB per day or 18 GB to store data for 4 weeks or 300 GB to store data for 65 weeks. Due to the large size of these data, only a tiny sample of them will be used to demonstrate usefulness of the PMDDM model.

Sales Data

The sales data cube C_{Sales} is defined as:

- The characteristic set C = {TIME, PRODUCT, LOCATION, SALES, BELIEF}, (m = 5)
- The attribute set A = {day, month, year, product_name, city, state, price, quantity, pS}, (t = 9)
- schema of C:
  - f(TIME) = {day, month, year}
  - f(PRODUCT) = {product_name}
  - f(LOCATION) = {city, state}
  - f(SALES) = {price, quantity}
  - f(BELIEF) = {pS}
- dimension function d:
  - d (TIME) = 1 i.e., TIME is a dimension

  - d (PRODUCT) = 1 i.e., PRODUCT is a dimension
  - d (LOCATION) = 1 i.e., LOCATION is a dimension
  - d (SALES) = 0 i.e., SALES is a measure
  - d (BELIEF) = 0 i.e., BELIEF is a measure

- A partial order on the Sales cube is as follows:
  - O_{TIME} = {⟨day, month⟩, ⟨day, year⟩, ⟨month, year⟩}
  - $O_{PRODUCT}$ = {}
  - $O_{LOCATION}$ = {⟨city, state⟩}
  - O_{SALES} = {}
  - O_{BELIEF} = {}
- L is as follows:
  - Let us assume the following domains for the attributes
    - A = {year, month, day, product_name, city, state, price, quantity, pS}
    - dom year = {2004, 2003, 2002, 2001}
    - dom product_name = {DIET PEPSI, PEPSI, COKE}
    - dom city = {Boston, New York, Dallas, San Francisco, Chicago}
    - dom state = {MA, NY, TX, CA, IL}
    - dom price = {0, 1, 2, …}
    - dom quantity = {0, 1, 2, …}
    - pS = 1 (Belief strength of all the cells represented is true, making this a deterministic cube)

Sales data for the last 7 days for DIET PEPSI is specified as follows.

C_{Sales} = {⟨⟨2004,01,01, DIET PEPSI, Boston, MA⟩, ⟨100, 10⟩⟩,
⟨⟨2004,01,02, DIET PEPSI, Boston, MA⟩, ⟨125, 10⟩⟩,
⟨⟨2004,01,03, DIET PEPSI, Boston, MA⟩, ⟨150, 15⟩⟩,
⟨⟨2004,01,04, DIET PEPSI, Boston, MA⟩, ⟨110, 15⟩⟩,
⟨⟨2004,01,05, DIET PEPSI, Boston, MA⟩, ⟨160, 20⟩⟩,
⟨⟨2004,01,06, DIET PEPSI, Boston, MA⟩, ⟨130, 15⟩⟩,
⟨⟨2004,01,07, DIET PEPSI, Boston, MA⟩, ⟨180, 30⟩⟩,
⟨⟨2004,01,01, DIET PEPSI, New York, NY⟩, ⟨105, 10⟩⟩,
⟨⟨2004,01,02, DIET PEPSI, New York, NY⟩, ⟨125, 10⟩⟩,
⟨⟨2004,01,03, DIET PEPSI, New York, NY⟩, ⟨105, 20⟩⟩,
⟨⟨2004,01,04, DIET PEPSI, New York, NY⟩, ⟨140, 5⟩⟩,
⟨⟨2004,01,05, DIET PEPSI, New York, NY⟩, ⟨110, 20⟩⟩,
⟨⟨2004,01,06, DIET PEPSI, New York, NY⟩, ⟨120, 15⟩⟩,
⟨⟨2004,01,07, DIET PEPSI, New York, NY⟩, ⟨110, 10⟩⟩,
⟨⟨2004,01,01, DIET PEPSI, Chicago, IL⟩, ⟨200, 10⟩⟩,
⟨⟨2004,01,02, DIET PEPSI, Chicago, IL⟩, ⟨105, 10⟩⟩,
⟨⟨2004,01,03, DIET PEPSI, Chicago, IL⟩, ⟨110, 25⟩⟩,
⟨⟨2004,01,04, DIET PEPSI, Chicago, IL⟩, ⟨150, 10⟩⟩,
⟨⟨2004,01,05, DIET PEPSI, Chicago, IL⟩, ⟨100, 20⟩⟩,
⟨⟨2004,01,06, DIET PEPSI, Chicago, IL⟩, ⟨100, 15⟩⟩,
⟨⟨2004,01,07, DIET PEPSI, Chicago, IL⟩, ⟨130, 30⟩⟩,
⟨⟨2004,01,01, DIET PEPSI, San Francisco, CA⟩, ⟨190, 10⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, San Francisco, CA⟩, ⟨165, 10⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, San Francisco, CA⟩, ⟨155, 25⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, San Francisco, CA⟩, ⟨145, 15⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, San Francisco, CA⟩, ⟨130, 20⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, San Francisco, CA⟩, ⟨115, 25⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, San Francisco, CA⟩, ⟨135, 25⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Dallas, TX⟩, ⟨100, 10⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Dallas, TX⟩, ⟨125, 10⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Dallas, TX⟩, ⟨100, 15⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Dallas, TX⟩, ⟨140, 5⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Dallas, TX⟩, ⟨110, 10⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Dallas, TX⟩, ⟨130, 15⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Dallas, TX⟩, ⟨120, 10⟩⟩}

Similarly, the sales data for other individual products in this category is specified. Since the forecast method needs the average sales information, the Cube operations can be used to calculate this.

1. Average Price of DIET PEPSI in Boston, MA for the duration 2004/01/01-2004/01/07

$= \mathit{\Gamma}_{[\text{AVG}, \{\text{product\_name,city,state }\}, \text{price}]} \left(\Sigma_{(\text{year=2004} \wedge \text{month=01} \wedge \text{day} \geq \text{01} \wedge \text{day} \leq \text{01} \wedge \text{product\_name='DIET PEPSI'} \wedge \text{state='MA'} \wedge \text{city='Boston'})}(\text{C}_{\text{Sales}})\right)$

$= 136.43$

Average quantity of DIET PEPSI sold in Boston, MA for the duration 2004/01/01-2004/01/07

$$= \Gamma_{[\text{AVG},\{\text{product\_name,city,state }\},\text{quantity}]}\ (\Sigma_{(\text{year}=2004\ \wedge\ \text{month}=01\ \wedge\ \text{day}\geq 01\ \wedge\ \text{day}\leq 01\ \wedge\ \text{product\_name}=\text{'DIET PEPSI'}\ \wedge\ \text{state}=\text{'MA'}\ \wedge\ \text{city}=\text{'Boston'})}(C_{\text{Sales}}))$$

$= 16.43$

2. Average Price of DIET PEPSI in New York, NY for the duration 2004/01/01-2004/01/07

$$= \Gamma_{[\text{AVG},\{\text{product\_name,city,state }\},\text{price}]}\ (\Sigma_{(\text{year}=2004\ \wedge\ \text{month}=01\ \wedge\ \text{day}\geq 01\ \wedge\ \text{day}\leq 01\ \wedge\ \text{product\_name}=\text{'DIET PEPSI'}\ \wedge\ \text{state}=\text{'NY'}\ \wedge\ \text{city}=\text{'New York'})}(C_{\text{Sales}}))$$

$= 116.43$

Average quantity of DIET PEPSI sold in New York, NY for the duration 2004/01/01-2004/01/07

$$= \Gamma_{[\text{AVG},\{\text{product\_name,city,state }\},\text{quantity}]}\ (\Sigma_{(\text{year}=2004\ \wedge\ \text{month}=01\ \wedge\ \text{day}\geq 01\ \wedge\ \text{day}\leq 01\ \wedge\ \text{product\_name}=\text{'DIET PEPSI'}\ \wedge\ \text{state}=\text{'NY'}\ \wedge\ \text{city}=\text{'New York'})}(C_{\text{Sales}}))$$

$= 12.86$

3. Average Price of DIET PEPSI in Chicago, IL for the duration 2004/01/01-2004/01/07

$$= \Gamma_{[\text{AVG},\{\text{product\_name,city,state }\},\text{price}]}\ (\Sigma_{(\text{year}=2004\ \wedge\ \text{month}=01\ \wedge\ \text{day}\geq 01\ \wedge\ \text{day}\leq 01\ \wedge\ \text{product\_name}=\text{'DIET PEPSI'}\ \wedge\ \text{state}=\text{'IL'}\ \wedge\ \text{city}=\text{'Chicago'})}(C_{\text{Sales}}))$$

$= 127.86$

Average quantity of DIET PEPSI sold in Chicago, IL for the duration 2004/01/01-2004/01/07

$$= \mathit{\Gamma}_{[\text{AVG},\{\text{product\_name,city,state }\},\text{quantity}]}\left(\Sigma_{(\text{year}=2004 \wedge \text{month}=01 \wedge \text{day}\geq 01 \wedge \text{day}\leq 01 \wedge \text{product\_name}=\text{'DIET PEPSI'} \wedge \text{state}=\text{'IL'} \wedge \text{city}=\text{'Chicago'})}(C_{\text{Sales}})\right)$$

= 17.14

4. Average Price of DIET PEPSI in San Francisco, CA for the duration 2004/01/01-2004/01/07

$$= \mathit{\Gamma}_{[\text{AVG},\{\text{product\_name,city,state }\},\text{price}]}\left(\Sigma_{(\text{year}=2004 \wedge \text{month}=01 \wedge \text{day}\geq 01 \wedge \text{day}\leq 01 \wedge \text{product\_name}=\text{'DIET PEPSI'} \wedge \text{state}=\text{'CA'} \wedge \text{city}=\text{'San Francisco'})}(C_{\text{Sales}})\right)$$

