Paper Reviewed (1)

- Chris Stolte, Diane Tang, Pat Hanrahan
  "Query, Analysis, and Visualization of Hierarchically Structured Data Using Polaris"

Overview

- Hierarchical Structure of Data
- Relational Databases VS. Data Cubes
- Nest Operand VS. Dot Operand
- New Interface in support of data cube
- Critiques

Hierarchical Structure of Data

- How to derive the Hierarchical Structure of Data
  - Known hierarchical structure (country, province, city)
  - Using data mining algorithm (decision trees, clustering technique)
- Benefit of hierarchical structure over relational structure
  - Flexible and efficient in obtaining data summaries of different aspects of data during data exploration process.
  - Support "semantic zooming" visualization
- Realization of organizing data into hierarchical structure
  - Concept of Data Cube

Relational Database VS Data Cubes

- Aspects of data dimensions
  - Relational Database: Dimensions are independent
  - Data Cube: Dimensions can be hierarchically dependent
- Aspect of data summary
  - Relational Database: Use SQL queries to retrieve
  - Data Cube: Aggregated values (summation, average, etc.) are readily stored in the cells of data cube

Critiques

- Pros
  - Provides interfaces for non-expert to retrieve data that involve complex data query algebra
  - Construct a robust formalism for presenting data cubes, which help reveal many aspects of data summary (different abstraction level of data and different detailed level of data)
  - Can also be an visualization tool for understanding the data mining model, which configure the hierarchical data structure.
- Cons
  - Did not use intuitive navigation techniques to facilitate changing views of data
  - Systems designed heavily focus on presenting summary of data. Could lead users only concentrate on this part of data analysis

New Interface in support of data cube

- Display dimensions hierarchies for more quickly configuring the table (determine the number of panes)
  - On the schema
  - On the "shelves" of table
- Distinguish between "Node" and "Path"
  - Example: When selecting dimension "Month" from schema, Default is Year.Month. But can change to "Month" or "Year.Month" or "Quarter.Month",
- Change level of detail within panes to reflect the change of dimension hierarchy (will change number of marks within panes as well)

Nest Operand VS. Dot Operand

- Nest operand (no hierarchy implication)
  - Relational databases: SELECT A, D, sum (b) FROM table GROUP BY A, D
- Dot operand (hierarchy implication)
  - Data Cubes: The datasets do not have any data of October. So after nesting, we do not see Oct nested under Qtr4
  - Data Cubes: Semantically, Quarter and Month have hierarchy implications. So after doting, Oct is still displayed under Qtr4 even that there is no corresponding data

Critiques

- Pros
  - Provides interfaces for non-expert to retrieve data that involve complex data query algebra
  - Construct a robust formalism for presenting data cubes, which help reveal many aspects of data summary (different abstraction level of data and different detailed level of data)
  - Can also be an visualization tool for understanding the data mining model, which configure the hierarchical data structure.
- Cons
  - Did not use intuitive navigation techniques to facilitate changing views of data
  - Systems designed heavily focus on presenting summary of data. Could lead users only concentrate on this part of data analysis

Paper Reviewed (2)

- Chris Stolte, Diane Tang, Pat Hanrahan
  "Multi-scale visualization using data cubes"
**Overviews**

- **Features Supported**
  - Data abstraction and visual abstraction
  - Allow independently zooming along one or more dimensions
- **Formalism guiding the Multi-scale visualization**
  - Zoom graph
  - Polaris specification
- **Proved effective design pattern**

**Critique**

**Paper Reviewed (3)**

- **Mihael Ankerst, David H. Jones, Anne Kao, Changzhou Wang**

"DataJewel: Tightly Integrating Visualization with Temporal Data Mining"

**Overviews**

- **Temporal Databases**
- **Information Tasks of Temporal Data Mining**
- **Non-expert integrated Solutions - DataJewel**
- **Aircraft Maintenance Data Scenario**

**Critiques**

- **Pros**
  - Support normal zooming and semantic zooming (make use of the "structured" nature of data) on databases visualization
  - Try to formalize the relationship between zooming and data semantics. Not just treat zooming as a HCI technique
- **Cons**
  - The generality of proposed formalism for zooming has not been proved (currently applicable to 4 design patterns)
  - Did not address Focus+Context or retaining original visualization for referencing after zooming

