LA-UR- 09-00378

Approved for public release; distribution is unlimited.

Title: View Discovery in OLAP Databases through Statistical Combinatorial Optimization

Author(s): Cliff Joslyn,
John Burke,
Terence Critchlow,
Nick Hengartner,
Emilie Hogan

Intended for: Publication in the proceeding of SSDBM

Los Alamos
NATIONAL LABORATORY
EST. 1943

View Discovery in OLAP Databases through Statistical Combinatorial Optimization

Cliff Joslyn[1], John Burke[1], Terence Critchlow[1],
Nick Hengartner[2], and Emilie Hogan[1,3]

[1] Pacific Northwest National Laboratory
[2] Los Alamos National Laboratory
[3] Mathematics Department, Rutgers University

Abstract. The capability of OLAP database software systems to handle data complexity comes at a high price for analysts, presenting them a combinatorially vast space of views of a relational database. We respond to the need to deploy technologies sufficient to allow users to guide themselves to areas of local structure by casting the space of "views" of an OLAP database a combinatorial object of all projections and subsets, and "view discovery" as an optimization process over that lattice. We equip the view lattice with statistical information theoretical measures sufficient to support a combinatorial search process. We outline "hop-chaining" as a particular view discovery algorithm over this object, wherein users are guided across a permutation of the dimensions by searching for successive two-dimensional views, pushing seen dimensions into an increasingly large background filter in a "spiraling" search process. We illustrate this work in the context of data cubes recording summary statistics for radiation portal monitors at US ports.

1 Introduction and Related Work

OnLine Analytical Processing (OLAP) [1, 7, 8] is a relational database technology providing users with rapid access to summary, aggregated views of a single large database, and is widely recognized for knowledge representation and discovery in high-dimensional relational databases. OLAP technologies provide intuitive and graphical access to the massively complex set of possible summary views available in large relational (SQL) structured data repositories [21]. But the ability of OLAP database software systems, such as the industry-leading Hyperion[4] and ProClarity[5] platforms, to handle data complexity comes at a high price for analysts. The available portions and projections of the overall data space present a bewilderingly wide-ranging, combinatorially vast, space of options. There is an urgent need for knowledge discovery techniques that guide users' knowledge discovery tasks; to find relevant patterns, trends, and anomalies; and to do so within the intuitive interfaces provided by "business intelligence" OLAP tools.

Through the Generalized Data-Driven Analysis and Integration (GDDAI) Project [10], our team has been developing both pure and hybrid OLAP data

[4] `http://www.oracle.com/technology/products/bi/essbase/visual-explorer.html`

[5] `http://www.microsoft.com/bi/products/ProClarity/proclarity-overview.aspx`

analysis capabilities for a range of homeland security applications. The overall GDDAI goal is to provide a seamless integration of analysis capabilities, allowing analysts to focus on understanding the *data* instead of the *tools*. We describe GDDAI's approach to knowledge discovery in OLAP data cubes using information-theoretical combinatorial optimization, and as applied in the ProClarity platform on databases of surveillance data from radiation monitors at US ports of entry. We aim at a formalism for user-assisted knowledge discovery in OLAP databases around the fundamental concept of **view chaining**. Users are provided with analytical feedback to guide themselves to areas of high local structure within view space: that is, to significant collections of dimensions and data items (columns and rows, respectively), in an OLAP-structured database.

OLAP is fundamentally concerned with a collection of N variables X^i and a multi-dimensional data relation over their Cartesian product $\mathbf{X} := \bigtimes_{i=1}^{N} X^i$. Thus formalisms for OLAP data analysis are naturally rooted in relational database theory, and OLAP formalisms [2, 9, 30] extend relational calculi and algebras, for example extending the SQL language to its multi-dimensional analog MDX[6]. But OLAP shares mathematical connections with a range of multivariate analytical approaches operable on the space $\mathbf{X}$, for example statistical databases [27]; the analysis of contingency tables [4]; hierarchical log-linear modeling [18]; grand tour methods in multi-variate data visualization [3]; projection pursuit [19]; data tensor analysis [13]; and reconstructibility analysis [12, 20].

Our ultimate goal is to place OLAP knowledge discovery methods within a mathematical context of combinatorial optimization in such a manner as to be realizable within existing industry-standard database patforms. Specifically:

- Given a foundational OLAP database engine platform (e.g. EssBase[7], SSAS[8]);
- And an OLAP client with technology for graphical display and an intuitive interface (e.g. ProClarity, Hyperion);
- Cast the space of **views** (sub-cubes) of an OLAP database as a combinatorial, lattice-theoretical structure;
- Equipped with statistical measures reflecting the structural relations among views (their dimensional scope, depth, etc.) in the context of the data observed within them;
- To support both automated search to areas of high local structure;
- And user-guided exploration of views in the context of these measures.

While we believe that our emphasis on a combinatorial approach is distinct, our work resonates with that of a number of others. Our process of "view chaining", or moving from one projected subset of a data cube to another intersecting in dimensionality, is similar to the navigational processes described by others [23–26], and anticipated in some of our prior work [11]. But we note that approaches which seek out "drill-down paths" [6] only "descend" the view lattice along one "axis" of views with increasing dimensional extension, sequentially adding variables to the view at each step. In contrast, our "hop-chaining" technique chains

[6] `http://msdn.microsoft.com/en-us/library/ms145506.aspx`

[7] `http://www.oracle.com/appserver/business-intelligence/essbase.html`

[8] `http://msdn.microsoft.com/en-us/library/ms175609(SQL.90).aspx`

through a sequence of two-dimensional views, affecting a permutation of the variables X^i.

Our overall approach is consistent with an increasingly large body of similar work drawing on information theoretical statistical measures in data cubes to provide quantities for making navigational choices [20, 22]. However, some other researchers have used different statistical approaches, for example variance estimation [25] or skewness measures [15]. Our primary departure from traditional OLAP analysis is the extension to conditioning and conditional probability measures over views. This not only provides the basis for optimization and navigation, it also creates a strong connection to graphical or structural statistical models [5, 29], graphoid logics [16, 28], as well as systems-theory based structural model induction methodologies [12, 14].

We begin by establishing concepts and notation for operation of OLAP databases as hierarchical data tensors, and then define the **view lattice** over (non-hierarchical, unconditional) views. This brings us to a point where we can explicate the fundamental OLAP operations: projection, extension, filtering, and "flushing" (decreasing a filter). We introduce **conditional views** and the complex combinatorial object which is the **conditional view space**. This prepares us to introduce "hop-chaining" as a particular view discovery algorithm over this combinatorial object, wherein users are guided by conditional information measures across a permutation of the dimensions by searching for successive two-dimensional views, pushing seen dimensions in a "spiraling" search process into an increasingly large background filter. We then illustrate hop-chaining on databases of surveillance data from radiation monitors at US ports of entry.