= 147.86

Average quantity of DIET PEPSI sold in San Francisco, CA for the duration 2004/01/01-2004/01/07

$$= \mathit{\Gamma}_{[\text{AVG},\{\text{product\_name,city,state }\},\text{quantity}]}\left(\Sigma_{(\text{year}=2004 \wedge \text{month}=01 \wedge \text{day}\geq 01 \wedge \text{day}\leq 01 \wedge \text{product\_name}=\text{'DIET PEPSI'} \wedge \text{state}=\text{'CA'} \wedge \text{city}=\text{'San Francisco'})}(C_{\text{Sales}})\right)$$

= 18.57

5. Average Price of DIET PEPSI in Dallas, TX for the duration 2004/01/01-2004/01/07

$$= \mathit{\Gamma}_{[\text{AVG},\{\text{product\_name,city,state }\},\text{price}]}\left(\Sigma_{(\text{year}=2004 \wedge \text{month}=01 \wedge \text{day}\geq 01 \wedge \text{day}\leq 01 \wedge \text{product\_name}=\text{'DIET PEPSI'} \wedge \text{state}=\text{'TX'} \wedge \text{city}=\text{'Dallas'})}(C_{\text{Sales}})\right)$$

= 117.86

Average quantity of DIET PEPSI sold in Dallas, TX for the duration 2004/01/01-2004/01/07

$= \Gamma_{[\text{AVG},\{\text{product\_name,city,state }\},\text{quantity}]} (\Sigma_{(\text{year=2004} \wedge \text{month=01} \wedge \text{day}\geq 01 \wedge \text{day}\leq 01 \wedge \text{product\_name='DIET PEPSI'} \wedge \text{state='TX'} \wedge \text{city='Dallas'})}(C_{\text{Sales}}))$

$= 10.71$

The result of these aggregation operations to calculate average price and quantity can be organized in cube format with cell structure ⟨⟨product_name, city, state⟩, ⟨Average_Price, Average_Quantity⟩⟩ as follows:

$C_{\text{AverageSales}}$ = {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨136.43, 16.43⟩⟩,
⟨⟨DIET PEPSI, New York, NY⟩, ⟨116.43, 12.86⟩⟩,
⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨127.86, 17.14⟩⟩,
⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨147.86, 18.57⟩⟩,
⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨117.86, 10.71⟩⟩}

Promotion Data

The Promotion data cube C_{Promo} is defined as:

- The characteristic set C = {TIME, PRODUCT, LOCATION, PROMOTION, BELIEF}, (m = 5)
- The attribute set A = {day, month, year, product_name, city, state, promotion_type, number, pS}, (t = 9)
- schema of C:
  - f(TIME) = {day, month, year}
  - f(PRODUCT) = {product_name}
  - f(LOCATION) = {city, state}

  - f(PROMOTION) = {promotion_type, number}
  - f(BELIEF) = {pS}
- dimension function d:
  - d (TIME) = 1 i.e., TIME is a dimension
  - d (PRODUCT) = 1 i.e., PRODUCT is a dimension
  - d (LOCATION) = 1 i.e., LOCATION is a dimension
  - d (PROMOTION) = 0 i.e., PROMOTION is a measure
  - d (BELIEF) = 0 i.e., BELIEF is a measure
- A partial order on the Promotion cube is as follows:
  - O_{TIME} = {⟨day, month⟩, ⟨day, year⟩, ⟨month, year⟩}
  - $O_{PRODUCT}$ = {}
  - $O_{LOCATION}$ = {⟨city, state⟩}
  - $O_{PROMOTION}$ = {}
  - O_{BELIEF} = {}
- L is as follows:
  - Let us assume the following domains for the attributes
    - A = {year, month, day, product_name, city, state, promotion_type, number, pS}
    - dom year = {2004,2003,2002,2001}
    - dom product_name = {DIET PEPSI, PEPSI, COKE}
    - dom city = {Boston, New York, Dallas, San Francisco, Chicago}
    - dom state = {MA, NY, TX, CA, IL}

- dom promotion_type = {1=Aisle_Display, 2=Front_Display, 3=Discount_Coupon, 4=SuperBowl_Ad}
- dom number = {0, 1, 2, …}
- pS = 1 (Belief strength of all the cells represented is true, making this a deterministic cube)

Promotion data for the last 7 days for DIET PEPSI is specified as follows.

C_{Promo} = {⟨⟨2004,01,01, DIET PEPSI, Boston, MA⟩, ⟨1, 10⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨2, 10⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨1, 15⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨3, 15⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨2, 10⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston, MA ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, New York, NY⟩, ⟨1, 10⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, New York, NY ⟩, ⟨2, 10⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, New York, NY ⟩, ⟨1, 20⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, New York, NY ⟩, ⟨3, 5⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, New York, NY ⟩, ⟨3, 20⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, New York, NY ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, New York, NY ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Chicago, IL⟩, ⟨2, 10⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Chicago, IL ⟩, ⟨1, 10⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Chicago, IL ⟩, ⟨1, 25⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Chicago, IL ⟩, ⟨1, 10⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Chicago, IL ⟩, ⟨3, 20⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Chicago, IL ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Chicago, IL ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, San Francisco, CA⟩, ⟨1, 10⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, San Francisco, CA ⟩, ⟨2, 10⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, San Francisco, CA ⟩, ⟨3, 25⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, San Francisco, CA ⟩, ⟨2, 15⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, San Francisco, CA ⟩, ⟨1, 20⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, San Francisco, CA ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, San Francisco, CA ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Dallas, TX⟩, ⟨1, 10⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Dallas, TX ⟩, ⟨1, 10⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Dallas, TX ⟩, ⟨2, 15⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Dallas, TX ⟩, ⟨3, 5⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Dallas, TX ⟩, ⟨2, 10⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Dallas, TX ⟩, ⟨4, 1⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Dallas, TX ⟩, ⟨4, 1⟩⟩}

Similarly, the promotion data for other individual products in this category is specified. The following calculation of the promotional factors for all the cities for each product shows the versatility of the cube operations by retrieving all the values in a cube format.

$C_{PromoFactors} = \Gamma_{[PromoFactorFunction,\{product\_name,city,state\},PF]}(C_{Promo})$

where,

PF is the Promotional Factor, and

PromoFactorFunction is the aggregate function defined to calculate the promotional factor PF for each grouping on temporary cube C_{PF}. For this fictitious application, promotional factor function is defined as follows:

Predicate P is constructed from the grouping attributes.

$$C_{temp} = \prod^{M}_{promo\_type,number} \sum_{(P=product\_name=?\wedge state=?\wedge city=?)} (C_{Promo})$$

$$C_{PF} = \Gamma_{[PROMO\_FORMULA=(1+((promo\_type*10)+number)/100),\{promo\_type,number\},PF]}(C_{temp})$$

$$\text{PromotionFactorFunction} = \sum \prod^{M}_{PF} (C_{PF})$$

Using this aggregate function definition for Boston city with Predicate P=(product_name='DIET PEPSI'^city='Boston'^state='MA') results in a promotional factor of 2.32. Similarly, it will result in 2.47 for New York, 2.37 for Chicago, 2.52 for San Francisco, and 2.22 for Dallas. The actual result of the aggregation operation to calculate promotional factors in cube format will be:

$C_{PromoFactors}$ = {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨2.32⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨2.52⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨2.47⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨2.37⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨2.22⟩⟩}

Competition Data

The competition analysis for a product can be very challenging in real world situations. This is because the competition is multi-faceted for an individual product. All the products with similar utility value compete. At the same time brand names try to add value to the individual product within their brand. In addition, general competition comes from other sellers of the same product. Because of the complexity involved in competition analysis, only the competition from other retailers is used for this fictitious application. To calculate the final demand, the sales of other retailers are guessed and deducted from the preliminary estimates calculated from historical sales data and promotion data. The agents are allowed to guess the competitors' sales and to associate strength of belief with each guess. The Probabilistic Multidimensional Data Model is capable of storing the data containing uncertainty. To store the competition data, the 'Competition cube' is defined as:

- The characteristic set C = {TIME, PRODUCT, LOCATION, SALES, BELIEF}, (m = 5)
- The attribute set A = {day, month, year, product_name, city, state, quantity, amount, pS}, (t = 9)
- schema of C:
    - f(TIME) = {day, month, year}
    - f(PRODUCT) = {product_name}
    - f(LOCATION) = {city, state}
    - f(SALES) = {quantity, amount}
    - f(BELIEF) = {pS}

- dimension function d:
  - d (TIME) = 1 i.e., TIME is a dimension
  - d (PRODUCT) = 1 i.e., PRODUCT is a dimension
  - d (LOCATION) = 1 i.e., LOCATION is a dimension
  - d (SALES) = 0 i.e., SALES is a measure
  - d (BELIEF) = 0 i.e., BELIEF is a measure
- A partial order on the Competition cube is as follows:
  - O_{TIME} = {⟨day, month⟩, ⟨day, year⟩, ⟨month, year⟩}
  - $O_{PRODUCT}$ = {}
  - $O_{LOCATION}$ = {⟨city, state⟩}
  - O_{SALES} = {⟨quantity, amount⟩}
  - O_{BELIEF} = {All partial orders defined on real numbers}
- L is as follows:
  - Let us assume the following domains for the attributes
    - A = {year, month, day, product_name, city, state, quantity, amount, pS}
    - dom year = {2004,2003,2002,2001}
    - dom product_name = {DIET PEPSI, PEPSI, COKE}
    - dom city = {Boston, New York, Dallas, San Francisco, Chicago}
    - dom quantity = {0, 1, 2, …}
    - dom amount = {0, 1, 2, …}
    - dom pS = {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

Competition data for the previous seven days are required to calculate final demand. However, due to the large size of these data, only one day’s data for DIET PEPSI are specified below. Appendix X contains data for the previous seven days. These data are collected from reports agents prepared based on their guesswork, and contain strength of belief for each statement or guess. These data are formulated as cube, with the above definition. For each cell, total belief strength is less than or equal to 1.