**Effective Design Pattern**

- **Thematic map**
- **Chart stack**
- **Scatter plot**
- **Matrices**

**Temporal Databases**

<table>
<thead>
<tr>
<th>Date</th>
<th>Aircraft</th>
<th>Model</th>
<th>Problem ID</th>
<th>System Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Dec</td>
<td>Air Canada</td>
<td>737</td>
<td>Engine Fail</td>
<td>Engine Fuel</td>
</tr>
<tr>
<td>1-Dec</td>
<td>Air Canada</td>
<td>737</td>
<td>Communication</td>
<td>Communication</td>
</tr>
<tr>
<td>1-Dec</td>
<td>United Airlines</td>
<td>747</td>
<td>Engine Fail</td>
<td>Engine Fuel</td>
</tr>
<tr>
<td>1-Dec</td>
<td>United Airlines</td>
<td>747</td>
<td>Communications</td>
<td>Communication</td>
</tr>
<tr>
<td>1-Dec</td>
<td>United Airlines</td>
<td>737</td>
<td>Engine Fail</td>
<td>Engine Fuel</td>
</tr>
<tr>
<td>1-Dec</td>
<td>United Airlines</td>
<td>737</td>
<td>Communications</td>
<td>Communication</td>
</tr>
</tbody>
</table>
Information Tasks of Temporal Data Mining

Which event has anomaly during the a certain period of time?
Is there any other event that has the similarly abnormal pattern like the already observed event?

Example:
During 1990 to 2000:
Which airplane system has significantly low or high relative frequency of being affected by problems reported?
Which else airplane system has the similar troublesome situation? (within event attribute)
Which model, airline, etc has the similar troublesome situation? (cross event attribute)

Critiques

Pros
- Interaction between data mining and data visualization for efficiently exploring huge databases
- Non data mining experts can mine more meaningful information
- Application specific
- Number of events per attribute <10; number of events per event attribute >200; smallest time unit is day
- Limited tasks
- Limited to find anomalies and correlations
- Limited Data Type

Cons
- Interaction between data mining and data visualization
- Limited to find anomalies and correlations
- Limited Data Type

Aggregated Databases

Original relational tables are compressed by computing the summary statistics: count(), sum(), ... per day: 402
Average # of distinct events per day (by aggregation): 32
Greatly reduce memory capacity requirement!

CalendarView(2)
data of each day is encoded in the calendar day as a histogram where height indicates occurring frequency while colors means different events
Event dates is represented by visual metaphor of a calendar

Visualization Interaction(1)

Select Date Range
Ascending/Descending order

By LongestStreak and then visualization, the high occurrences of engine fuel problem are spotted during the end of July 2000

Aircraft Maintenance Data Scenario (1)
2000
June
July
August
September
By LongestStreak and then visualization, the high occurrences of engine fuel problem are spotted during the end of July 2000

Data Mining algorithm
- LongestStreak:
  - Calculate "relative frequency" of event E of each day
  - Days in which the relative frequency of event E is significantly lower or over the mean value are labeled "significant day"

- "Box" represents primitives of program operations and database operations
- "Arrow" represents the sequencing of the primitives.

- Visualization guided + Domain expert centric
  - Overview of data are first given by visualization
  - Domain expert iteratively takes following actions based on his knowledge and the visualized overview of data
    - Filter data by selecting date range, or
    - Interact with the visualization to explore patterns, or
    - Initiate data mining when spotting suspicious patterns
  - Also can select different visualization techniques in accordance with the data size

Paper Reviewed (4)

Alexander Aiken Jolly Chen Michael Stonebraker Allison Woodruff
"Tioga-2: A Direct Manipulation Database Visualization Environment"

Overview

Intro. Of Tioga-2
User Interface of Tioga-2
Model of Presenting Data of Tioga-2
Details of Presenting Data of Tioga-2
Miscellaneous of Presenting Data of Tioga-2
Critiques

Overview

Tioga-2
- An visual SDK environment for databases applications
- