Exploration and Visualization of OLAP Cubes with Statistical Tests

Carlos Ordonez
University of Houston
Dept. of Computer Science
Houston, TX 77204, USA

Zhibo Chen
University of Houston
Dept. of Computer Science
Houston, TX 77204, USA

ABSTRACT

In On-Line Analytical Processing (OLAP), users explore a database cube with roll-up and drill-down operations in order to find interesting results. Most approaches rely on simple aggregations and value comparisons in order to validate findings. In this work, we propose to combine OLAP dimension lattice traversal and statistical tests to discover significant metric differences between highly similar groups. A parametric statistical test allows pair-wise comparison of neighboring cells in cuboids, providing statistical evidence about the validity of findings. We introduce a two-dimensional checkerboard visualization of the cube that allows interactive exploration to understand significant measure differences between two cuboids differing in one dimension along with associated image data. Our system is tightly integrated into a relational DBMS, by dynamically generating SQL code, which incorporates several optimizations to efficiently explore the cube, to visualize discovered cell pairs and to view associated images. We present an experimental evaluation with medical data sets focusing on finding significant relationships between risk factors and disease.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*

General Terms

Algorithms, Human Factors, Performance

Keywords

Parametric test, cube, visualization

1. INTRODUCTION

In a modern data mining environment users have a wide spectrum of options to analyze a data set going from simple queries to building machine learning and statistical models. On-Line Analytical Processing (OLAP) [5, 11] is an important application of exploratory database analysis that is complementary to such approaches. In an OLAP database users generally explore a large fact table with aggregations performed at multiple granularity levels trying to find interesting results. In OLAP most computations return simple univariate statistics such as sums, row counts, means and standard deviations. On the other hand, data mining [7, 11], statistical [13] and machine learning [17] techniques generally build models of varied complexity on a data set, depending on problem requirements. A comprehensive family of statistical tests sit somewhere in the middle between univariate statistical analysis and complex statistical models. Our tool shows parametric statistical tests are a promising technique to explore OLAP cubes.

Statistical tests [22] exhibit several advantages over statistical and machine learning models. They have simple and weak assumptions about the probability distribution behind the data set. In our case, such distribution generally comes in the form of a normal (Gaussian) distribution, or a closely related probability distribution function. Statistical tests use mathematically simple equations that can be efficiently evaluated with SQL queries because they generally do not require vector or matrix manipulation. Statistical tests can produce statistically reliable results with both large data sets and small data sets, whereas many data mining and machine learning techniques require large data sets in order to find significant results. It is important to notice that small data sets may appear even when working with large databases, due to analyzing a database at coarse aggregation levels (e.g. grouping by store) or when the distribution behind some selection attribute is skewed (e.g. zipf). On the other hand, compared to standard exploratory OLAP analysis, statistical tests provide more evidence that a finding is indeed significant, going beyond simple comparisons or getting variance proportions. Nevertheless, statistical tests generally require many trial and error runs before a plausible finding is made and each run requires varying parameters or selecting subsets of the data set. With such motivation in mind, our tool automates the process of exploring cuboids from a low dimensional cube trying to find significant measure differences supported by statistical tests. Since the lattice behind the cube dimensions represents a combinatorial search space the problem is computationally challenging; several optimizations are incorporated to make the exhaustive comparison process faster. Our current application is in medical databases.

This is an outline of the rest of the article. Section 2 introduces basic definitions for OLAP databases and statistical tests. Section 3 explains how our tool automatically applies statistical tests on all cuboids from a cube. Section 4 presents an experimental evaluation of visualization

http://doi.acm.org/10.1145/1562849.1562855

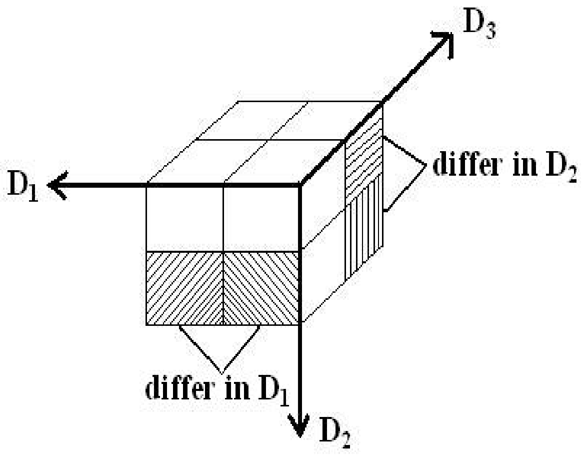


Figure 1: Discovering group pairs with significant measure differences.

and database optimizations with medical data sets. Related work is discussed in Section 5. The conclusions are presented in Section 6.

2. DEFINITIONS

Let F be a fact table with n records having d cube dimensions [11], $D = \{D_1, \ldots, D_d\}$, a set of e measure [11] attributes $A = \{A_1, A_2, \ldots, A_e\}$ and an additional set of f image attributes $I = \{I_1, \ldots, I_f\}$. This set I is required only for visualization. The data structure representing all subsets of dimensions and their containment is called the dimension lattice [11]. Due to their simplicity and wide application in the medical domain we restrict dimensions to be binary. The set of image attributes represents a single image, where each attribute can be a pixel, an image segment or an image region. In OLAP processing, the basic idea is to compute aggregations (sum(), count()) on measures A_i by subsets of dimensions (i.e. cuboids or cuboids) G s.t. $G \subseteq D$, effectively performing aggregations at different granularity levels. The set of all potential aggregations at a certain level is called a cuboid and one specific group is called a cell. In our case, aggregations are used to derive univariate statistics such as μ, σ, which in turn are the basic elements in the equations of a parametric statistical test, introduced in Section 3.

Example

In Figure 1 we present an example of a cube having three dimensions D_1, D_2, D_3. Each face represents a 2-dimensional cuboid. As can be seen, there exist two sets of cell pairs within one cuboid that differ in exactly one dimension. The difference in fill pattern is indicating there is a significant difference on a measure attribute.

3. APPLYING STATISTICAL TESTS ON OLAP CUBES

This section introduces our main technical contributions. We explain how to apply statistical tests to explore OLAP cubes. We propose an algorithm that explores the cube at several dimension granularities to compare highly similar groups. Since we consider data sets with alphanumeric and imaging attributes, our algorithm performs processing of image attributes in order to provide visualization. We introduce optimizations to generate efficient SQL code. We discuss the application of our research in medical databases.

3.1 Statistical Tests

Instead of looking for unusual patterns in cuboids like previous work [8, 10, 9], we propose to use a statistical test to compare pairs of cells in cuboids, providing a more reliable discovery. Most existing work relies on simple comparisons and multi-level aggregations to find interesting findings. Instead, in our approach we exploit a parametric statistical test comparing populations means [22]. Such approach provides the following advantages:

- Two large groups of any size can be compared. Two groups with very different number of elements can be compared (e.g. a large and a small group).
- The means comparison takes into account data variance, which measures measure overlap between the corresponding subpopulations.
- In the case of OLAP, dimensions can be used to focus on highly similar groups, differing in a few dimensions.
- It represents a natural extension of OLAP computations since it relies on distributive aggregations [9].
- Measures are assumed to have a normal distribution, which is applicable in most cases.

We now describe the parametric statistical test in more formal terms. We use a statistical comparison test of the means μ_1, μ_2 from two data subsets (populations), where the size of each data subset is N_1, N_2. Each data subset is assumed to be an independent sample. In this case the null hypothesis H_0 states that $\mu_1 = \mu_2$ and the goal is to find cells where H_0 can be rejected (deemed false) with high confidence $1 - p$, where p generally takes the following thresholds, $p \in \{0.01, 0.05, 0.10\}$. Therefore, the so-called alternative hypothesis H_1 asserts $\mu_1 \neq \mu_2$. When H_0 can be rejected the test will return the significance level p; such outcome will allow us to provide strong statistical evidence supporting $H_1 : \mu_1 \neq \mu_2$. Otherwise, when $p > 0.1$ there does not exist a significant means difference. We use a two-tailed test which allows finding a significant difference on both tails of the Gaussian distribution. The statistical test relies on Equation 1 to compute a random variable z with pdf $N(0, 1)$:

$$z = \frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2/N_1 + \sigma_2^2/N_2}}, \tag{1}$$

where μ_i, σ_i correspond to the estimated mean and standard deviation from group $1, 2$, respectively. When both groups are large the z value just needs to be compared with $z_{p/2}$ in the cumulative probability table for $N(0, 1)$ (e.g. $z_{0.495}$) When either group is small or both groups are small the statistical test requires computing the degrees of freedom as

$$df = N_1 + N_2 - 2 \tag{2}$$

and then looking up z on the t-student distribution table according to df. This implies that either $N_1 > 1$ or $N_2 > 1$ because $df \geq 1$. If one group is much larger than the other one then there exists a row for $df = \infty$. For instance, it is possible to compare a singleton set with a large set of records.

Applying the statistical test on cubes

Applying the statistical test has two goals: (1) finding significant differences between two groups in a cuboid on at least one measure. Finding two or more significant measure differences is desirable, but rare. (2) When there exists a significant difference we focus on groups that differ in one dimension, which can explain cause-effect. Even though dimensions are considered independent the aggregation automatically groups records with correlated dimensions together. Therefore, if a high correlation exists in binary dimensions it will be automatically considered. With respect to the first goal, when applying a statistical test a significant difference can only be supported by a small p-value which takes into account both the means and the standard deviation of the distributions. The smaller the p-value the more likely the difference between both groups is significant. It is expected that many differences will not be significant, making the search problem expensive. Regarding the second goal, we are interested in finding significant differences in highly similar groups because that helps explain which specific dimension "triggers" a significant change on the cuboid measure. For instance, finding a significant measure difference, between two highly dissimilar groups, makes causal explanation difficult, since such difference may be attributed to two or more dimensions. Nevertheless, such less significant findings can be stored on additional tiers of group pairs.

3.2 Algorithm

We introduce an algorithm that integrates cube exploration, statistical tests and visualization. Our algorithm has the following goals: (1) exploring all cuboids from F when d is medium or low. Otherwise, when d is large, exploring all cuboids based on k dimensions selected by the user s.t. $k < d$. (2) running the statistical test for every pair. (3) selecting significant pairs differing in δ cube dimensions. (4) interactive visual exploration of the cube together with statistically significant results. (5) efficient visualization of image data associated with each group.

Our algorithm basically computes the entire cube, exploring the entire dimension lattice and then applies statistical tests for every pair. The algorithm assumes a low d or alternatively low k, binary dimensions, which is common in medical databases. Our tool applies a top-down approach exploring all cuboids from a cube, working level-wise.

The algorithm input and output is as follows:

- Input parameters: maximum p-value, δ threshold of maximum # of different dimensions (generally $\delta = 1$).
- Output: a table C containing all cell pairs differing in δ dimensions.

The algorithm steps are the following:

1. Precompute cube with d dimensions on all e measures getting groups at the finest granularity level.
2. Traverse the lattice. Compute sufficient statistics N, L, Q for every group in the dimension lattice.
3. Create group pairs with groups differing in at most δ dimensions.
4. Compute subpopulation parameters $\mu_1, \sigma_1, \mu_2, \sigma_2$, based on n, L, Q.
5. Compute a statistical test for every cell pair on the same level of aggregation.
6. Select pairs having significance value $< p$ with at most δ different dimensions. Categorize results into tiers having 1,2,3 dimensions differences.

When d is small or medium (e.g. our medical data sets), the table F is first aggregated at the finest granular level as given by $D_1, \ldots, D_d$ and then the cube exploration proceeds with subsets of dimensions having $d-1, d-2, \ldots, 1$ dimensions. Otherwise, we assume the user selects k dimensions s.t. $k < d$. Cube pre-computation automatically eliminates groups with zero records (empty groups). When d is large the user specifies a small subset of k dimensions that can be analyzed interactively. Given the combinatorial number of dimension subsets and hence pairs, it is infeasible to explore all of them. Given one level of aggregation the algorithm performs a pair-wise test-based comparison of all group pairs. Such comparison is made on each of the measure attributes. The algorithm tries a list of increasing p-values, in order to find the smallest p value at which H_0 can be rejected. If $p > 0.1$ then the test shows there is little evidence $\mu_1 \neq \mu_2$.

3.3 Exploration and Visualization

After the cube has been built, either with the d or k dimensions, the user can explore it and visualize results. Our prototype allows visualizing pre-processed images through summarization, sampling or visualizing the entire set for a group. We are currently exploring the visualization of individual raw images, when images are not uniform or are bigger. For the remainder of the article we will assume F has pre-processed images at a low resolution. In the case of medical databases image attributes are generally standardized, allowing uniform techniques for visualization. That is, image attributes have been already pre-processed when they are stored in the DBMS.

Our prototype can do the following:

1. Visualizing the cube and all cuboids in a 2D representation.
2. Highlighting significant pairs of cells indicating their different dimension and equal dimensions.
3. Isolating the measure attribute where the significant difference was found.
4. Visualizing associated image data based on image summaries or randomly selected samples (to speedup visualization)

5. Interactively navigating the cube, allowing the user to jump from one cell to another cell.

The 2D representation of the cube follows a checkerboard display similar to a Karnaugh map (as used in digital design), where cells differing in one dimension are visually linked. Each cell has a fill pattern determined by the 1s and 0s determined by the specific aggregation group it represents. Image summaries provide an accurate representation on what an average record looks like; this is enabled by image standardization. Samples are needed to inspect a few images coming from a large group since it would be infeasible to manually inspect a large number of images. When the user switches from one pair to another pair the program dynamically retrieves and displays the corresponding average, sample or "all" images, based on user's selection. In general, our prototype retrieves records and imaging attributes from tables in the DBMS.

3.4 Optimizations

Our tool generates standard SQL code and it can connect to any relational DBMS through the JDBC interface. After exploring a cube the tool produces an output table with the most important group pairs where mean differences are significant, indicating which dimensions are equal and which specific dimension is different.

Our algorithm was implemented by dynamically generating SQL queries to build the d-dimensional cube, traverse the lattice, form pairs and compute the statistical test. We found several issues in trying to optimize SQL queries. (1) It is not possible to pre-define a general-purposes primary index for the cube because d may vary and the corresponding columns will be different, given different fact tables F. Instead the cube table has either a simple primary key to uniquely identify groups (cube cells) or a primary index on the dimensions. (2) A search for a specific cell requires handling nulls and "All" separately. In particular they were coded as negative integer values (i.e. codes different from 0 and 1). (3) Traversing the entire dimension lattice is the most time-consuming stage, especially for subsets of dimensions around $d/2$. (4) OLAP cubes combined with statistical tests are not an "association rule" problem. (5) When visualization is required, it may be necessary to inspect individual record images to get a more concrete idea about statistical findings or when images have not been pre-processed. Therefore, image sampling is required when N is large for some group.

Computing statistical tests on cubes although it seems a similar problem to association rules [11], it is different because there is no "minimum support" threshold and binary dimensions are not equivalent to items [2]. This is because both 1 and 0 are used to get groups, whereas association rule algorithms generally consider only 1s. Each group may have any combination of 0s and 1s in the dimensions.

We introduce the following optimizations: (1) All aggregations are stored on the same table. This table contains all cuboids at different aggregation granularities. The table has a secondary index on all dimensions coding "All" and "null" separately. (2) Search for a specific cell is indexed. The secondary index allows efficient retrieval of cell pairs and associated image data in one indexed search per group (i.e. two indexed searches per pair). (3) All measure and image attributes are uniformly manipulated with sufficient statistics N, L, Q, to be explained below. We are not currently interested in finding significant differences in image regions, but sufficient statistics open that possibility. (4) When there are image attributes we introduce two optimizations. Image attributes are aggregated per cell for visualization. Each cell has its "average" image. That is, OLAP computations are extended to image attributes. Individual images can be visualized performing sampling from a specific group from F, or all images can be visualized when the group (cell) has a small N. Image retrieval is done in a single query, retrieving all image information for display purposes. Once images are retrieved they are held in main memory for visualization.

Contrary to common intuition, when d is small a "bottom-up" level-wise algorithm is not used. Since dimensions are assumed to be independent and the statistical test relies on averages rather than counts (frequencies), downward closure ("association rule"-like) optimizations are not applicable. That is, we cannot explore $k-1$-dimensional cuboids in order to prune out k-dimensional cuboids. In fact, a significant means difference might be found between cuboids having very different N_1, N_2. This leads to an exhaustive search process, computationally expensive for a high-d cube.