$C_{Competition}$ = {⟨⟨2004,01,01, DIET PEPSI, Boston, MA⟩, ⟨110, 10, 0.2⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨120, 10, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨120, 5, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨120, 15, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨130, 10, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston, MA ⟩, ⟨130, 15, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨125, 10, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨120, 10, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨120, 5, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨120, 15, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨130, 10, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston, MA ⟩, ⟨130, 15, 0.2⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨120, 10, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨120, 5, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨120, 15, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨130, 10, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨130, 15, 0.2⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston, MA ⟩, ⟨150, 15, 0.1⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.7⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨110, 5, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨120, 10, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨120, 5, 0.05⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨120, 15, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨130, 10, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston, MA ⟩, ⟨130, 15, 0.1⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨160, 20, 0.3⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.2⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨190, 10, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨120, 5, 0.05⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨120, 15, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨130, 10, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨130, 5, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston, MA ⟩, ⟨130, 15, 0.2⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨130, 15, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨110, 5, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨120, 10, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨120, 5, 0.02⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨120, 15, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨130, 10, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨130, 5, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨130, 15, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston, MA ⟩, ⟨140, 10, 0.9⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston, MA ⟩, ⟨180, 30, 0.8⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston, MA ⟩, ⟨110, 10, 0.02⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston, MA ⟩, ⟨110, 5, 0.02⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston, MA ⟩, ⟨110, 15, 0.06⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston, MA ⟩, ⟨120, 10, 0.1⟩⟩}

Similarly, competition data for other individual products in this category is specified.

The above data cannot be represented using a deterministic data model. Users of these data will be able to formulate queries using PMDDM operations. For example, a query can be formulated to get maximum sales, while another can be used to attract sales with highest confidence. This ability to query data provides flexibility in their use.

To forecast demand using both method and data, the preliminary demand forecast has to be calculated using historical and promotion data. According to the forecast method, demand is directly proportional to the previous week's average sales multiplied by the promotional factors. Therefore, preliminary demand forecast cube, $C_{PreForecast}$, is given by sequence of operations below:

1. $C_1 = C_{AverageSales}\ \Theta_P\ C_{PromoFactors}$

Where P is the join predicate made of product_name, city, and state

C_1 = {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨136.43, 16.43, 2.32⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨116.43, 12.86, 2.47⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨127.86, 17.14, 2.37⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨147.86, 18.57, 2.52⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨117.86, 10.71, 2.22⟩⟩}

2. C_2 =

$\Gamma_{[\text{PRE\_FORECAST1=(average\_price*promo\_factor),\{average\_price,promo\_factor\},PRE\_PRICE}]}(C_1)$

= {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨316.52⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨287.58⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨303.03⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨372.61⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨261.65⟩⟩}

3. C_3 =

$\mathit{\Gamma}_{[\text{PRE\_FORECAST2=(average\_quantity*promo\_factor),\{average\_quantity,promo\_factor\},PRE\_QUANT}]}(C_1)$

= {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨38.12⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨31.76⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨40.62⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨46.80⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨23.78⟩⟩}

4. $C_4 = C_2 \Theta_P C_3$

= {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨316.52, 38.12⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨287.58, 31.76⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨303.03, 40.62⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨372.61, 46.80⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨261.65, 23.78⟩⟩}

5. $C_{\text{PRE\_FORECAST}} = \mathit{\Lambda}_{\mathit{PRE\_FORECAST}}(C_4)$ - This renaming operation just changes the attribute names to PRE_FORECAST references without any data changes to the result obtained in step 4 above.

This preliminary demand forecast is based on the previous week’s average sales and promotional data and needs to be adjusted by deducting competitors’ sales. Since competition data are uncertain, there are several choices available. For this fictitious application, *highest most likely sales* and *maximum sales* will be used separately to derive two different forecasts. The first forecast uses the highest most likely sales and the second forecast uses maximum sales.

First Forecast: Most likely sales are retrieved using a query like “Select most likely sales from Competition cube.” This requires a fuzzy membership function defined for this cube mapping fuzzy sets *certain, most likely, very likely, likely, unlikely,* and *very unlikely* to crisp sets 1.00, 0.99-0.70, 0.75-0.55, 0.60-0.40, 0.45-0.25, and 0.30-0.00. Using this mapping *most likely* is described with a strength of belief between 0.99 – 0.75. Therefore, the selection predicate to get *most likely* sales will be “pS >= 0.75 and pS < 1.00”. This selects cells with probability greater than 0.75. Among resultant cells, one can select the cell with the highest quantity to get the highest most likely sales. For the city of Boston, this operation on the competition cube will be.

$C_{COMP\_QUANT} = \Gamma_{[MAX,\{product\_name,city,state\},quantity]}(\Sigma_{(P=(pS >= 0.75 \wedge pS < 1.00))}(C_{Competition}))$

= $\Gamma_{[MAX,\{product\_name,city,state,quantity\},quantity]}$ {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨180, 30, 0.8⟩⟩, ⟨⟨DIET PEPSI, Boston, MA⟩, ⟨140, 10, 0.9⟩⟩}

= {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨180⟩⟩}

Similar operations for other cities on the data presented in Appendix B result in the following highest most likely sales:

$C_{COMP\_QUANT} = \Lambda_{COMP\_QUANT}$ {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨180⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨160⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨120⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨130⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨110⟩⟩}

To get the demand forecast, C_{COMP_QUANT} quantities need to be subtracted from $C_{PRE\_FORECAST}$ quantities, using the operation below:

$C_{FINAL\_FORECAST} = \Gamma_{[FIN\_FORECAST=(PRE\_FORECAST.quantity - COMP\_QUANT.quantity),\{product\_name,city,state\},quantity]}(C_{PRE\_FORECAST}\ \Theta_P\ C_{COMP\_QUANT})$

= {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨136.52, 38.12⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨127.58, 31.76⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨183.03, 40.62⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨242.61, 46.80⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨151.65, 23.78⟩⟩}

Second Forecast: This forecast uses the maximum sales quantity estimates for the competitors. To calculate the maximum sales, the following operation on the Competition cube can be used:

$C_{COMP\_QUANT} = \Gamma_{[MAX,\{product\_name,city,state\},quantity]}(C_{Competition})$

= {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨190, 10, 0.01⟩⟩}

Note that the maximum sales estimate of 190 for Boston city has very little likelihood. Similar operations for other cities on the data presented in Appendix B result in the following highest most likely sales:

$C_{COMP\_QUANT} = \Lambda_{COMP\_QUANT}$ {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨190⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨170⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨145⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨140⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨140⟩⟩}

To get the demand forecast, C_{COMP_QUANT} quantities need to be subtracted from $C_{PRE\_FORECAST}$ quantities, using the operation below:

$C_{FINAL\_FORECAST} = \mathit{\Gamma}_{[FIN\_FORECAST=(PRE\_FORECAST.quantity\ -\ COMP\_QUANT.quantity),\{product\_name,city,state\},quantity]}(C_{PRE\_FORECAST}\ \Theta_P\ C_{COMP\_QUANT})$

= {⟨⟨DIET PEPSI, Boston, MA⟩, ⟨126.52, 38.12⟩⟩,

⟨⟨DIET PEPSI, New York, NY⟩, ⟨117.58, 31.76⟩⟩,

⟨⟨DIET PEPSI, Chicago, IL⟩, ⟨158.03, 40.62⟩⟩,

⟨⟨DIET PEPSI, San Francisco, CA⟩, ⟨232.61, 46.80⟩⟩,

⟨⟨DIET PEPSI, Dallas, TX⟩, ⟨121.65, 23.78⟩⟩}

The above result can be used as the final forecast for demand. The above fictitious application described methods of using PMDDM to store different kinds of data, exercising various algebraic operations, and results of these operations. PMDDM is at least as expressive as the relational data model, and hence flexibility provided by it is at least as much as the relational data model provides. This can be seen also from operations on cubes of the above fictitious application.

Bayesian Belief Networks and PMDDM

In this section, a brief description of conditional independence and a formal definition of Bayesian Belief Networks (BBNs) and their properties are presented. Comparison and contrast of PMDDM and BBNs is also presented in this section.

Conditional Independence

In probability theory, the likelihood of a statement holding true or an event taking place is represented as a number between 0 and 1. A 0 represents impossibility and a 1 represents complete certainty. Probability is also expressed as percentage. Probability is generally used to represent uncertainty and frequency. For example, physicians make statements such as, “Smoking causes cancer with a probability of 0.7.” This statement reflects that cause effect relationship between smoking and cancer is not certain, but more likely to be true. Similarly, “There is 80% chance of rain today” or equivalently “The probability of raining today is 0.8” reflects that event is very likely, but not certain. Humans deal with this kind of uncertain information effectively. Probability numbers assigned to these statements and events may be obtained using various methods. Empiricists interpret probabilities as frequencies, while rationalists interpret them as belief strengths (Pearl, 1988). The probability of an event or a set of events, A, is represented as P(A). The probability assignment must satisfy the following four basic axioms of probability theory:

1. $0 \geq P(A) \leq 1$
2. P(FALSE) = 0 & P(TRUE) = 1
3. P(A and B) = 0 & P(A or B) = P(A) + P(B), where A and B are mutually exclusive events
4. If $\{E_1, E_2, E_3, \ldots, E_n\}$ is a mutually exclusive and exhaustive set of events, then $\sum_{i=1}^{n} P(E_i) = 1$

The first axiom indicates that probability is a number between 0 and 1. The second axiom states that probability of an impossible event is 0 while that of a certain event is 1.

The third and fourth axioms are based on sets of events. A set of mutually exclusive events {A,B} implies that A and B do not occur together, hence (from axiom 2), the probability for them to occur is 0. The probability of either one of the events A or B is the sum of their individual probabilities. The fourth axiom restricts the sum of probabilities of all possible events to 1.

The probability of an event may not always be useful or meaningful. Events may not be independent in some circumstances. Suppose one assumes that the chances of pavement being wet are 20%, i.e. P(Pavement being wet)=0.20. Then it is learned that it rained. Now belief in the pavement being wet increases. Bayesian formalism considers this new probability of "pavement being wet" after learning "it rained" as probability of the conditional event, "pavement being wet *given that* it rained." It is written as P(Pavement being wet | It rained). If P(A|B) = P(A), then A and B are said to be *independent*. For example, consider P(Pavement being wet | Gas bill is high). Assessment of this probability would be the same as P(Pavement being wet). That is, the fact that the "gas bill is high" has no impact on the chances of "pavement being wet," therefore these two events are independent. Similarly, if P(A|B AND C) = P(A|C), then A and B are *conditionally independent* given C. For example, consider P(Pavement being wet | Weather forecast called for rain *and* It rained). Assessment of this probability would be the same as P(Pavement being wet | It rained), because the fact that "weather forecast called for rain" does not matter once it is known that "it rained." Therefore, "pavement being wet" is *conditionally independent* of "weather forecast called for rain" *given* "it rained." P(A|B AND C) is also written as P(A|B,C). This information can be represented graphically in several ways. Graph in Figure 3 is an example of graphical representation using directed edges for dependency.

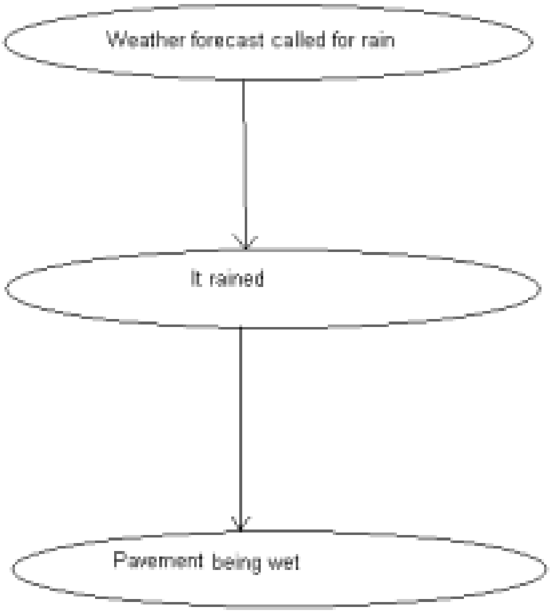

Figure 3. Graphical Representation of Pavement Example.

As can be seen from Figure 3, “It rained” blocks the path between “weather forecast called for rain” and “pavement being wet.” The Bayesian belief networks framework formalizes this concept of graphical representation. The following definitions from Pearl (1988) formally define the Bayesian belief networks framework. In the following definitions, ‘variable’, ‘event’, and ‘node’ are used interchangeably.

Independence Statement: U is Universe of events. An event e_i is statistically independent of another event e_j if $P(e_i \mid e_j) = P(e_i)$. Similarly, e_j is statistically independent of e_i if $P(e_j \mid e_i) = P(e_j)$. If both are true, then $P(e_i e_j) = P(e_i)P(e_j)$, which implies e_j and e_i are mutually statistically independent. Similarly, if $P(e_i, e_j \mid S) = P(e_i \mid S)P(e_j \mid S)$ when $P(S) \neq 0$, then e_i and e_j are statistically independent given S, where S is any subset of U that does not contain e_i and e_j. This is also written as I(e_i, S, e_j) and called an *Independence Statement*.

Dependency Model, Graph Isomorphism, and DAG Isomorphism: A *Dependency Model* is a list M of independence statements of the form I(X,Z,Y), also written as $I(X,Z,Y)_M$.. M is *graph isomorphic* if all independencies in M and no independencies outside M can be represented using an undirected graph G. Similarly, M is *DAG isomorphic* if it can be represented in this manner using a Directed Acyclic Graph (DAG).

Converging Arrows and Diverging Arrows: If a, b, and c are three nodes in a DAG D, structure a→b←c has converging arrows and a←b→c has diverging arrows.

d-separation: If X, Y, and Z are three disjoint subsets of nodes in a DAG D, then Z is said to *d-separate* X from Y, denoted $<X|Z|Y>_D$, if along every path between a node in X and a node in Y there is node w satisfying one of the following two conditions: (1) w has converging arrows and none of w or its descendants are in Z, or (2) w does not have converging arrows and w is in Z.

I-map and Minimal I-map: A DAG D is said to be *I-map* of a dependency model M if every d-separation condition displayed in D corresponds to a valid conditional independence relationship in M, i.e. if for every three disjoint sets of vertices X, Y, and Z we have $<X|Z|Y>_D \Rightarrow I(X,Z,Y)_M$. A DAG is a *minimal I-map* of M if none of its arrows can be deleted without destroying its I-mapness.

Bayesian belief network: Given a probability distribution P on a set of variables V, a DAG D=(V,E) where E is an ordered pair of variables (each of which corresponds to a vertex in graphical representation) of V is called a *Bayesian belief network* of P if and only if D is a minimal I-map of P.

Comparison and Contrast between Bayesian Belief Networks and PMDDM

Bayesian belief networks have been researched since early 1980s. There is a wealth of research conducted on a variety of topics related to Bayesian belief networks. This research includes general introduction (Charniak, 1991; Geiger & Heckerman, 1991; Jaynes & Bretthorst, 2003; Jeffrey, 1983; Jensen, 2001; Kemp, 2003; Pearl, 1988; Power, 2003; Rao, 1973), classification (Breiman, Friedman, Olshen, & Stone, 1984; Chow & Liu, 1968; Rao, 1973; Rebane & Pearl, 1987), construction/recovery (Charniak, 1991; Cooper & Herskovits, 1991, 1992; Fung & Crawford, 1990; Geiger & Heckerman, 1991; Geiger, Paz, & Pearl, 1990; Jensen, 2001; Lee, Barua, & Whinston, 1997; Lilford & Braunholtz, 2003; Mechling, 1992; Mechling & Valtorta, 1994; Moole & Valtorta, 2002; Moole, 1997; Moole & Valtorta, 2003; Neapolitan, 2004; Pearl, 1988; Pearl & Verma, 1991; Rebane & Pearl, 1987; Spirtes & Glymour, 1991; Verma & Pearl, 1992), reasoning and inference (Ambrosio, 1990; Chang & Fung, 1991; Geiger & Heckerman, 1991; Meek, 1995; Moole, 1997; Moole & Valtorta, 2003; Pearl, 1988; Pearl & Verma, 1991), complexity analysis (Cooper, 1990; Lam & Segre, 2002; Moole, 1997; Neapolitan, 2004; Valtorta & Loveland, 1989, 1992), and applications (Charniak, 1991; Geiger & Heckerman, 1991; Huyn, 2001; Jensen, 2001; Motro & Smets, 1997; Rao, 1973; Sreenivasan, 2003; Turban & Aronson, 2001). Salient features of BBNs are described below, while comparing and contrasting them with PMDDM.

Independence Assumptions

Both BBNs and PMDDM represent the probability distribution of an event set. BBNs contain implicit independence assumptions, while PMDDM does not. d-separation condition identifies dependency relationships among random variables in BBNs. More specifically, random variables that are not relevant in calculating probability of an event are all assumed d-separated from event of interest. An advantage of this assumption is that the specification of probabilities for each event becomes simpler. All conditional probabilities for the sets of irrelevant variables become unnecessary. This results in significant storage space savings and also facilitates automated inference. A disadvantage of this approach is that the underlying probability distribution for a set of events may not satisfy independence assumptions. In contrast, PMDDM is capable of storing and manipulating large amounts of uncertain data or probability distributions.

Graphical Representation

Graphical representation of dependency relationships among variables is a prominent characteristic of BBNs. In contrast, PMDDM does not have a graphical representation, even though a multidimensional concept called Cube is defined. Probability distributions that can be represented by graphical models are depicted in Figure 4 below.

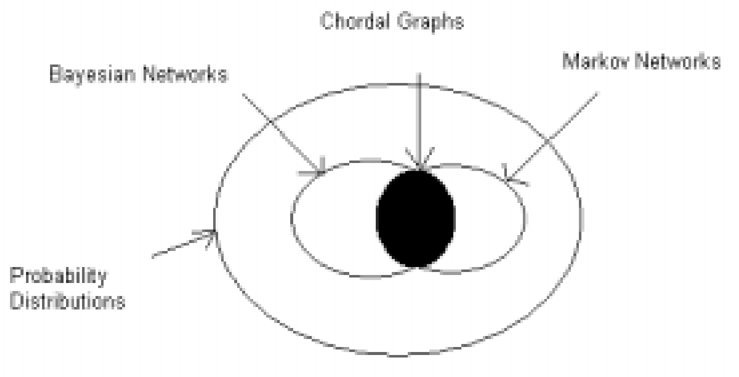

Figure 4. Graphical representation of probability distributions. Adapted from Pearl (1988) with permission from author.

As can be seen from Figure 4, only a portion of all probability distributions can be represented using graphical models. Only the DAG Isomorphic class of probability distributions can be represented using BBNs. In contrast, PMDDM has no limitation on the class of probability distributions.

Expressiveness

BBNs are able to represent probability distributions that are DAG isomorphic. In addition, they express dependency models with the following characteristics (Pearl, 1988):

1. Symmetry: $I(X,Z,Y) \Leftrightarrow (Y,Z,X)$
2. Composition/Decomposition: $I(X,Z,Y \cup W) \Leftrightarrow I(X,Z,Y)$ & $I(X,Z,W)$
3. Intersection: $I(X, Z \cup W, Y)$ & $I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W)$
4. Weak Union: $I(X, Z, Y \cup W) \Rightarrow I(X, Z \cup W, Z)$
5. Contraction: $I(X, Z \cup Y, W)$ & $I(X,Y,Z) \Rightarrow I(X, Z, Y \cup W)$

6. Weak Transitivity: $I(X,Z,Y)$ & $I(X, Z \cup \gamma, Y) \Rightarrow I(X, Z, \gamma)$ or $I(\gamma, Z, Y)$
7. Chordality: $I(\alpha, \gamma \cup \delta, \beta)$ & $I(\gamma, \alpha \cup \beta, \delta) \Rightarrow I(\alpha, \gamma, \beta)$ or $I(\alpha, \delta, \beta)$

In addition to these characteristics, d-separation operation is weakly transitive. These characteristics of BBNs are amenable to causal interpretation and are more intuitive for knowledge representation (Neapolitan, 2004; Pearl, 1988). However, expressive power of BBNs is limited to DAG isomorphic probability distributions. In contrast, PMDDM has at least the expressive power of the relational model. This means it is not only able to express general probability distributions that cannot be expressed by BBNs, but also general relational restrictions on sets.

Data Views

The data represented in BBNs is DAG isomorphic and can be represented graphically. In contrast, PMDDM is multi-dimensional data. A graphical view of the data is more useful to inference and reasoning tasks, while a multi-dimensional view is more useful to interpretation of the data. This is reflected also in operations defined on the BBNs and the PMDDM. The BBNs have operations to identify relevancy and dependency helping the user to infer consequences and reason on cause effect relationships. The PMDDM has operations to investigate data by querying, drilling down, and merging cubes.

Space

Encoding an arbitrary joint probability distribution for *n* propositional variables requires 2^n entries. For example, a domain with five variables requires 32 entries and one with 10 variables requires 1024 entries to represent complete joint probability distribution. This exponential growth in space is a major concern. BBNs capture only conditional probabilities relevant to a given variable (relevancy as determined by the d-separation condition). This results in storage space savings. In general, BBNs can be represented using polynomial space complexity (Cooper, 1990; Valtorta & Loveland, 1989, 1992). Comparing this with PMDDM, which attempts to store entire joint probability distribution may seem a waste of space. But, as described in the *Application* of the *Model in Business Management* section above, the data captured may not have the entire joint probability distribution in most applications. Even when data are available, it is possible to reduce space requirements by not storing propositions whose probability is below a given threshold, for example 0.001, which represents a negligible chance of being true. In addition, PMDDM can handle the probability mass assigned to missing propositions or unknown propositions effectively.

Time

The time complexity of sequential algorithms to construct, infer, or update BBNs is in general polynomial or exponential (Cooper, 1990; Valtorta & Loveland, 1992). Since most algorithms examine all input, size of the data stored naturally increases the time to perform operations on it. An exception is approximate algorithms, which may not consider all input and/or all possible operations on the data. The reduced size of data input to the BBN algorithms results in savings of computational time as well. In contrast, PMDDM does not have such privilege to reduce size of input, as their primary purpose is to store large amounts of uncertain data. As shown in *Time and Space Complexities of the Algorithm* section above, operations on PMDDM are of exponential time complexity. This observation should lead to the practicality of PMDDM for a particular problem, which should be determined based on number of variables in the domain as well as computational resources available.

Parallelism

The period of time from 1975 to 1990 witnessed a rapid advancement of parallel architectures. The relative lull of 1990s was followed by massive parallelization (cluster computing, symmetric multi processing (SMP), grid computing). Current trends point to an expansive use of parallelization. Research on BBNs has been keeping up with this trend and several algorithms have been parallelized effectively (Lam & Segre, 2002; Mechling & Valtorta, 1994; Moole & Valtorta, 2003). These researchers pointed out inherent parallelism in operations on graphical models and probability distributions. Operations on PMDDM may benefit by researching parallelization. This is because operations are exponential and parallelization reduces time required to compute, making PMDDM more attractive.

Summary of Similarities Between BBNs and PMDDM

There are several similarities between BBNs and PMDDM. They are:

1. Both represent probability distributions (i.e. uncertain information);
2. Both can represent data visually, BBNs using graphs and PMDDM using cubes;
3. Both can be used for Decision Support Systems; and
4. Both have inherent parallelism.

Table 2.
Summary of Differences Between BBNs and PMDDM

	BBNs	PMDDM
1	Only DAG isomorphic probability distributions can be represented.	Any probability distribution as well as relational data can be represented.
2	Suitable to represent causal relationships.	Suitable to represent hierarchical and relational concepts.
3	More useful for inference and reasoning.	More useful for data analysis.
4	Requires polynomial time and space.	Requires exponential time and space.
5	Well-established concept.	Relatively recent advancement.

Summary

In this chapter, algebraic operations on PMDDM, a Bayesian Framework for modification of probabilistic multidimensional data, and comparison and contrast between PMDDM and BBNs are presented. An algorithm for data modification is presented along with proof of correctness. Space and time complexities of this algorithm were analyzed. In Chapter 5, a summary of research results and conclusions is presented.

CHAPTER 5

SUMMARY, CONCLUSIONS, AND FURTHER RESEARCH

Decision sciences focus on improving managerial decision-making process. Decision Support Systems is an important area of study in decision sciences. Business intelligence and decision-making in today's business world require extensive use of huge volumes of real-world data, which contain uncertainty and change over time. Organizations need to handle uncertain information. Ignoring it is not a good option. Many organizations use Decision Support Systems to enhance managerial decisions. These systems should be able to handle efficiently large amounts of data, as well as uncertainty, and modifications to uncertain data. Relational database product vendors have provided several extensions and features to support these requirements, but these extensions lack support of conceptual models, which impedes growth of the software product market. Limited availability of Decision Support Systems to business could result in inconsistent and sub-optimal decisions.

Decision Support Systems that cannot handle large volumes of data containing uncertainty are less useful for decision-making. Data representation and uncertainty representation are crucial parts of these Decision Support Systems (Moole, 2003). Currently, Decision Support Systems products lack a standard conceptual data model for supporting data representation and operations. They also lack a framework to represent uncertainty, modify uncertain data, and perform imprecise queries. Conceptual models and frameworks supported by theories founded in mathematics enable users of product to understand better the claims of product manufacturers (Codd et al., 1993). They also enable researchers to contribute independently to technology (Agarwal et al., 1997; Thomas & Datta, 2001). Recently, researchers focused on this problem, and Moole (2003) proposed a probabilistic multidimensional data model; however, this model lacked the framework for probabilistic data modification. Lack of a framework to modify data diminishes importance of data models and their usefulness (Dey & Sarkar, 2000; Moole, 2003). This dissertation research was aimed at providing such enhanced data model to enable Decision Support Systems product development.

This study developed a framework and an algorithm for probabilistic data modification to enhance importance and usefulness of probabilistic multidimensional data model of Moole. This framework and algorithm can update uncertain data consistent with the model (*consistent*), resulting in valid data (*closed*), and is reliable in all possible update scenarios (*complete*).

The solution developed by the investigator is significant because a $4 billion software products development market is not achieving its potential (Pendse, 2003). This solution will contribute to data models research and the knowledge base and may result in better DSS tools for business. This solution may help standardize multidimensional database products and related tools. Such standardization can facilitate widespread adoption of these products and tools by business as happened in case of relational databases (Date, 2000). Standardized products are easier to understand and generally cheaper than custom-built and proprietary products (Thomas & Datta, 2001). DSS products developed as a result of this research may reduce overall cost of ownership of DSS products to business. Reduced prices of goods and services for consumers due to enhanced decision-making improve the standard of living (Gairdner, 2000). The investigator performed research activities summarized below.

Probabilistic multidimensional data model enhancements

Investigator integrated the multidimensional data model (MDD) and probabilistic data model and provided additional required algebraic operations for the Probabilistic MDD Model. The probabilistic multidimensional data model definition and its algebra were enhanced to include the following definitions and operators:

Predicate P: A predicate is a well-formed formula in first-order predicate logic.

An *atomic predicate* is a restriction on the domain of a single attribute or characteristic, e.g. (year = 1994).

A *compound predicate* is a logical expression of atomic predicates. Logical operators are ∧ (and), ∨ (or), ¬ (not), → (implies), and ↔ (equivalent to). It is of form: P = p_1 ⟨op⟩ p_2 ⟨op⟩ … ⟨op⟩ p_n. e.g. (year = 1994) ∧ ((quantity < 15) ∨ (amount > 100))

***l* satisfies P**: *l*, an instance of L, with structure <address, content> satisfies predicate P if and only if:

Case 1: If an element of *l* is a dimension, then *l*.AC satisfies P, otherwise *l*.CC satisfies P, if P is atomic and truth-value is TRUE.

$$P(l) = TRUE \quad \begin{cases} a \ \ in \ \ P, a \in f(d_i), d_i \in D, \\ P(l.AC\ [a]) = TRUE \\ OR \\ a \ \ in \ \ P, a \in f(m_i), m_i \in M, \\ P(l.CC\ [a]) = TRUE \end{cases}$$

e.g. Upper left most corner cell in cube of Figure 1 satisfies P=(year=1993)

Case 2: If P is a compound predicate, *l* satisfies P, when all truth-values evaluated together with connecting operators results in TRUE.

$$\forall p_i \in P, a \ in \ p_i, p_i(l) = Q_i$$
$$P(l) = Q_1 \langle op \rangle Q_2 \langle op \rangle \ldots \langle op \rangle Q_n$$

e.g. Upper left most corner cell in cube of figure 1 satisfies

P=(year=1993)∧(city="Boston")∧(product_name="P1")

Cardinality of Predicate (η_P): Cardinality, denoted by η, of a predicate is defined as number of unique attributes appearing in predicate. An atomic predicate has cardinality of 1. Cardinality of a compound predicate is ≥1 (a compound predicate may be constructed using a single attribute, hence η=1).

Selectivity of Predicate P on a cube C ($\delta_{[P,C]}$): Selectivity of a predicate P, denoted by δ, for a given cube C, is size of subset of cube cells in L of C that satisfy P.

Cardinality and selectivity are useful in ordering and identifying cube cells. The update algorithm uses this ordering capability to ensure correct handling of marginal probability specifications.

Data modification framework

Provided a comprehensive (*complete*, *consistent*, and *closed*) framework to update probabilistic data. This framework is based on Bayesian framework. This data modification framework described four possible cases of data modification, summarized below.

Let one denote the class of L *c* of a Cube C_I as a set of attributes AXY ∪ {pS} or {A, X, Y, pS}, where A is the address component, X and Y are mutually exclusive subsets of the set $\{\{A_d \cup A_m\} - \{A \cup pS\}\}$, one of which may be empty. An object of this class is denoted as {A=a, X=x, Y=y, pS=q}. In this representation, A corresponds to L.AC and X and Y are subsets of remaining attributes in the class of L without including pS. For example, if one has the cube of figure 1, then A = L.AC = {year, product, city}, X may be {amount} and Y may be {quantity}. This can also be represented as the union of all these attributes {year, product, city, amount, quantity, pS} in which pS represents joint probability of the remaining attributes. Let one suppose receipt of new information consisting of objects representing new beliefs. Assume that new information is specified as a cube C_{new} with the same structure as the existing cube C_{old}. In following sections, one says ”the new set of objects matches the existing set of objects” to indicate a selection predicate P constructed on $\{X \cup Y\}$ evaluating to true for both sets. When there is no match, there does not exist a selection predicate that satisfies both sets of objects. A special case is when all attributes of an object are unknown, i.e. ⟨A=a, X=*, Y=*, pS=q⟩. In this case, there are an infinite number of predicates that match. This is considered as not matching. The resulting cube after applying the updates described in each of the following cases is denoted by $C_{updated}$.

Case 1: There exists a set of objects $\langle a, x, y, q\rangle \in C_{new}$ that specifies complete joint probability distribution for A=a, i.e. $\Gamma_{SUM,pS,P}\left(\prod_{pS}^{M}\sum_{[A=a]}C_{new}\right)$ = 1. In this case, all existing objects must be replaced with the new set of objects. The remaining cases assume the new probability distribution specified is incomplete, It should be noted that partial distributions can be made complete distributions by assigning unspecified probability to unknown values.

Case 2: There exists an object $\langle a, x, y, q\rangle \in C_{new}$ that does not match with any object in existing data. In this case one has to create a new object. When a new object with an address component "a" is created, one has to adjust strength of belief in other objects with the same address component. An extremity is when the new object has q = 1, in which case it replaces all existing objects, because the data model restricts the sum of beliefs for an address not to exceed 1 (this is handled by case 1). Cases 3 and 4 handle allocation of residual probability when q < 1.

Case 3: There exists an object $\langle a, x, y, q\rangle \in C_{old}$, for some $q \in (0, 1]$, such that $x = x_i$, for some $i \in \{1, 2, \ldots, m\}$. This is a case where an existing object matches an object in the new information on attributes A and X. It is possible for an existing object to match new information based on more than one predicate. In such cases, the predicate selection is made by maximizing η and minimizing δ. The rationale behind this is that when η is maximum, there is a greater number of attributes in a predicate, which indicates a more precise match of objects (less marginalization) and a minimal δ indicates less number of objects matched (an exact match will have δ=1). This is essential in order to handle marginal probability specifications.

In this case, the new probability Q for the matching object is calculated, using Jeffrey's Rule, as follows:

$$
\begin{aligned}
Q &= \mathrm{PROB}[A = a, X = x_i, Y = y] \\
&= \mathrm{prob}[Y = y \mid A = a, X = x_i] * \mathrm{PROB}[A = a, X = x_i] \\
&= (\mathrm{prob}[A = a, X = x_i, Y = y] / \mathrm{prob}[A = a, X = x_i]) * \mathrm{PROB}[A = a, X = x_i] \\
&= \frac{q}{\Gamma_{SUM,pS,P}\left(\prod_{pS}^{M} \sum_{[A=a,X=xi]} C_{old}\right)} * p_i \qquad \rightarrow \qquad \text{Equation } (\mathbf{I})
\end{aligned}
$$

where p_i is $\mathrm{PROB}[A = a, X = x_i]$ for $i = 1, 2, \ldots, m$.

Then object $\langle a, x, y, q\rangle$ should be replaced with $\langle a, x, y, Q\rangle$.

Case 4: There exists an object ⟨a, x, y, q⟩ ∈ C_{old}, for some q ∈ (0, 1], such that x ≠ xi, for all i ∈ {1, 2, …, m}. In this case, an existing object does not have a matching object in the new information. The object ⟨a, *, *, q⟩ also has no match, therefore it will be handled by this case.

In this case, new probability for objects without a match is calculated by proportionately distributing the difference between old residual probability and new residual probability after resolving the objects of above cases, if any. The old residual probability P_{oldRes} of objects with A = a and $X \neq x_i$ before applying cases 1, 2, and 3 is calculated by:

$$P_{oldRes} = \text{prob}[A{=}a, X{=}x] = 1 - \Gamma_{SUM, pS, P_{old\text{Res}}}\left(\prod\nolimits_{pS}^{M} \sum\nolimits_{[A = a, X \neq xi]} C_{old}\right)$$

The new residual probability after previous cases P_{newRes} of objects with A=a and $X \neq x_i$ is calculated by:

$$P_{newRes} = \text{PROB}[A{=}a, X{=}x] = 1 - \Gamma_{SUM, pS, P_{new\text{Res}}}\left(\prod\nolimits_{pS}^{M} \sum\nolimits_{[A=a, X \neq xi]} C_{updated}\right)$$

One proportionately distributes residual probability $P_{oldRes} - P_{newRes}$ based on old probabilities. This distribution has to be to objects other than x_i, i = 1, 2, …, m.

With this, one can now calculate new probability Q associated with ⟨a, x, y, q⟩ as below:

$$\begin{aligned} Q &= \text{PROB}[A = a, X = x, Y = y] \\ &= \text{prob}[Y = y \mid A = a, X = x] * \text{PROB}[A = a, X = x] \\ &= (\text{prob}[A = a, X = x, Y = y] / \text{prob}[A = a, X = x]) * \text{PROB}[A = a, X = x] \\ &= (q / P_{oldRes}) * P_{newRes} \quad \rightarrow \quad \text{Equation } (\mathbf{II}) \end{aligned}$$

Then object ⟨a, x, y, q⟩ should be replaced with ⟨a, x, y, Q⟩.

Data modification algorithm

Investigator developed an algorithm to modify the data. This algorithm conforms to the data modification framework described above. It can perform data modifications in all four possible cases described in the data modification framework. This algorithm accepts an old cube and a new cube to produce an updated cube as the output. The algorithm was proved to be correct, by proving it is: *consistent* (with PMDDM), *complete* (handles all possible cases of data modification), and *closed* (results only in valid objects). Time and space complexity analysis was performed. The worst case time complexity was determined to be $\max(S_o,S_n) * O(S_nS_u + S_n*(\log_2 S_o)^2)$ and the worst case space complexity was $O(S_u)$, where S_o, S_n and S_u are number of objects in old cube, new cube, and the updated cube respectively.

Application of the Model in Business Management

Probabilistic multidimensional data model was applied to solve a business management problem. Forecasting demand for grocery products was described for a fictitious retail chain. Forecast method used for this business application uses historical data, product promotion data, and competition data. The forecast method utilizes heuristic rules specified below to forecast demand for each product in the category for the next day.

1. Average sales for the previous week (seven days): Demand is directly proportional to average sales M.
2. Promotion data for the previous week: Promotional factor, P, multiplies M. P is derived by incorporating total number of promotions.

3. Competition data (from other retailers in area as reported by agents) for the previous week: Competition, C, decreases sales.
4. Category sales forecast information: Final category demand, T_f, is equal to average category demand within the previous four weeks, T_a. This means, if tentative category demand, T_t (calculated by aggregating individual product demand in that category), is different from T_a, then the individual product demand has to be adjusted to make T_t equal to T_a.

To forecast demand using the above forecast method, the following data are needed:

1. Sales data for the previous week for the category;
2. Product promotion data for the previous week;
3. Competitor sales information for the previous week, as reported by agents; and
4. Daily category sales forecasts for the previous week.

Probabilistic multidimensional data model was shown capable of storing, manipulating, and retrieving the above data. Usage of model and the algebraic operators to handle above data types was described using sample data.

Comparison and Contrast with Bayesian Belief Networks

PMDDM was compared and contrasted with Bayesian belief networks. This analysis identified several similarities between BBNs and PMDDM. They are:

1. Both represent probability distributions (i.e. uncertain information);
2. Both can represent data visually, BBNs using graphs and PMDDM using cubes;

3. Both can be used for Decision Support Systems; and
4. Both have inherent parallelism.

Differences summarized in the following table were identified.

Table 3.
Differences Between BBNs and PMDDM

	BBNs	PMDDM
1	Only DAG isomorphic probability distributions can be represented.	Any probability distribution as well as relational data can be represented.
2	Suitable to represent causal relationships.	Suitable to represent hierarchical and relational concepts.
3	More useful for inference and reasoning.	More useful for data analysis.
4	Requires polynomial time and space.	Requires exponential time and space.
5	Well-established concept.	Relatively recent advancement.

In conclusion, this research achieved all intended results as specified in scope of this research. Analytical method was identified as the appropriate method and was used for these research activities. Advantages of analytical method were found to be helpful. Disadvantages of analytical method include logical errors, problems of semantics, and possibility to focus on trivial and irrelevant problems. Paying close attention to the derivation of formulas eliminated logical errors. Problems of semantics were reduced by using the semantics established by reputed researchers such as Chow & Liu (1968), Codd (1971), Dubois (Dubois & Prade, 1988), Jeffrey (1983), Knuth (1973), Pearl (Pearl, 1988, 2001; Pearl, Geiger, & Verma, 1990; Pearl & Verma, 1991), and Shafer (1990). Parts of the research results were published in the proceedings of IEEE SoutheastCon 2004, Greensboro, North Carolina, USA, in refereed section (Moole & Korrapati, 2004a), which also received Walden University Presentation Honorarium. An application of PMDDM in managerial decision-making is described in another paper published in IEEE SoutheastCon 2005 (Moole & Korrapati, 2005).IEEE SoutheastCon refereed section papers are blind peer-reviewed for technical accuracy, relevancy, significance, and contribution to state-of-the-art. This reduced the possibility of these results being trivial or irrelevant. In addition, parts of these results were also presented at Allied Academics National Conference (Moole & Korrapati, 2004b).

Further Research

Probabilistic multidimensional data models is relatively a recent development compared to the research in relational data models that dates back to the early 1970s. This dissertation research is only a step into the complete definition of probabilistic multidimensional data model. There is plenty of scope to research futher, only limited by the imagination. To list a few topics for further research in this area, current PMDDM can be extended to include parallelization of algebraic operations, implementation issues, applicability in various fields, and feasibility studies. Alternative data modification algorithms can be devised using algebraic operations. These topics are complex and are beyond the scope of this research.

