ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 6, No. 4, 2011, *pp*. 269-278

A Framework for Reducing Multidimensional Database to Two Dimensions

Adio Akinwale [1], Kolawole Adesina[2] and Olusegun Folorunso[3]

Department of Computer Science, University of Agriculture, Abeokuta, Nigeria

(*Received April 8, 2011, accepted May 17, 2011*)

Abstract. This work used a method of Matrix Decomposition Algorithm to obtain a new dataset of genetic epistasis as a surrogate for a multidimensional dataset which transformed multidimensional database to a 2-dimensional database. It employed decomposition algorithms based on Boyce Codd Normal Form for minimizing anomalies. The decomposition and reversible algorithms were used on relationship among object attributes and were implemented. The implemented program ran on sample genetic epistasis datasets of up to 10 dimensions and it was shown that multidimensional datasets can be reduced to two dimensions. It was established that the time taken to generate a sequence of tuples from multidimensional database to a 2-dimensional dataset was directly proportional to the number of genes considered. The result showed that the reduced 2-dimensional database did not require any in-built functions which take long processing time for generating query result as against querying of multidimensional dataset. The reduced 2-dimensional dataset was reversible to the original multidimensional dataset for lossless join operation which indicated that there was no loss of data values or tuple. The method was compared with existing reduction techniques and it was found that data access was very fast with decomposition algorithm than relational model.

Keywords: Matrix decomposition algorithm, multidimensional database, genetic epistatis, principal component analysis, project pursuit method, relational model

1. Introduction

Nowadays, emerging data are multidimensional in nature. Multidimensional database technology is being applied to distributed data and to new types of data that current technology often cannot adequately analyze. The global reason for the multidimensional database's rise is to facilitate flexible, high performance access and analysis of large volumes of complex and interrelated data [24]. Multidimensional database structure has evolved to match closely the way people visualize data [17]. Thus, people think of their businesses in multidimensional terms. For example, a sales manager will want to know the sales of a particular product over a period of time and may be in a particular region or location. This is a multidimensional view of sales data. The higher the dimensionality, the higher the volume of the data, and consequently, the more complex the data is to manage.

To lessen the scalability problem of processing large datasets, data reduction technique is proffered. Data reduction techniques are useful to reduce the scalability problem of processing large datasets. In order to efficiently archive and process growing multidimensional datasets, the work presents and compares reduction and reversibility technique algorithms that reduce multidimensional database to a two dimensional database and also reverse the two dimensional back to its original form without loss of data.

2. Literature Review

Multidimensional database is stored in such a way as to be represented to the user as a hypercube or multidimensional array, where each core value or fact occupies a cell indexed by a unique set of dimension values. Agrawal et. al. also asserted that multidimensional database is a key technology in the enabling of interactive analyses of large amounts of data for decision-making purposes [1]. It is used to process and analyze large and complex datasets. Multidimensional database originated from the multidimensional matrix

[1] *E-mail address*: aatakinwale@yahoo.com;
[2] *E-mail address*: kolawole_adesina@yahoo.co.uk;
[3] *E-mail address*: folorunsolusegun@yahoo.com

algebra, which has been used for (manual) data analysis since the late 19th century. An example of multidimensional software is On-Line Analytical Processing (OLAP) which supports multidimensional view of data in various ways. Sarawagi summarized OLAP as **F**ast **A**nalysis of **S**hared **M**ultidimensional **I**nformation [27]. Gupta et.al. cited that there is no theoretical limit to the number of dimensions although, current multidimensional database tools experience performance problems when the number of dimensions is more than 10. Multidimensional databases do not support the logical joining of multiple multidimensional arrays. The inability to join multidimensional databases with relational databases, limits query flexibility by eliminating the possibility of using characteristics tables to dynamically segment data [12], [23].

Colliat cited that there are two main approaches to building a multidimensional database. One approach maintains the data as a k-dimensional cube based on a non-relational specialized storage structure for storing k-dimensional data. Another approach uses a relational model wherein operations on the data cubes are translated to relational queries. This means it is posed in a possibly enhanced dialect of structural query language. This approach thus calls for dimensionality reduction [11].

The problem of dimension reduction is introduced as a way to overcome the curse of the dimensionality when dealing with vector data in high-dimensional spaces and as a modeling tool for such data. It is defined as the search for a low-dimensional manifold that embeds the high-dimensional data. It is basically a mapping from a D-dimensional space onto an L-dimensional space, for D > L. Lindsay devised a method that automates the task of projection pursuit. He characterized a given projection by a numerical index that indicated the amount of structure that is present. The index can be used as the basis for a heuristic search to locate the interesting projection [20].

Jensen et. al. considered analyzing multidimensional database with a two-dimensional spreadsheet. However, a table is considerable for only two-dimensional dataset. If there is need to go further to three dimensions, the obvious solution is to use separate worksheet for each extra dimension, with one worksheet for each dimension values and only to some extent [16]. This implies that there will be need to continuously create extra sheets for additional dimension. Analyses involving several values of the extra dimensions are cumbersome, and with many thousands of dimension values, the solution becomes infeasible or complex thus, attention is necessary to resolve the problem.

3. Theoretical Form of the System

A gene consists of 2 alleles – dominant and recessive denoted by **A** and **a** respectively. Each allele has a biological population frequency of $p = q = 0.5$ with genotype frequencies of p^2 for **AA**, 2pq for **Aa** and q^2 for **aa** following Hardy-Weinberg Equilibrium. Alleles **A** and **a** can combine in the following forms: **AA**, **Aa**, **aA**, **aa** [6]. Because the order of the alleles is unimportant, a genotype can have one of 3 values **AA**, **Aa** and **aa**. The penetrance function defines the probability (P) of disease (D) for genotypes (G) for one or more genetic variations, denoted by P[D/G]. Examples are P[D/AA] and P[D/AA,BB]. Thus for **n** genotypes interaction there will be $\mathbf{3^n}$ penetrance functions. The fixed dimensional length n-genotype Single Nucleotide Polymorphism (SNP) gives a total number of $\mathbf{3^n}$ sequences with the same number of penetrances, thus 3 SNP gives $3^2 = 9$ sequences.

Examples: 2 genotypes have 3^2 functions = 9 functions, i.e., 3 x 3 array.

3 genotypes have 3^3 functions = 27 functions, i.e., 3 x 3 x 3 array.

4 genotypes have 3^4 functions = 81 functions, i.e., 3 x 3 x 3 x 3 array.

26 genotypes have 3^{26} functions = 67,108,864 functions, i.e., 3 x 3 x 3 x … x 3 array.

n genotypes have 3^n functions = 3 x 3 x 3 x 3,…, x n array.

Equally, business sales outlet may have interesting data to manipulate and evaluate the efficient of the algorithm. These types of data will have varying dimensions. The achievement was to turn any of these multidimensional arrays into a 2-dimensional list, which has been detailed and described in figure 1a-d.