The tool uses the following optimizations for efficient statistical test computation: (1) F is aggregated at the finest granular level producing a cube table C and further coarser-grained aggregations are computed from C. This step is extremely important to eliminate empty cells in the cube. Only groups (cells) with $N > 0$ participate in further processing. (2) Sufficient statistics are computed on each cell so that univariate statistics can be derived in one pass. Such statistics include count(*), sum(A_i) and sum(A_i^2) for a measure column A_i. More specifically, $N_i, L_i = \sum A_i, Q_i = \sum A_i^2$ for each A_i. This idea is also applied to image data: $N_i^I, L_i^I = \sum I_i, Q_i^I = \sum I_i^2$ for I_i. Then $\mu_1, \mu_2, \sigma_1, \sigma_2$ are easily derived from sufficient statistics. (3) In general, for large groups it is required to compare the test statistic against a given z value using the normal $N(0,1)$ distribution, which generally requires looking up a value in a small table. We introduce a simple optimization, finding the significance of the test statistic without visiting such table with a CASE statement based on the specific p-values commonly used in the medical domain ($p \in \{0.01, 0.05, 0.10\}$). That is, finding $z_{p/2}$ for the two-tailed test is done in main memory. On the other hand, when the group is small (say < 30) we perform an indexed search on the t-student distribution using df as the search key. (4) Depending on input parameters, the SQL code will compare only groups having up to δ dimension differences, being $\delta = 1$ the default. This means groups in a pair must differ in up to δ dimensions in order to be compared.

3.5 Medical Application: Finding Differences between Similar Groups

Table 1 shows actual significant findings on a medical data set, explained in more detail in Section 4. Each row represents the comparison between two patient groups, differing in one dimension indicated by "0/1". Cube dimensions are well-known risk factors for heart disease including family history of heart disease, diabetes, gender, high cholesterol and high blood pressure. For a given group, each matching dimension will be 0, indicating absence of a risk factor, 1 indicating presence of a risk factor, or "All" indicating such dimension was ignored in the aggregation. When p is small it indicates two highly similar groups of patients, differing

D_1 FamHist	D_2 Diab	D_3 Gender	D_4 HighChol	D_5 highBP	N_1	N_2	A_1 LAD	A_2 RCA
0	All	All	1	0/1	35	23	$p > 0.1$	$p \in [0.01 - 0.05]$
0/1	All	All	1	0	35	26	$p > 0.1$	$p < 0.01$
All	0/1	All	All	All	47	157	$p < 0.01$	$p < 0.01$

Table 1: Medical database: group pairs with significant measure differences.

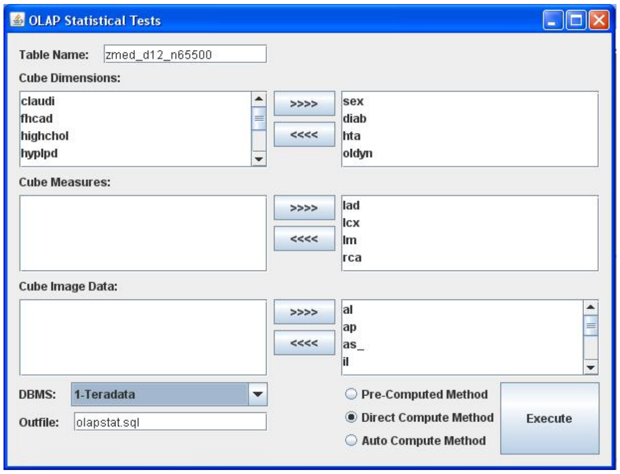


Table Name: zmed_d12_n65500

Cube Dimensions:

claudi
fhcad
highchol
hyplpd

sex
diab
hta
oldyn

Cube Measures:

lad
lcx
lm
rca

Cube Image Data:

al
ap
as_
il

DBMS: 1-Teradata

Outfile: olapstat.sql

○ Pre-Computed Method
● Direct Compute Method
○ Auto Compute Method

Figure 2: OLAP statistical test GUI.

in exactly one risk factor have a high likelihood of having a different artery health (medical measurement of artery narrowing in this case).

3.6 GUI for Visualization and Exploration

We are currently applying our prototype on a medical data set having heart images, where such images are normalized. The images are pre-processed to get 32 regions, describing an 8×4 2-dimensional map of the heart muscle. For medical interpretation these 32 regions are condensed into 9 regions that are in the range [-1,1]. A value close to -1 indicates a healthy region, whereas numbers closer to +1 indicate a severe degree of disease, with 0 being neutral. These numbers allow visualizing a spectrum of patients' health in color. This scale also allows a uniform visualization technique for heart images of diverse patients.

Our tool GUI and visualization aids is illustrated in Figure 2, where the medical doctor can select cube dimensions, cube measures and image attributes. The output from the tool is shown in Figure 3 (for an average image per group) and Figure 4 (with a sample of heart images per group). The tool allows 2D visualization for the dimension lattice on the left panel for a specific dimensionality k, where $2 \leq k \leq d$. This 2D visualization is based on a checker board display, where red means "1" (risk factor present) and blue means "0" (risk factor absent). Since this checkerboard is based on a set of k selected binary dimensions there are up to $\binom{d}{k}$ cells. A pair of cells linked by the green lines means they differ in one dimension, which highlights a specific risk factor triggering heart disease. The left upper part indicates dimensions which are equal, while the right panel indicates the specific dimension that is different for each group (diabetes in this case). For heart image attributes, on the right two windows the medical doctor can visualize for a group with many records a summary of all its images, or alternatively, a sample dynamically retrieved from the database. Otherwise, when the group is small enough all its images can be visualized (perhaps as thumbnails). In short, these two windows enable visualization of image data, with a scale going from -1 (very sick) to +1 (completely healthy).