REFERENCES

Agarwal, R., Gupta, A., & Sarawagi, S. (1997). *Modeling multidimensional databases.* Paper presented at the Proceedings of 13th ICDE Conference, Birmingham, U.K.

Ambrosio, B. D. (1990). *Symbolic Probabilistic Inference in Belief Networks*. Corvallis, OR: Oregon State University.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth.

Chang, K. C., & Fung, R. (1991). *Symbolic Probabilistic Inference with continuous variables.* Paper presented at the Proceedings of 7th Conference on Uncertainty in AI, San Mateo, CA.

Charniak, E. (1991). Bayesian Networks without Tears. *AI Magazine, 12*(4), 50-63.

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM SIGMOD Record, 26*(1), 65-74.

Chen, Q., Hsu, M., & Dayal, U. (2000). *A Data-Warehouse / OLAP Framework for Scalable Telecommunication Tandem Traffic Analysis.* Paper presented at the 16th International Conference on Data Engineering, San Diego, CA.

Chow, C. K., & Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory, 14*, 462-467.

Codd, E. F. (1971). A Database Sublanguage Founded on the Relational Calculus. *In Proceedings of the 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control*, 35-68.

Codd, E. F., Codd, S. B., & Sally, C. T. (1993). *Providing OLAP to User-Analysts: An IT Mandate* [Web Site]. E. F. Codd & Associates. Retrieved 09/10/2002, 2002, from the World Wide Web: http://www.arborsoft.com/papers/coddTOC.html

Cooper, G. F. (1990). The complexity of probabilistic inference using Bayesian Belief Networks. *Artificial Intelligence, 42*, 393-405.

Cooper, G. F., & Herskovits, E. (1991). *A Bayesian Method For the Induction of Probabilistic Networks from Data* (Report # SMI-91-01). Pittsburgh, PA: University of Pittsburgh.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning, 9*, 309-347.

Date, C. J. (2000). *The Database Relational Model: A Retrospective Review and Analysis: A Historical Account and Assessment of E. F. Codd's Contribution to the Field of Database Technology* (1st ed.): Addison-Wesley Pub Co.

Date, C. J. (2003). *An Introduction to Database Systems* (8th ed.): Pearson Addison Wesley.

Dey, D., & Sarkar, S. (2000). Modifications of Uncertain Data: A Bayesian Framework for Belief Revision. *Information Systems Research, 11*(1), 1-16.

Dezert, J. (Ed.). (2002). *Foundations for a new theory of plausible and paradoxical reasoning*. Sofia: Bulgarian Academy of Sciences.

Dubois, D., & Prade, H. (1988). *Possibility Theory*. New York: Plenum Press.

E. D. Hirsch, Joseph F. Kett, Trefil, J., & Trefil, J. S. (2002). *The New Dictionary of Cultural Literacy* (3 ed.): Houghton Mifflin Company.

Foote, P. S., & Krishnamurthi, M. (2001). Forecasting using data warehousing model: Wal-Mart's experience. *The Journal of Business Forecasting Methods & Systems, 20*(3), 13-17.

Fung, R. M., & Crawford, S. L. (1990). *Constructor: A system for the induction of probabilistic models.* Paper presented at the Proceedings of AAAI, Boston, MA.

Gairdner, A. (2000, 05/06/2004 17:05:57). *G20 Bulletin: 25 October 2000* [WebSite]. University of Toronto Library & G8 Research Group at the University of Toronto. Retrieved 05/06/2004, 2004, from the World Wide Web: http://www.g7.utoronto.ca/g20bulletin/2000/bull2.htm

Geiger, D., & Heckerman, D. (1991). *Advances in Probabilistic Reasoning.* Paper presented at the Proceedings of the 7th Conference on Uncertainty in AI.

Geiger, D., Paz, A., & Pearl, J. (1990). *Learning causal trees from dependence information.* Paper presented at the Proceedings of AAAI, Boston, MA.

Gyssens, M., & Lakshmanan, L. V. S. (1997). *A foundation for multidimensional databases.* Paper presented at the Proceedings of 23rd VLDB Conference, Athens, Greece.

Huyn, N. (2001). Data analysis and mining in the life sciences. *ACM SIGMOD Record, 30*(3), 76-85.

Inmon, W. H. (1990). *Using Oracle to Build Decision Support Systems*: QED Press.

ITG. (2000). *Strategies for E-Financials: Competitive Impact of Financial Systems in e-Business* [Web Site]. International Technology Group, 885 North San Antonio Road, Suite C, Los Altos, California 94022-1305. Retrieved July 19, 2003, 2003, from the World Wide Web: http://www-1.ibm.com/servers/eserver/zseries/library/whitepapers/pdf/gf225164.pdf

Jaynes, E. T., & Bretthorst, G. L. (2003). *Probability Theory: The Logic of Science*: Cambridge University Press.

Jeffrey, R. (1983). *The logic of decisions* (2nd ed.). Chicago, IL: University of Chicago Press.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. New York, NY: Springer.

Kemp, F. (2003). A Course in Probability Theory. *J Royal Statistical Soc D, 52*(2), 239-239.

Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (1998). *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*. New York, NY: John Wiley & Sons, Inc.

Klir, G. J., & Wierman, M. J. (1998). *Uncertainty-Based Information - Elements of Generalization Information Theory*: Physica-Verlag Heidelberg.

Knuth, D. (1973). *Fundamentals of Algorithms* (2 ed. Vol. 1): Addison-Wesley Publishing Company.

Lam, W., & Segre, A. M. (2002). A Parallel Learning Algorithm for Bayesian Inference Networks. *IEEE Transactions on Knowledge and Data Engineering, 14*(1), 159-208.

Lancaster, G., & Lomas, R. (1986). A Managerial Guide to Forecasting. *International Journal of Physical Distribution & Materials Management, 16*(6), 1-37.

Lee, B. T., Barua, A., & Whinston, A. B. (1997). Discovery and Representation of Causal Relationships in MIS Research: A Methodological Framework. *MIS Quarterly*.

Li, C., & Wang, X. S. (1996). *A data model for supporting On-Line Analytical Processing.* Paper presented at the Proceedings of the Conf. Information and Knowledge Management, Baltimore, MD, USA.

Lilford, R. J., & Braunholtz, D. (2003). Reconciling the quantitative and qualitative traditions--the Bayesian approach. *Public Money & Management, 23*(3), 203-209.

Mantrala, M. K., & Raman, K. (1999). Demand uncertainty and supplier's returns policies for a multi-store style-good retailer. *European Journal of Operational Research, 115*(2), 270-275.

Martin, J. R. (2004, 2/8/2002). *Buckley, J.W., M.H. Buckley and H. Chiang. 1976. Research Methodology & Business Decisions. National Association of Accountants. - Summary by James R. Martin* [WebSite]. James R. Martin, University of South Florida. Retrieved July 10, 2003, from the World Wide Web: http://www.maaw.info/ArtSumBuckley76.htm

Mechling, R. (1992). *PaCCIN: A Parallel Constructor of Causal Independence Networks.* Unpublished Research, University of South Carolina, Columbia, SC.

Mechling, R., & Valtorta, M. (1994). A Parallel Constructor of Markov Networks. In P. Cheeseman & R. W. Oldford (Eds.), *Selecting Models from Data: Artificial Intelligence and Statistics IV* (pp. 255-261). New York, NY: Springer.

Meek, C. (1995). *Causal Inference and causal explanation with background knowledge.* Paper presented at the Proceedings of 11th Conference on Uncertainty in AI, San Mateo, CA.

Moole, B., & Valtorta, M. (2002). *Causal Explanation with Background Knowledge* (Technical Report TR-2002-017). Columbia, SC 29208, USA: Dept of CSE, University of South Carolina.

Moole, B. R. (1997). *Parallel Construction of Bayesian Belief Networks (Research Thesis).* Unpublished Research, University of South Carolina, Columbia, SC, USA.

Moole, B. R. (2003, April 04-06). *A Probabilistic Multidimensional Data Model and Algebra for OLAP in Decision Support Systems.* Paper presented at the Proceedings of the IEEE SoutheastCon'03, Ocho Rios, Jamaica.

Moole, B. R., & Korrapati, R. B. (2004a, March 26-28). *A Bayesian Framework for Modification of Uncertain Data in Probabilistic Multidimensional Data Model.* Paper presented at the IEEE SoutheastCon 2004, Greensboro, NC, USA.

Moole, B. R., & Korrapati, R. B. (2004b, Fall 2004). *A Decision Support System Model for Forecasting in Inventory Management Using Probabilistic Multidimensional Data Model (PMDDM).* Paper presented at the Proceedings of the Allied Academies National Conference, Academy of Information and Management Sciences, Maui, Hawaii.

Moole, B. R., & Korrapati, R. B. (2005, April 8-10). *Forecasting Demand Using Probabilistic Multidimensional Data Model.* Paper presented at the IEEE Southeast Conference 2005, Ft. Lauderdale, Florida.

Moole, B. R., & Valtorta, M. G. (2003, December 18-20). *Causal Explanation with Background Knowledge.* Paper presented at the 1st Indian International Conference on Artificial Intelligence 2003 (IICAI'03), Hyderabad, India.

Motro, A., & Smets, P. (1997). *Uncertainty Management in Information Systems: From needs to solutions*: Kluwer Academic Publishers.

Neapolitan, R. E. (2004). *Learning Bayesian Networks*. Upper Saddle River, NJ: Pearson Prentice Hall.

PCG. (1998). *Transforming Supermarkets: The Fresh Market Manager Solution* (White Paper). Park City, Utah 84060: Park City Group, Inc.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Pearl, J. (2001). *Causality*. Cambridge, UK: Cambridge University Press.

Pearl, J., Geiger, D., & Verma, T. (1990). Conditional Independence and its representation.

Pearl, J., & Verma, T. (1991). *A Theory of Inferred Causation.* Paper presented at the Principles of Knowledge Representation and Reasoning, Proceedings of the 2nd International Conference, San Mateo, CA.

Pendse, N. (2003). *Market Share Analysis: Minimal growth in OLAP revenues in 2002* [Web Site]. The OLAP Report. Retrieved July 10, 2003, 2003, from the World Wide Web: http://www.olapreport.com/Market.htm

Power, D. J. (2003). *A Brief History of Decision Support Systems, version 2.8* [World Wide Web]. DSSResources.COM. Retrieved 01/30/2004, 2004, from the World Wide Web: http://dssresources.com/history/dsshistory.html

Rao, C. R. (1973). *Linear Statistical Inference And Its Applications*: John Wiley & Sons.

Rebane, G., & Pearl, J. (1987). *The recovery of causal poly-trees from statistical data.* Paper presented at the Proceedings of Workshop on Uncertainty in AI.

Shafer, G. (1990). Perspectives on the theory and practice of belief functions. *International Journal of Approximate Reasoning, 4*(323-362).

Spirtes, P., & Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review, 9*(1).

Sreenivasan, R. (2003). Modeling in Medical Decision Making: a Bayesian Approach. *J Royal Statistical Soc D, 52*(2), 246-246.

Steinmetz, S. (Ed.). (1997). *Random House Webster's College Dictionary*. New York: Random House.

SUN-Microsystems. (1999). *Sun Enterprise 1000 Scores Big in Informix Red Brick Performance Test for the Retail Enterprise* [Web Site]. Sun Microsystems. Retrieved July 19, 2003, 2003, from the World Wide Web: http://www.sun.com/smi/Press/sunflash/9908/sunflash.990809.3.html

Symbol. (2004). *Computer Automated Ordering Provides Faster, More Accurate Replenishment: Executive Summary* [Web Site]. http://www.symbol.com. Retrieved March 11, 2004, 2004, from the World Wide Web: http://www.symbol.com/solutions/retail/food_drug_whtppr_cao.html

Thomas, H., & Datta, A. (2001). A Conceptual Model and Algebra for On-Line Analytical Processing in Decision Support Databases. *Information Systems Research, 12*(1), 83-102.

Turban, E., & Aronson, J. E. (2001). *Decision Support Systems and Intelligent Systems*. Upper Saddle River, New Jersey: Prentice Hall.

Valtorta, M., & Loveland, D. W. (1989). *On the complexity of Belief Network Synthesis and Refinement* (TR-89011). Columbia, SC: Dept of CS, University of South Carolina.

Valtorta, M., & Loveland, D. W. (1992). On the complexity of Belief Network Synthesis and Refinement. *International Journal of Approximate Reasoning, 7*(3-4), 121-148.

Vassiliadis, P., & Sellis, T. K. (1999). A Survey of Logical Models for (OLAP) Databases. *ACM SIGMOD Record, 28*(4), 64-69.

Verma, T., & Pearl, J. (1992). *An Algorithm for deciding if a set of observed independencies has a causal explanation.* Paper presented at the Proceedings of the 8th Conference on Uncertainty in AI, Stanford, CA.

Zadeh, L. A. (1965). Fuzzy Sets. *Information Control, 8*(3), 338-353.

APPENDIX A

CURRICULUM VITAE

Bhaskara Reddy Moole
1116 Cypress Tree Place
Herndon, VA 20170
bhaskarareddy@wondertechnology.com

SUMMARY

Experience in Architecture, Design and Development of Internet and Intranet Portals and Client/Server applications using Java/J2EE/C++ on UNIX and Windows NT platforms.

CERTIFICATIONS

Sun: Sun Certified Java2 Programmer
Brain Bench: Java2, XML, HTML, Unix
IBM: IBM Certified Solution Developer - IBM WebSphere Portal for Multiplatforms v4.1

EDUCATION

M.S. (CS), University of South Carolina, Columbia, SC, USA
M. Tech. (AI), University of Hyderabad, Hyderabad, AP, India
B.E. (CSE), University of Mysore, Mysore, KRN, India

PROFESSIONAL EXPERIENCE

Bhaskara Moole has many years of cumulative experience in software development, design, architecture, and consulting. He is currently a consultant on enterprise portal technologies to the Department of Homeland Security.

PUBLICATIONS

Moole, B. R., & Valtorta, M. G. (2003, December 18-20). *Causal Explanation with Background Knowledge.* Paper presented at the 1st Indian International Conference on Artificial Intelligence 2003 (IICAI'03), Hyderabad, India.

Moole, B. R. (2003, April 04-06). *A Probabilistic Multidimensional Data Model and Algebra for OLAP in Decision Support Systems.* Paper presented at the Proceedings of the IEEE SoutheastCon'03, Ocho Rios, Jamaica.

Research Work referenced in a pioneering book by Prof. Judea Pearl of UCLA titled "Causality: Models, Reasoning, and Inference" from Cambridge University Press 2000. ISBN 0-521-77362-8.

Moole, B., & Valtorta, M. (2002). *Causal Explanation with Background Knowledge* (Technical Report TR-2002-017). Columbia, SC 29208, USA: Dept of CSE, University of South Carolina.

Moole, B. R., & Reddy, A. S. (1991, 27-29 December). *A Cognitive Model of Language and Thought [CMOLT].* Paper presented at the Proceedings of ICC-91, Indian Computing Congress, Hyderabad, India.

THESES

Parallel Construction of Bayesian Belief Networks, MS (R) Thesis, University of South Carolina, Columbia, SC, USA.

A Cognitive Model of Language and Thought, MS (R) Thesis, University of Hyderabad, Hyderabad, AP, India.

An expert system to diagnose the problems in automobiles, BS (P) Thesis, BIET, University of Mysore, Karnataka, India.

MEMBERSHIP IN PROFESSSIONAL ORGANIZATIONS

IEEE member (and Reviewer for SoutheastCon)

Upsilon Pi Epsilon International Honor Society Member

APPENDIX B

COMPETITION DATA

The complete competition data is specified as follows:

{⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨110, 10, 0.2⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨110, 15, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨120, 10, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨120, 5, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨120, 15, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨130, 10, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Boston⟩, ⟨130, 15, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨125, 10, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨110, 15, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨120, 10, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨120, 5, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨120, 15, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨130, 10, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Boston⟩, ⟨130, 15, 0.2⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨110, 15, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨120, 10, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨120, 5, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨120, 15, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨130, 10, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨130, 15, 0.2⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Boston⟩, ⟨150, 15, 0.1⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨110, 15, 0.7⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨110, 5, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨110, 15, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨120, 10, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨120, 5, 0.05⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨120, 15, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨130, 10, 0.01⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨130, 5, 0.1⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Boston⟩, ⟨130, 15, 0.1⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨160, 20, 0.3⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨110, 15, 0.2⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨110, 15, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨120, 10, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨120, 5, 0.05⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨120, 15, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨130, 10, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨130, 5, 0.01⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Boston⟩, ⟨130, 15, 0.2⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨130, 15, 0.1⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨110, 15, 0.1⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨110, 5, 0.1⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨120, 10, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨120, 5, 0.05⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨120, 15, 0.02⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨130, 10, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨130, 5, 0.01⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨130, 15, 0.2⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Boston⟩, ⟨140, 10, 0.4⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston⟩, ⟨180, 30, 0.3⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston⟩, ⟨110, 10, 0.2⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston⟩, ⟨110, 5, 0.2⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston⟩, ⟨110, 15, 0.2⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Boston⟩, ⟨120, 10, 0.1⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, New York⟩, ⟨105, 10, 0.3⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, New York⟩, ⟨125, 10, 0.5⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, New York⟩, ⟨105, 20, 0.7⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, New York⟩, ⟨140, 5, 0.4⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, New York⟩, ⟨110, 20, 0.6⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, New York⟩, ⟨120, 15, 0.1⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, New York⟩, ⟨110, 10, 0.3⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Chicago⟩, ⟨200, 10, 0.3⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Chicago⟩, ⟨105, 10, 0.3⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Chicago⟩, ⟨110, 25, 0.4⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Chicago⟩, ⟨150, 10, 0.3⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Chicago⟩, ⟨100, 20, 0.4⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Chicago⟩, ⟨100, 15, 0.4⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Chicago⟩, ⟨130, 30, 0.3⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, San Francisco⟩, ⟨190, 10, 0.2⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, San Francisco⟩, ⟨165, 10, 0.3⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, San Francisco⟩, ⟨155, 25, 0.4⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, San Francisco⟩, ⟨145, 15, 0.5⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, San Francisco⟩, ⟨130, 20, 0.6⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, San Francisco⟩, ⟨115, 25, 0.7⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, San Francisco⟩, ⟨135, 25, 0.8⟩⟩,

⟨⟨2004,01,01, DIET PEPSI, Dallas⟩, ⟨100, 10, 0.2⟩⟩,

⟨⟨2004,01,02, DIET PEPSI, Dallas⟩, ⟨125, 10, 0.3⟩⟩,

⟨⟨2004,01,03, DIET PEPSI, Dallas⟩, ⟨100, 15, 0.1⟩⟩,

⟨⟨2004,01,04, DIET PEPSI, Dallas⟩, ⟨140, 5, 0.2⟩⟩,

⟨⟨2004,01,05, DIET PEPSI, Dallas⟩, ⟨110, 10, 0.4⟩⟩,

⟨⟨2004,01,06, DIET PEPSI, Dallas⟩, ⟨130, 15, 0.5⟩⟩,

⟨⟨2004,01,07, DIET PEPSI, Dallas⟩, ⟨120, 10, 0.3⟩⟩}

APPENDIX C

GLOSSARY

OLAP (On-Line Analytical Processing): It is a term used to describe the nature of data processing using the queries to analyze data online, as opposed to offline or batch processing.

Multi-Dimensional Data View: A view of data that involves multiple dimensions or attributes. Relational data is organized in two-dimensional tables, which is inadequate for OLAP applications.

DSS (Decision Support Systems): DSS are used to assist human decision makers in complex decision-making scenarios.

DAG (Directed Acyclic Graph): A data structure containing nodes and directed edges that contains no cycles.