2 OLAP Formalism

Although the mathematical tools required to analyze OLAP databases are relatively simple, their notational formalisms are inevitably, and regretably, not [2, 9, 30]. While our formalism is similar, it differs in a number of ways as well:

- We combine projections I on dimensions and restrictions J on records into a lattice-theoretical object called a **view** $\mathcal{D}_{I,J}$.
- For this work we consider only a single "measure" (a quantity tracked and aggregated in the data cube) as an integral count of a number of records. Such count measures support the development of frequency distributions, and thus information-theoretical measures, over cube views. Thus we do not deal with other quantitative measures, or the general distinction and interaction between measures and dimensions.
- Our view discovery method is available on flat dimensions which are not hierarchically-structured to support roll-up aggregation and drill-down disaggregation operations. But for completeness within an OLAP context, we introduce the notation and concepts for hierarchical dimensions, develop our methodology without them, and then return to discuss how we handle drill-down in our practical application, and suggest the way forward to future extension to the full hierarchical case.

2.1 Data Cubes as Hierarchical Data Tensors

Let $\mathbb{N} = \{1, 2, \ldots\}, \mathbb{N}_N := \{1, 2, \ldots, N\}$. For some $N \in \mathbb{N}$, define a **data cube** as an N-dimensional hierarchical data tensor $\mathcal{D} := \langle \mathbf{X}, \mathcal{X}, \mathcal{Q}, c \rangle$ where:

- $\mathcal{X} := \{X^i\}_{i=1}^N$ is a collection of N **variables** or **columns** with $X^i := \{x_{k^i}\}_{k^i=1}^{L^i} \in \mathcal{X}$;
- $\mathbf{X} := \times_{X^i \in \mathcal{X}} X^i$ is an overall **data space** or **data schema** whose members are N-dimensional vectors $\boldsymbol{x} = \langle x_{k^1}, x_{k^2}, \ldots, x_{k^N} \rangle = \langle x_{k^i} \rangle_{i=1}^N \in \mathbf{X}$ called **slots**;
- $\mathcal{Q} = \{\mathcal{P}^i\}_{i=1}^N$ is a collection of N partially-ordered hierarchical **dimensions** $\mathcal{P}^i = \langle P^i, \leq^i \rangle$ with **members** $p^i \in P^i$. Each partially-ordered set (poset) $\mathcal{P}^i$ is isomorphic to a sub-poset of the Boolean lattice $\mathcal{B}^i = \left\langle 2^{X^i}, \subseteq \right\rangle$ which is the power set of the values of the variable X^i ordered by set inclusion. We map each subset $Y^i \subseteq X^i$ to its corresponding member $p^i \in P^i$, and denote $Y^i \in P^i$ for simplicity. In particular for each $x^i \in X^i, \{x^i\} \in P^i$ is an atom of $\mathcal{P}^i$. The order relation $\leq^i$ is then isomorphic to a sub-relation of the subset relation $\subseteq$. Additionally, each partial order $\leq^i$ implies the **covering relation** $\prec^i$, where $p \prec^i p'$ means that $p \leq^i p'$ and p is an immediate child of p'. While in principle each poset $\mathcal{P}^i$ could be as large as the full power set, in practice, they are **trees**, with $X^i \in P^i$ and $\forall x^i \in X^i, \{x^i\} \in P^i$.
- $c : \mathbf{X} \to \{0, 1, \ldots\}$ is a **count** function.

Let $M := \sum_{\boldsymbol{x} \in \mathbf{X}} c(\boldsymbol{x})$ be the total number of records in the database. Then $\mathcal{D}$ also has relative frequencies f on the cells, so that $f : \mathbf{X} \to [0, 1]$, where $f(\boldsymbol{x}) = \frac{c(\boldsymbol{x})}{M}$, and thus $\sum_{\boldsymbol{x} \in \mathbf{X}} f(\boldsymbol{x}) = 1$.

We can identify the **OLAP space** or **cube schema** as $\mathbf{P} := \times_{i=1}^N \mathcal{P}^i$, so that each $\boldsymbol{p} \in \mathbf{P}$ is a **cell**. We then have the hierarchical count function $\hat{c} : \mathbf{P} \to \mathbb{N}$, where $\hat{c}(\boldsymbol{p}) := \sum_{\boldsymbol{x} \leq \boldsymbol{p}} c(\boldsymbol{x})$, and $\leq := \times_{i=1}^N \leq^i$, the product order of the hierarchies. Note that we can also define $\hat{c}$ recursively,

$$\hat{c}(\boldsymbol{p}) := \begin{cases} \sum_{\boldsymbol{p}' \prec \boldsymbol{p}} \hat{c}(\boldsymbol{p}') & \nexists \{\boldsymbol{x}\} = \boldsymbol{p} \\ c(\boldsymbol{p}) & \exists \{\boldsymbol{x}\} = \boldsymbol{p} \end{cases} .$$

There is also the hierarchical frequency function $\hat{f} : \mathbf{P} \to [0, 1]$, with $\hat{f}(\boldsymbol{p}) := \frac{\hat{c}(\boldsymbol{p})}{M}$.

In an OLAP database context, the variables $X^i \in \mathcal{X}$ correspond to columns in a relational data table; the dimensions $\mathcal{P}^i \in \mathcal{Q}$ to the dimensions of an OLAP cube; vectors $\boldsymbol{x} \in \mathbf{X}$ to the rows of a fact table; the vectors $\boldsymbol{p} \in \mathbf{P}$ to the cells in an OLAP cube; and the counts $\hat{c}(\boldsymbol{p})$ to the values stored in the $\boldsymbol{p}$ cell of a cube.

An example of a data tensor is shown in Table 1, for $\mathcal{X} = \{X^1, X^2, X^3\} = \{X, Y, Z\}$, with $X = \{\alpha, \beta\}, Y = \{a, b, c\}. Z = \{x, y, z, w\}$, so that $N = 3$. The table shows the counts $c(\boldsymbol{x})$, so that $M = 74$, and the frequencies $f(\boldsymbol{x})$. In general we can assume the entire power set $\mathcal{B}^i = 2^{X^i}$ as a natural hierarchy on each X^i, but we can additionally assert a more specific tree hierarchy $\mathcal{P}^3 = \langle P^3, \leq^3 \rangle$ on Z, with $P^3 = \{x, y, z, w, r = \{x, y\}, s = \{z, w\}, t = Z\}$. Then we have e.g. a

X	Y	Z	$c(x)$	$f(x)$
α	a	x	1	0.014
		z	3	0.041
		w	7	0.095
	b	x	9	0.122
		w	15	0.203
	c	x	2	0.027
		y	8	0.108
		z	4	0.054
		w	1	0.014

X	Y	Z	$c(x)$	$f(x)$
β	a	x	1	0.014
	b	y	4	0.054
		z	3	0.041
		w	3	0.041
	c	x	6	0.081
		y	2	0.027
		z	4	0.054
		w	1	0.014

Table 1. An example data tensor. Blank entries repeat the elements above, and rows with zero counts are suppressed.

typical slot $\boldsymbol{x} = \langle \beta, b, y \rangle \in \mathbf{X}$ and a typical cell $\boldsymbol{p} = \langle \{\beta\}, \{b\}, s \rangle \in \mathbf{P}$. Note that $\hat{c}(\boldsymbol{p}) = c(\langle \beta, b, z \rangle) + c(\langle \beta, b, w \rangle) = 6, \hat{f}(\boldsymbol{p}) = 0.082$.

2.2 Views and Chaining Operations in the View Lattice

In this section we consider only the non-hierarchical case where $\forall \mathcal{P}^i \in \mathcal{Q}, P^i = X^i$, and $\leq^i = \emptyset$. Then $\mathbf{X} = \mathbf{P}$, and $\mathcal{D}$ becomes a regular data tensor, or an N-dimensional contingency table. We return to hierarchical data cubes in Sec. 4.3.

At any time, we may look at a projection of $\mathcal{D}$ along a sub-cross-product involving only certain dimensions $I \subset \mathbb{N}_N$. Call I a **projector**, and denote $\boldsymbol{x} \downarrow I = \langle x_{k'} \rangle_{i \in I} \in \mathbf{X} \downarrow I$ where $\mathbf{X} \downarrow I := \bigtimes_{i \in I} X^i$, as a projected vector and data cube. We write $\boldsymbol{x} \downarrow i$ for $\boldsymbol{x} \downarrow \{i\}$, and for projectors $I \subseteq I'$ and vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{X}$, we use $\boldsymbol{x} \downarrow I \subseteq \boldsymbol{y} \downarrow I'$ to mean $\forall i \in I, \boldsymbol{x} \downarrow i = \boldsymbol{y} \downarrow i$.

Count and frequency functions convey to the projected count and frequency functions $c[I] : \mathbf{X} \downarrow I \rightarrow \mathbb{N}$ and $f[I] : \mathbf{X} \downarrow I \rightarrow [0, 1]$, so that

$$c[I](\boldsymbol{x} \downarrow I) = \sum_{\boldsymbol{x}' \downarrow \mathbb{N}_N \supseteq \boldsymbol{x} \downarrow I} c(\boldsymbol{x}') \tag{1}$$

$$f[I](\boldsymbol{x} \downarrow I) = \sum_{\boldsymbol{x}' \downarrow \mathbb{N}_N \supseteq \boldsymbol{x} \downarrow I} f(\boldsymbol{x}'), \tag{2}$$

and $\sum_{\boldsymbol{x} \downarrow I \in \mathbf{X} \downarrow I} f[I](\boldsymbol{x} \downarrow I) = 1$. In words, we add the counts (resp. frequencies) over all vectors in $\boldsymbol{y} \in \mathbf{X}$ such that $\boldsymbol{y} \downarrow I = \boldsymbol{x} \downarrow I$. This is just the process of building the I-marginal over f, seen as a joint distribution over the X^i for $i \in I$.

Any set of record indices $J \subseteq \mathbb{N}_M$ is called a **filter**. Then we can consider the filtered count function $c^J : \mathbf{X} \rightarrow \{0, 1, \ldots\}$ and frequency function $f^J : \mathbf{X} \rightarrow [0, 1]$ whose values are reduced by the restriction in $J \subseteq \mathbb{N}_M$, now determining

$$M' := \sum_{\boldsymbol{x} \in \mathbf{X}} c^J(\boldsymbol{x}) = |J| \leq M. \tag{3}$$

We renormalize the frequencies f^J over the resulting M' to derive

$$f^J(\boldsymbol{x}) = \frac{c^J(\boldsymbol{x})}{M'}, \tag{4}$$

so that still $\sum_{\boldsymbol{x} \in \mathbf{X}} f^J(\boldsymbol{x}) = 1$.

Finally, when both a selector I and filter J are available, then we have $c^J[I] : \mathbf{X} \downarrow I \to \{0, 1, \ldots\}, f^J[I] : \mathbf{x} \downarrow I \to [0, 1]$ defined analogously, where now $\sum_{\boldsymbol{x} \downarrow I \in \mathbf{X} \downarrow I} f^J[I](\boldsymbol{x} \downarrow I) = 1$. So given a data cube $\mathcal{D}$, denote $\mathcal{D}_{I,J}$ as a **view** of $\mathcal{D}$, restricting our attention to just the J records projected onto just the I dimensions $\mathbf{X} \downarrow I$, and determining count $c^J[I]$ and frequency $f^J[I]$ functions.

In a lattice theoretical context, each projector $I \subseteq \mathbb{N}_N$ can be cast as a point in the Boolean lattice $\mathcal{B}^N$ of dimension N called a **projector lattice**. Similarly, each filter $J \subseteq \mathbb{N}_M$ is a point in a Boolean lattice $\mathcal{B}^M$ called a **filter lattice**. Thus each view $\mathcal{D}_{I,J}$ maps to a unique node in the **view lattice** $\mathcal{B} := \mathcal{B}^N \times \mathcal{B}^M = 2^N \times 2^M$, the Cartesian product of the projector and filter lattices.

We then define **chaining** operations as transitions from an initial view $\mathcal{D}_{I,J}$ to another $\mathcal{D}_{I',J}$ or $\mathcal{D}_{I,J'}$, corresponding to a move in the view lattice $\mathcal{B}$:

Projection: Removal of a dimension so that $I' = I \setminus \{i\}$ for some $i \in I$. This corresponds to moving a single step down in $\mathcal{B}^N$, and to marginalization in statistical analyses. We have $\forall \boldsymbol{x}' \downarrow I' \in \mathbf{X} \downarrow I'$,

$$c^J[I'](\boldsymbol{x}' \downarrow I') = \sum_{\boldsymbol{x} \downarrow I \supseteq \boldsymbol{x}' \downarrow I'} c^J[I](\boldsymbol{x}). \tag{5}$$

Extension: Addition of a dimension so that $I' = I \cup \{i\}$ for some $i \notin I$. This corresponds to moving a single step up in $\mathcal{B}^N$. Rather than aggregating, we're now *disaggregating* or *distributing* information about the I dimensions over the $I' \setminus I$ dimensions. Notationally, we have the converse of (5), so that $\forall \boldsymbol{x} \downarrow I \in \mathbf{X} \downarrow I$,

$$\sum_{\boldsymbol{x}' \downarrow I' \supseteq \boldsymbol{x} \downarrow I} c^J[I'](x') = c^J[I](\boldsymbol{x} \downarrow I).$$

Filtering: Removal of records by strengthening the filter, so that $J' \subseteq J$. This corresponds to moving potentially multiple steps down in $\mathcal{B}^M$.

Flushing: Addition of records by weakening (reversing, flushing) the filter, so that $J' \supseteq J$. Corresponds to moving potentially multiple steps up in $\mathcal{B}^M$.

Repeated chaining operations thus map to trajectories in $\mathcal{B}$. Consider the very small example shown in Fig. 1 for $N = M = 2$ with dimensions $\mathcal{X} = \{X, Y\}$ and two N-dimensional data vectors $\boldsymbol{a}, \boldsymbol{b} \in X \times Y$, and denote e.g. $X/\boldsymbol{ab} = \{\boldsymbol{a} \downarrow \{X\}, \boldsymbol{b} \downarrow \{X\}\}$. The left side of Fig. 1 shows the separate projector and selector lattices (bottom nodes $\emptyset$ not shown), with extension as a transition to a higher rank in the lattice and projection as a downward transition. Similarly, filtering and flushing are the corresponding operations in the filter lattice. The view lattice is shown on the right, along with a particular chain operation $\mathcal{D}_{\{X,Y\},\{\boldsymbol{a}\}} \mapsto \mathcal{D}_{\{Y\},\{\boldsymbol{a}\}}$, which projects the subset of records $\{\boldsymbol{a}\}$ from the two-dimensional view $\{X, Y\} = \mathcal{X}$ to the one-dimensional view $\{X\} \subseteq \mathcal{X}$.

2.3 Relational Expressions and Background Filtering

Note that usually $M \gg N$, so that there are far more records than dimensions (in our example, $M = 74 > 3 = N$). In principle, filters J defining

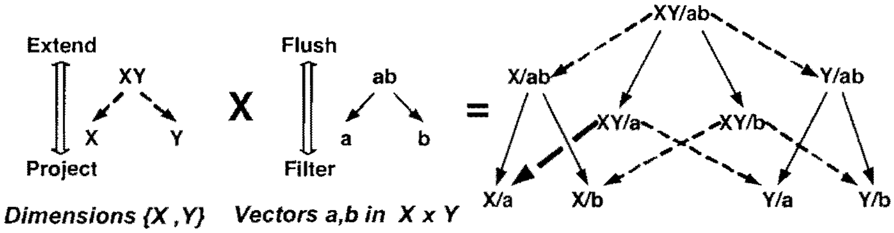


Fig. 1. The lattice theoretical view of data views. (Left) The projector and filter lattices $\mathcal{B}^N, \mathcal{B}^M$ (global lower bounds $\emptyset$ not shown). (Right) The view lattice $\mathcal{B}$ as their product. The projection chain operation $\mathcal{D}_{\{X,Y\},\{a\}} \mapsto \mathcal{D}_{\{X\},\{a\}}$ is shown as a bold link.

which records to include in a view can be specified arbitrarily, for example through any SQL or MDX "`where`" clause, or through OLAP operations like "`top` n", including the n records with the highest value of some feature. In practice, filters are specified as relational expressions in terms of the dimensional values, as expressed in MDX `where` clauses. In our example, we might say `where X` $= \alpha$ `and` (`Z` $<=$ `y and Z` $>=$ `x`), using lexicographic order on the Z variable to determine a filter J specifying just those 20 out of the total possible 74 records. For notational purposes, we will therefore sometimes use these relational expressions to indicate the corresponding filters.

Note that each relational filter expression references a certain set of variables, in this case X and Z, denoted as $R \subseteq \mathbb{N}_N$. Compared to our projector I, R naturally divides into two groups of variables:

Foreground: Those variables in $R^f := R \cap I$ which appear in both the filter expression and are included in the current projection.

Background: Those variables in $R^b := R \setminus I$ which appear only in the filter expression, but are not part of the current projection.

The portions of filter expressions involving foreground variables restrict the rows and columns displayed in the OLAP tool. Filtering expressions can have many sources, such as `Show Only` or `Hide`. It is common to select a collection of siblings within a particular sub-branch of the $\mathcal{P}^i$ tree. For example for a spatial dimension $\mathcal{P}$, the user within the ProClarity tool might select "`All -> USA -> California`", or its children "`California ->` Cities", all siblings. But those portions of filter expressions involving background variables do not change which rows or columns are displayed, but only serve to reduce the values shown in cells. In ProClarity, these are shown in the Background pane.

2.4 Example

Table 2 shows the results of four chaining operations from our original example in Table 1, including a projection $I = \{1,2,3\} \mapsto I' = \{1,2\}$, a filter using relational expressions, and a filter using a non-relational expression. The bottom right shows a hybrid result of applying both the projector $I' = \{1,2\}$ and the relational filter expression `where X` $= \alpha$ `and` (`Z` $<=$ `y and Z` $>=$ `x`). Compare this to the top left, where there is only a quantitative restriction for the same dimensionality because of the use of a background filter. Here $I = \{X, Y\}, R = \{X, Z\}, R^f = \{X\}, R^b = \{Z\}, M' = 20$.

X	Y	$c[I'](\boldsymbol{x})$	$f[I'](\boldsymbol{x})$
α	a	11	0.150
	b	24	0.325
	c	15	0.203
β	a	1	0.014
	b	10	0.136
	c	13	0.176

X	Y	Z	$c^{J'}(\boldsymbol{x})$	$f^{J'}(\boldsymbol{x})$
α	a	x	1	0.050
	b	x	9	0.450
	c	x	2	0.100
		y	8	0.400

X	Y	Z	$c^{J'}(\boldsymbol{x})$	$f^{J'}(\boldsymbol{x})$
α	b	w	15	0.333
		x	9	0.200
	c	y	8	0.178
	a	w	7	0.156
β	c	x	6	0.133

X	Y	$c^{J'}[I'](\boldsymbol{x})$	$f^{J'}[I'](\boldsymbol{x})$
α	a	1	0.050
	b	9	0.450
	c	10	0.500

Table 2. Results from chaining operations $\mathcal{D}_{\mathbb{N}_N, \mathbb{N}_M} \mapsto \mathcal{D}_{I', J'}$ from our original data cube in Table 1. (Top Left) Projection: $I' = \{1, 2\}, M' = M = 74$. (Top Right) Filter: $J' =$ `where X` $= \alpha$ **and** (`Z` $<=$ `y` **and** `Z` $>=$ `x`). $M' = 20$. (Bottom Left) Filter: J' determined from top 5 most frequent entries, $M' = 45$. (Bottom Right) $I' = \{1, 2\}$ and J' determinued by the relational expression `where X` $= \alpha$ **and** (`Z` $<=$ `y` **and** `Z` $>=$ `x`).

3 Information Measures on Conditional Views

For this section, we will generally consider the filter J to be fixed, and supress the superscript on f, unless otherwise needed.

3.1 Conditional Views

We have seen that the frequencies $f : \mathbf{X} \to [0, 1]$ represent joint probabilities $f(\boldsymbol{x}) = f(x_{k^1}, x_{k^2}, \ldots, x_{k^N})$, so that from (2) and (5), $f[I](\boldsymbol{x} \downarrow I)$ expresses the I-way marginal over a joint probability distribution f. Now consider two projectors $I_1, I_2 \subseteq \mathbb{N}_N$, so that we can define a conditional frequency $f[I_1 | I_2] : \mathbf{X} \downarrow I_1 \cup I_2 \to [0, 1]$ where $f[I_1 | I_2] := \frac{f[I_1 \cup I_2]}{f[I_2]}$. For individual vectors, we have

$$f[I_1 | I_2](\boldsymbol{x}) = f[I_1 | I_2](\boldsymbol{x} \downarrow I_1 \cup I_2) := \frac{f[I_1 \cup I_2](\boldsymbol{x} \downarrow I_1 \cup I_2)}{f[I_2](\boldsymbol{x} \downarrow I_2)}.$$

$f[I_1 | I_2](\boldsymbol{x})$ is the probability of the vector $\boldsymbol{x} \downarrow I_1 \cup I_2$ restricted to the $I_1 \cup I_2$ dimensions given that we know we can only choose vectors whose restriction to I_2 is $\boldsymbol{x} \downarrow I_2$. We note that $f[I_1 | \emptyset](\boldsymbol{x}) = f[I_1](\boldsymbol{x}), f[\emptyset | I_2] \equiv 1$, and since $f[I_1 | I_2] = f[I_1 \setminus I_2 | I_2]$, in general we can assume that I_1 and I_2 are disjoint.

We can now extend our concept of a view to a **conditional view** $\mathcal{D}_{I_1 | I_2, J}$ as a view on $\mathcal{D}_{I_1 \cup I_2, J}$, which is further equipped with the conditional frequency $f^J[I_1 | I_2]$. Conditional views $\mathcal{D}_{I_1 | I_2, J}$ live in a different combinatorial structure than the view lattice $\mathcal{B}$. To describe $I_1 | I_2$ and J in a conditional view, we need three sets $I_1, I_2 \in \mathbb{N}_N$ and $J \in \mathbb{N}_M$ with I_1 and I_2 disjoint. So define $\mathcal{A} := 3^{[N]} \times 2^M$ where $3^{[N]}$ is a poset with the following structure:

- $N+1$ levels numbered from the bottom $0, 1, \ldots N$.
- The i^{th} level contains all partitions of each of the sets in $\binom{[N]}{i}$, that is the i-element subsets of $\mathbb{N}_N$, into two parts where
  1. The order of the parts is significant, so that $[\{1,3\},\{4\}]$ and $[\{4\},\{1,3\}]$ of $\{1,3,4\}$ are not equivalent.
  2. The empty set is an allowed member of a partition, so $[\{1,3,4\},\emptyset]$ is in the third level of $3^{[N]}$ for $N \geq 4$.
- We write the two sets without set brackets and with a $|$ separating them.
- The partial order is given by an extended subset relation: if $I_1 \subseteq I_{1'}$ and $I_2 \subseteq I_{2'}$, then $I_1|I_2 \prec I_{1'}|I_{2'}$, e.g. $1\ 2|3 \prec 1\ 2\ 4|3$.

An element in the poset $3^{[N]}$ corresponds to an $I_1|I_2$ by letting I_1 (resp. I_2) be the elements to the left (resp. right) of the $|$. We call this poset $3^{[N]}$ because it's size is 3^N and it really corresponds to partitioning $\mathbb{N}_N$ into three disjoint sets, the first being I_1, the second being I_2 and the third being $\mathbb{N}_N \setminus (I_1 \cup I_2)$. The structure $3^{[2]}$ is shown in Fig. 2.

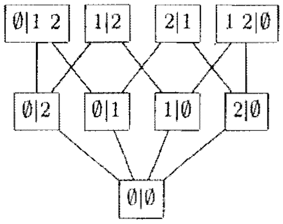


Fig. 2. The structure $3^{[2]}$.

3.2 Information Measures

So given a view $\mathcal{D}_{I,J} \in \mathcal{B}$ which we identify with its frequency $f^J[I]$, or a conditional view $\mathcal{D}_{I_1|I_2;J} \in \mathcal{A}$ which we identify with its conditional frequency $f^J[I_1|I_2]$, we are interested in measuring how "interesting", "unusual", or how much information content is present. Such measures can then be used for combinatorial search and optimization over the view structures $\mathcal{B}, \mathcal{A}$. We use some of the standard, and some not so widely used, measures from information theory.

First, for an unconditional view $\mathcal{D}_{I,J}$, we can define the entropy measure (no longer suppressing J)

$$H(f^J[I]) := -\sum_{x \in \mathbf{X}\downarrow I} f^J[I](x) \log(f^J[I](x)).$$

Maximum entropy corresponds to a "flat" uniform distribution, so other things being equal, users have an interest in views with lower H. Given a conditional view $\mathcal{D}_{I_1|I_2,J}$, we also have the **conditional entropy**, $H(f^J[I_1|I_2])$, where

$$H(f^J[I_1|I_2]) := H(f^J[I_1 \cup I_2]) - H(f^J[I_2]).$$

Given two views $\mathcal{D}_{I,J}, \mathcal{D}_{I,J'}$ of the same dimensionality I, but with different filters J and J', we have the **relative entropy** (directed divergence, or Kullback-Leibler divergence)

$$D(f^J[I] \| f^{J'}[I]) := \sum_{\boldsymbol{x} \in \mathbf{X}\downarrow I} f^J[I](\boldsymbol{x}) \log\left(\frac{f^J[I](\boldsymbol{x})}{f^{J'}[I](\boldsymbol{x})}\right),$$

and the **Hellinger distance**

$$G(f^J[I], f^{J'}[I]) := \sqrt{\sum_{\boldsymbol{x} \in \mathbf{X}\downarrow I} \left(\sqrt{f^J[I](\boldsymbol{x})} - \sqrt{f^{J'}[I](\boldsymbol{x})}\right)^2}.$$

We prefer G to D, since it is symmetric, less sensitive to zeros in the distribution, and it can be shown that for distributions that are bounded away from zero and close to one another, D is equal to first order to the square to one quarter of the square of G [17].

4 Hop-Chaining View Discovery

Given our basic formalism on data views, conditional views, and information measures on them, a variety of possible user-guided navigational tasks become possible. For example, above we discussed Cariou *et al.* [6], who develop methods for discovering "drill-down paths" in data cubes. We can describe this as creating a series of views with projectors $I_1 \supseteq I_2 \supseteq I_3$ of increasingly specified dimensional structure.

Our approach is motivated by the idea that most users will be challenged by complex views of high dimensionality, while still needing to explore many possible data interactions. We are thus interested in restricting our users to two-dimensional views only, producing a sequence of projectors I_1, I_2, I_3 where $|I_k| = 2$ and $|I_k \cap I_{k+1}| = 1$, thus affecting a permutation of the variables X^i.

4.1 Preliminaries

We assume a fixed but arbitrary permutation of the $i \in \mathbb{N}_N$ so that we can refer to the dimensions $X^1, X^2, \ldots, X^N$ in order. The choice of the initial variables X^1, X^2 is a free parameter to the method, acting as a kind of "seed".

One thing that is critical to note is the following. Consider a view $\mathcal{D}_{I,J}$ which is then filtered to include only records for a particular member $x_0^{i_0} \in X^{i_0}$ of a particular dimension $X^{i_0} \in \mathcal{X}$; in other words, let J' be determined by the relational expression $\texttt{where } \mathtt{X}^{i_0} = \mathtt{x}_0^{i_0}$. Then in the new view $\mathcal{D}'_{I,J'}$, $f^{J'}[I]$ is positive only on the fibers of the tensor $\mathbf{X}$ where $X^{i_0} = x_0^{i_0}$, and zero elsewhere. Thus the variable X^{i_0} is effectively removed from the dimensionality of $\mathcal{D}'$, or rather, it is removed from the *support* of $\mathcal{D}'$.

Notationally, we can say $\mathcal{D}_{I,X^{i_0}=x_0^{i_0}} = \mathcal{D}_{I\setminus\{i_0\},X^{i_0}=x_0^{i_0}}$. Under the normal convention that $0 \cdot \log(0) = 0$, our information measures H, D, and G above are insensitive to the addition of zeros in the distribution. This allows us to compare the view $\mathcal{D}_{I.X^{i_0}=x_0^{i_0}}$ to any other view of dimensionality $I \setminus \{i_0\}$.

This is illustrated in Table 3 through our continuing example, now with the filter `where` Y = b. Although formally still a cube $X \times Y \times Z$, in fact this view lives in the $X \times Z$ plane, and so can be compared to the $X \times Z$ marginal.

X	Y	Z	$c(x)$	$f(x)$
α	b	x	9	0.265
		w	15	0.441
β		y	4	0.118
		z	3	0.088
		w	3	0.088

Table 3. Our example data tensor from Table 1 under the filter `where` Y = b; $M' = 34$.

Finally, some caution is necessary when the relative entropy $H(f^J[I_1|I_2])$ is calculated from data, as the magnitude of the relative entropy between empirical distributions is strongly influenced by small sample sizes. To counter such spurious effects, we supplement each calculated entropy with the probability that under the null distribution that the row has the same distribution as the marginal, of observing an empirical entropy larger or equal to actual value. When that probability is large, say greater than 5%, then we consider consider its value spurious and set it to zero before proceeding with the algorithm.

4.2 Method

We can now state the hop-chaining methodology.

1. Set the initial filter to $J = \mathbb{N}_M$. Set the initial projector $I = \{1, 2\}$, determining the initial view $f^J[I]$ as just the initial $X^1 \times X^2$ grid.
2. For each row $x_{k^1} \in X^1$, we have the distribution $f^{X^1=x_{k^1}}[I]$ of that row, and also the marginal $f^J[I \setminus \{X^1\}]$ over all the rows. In light of the discussion above, we can calculate all the Hellinger distances between each of the rows and this row marginal:
$$G(f^{X^1=x_{k^1}}[I], f^J[I \setminus \{X^1\}]) = G(f^{X^1=x_{k^1}}[I \setminus \{X^1\}], f^J[I \setminus \{X^1\}]),$$
and retain the maximum value $G^1 := \max_{x_{k^1} \in X^1} G(f^{X^1=x_{k^1}}[I], f^J[I \setminus \{X^1\}])$. We can dually do so for columns against the column marginal:
$$G(f^{X^2=x_{k^2}}[I], f^J[I \setminus \{X^2\}]) = G(f^{X^2=x_{k^2}}[I \setminus \{X^2\}], f^J[I \setminus \{X^2\}]),$$
retaining the maximum value $G^2 := \max_{x_{k^2} \in X^2} G(f^{X^2=x_{k^2}}[I], f^J[I \setminus \{X^2\}])$.
3. The user is prompted to select either a row $x_0^1 \in X^1$ or a column $x_0^2 \in X^2$. Since G^1 (resp. G^2) represents the row (column) with the largest distance from its marginal, perhaps selecting the global maximum $\max(G^1, G^2)$ is most appropriate; or this can be selected automatically. Letting x'_0 be the selected value from the selected variable (row or column) $i' \in I$, then J' is set to `where` $\mathtt{X}^{i'} = \mathtt{x}'_0$, and this is placed in the *background* filter.

4. Let $i'' \in I$ be the variable *not* selected by the user, so that $I = \{i', i''\}$.
5. For each dimension $i''' \in \mathbb{N}_N \setminus I$, that is, for each dimension which is neither in the background filter $R^b = \{i'\}$ nor retained in the view through the projector $\{i''\}$, calculate the conditional entropy of the retained view $f^{J'}[\{i''\}]$ against that variable: $H(f^{J'}[\{i''\}|\{i'''\}])$.
6. The user is prompted to select a new variable $i''' \in \mathbb{N}_N \setminus I$ to add to the projector $\{i''\}$. Since $\underset{i''' \in \mathbb{N}_N \setminus I}{\operatorname{argmin}} H(f^{J'}[\{i''\}|\{i'''\}])$ represents the variable with the most constraint against i'', that may be the most appropriate selection, or it can be selected automatically.
7. Let $I' = \{i'', i'''\}$. Note that I' is a sibling to I in $\mathcal{B}^N$, thus the name "hop-chaining".
8. Let I', J' be the new I, J and go to step 2.

Keeping in mind the arbitrary permutation of the X^i, then the repeated result of applying this method is a sequence of hop-chaining steps in the view lattice, building up an increasing background filter:

1. $I = \{1, 2\}, J = \mathbb{N}_M$
2. $I' = \{2, 3\}, J' =$ `where` $\mathtt{X}^1 = \mathtt{x}_0^1$
3. $I'' = \{3, 4\}, J'' =$ `where` $\mathtt{X}^1 = \mathtt{x}_0^1, \mathtt{X}^2 = \mathtt{x}_0^2$
4. $I''' = \{4, 5\}, J''' =$ `where` $\mathtt{X}^1 = \mathtt{x}_0^1, \mathtt{X}^2 = \mathtt{x}_0^2, \mathtt{X}^3 = \mathtt{x}_0^3$

4.3 Extension to Hierarchical Data Cubes

In Sec. 2.1 we introduced OLAP databases as data tensors with a hierarchical structure, but in Sec. 2.2 we developed view discovery for non-hierarchical tensors. We return now to consider view discovery in general for hierarchical OLAP databases, and how we accommodate hierarchical structure in hop-chaining.

In any given two-dimensional OLAP view on say the projector $I = \{1, 2\}$, the entries actually correspond not to slots $\boldsymbol{x} \in \mathbf{X}$, but to cells $\boldsymbol{p} \in \mathbf{P}$; and the rows and columns not to collections of data items $Y^i \subseteq X^i$, but of members $Q^i \subseteq P^i$. Q^i is then reflected in the (foreground) filter J. In fact, these can be arbitrary, drawing from different levels, perhaps showing `California` on one row, and `Detroit` on another, even within a `Country -> State -> City` hierarchy. The only restriction is that you cannot have two $p_1^i, p_2^i \in P^i$ with $p_1^i \leq^i p_2^i$, for example showing both `California` and `Los Angeles`. Mathematically, this forces our selection Q^i to determine an **antichain** of $\mathcal{P}^i$.

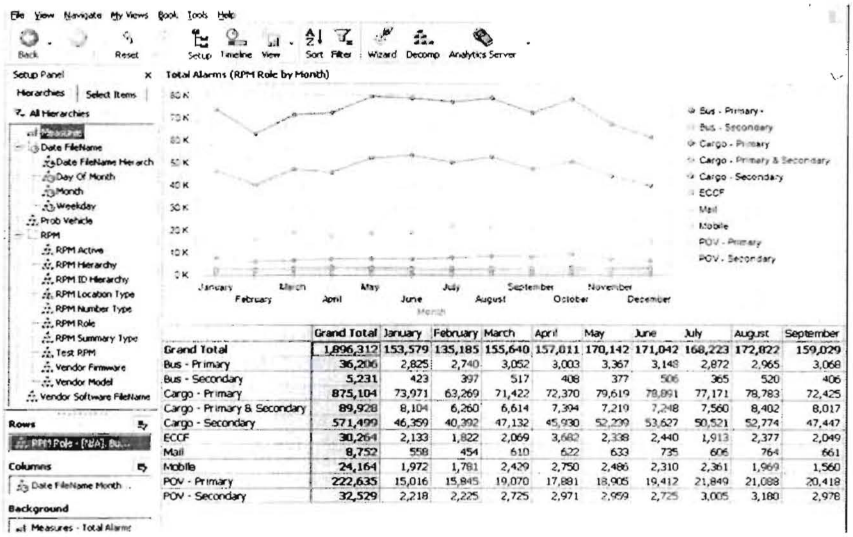
"Drilldown" and "rollup" are some of the primary operations available in OLAP. If $X^1 =$ "Location", and $p_0^1 =$ "California", then classical drilldown might take a row like `California` from a view, restrict J with the relational expression `where Location` $=$ `California`, and then replace Q^1 with all the children of p_0^1, so that $Q'^1 = \{p^1 \leq^1 p_0^1\}$.

We are experimenting with view discovery and hop-chaining formalisms which operate over these member collections Q^i, and in general over their Cartesian products $\bigtimes_{i \in I} Q^i \subseteq \mathbf{P} \downarrow I$. But in the current formulation, it is sufficient to consider each dimension X^i involved in a foreground view to be drilled-down to the immediate children of the top of $\mathcal{P}^i$, that is, the children of `All`.

5 Implementation

We have implemented the hop-chaining methodology in a prototype form for experimentation and proof-of-principle. ProClarity 6.2 is used in conjunction with SQL Server Analysis Services (SSAS) 2005 and the R statistical platform v. 2.7[9]. ProClarity provides a flexible and friendly GUI environment with extensive API support which we use to gather current display contents and query context for row, column and background filter selections. R is currently used in either batch or interactive mode for statistical analysis and development. Microsoft Visual Studio .Net 2005 is used to develop plug-ins to ProClarity to pass ProClarity views to R for hop-chain calculations.

A first view of the data set used for demonstrating this method is shown in Fig. 3, a screenshot from the ProClarity tool. The database is a collection of 1.9M records of events of personal vehicles, cargo vehicles, and others passing through radiation portal monitors (RPMs) at US ports of entry. The 15 available dimensions are shown on the left of the screen (e.g. "day of the month", "RPM hierarchy"), tracking such things as the identities and characteristics of particular RPMs, time information about events, and information about the hardware, firmware, and software used at different RPMs.



Rows
RPM Role - [N/A], Bu...

Columns
Date FileName Month ..

Background
Measures - Total Alarms

	Grand Total	January	February	March	April	May	June	July	August	September
Grand Total	1,896,312	153,579	135,185	155,640	157,011	170,142	171,042	168,223	172,822	159,029
Bus - Primary	36,206	2,825	2,740	3,052	3,003	3,367	3,148	2,872	2,965	3,068
Bus - Secondary	5,231	423	397	517	408	377	506	365	520	406
Cargo - Primary	875,104	73,971	63,269	71,422	72,370	79,619	78,891	77,171	78,783	72,425
Cargo - Primary & Secondary	89,928	8,104	6,260	6,614	7,394	7,219	7,248	7,560	8,402	8,017
Cargo - Secondary	571,499	46,359	40,392	47,132	45,930	52,239	53,627	50,521	52,774	47,447
ECCF	30,264	2,133	1,822	2,069	3,682	2,338	2,440	1,913	2,377	2,049
Mail	8,752	558	454	610	622	633	735	606	764	661
Mobile	24,164	1,972	1,781	2,429	2,750	2,486	2,310	2,361	1,969	1,560
POV - Primary	222,635	15,016	15,845	19,070	17,881	18,905	19,412	21,849	21,088	20,418
POV - Secondary	32,529	2,218	2,225	2,725	2,971	2,959	2,725	3,005	3,180	2,978

Fig. 3. Initial 2-D view of the alarm summary data cube, showing count distribution of RPM Role by months.

[9] http://www.r-project.org

6 Examples

Space limitations will allow showing only a single step for the hop-chaining procedure against the alarm summary data cube.

Fig. 3 shows the two-dimensional projection of the X^1 = "RPM Role" $\times X^2$ = "Month" dimensions within the 15-dimensional overall cube, drilled down to the first level of the hierarchies (see Sec. 4.3). Its plot shows the distributions of count c of alarms by RPM role (Busses Primary, Cargo Secondary, etc.) X^1, while Fig. 4 shows the distribution by Month X^2.

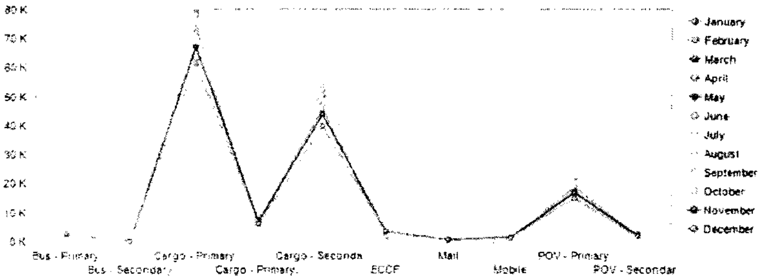


Fig. 4. Count distribution of months.

The distributions for roles seem to vary at most by overall magnitude, rather than shape, while the distributions for months appear almost identical. However, Fig. 5 and Fig. 6 show the same distributions, but now in terms of their frequencies f relative to their corresponding marginals, allowing us to compare the shapes of the distributions normalized by their absolute sizes. While the months still seem identical, the RPM roles are clearly different, although it is difficult to see which is most unusual with respect to the marginal (bold line).

The left side of Fig. 7 shows the Hellinger distance values $G(f^{X^i=x_{k^i}}[I], f^J[I \setminus \{X^i\}])$ for $i \in \{1, 2\}$, that is the Hellinger distance for each row or column against its marginal. The RPM roles "ECCF" and "Mail" are clearly the most significant, which can be verified by examining the anomolously shaped plots in Fig. 5. The most significant month is December, although this is hardly evident in Fig. 6. We select the maximal row-wise Hellinger value $G^1 = .011$ for ECCF, so that $i' = 1, x_0^1 =$ ECCF. $X^{i'} = X^1 =$ "RPM Role" is added to the backgound filter, $X^{i''} = X^2 =$ Months is retained in the view, and we calculate $H(f^{J'}[\{2\}|\{i'''\}])$ for all $i''' \in \{3, 4, \ldots, 15\}$, which are shown on the right of Fig. 7 for all significant dimensions. On that basis X^3 is selected as Day of Month with minimal $H = 3.22$.

The final view for X^2 = Months $\times X^3$ = Day of Month is then shown in Fig. 8. Note the strikingly divergent plot for April: it in fact does have the highest Hellinger distance at .07.

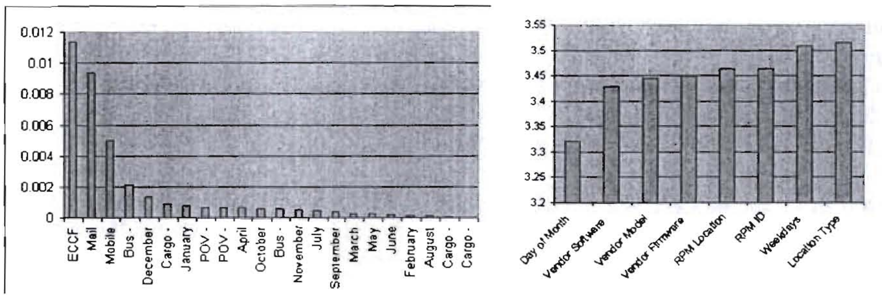
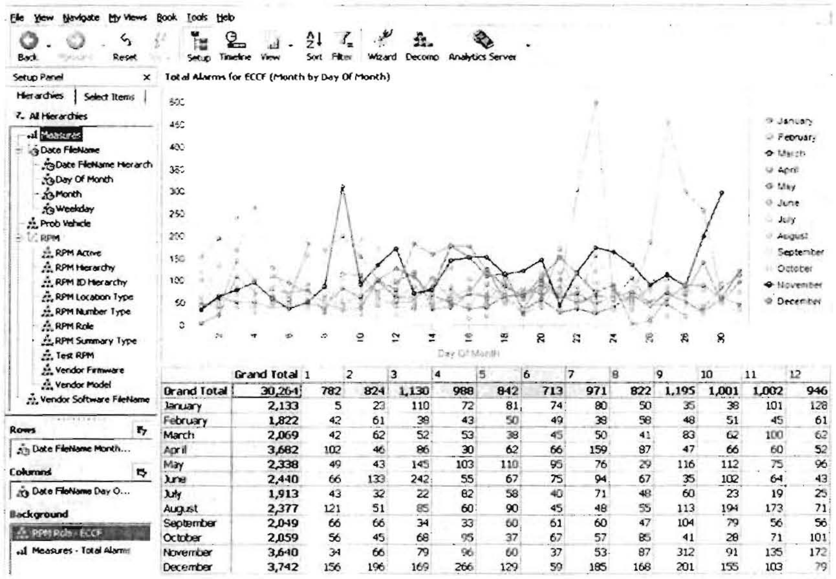


Fig. 7. (Left) Hellinger distances of rows and columns against their marginals. (Right) Relative entropy of months against each other significant dimension, given that RPM Role = ECCF.



	Grand Total	1	2	3	4	5	6	7	8	9	10	11	12
Grand Total	30,264	782	824	1,130	988	842	713	971	822	1,195	1,001	1,002	946
January	2,133	5	23	110	72	81	74	80	50	35	38	101	128
February	1,822	42	61	38	43	50	49	38	58	48	51	45	61
March	2,069	42	62	52	53	38	45	50	41	83	62	100	62
April	3,682	102	46	86	30	62	66	159	87	47	66	60	52
May	2,338	49	43	145	103	110	95	76	29	116	112	75	96
June	2,440	66	133	242	55	67	75	94	67	35	102	64	43
July	1,913	43	32	22	82	58	40	71	48	60	23	19	25
August	2,377	121	51	85	60	90	45	48	55	113	194	173	71
September	2,049	66	66	34	33	60	61	60	47	104	79	56	56
October	2,059	56	45	68	95	37	67	57	85	41	28	71	101
November	3,640	34	66	79	96	60	37	53	87	312	91	135	172
December	3,742	156	196	169	266	129	59	185	168	201	155	103	79

Fig. 8. Subsequent 2-D view of the alarm summary data cube.

7 Discussion, Analysis, and Future Work

In this paper, we have provided the fundemantals necessary to express view discovery in OLAP databases as a combinatorial search and optimization operation in general, aside from the specific hop-chaining method. What remains to be addressed is a precise formal expression of this optimization problem. This is dependent on the mathematical properties of our information measures H, D, and G over the lattices $\mathcal{B}, \mathcal{A}$. It is well known, for example, that H is a monotonic function in $\mathcal{B}$, in that $\forall I_1 \subseteq I_2, H(f^J[I_1]) \geq H(f^J[I_2])$. There should be ample literature (e.g. [29]) to fully explicate the behavior of these functions on these structures, and move on the properties of search algorithms.

Also as mentioned above, we are restricting our attention to OLPA cubes with a single "count" measure. Frequency distributions are available from other quantitative measures, and exploring the behavior of these algorithms in those contexts is of interest.

Finally, software implementations provide a tremendous value in performing this research, not only for practical validation by sponsors and users, but also for assisting with the methodological development itself. As our software platform matures, we look forward to incorporating other algorithms for view discovery [6, 15, 20, 23–26], for purposes of comparison and completeness.

8 Acknowledgements

We are grateful for the support of many colleagues, including Judith Cohn at the Los Alamos National Laboratory; David Gillen, Eric Stephan, Bryan Olsen, and Elena Peterson at the Pacific Northwest National Laboratory; Matt Damante and Robert Fernandez at the Lawrence Livermore National Laboratory; and our sponsors and colleagues at DHS/S&T, DHS/ICE, and DHS/CBP.

References

1. S Agarwal, R Agrawal, PM Deshpande, G Ashish, J Naughton, R Ramakrishnan, S Sarawagi: (1996) "On the Computation of Multidimensonal Aggregates", in: *Proc. 22nd VLDB Conference*, Bombay
2. Agrawal, Rakesh; Gupta, Ashish; and Sarawagi, Sunita: (1997) "Modeling Multidimensional Databases", *Proc. 13th Int. Conf. on Data Engineering*
3. Asimov, Daniel: (1985) "The Grand Tour: A Tool for Viewing Multidimensional Data", *SIAM J. Statistical Computing*, v. **6**:1
4. B Barak, K Chaudhuri, C Dwork, S Kale, F McSherry, K Talwar: (2007) "Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release", in: *Proc. 2007 Conf. Principles of Database Systems (PODS 07)*
5. Borgelt, Christian and Kruse, Rudolf: (2002) *Graphical Models*, Wiley, New York
6. V Cariou; J Cubille; C Derquenne; S Goutier, F Guisnel, H Klajnmic: (2008) "Built-In Indicators to Discover Interesting Drill Paths in a Cube", in: *DaWaK 2008, LNCS*, v. **5182**, ed. IY Song et al., pp. 33-44, Springer-Verlag, Berlin

29. Studeny, Milan: (2005) *Probabilistic Conditional Independence Structures*, Springer-Verlag, London

30. Thomas, Helen and Datta, Anindya: (2001) "A Conceptual Model and Algebra for On-Line Analytical Processing in Decision Support Databases", *Information Systems Research*, v. **12**:1, pp. 83-102