4. EXPERIMENTAL EVALUATION

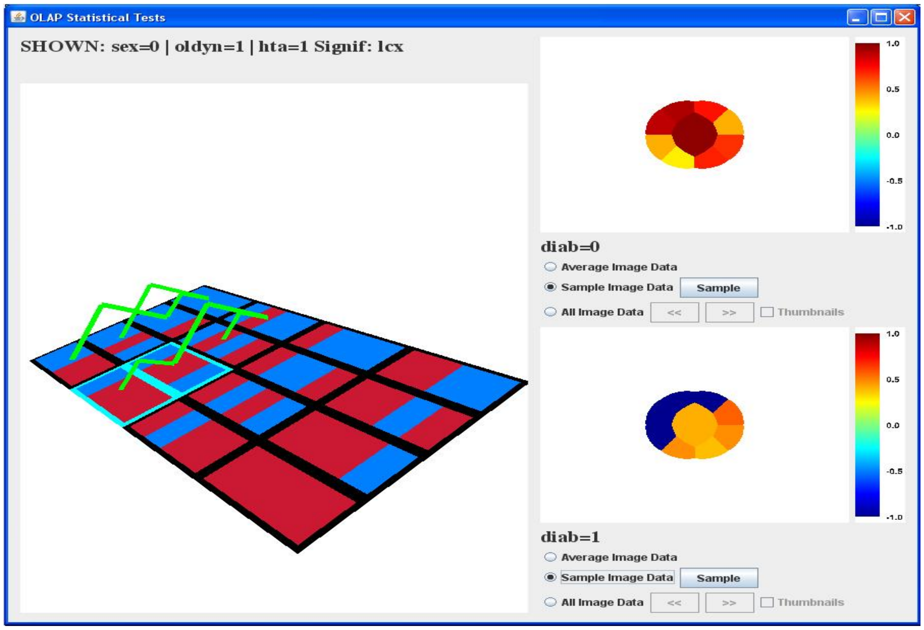
Our tool is an OLAP query generator developed in the Java language, where queries are written in ANSI SQL for maximum portability. Our tool connects to the DBMS via the standard JDBC (Java Database Connectivity) interface. We conducted our experiments on a modern DBMS. The DBMS was SQL Server running on a server with a CPU at 3.2GHz, 4GB of memory and 750GB on disk.

4.1 Data Sets

We performed experiments based on two real data sets coming from the medical domain. The first medical data set contains profiles of $n = 655$ patients and has 25 attributes containing categorical, numeric and image data. This data set was obtained from a hospital and we call it the "Heart" data set. There were medical measurements such as weight, heart rate, blood pressure and pre-existence of related diseases. Finally, the data set contains the degree of artery narrowing (stenosis) for the four heart arteries. All numeric attributes were converted to binary dimensions. There were $d = 12$ binary dimensions (e.g. gender, hypertension Y/N), $e = 4$ measures (artery disease measurement) $f = 9$ image attributes representing a standardized image of the heart. The second data set was obtained from the UCI Machine Learning repository [3] and we call it "Thyroid". The Thyroid data set contained the profiles of $n = 9,172$ patients. We transformed this data set to have $d = 10$ binary dimensions and $e = 5$ measures; this data set had no image attributes. These data sets were treated as the fact table F, defined in Section 2.

4.2 Default settings

For our medical data sets our goal was to explore the entire dimension lattice. Therefore, we used all d dimensions. The settings for parameters were as follows. $p = 0.01$, $\delta = 1$, which can be interpreted as follows. We want to find significant measure differences, with 99% confidence, on all group pairs differing in one dimension. A group pair in the cube



SHOWN: sex=0 | oldyn=1 | hta=1 Signif: lcx

diab=0

○ Average Image Data
● Sample Image Data
○ All Image Data

diab=1

○ Average Image Data
● Sample Image Data
○ All Image Data

Figure 3: Cube exploration in 2D and image visualization (average image).

can have from 1 to d dimensions, out of which one will be different. It is possible, but unlikely, that a group pair has significant differences in two or more measures.

4.3 Significant Group Pairs

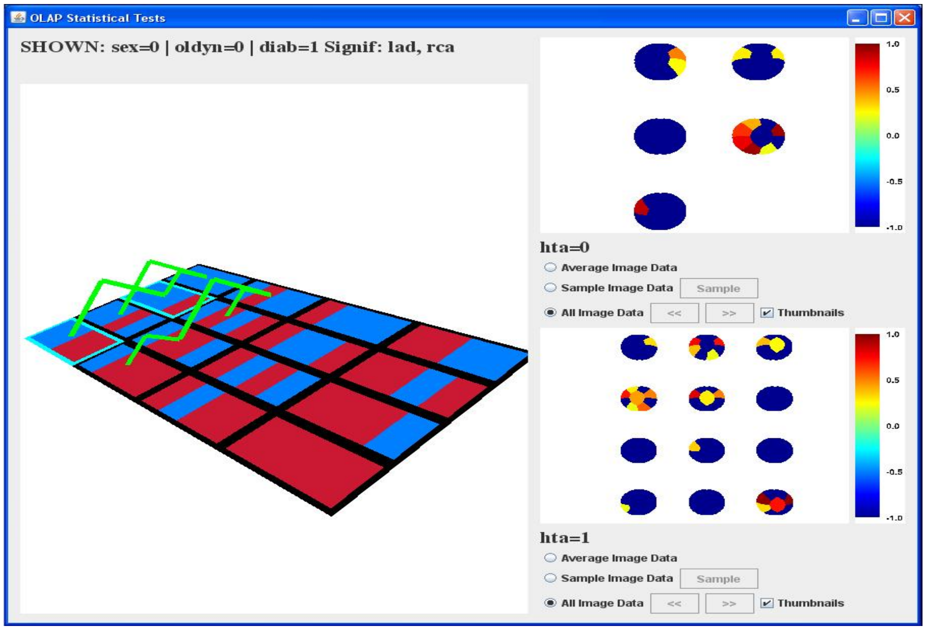
We provide a summary of pairs having a significant difference in at least one measure for both data sets. Table 2 summarizes significant pairs, which represent potentially valuable medical knowledge. Those pairs where $p < 0.01$ are valuable since they show a discriminating risk factor (binary dimension) causes a significant measure change. Those pairs whose p-value is between 0.05 and 0.10 are considered unimportant findings and therefore we do not show them. The most important pairs are those with a few equal dimensions (1 or 2) because they are general and involve larger groups, and those with many equal dimensions (8 for Heart data set [19], 10 for Thyroid data set [3]) because they summarize redundant subsets in the middle of the lattice, but they tend to be specific.

4.4 Image Visualization

We now discuss experiments for image visualization. These experiments illustrate the interactive response of our tool to explore the cube, isolate significant pairs and visualizing associated image data. We recomputed the cube at each d and then we measured the time to retrieve one group at each d. Based on expert opinion, we initially ranked dimensions in order of medical importance so that the most important dimension had rank one. Figure 5 shows time growth to retrieve images from the database. We compare the time to visualize the "average" image versus sampling one image

Table 2: Significant group pair differences.

Data Set	p-value	# equal dims	# pairs
Heart	<0.01	1	6
Heart	<0.01	2	76
Heart	<0.01	3	378
Heart	<0.01	4	974
Heart	<0.01	5	1436
Heart	<0.01	6	1287
Heart	<0.01	7	705
Heart	<0.01	8	201
Heart	<0.01	9	0
Thyroid	<0.01	1	8
Thyroid	<0.01	2	88
Thyroid	<0.01	3	406
Thyroid	<0.01	4	1068
Thyroid	<0.01	5	1780
Thyroid	<0.01	6	1966
Thyroid	<0.01	7	1441
Thyroid	<0.01	8	680
Thyroid	<0.01	9	188
Thyroid	<0.01	10	23



SHOWN: sex=0 | oldyn=0 | diab=1 Signif: lad, rca

hta=0

○ Average Image Data
○ Sample Image Data
● All Image Data ☑ Thumbnails

hta=1

○ Average Image Data
○ Sample Image Data
● All Image Data ☑ Thumbnails

Figure 4: Cube exploration in 2D and image visualization (sample of images).

from one cube cell (group); times are in milliseconds. In this case the cube is already pre-computed and we dynamically retrieve one average or one sample image for the group based on the following common medical dimensions: oldyn, sex, diab, hta, highchol. This represents a highly interesting group with specific risk factors. The left graph shows the time to retrieve images in this group varying n. In this case the time to retrieve an average image remains constant because the average image is already precomputed from sufficient statistics and the search is fully indexed for equality search. On the other hand, the time to retrieve one sample image grows with n because the group also grows in size. The plot shows time to retrieve all images as d grows, with C being recomputed at every d. We do not include the time to precompute C in the time to retrieve images. We can see the time to get the average image grows slowly as d grows showing an asymptotic behavior. On the other hand, we can see the time to get a sample image from one group gets increasingly faster because the group shrinks in size. The time to retrieve one sample should be greater than the time to retrieve the average image because the sample is dynamically obtained from one group which cannot be obtained through a fully indexed search.

Table 3: Image retrieval varying n (msecs).

n	Average	Sample	All
655	40	31	32
1310	40	37	41
2620	40	45	75
5240	41	46	100
10480	41	49	231
20960	42	51	487
41920	42	86	907
83840	43	146	1722

In order to give a complete experimental evaluation on image visualization, we now discuss experiments including the time to retrieve all images in one cell. Table 3 shows the times to retrieve images varying n and Table 4 has the times to retrieve images varying d. We include the times to retrieve the average image from the cube, one sample image from one group and the time to retrieve all images from one group. Notice retrieving all images from a group incurs on significant overhead, but it may still be acceptable for data sets having $n <$ 100k. Our image retrieval is done in a single query, retrieving all image information for display purposes. Once images are retrieved they are held in main memory for visualization.

4.5 Time complexity

Our previous experiments showed the challenge for our problem is d and not n. Nevertheless, we created larger data sets replicating F several times and we measured time for the entire process. The contribution of n to total time was so small that a plot varying n was a flat line. Therefore, we omit such plot.

Figure 6 illustrates time scalability as we vary d. The entire cube, cuboids, pairs and statistical test are computed at each d. The left graph corresponds to the Heart data set and the right graph corresponds to the Thyroid data

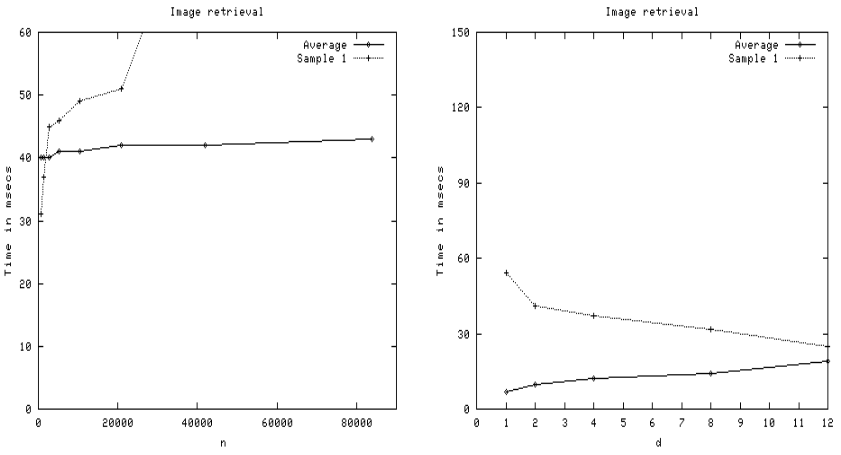


Figure 5: Time complexity for image retrieval; times in msecs.

Table 4: Image retrieval varying d (msecs).

d	Average	Sample	All
1	16	125	2100
2	30	106	1279
4	39	93	506
8	42	79	203
12	43	67	94

Table 5: Profile of cube exploration ($d = 12$ and $d = 10$, time in secs).

Step	Heart		Thyroid	
	Time	%	Time	%
Cube & NLQ	< 1	0	< 1	0
Get N, L, Q on lattice	495	31	116	34
Compute μ, σ from N, L, Q	173	11	28	8
Create group pairs based on δ	158	10	39	12
Compute test statistic	419	26	87	26
Categorize based on p-value	344	22	66	20

set. It is clear the time complexity grows exponentially as d increases, highlighting the expensive search process for significant pairs. However, for our medical data sets the entire lattice is explored in a matter of minutes.

4.6 Profiling Processing Steps

Table 5 gives a breakdown of the total execution time to explore the entire dimension lattice. As we can see most computations take significant time, despite F being a small data set. Traversing the dimension lattice is the slowest operation; this step requires creating an output group for each dimension combination and it is I/O bound since it requires visiting every F row. Computing the test statistic is the second slowest operation; this is mostly CPU time. Classifying pairs into tiers for final analysis comes slightly behind. The fastest operation is computing the cube at the finest granularity level to get groups having d dimensions.

Step	Data set	N	Y	Impr
cube & NLQ	Heart	na	< 1	-
cube & NLQ $n \times 1000$	Heart	na	11	-
lattice NLQ from cube	Heart	1396	1366	-2%
Primary index on dims	Heart	1366	400	-70%
cube & NLQ	Thyroid	na	< 1	-
cube & NLQ $n \times 1000$	Thyroid	na	82	-
lattice NLQ from cube	Thyroid	336	216	-36%
Primary index on dims	Thyroid	216	100	-54%

Table 6: Optimizations (na=not applicable, impr=improvement).

Table 6 compares optimizations. For the sake of completeness, we simulated a large fact table by replicating F 1000 times. This gives two large data sets having n =655k rows and $n \approx$ 9M rows, respectively. We want to understand how much longer it takes to compute the group-by query to get the cube on d dimensions. As we can see the cube can still be efficiently computed on large fact tables. The rest of the steps remain unaffected with the same measured time. Time for computing the cube directly from the data set F is not applicable in the first two rows. Precomputing the d cube and computing N, L, Q is compared with directly computing N, L, Q from F. That is, we want to understand if it is worth it to compress the (small) data set by precomputing the cube at the finest granularity level. As we can see there is an important performance improvement for the Thyroid data set. A primary index on dimensions is compared with a simple primary key for each group. In this case we want to understand the improvement on time to search each group efficiently. As we can see the index in dimension has a significant impact on performance.

5. RELATED WORK

Cube exploration is a well researched topic. OLAP and a classification of aggregations originates in the seminal paper [9]. Methods to increase the performance of multidimensional aggregations, by combining novel data structures and precomputation at different aggregation levels, are introduced in [1, 8]. The authors of [10] puts forward the plan

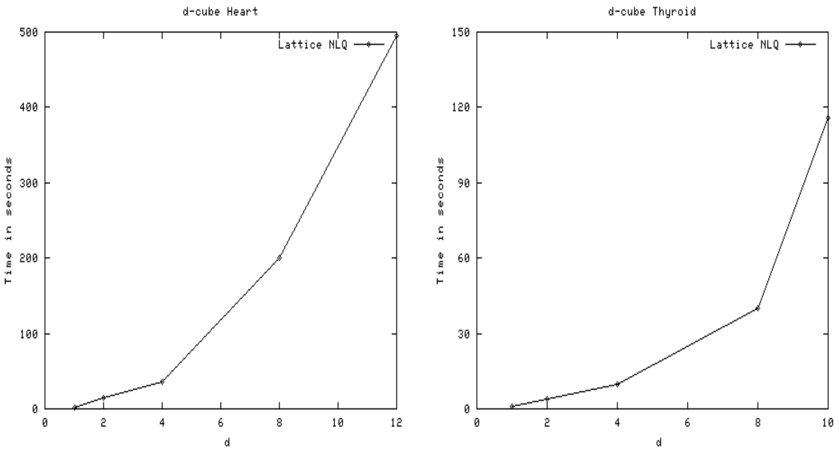


Figure 6: Time complexity to compute cube varying *d*.

of creating smaller, indexed summary tables from the original large input table to speed up aggregating executions. In [21] the authors explore tools to guide the user to interesting regions in order to highlight anomalous behavior while exploring large OLAP data cubes. This is done by identifying exceptions, that is, values in cells of a data cube that are significantly different from the value anticipated, based on a statistical model. In contrast, we propose to use statistical tests to do pair-wise comparison of neighboring cells in cuboids to discover significant metric differences between similar groups. We identify such differences giving statistical evidence about the validity of findings. Reference [12] proposes computing large lattices with a greedy algorithm.

Our work is related to applying data mining in medical data sets to improve heart disease diagnosis [19, 18]. These works discuss the medical significance of pairs to detect specific risks for heart disease. Neural networks are used to predict heart response based on exercise stress and heart muscle thickening images [4]. A basic set of search constraints is introduced in [20] and experimental results stress their importance.

There has been research on visualizing OLAP cubes. Reference [15] studies the problem of improving understanding through the use of visualization. Recent work can also be found on the use of tools to visually and interactively explore OLAP warehouses. The authors for [24] explore the requirements for analyzing a spatial database with an OLAP tool. This work shows the need to apply spatial data techniques, used in geographic information systems, for OLAP exploration, in which drill up/down, pivoting, and slicing and dicing provide a complementary perspective. In contrast, our work relies on statistical tests to explore OLAP cubes and can automatically detect significant metric differences between highly similar groups. Additional visualization work was completed in [16] where the mapping of the Cube Presentation Model, a display model for OLAP screens, involves visualization techniques from the Human-Computer Interaction field. In [14], the author presents a rigorous multidimensional visualization methodology for visualizing n-dimensional geometry and its applications to visual and automatic knowledge discovery. The application of visual knowledge discovery techniques is possible by transforming the problem of searching for multivariate relations among the variables into a two-dimensional pattern recognition problem. A framework for exploration of OLAP data with user-defined dynamic hierarchical visualizations is presented in [23]. While this study emphasizes the use of visualization tools to explore data warehouses, we are proposing a tool that not only gives the user visual aids to explore the data, but also to present the user with a novel method of highlighting interesting features of the cubes by means of statistical tests. Indexing is one method of increasing performance in the searching of images and the authors in [6] proposed a multilevel index structure that can efficiently handle queries on video data.

6. CONCLUSIONS

In this article, we presented a prototype which combines OLAP cube exploration, statistical tests and visualization. Statistical tests are applied on pairs of similar groups in order to find significant measure differences caused by some distinctive dimension. The cube is explored with automatically generated SQL code. Our tool can produce statistically reliable results on both large and small subsets. The internal aggregation algorithm incorporates several database optimizations. The cube is precomputed when the number of cube dimensions is low. Otherwise, the program works with a user-specified set of dimensions to pre-compute a lower dimensional cube. The cube incorporates a primary index on dimensions for efficient group search. We also introduced optimizations for interactive visualization of the cube, statistical test results and associated image data. The fact table is indexed for efficient image retrieval. Image attributes are uniformly treated as measures in order to get an average representative image per group through sufficient statistics or to collect sample images.

There are many research issues for future work. The combination of OLAP processing and visualization creates several research challenges. In particular, visualization of high-dimensional cubes requires novel cube data transformation techniques. In the case of medical databases image attributes can be efficiently visualized in a cell, but this could not be done for non-uniform images: dynamic image compression techniques could be required to interactively visualize them. We want to study how to further optimize pair creation and computation of statistical tests. There is a big family of statistical tests that may be applied in OLAP databases. The cube provides a natural way to summarize large databases which is more difficult if underlying records have image attributes.

Acknowledgments

We would like to thank the Emory University Hospital for providing the medical data set used in this work.

7. REFERENCES

[1] S. Agarwal, R. Agrawal, and P. Deshpande. On the computation of multidimensional aggregates. In *VLDB*, pages 506–521, 1996.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Conference*, pages 207–216, 1993.

[3] A. Asuncion and D.J. Newman. *UCI Machine Learning Repository*. University of California, Irvine. School of Inf. and Comp. Sci., 2007.

[4] L. Braal, N. Ezquerra, E. Schwartz, and Ernest V. Garcia. Analyzing and predicting images through a neural network approach. In *Proc. of Visualization in Biomedical Computing*, pages 253–258, 1996.

[5] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65–74, 1997.

[6] L. Chen, M. Ozsu, and V. Oria. Mindex: An efficient index structure for salient-object-based queries in video databases. *Multimedia Syst.*, 10(1):56–71, 2004.

[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, November 1996.

[8] Lixin Fu and J. Hammer. Cubist: a new algorithm for improving the performance of ad-hoc OLAP queries. In *Proc. ACM DOLAP Workshop*, 2000.

[9] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-total. In *ICDE Conference*, pages 152–159, 1996.

[10] H. Gupta, V. Harinarayan, A. Rajaraman, and J.D. Ullman. Index selection for OLAP. In *IEEE ICDE Conference*, 1997.

[11] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, 1st edition, 2001.

[12] V. Harinarayan, A. Rajaraman, and J.D. Ullman. Implementing data cubes efficiently. In *ACM SIGMOD Conference*, pages 205–216, 1996.

[13] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, New York, 1st edition, 2001.

[14] A. Inselberg. Visualization and knowledge discovery for high dimensional data. In *UIDIS*, pages 5–24, 2001.

[15] D. A. Keim, C. Panse, J. Schneidewind, M. Sips, M. C. Hao, and U. Dayal. Pushing the limit in visual data exploration: Techniques and applications. In *KI*, pages 37–51, 2003.

[16] A. S. Maniatis, P. Vassiliadis, S. Skiadopoulos, and Y. Vassiliou. Advanced visualization for OLAP. In *Proc. ACM DOLAP Workshop*, pages 9–16, New York, NY, USA, 2003. ACM Press.

[17] T.M. Mitchell. *Machine Learning*. Mac-Graw Hill, New York, 1997.

[18] C. Ordonez. Association rule discovery with the train and test approach for heart disease prediction. *IEEE Transactions on Information Technology in Biomedicine (TITB)*, 10(2):334–343, 2006.

[19] C. Ordonez, N. Ezquerra, and C.A. Santana. Constraining and summarizing association rules in medical data. *Knowledge and Information Systems (KAIS)*, 9(3):259–283, 2006.

[20] C. Ordonez, E. Omiecinski, Levien de Braal, Cesar Santana, and N. Ezquerra. Mining constrained association rules to predict heart disease. In *IEEE ICDM Conference*, pages 433–440, 2001.

[21] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In *EDBT*, pages 168–182. Springer-Verlag, 1998.

[22] M. Triola. *Essentials of Statistics*. Addison Wesley, 2nd edition, 2005.

[23] S. Vinnik and F. Mansmann. From analysis to interactive exploration: Building visual hierarchies from OLAP cubes. In *EDBT*, pages 496–514, 2006.

[24] A. Voß, V. Hernandez, H. Voß, and S. Scheider. Interactive visual exploration of multidimensional data: Requirements for CommonGIS with OLAP. In *DEXA Workshops*, pages 883–887, 2004.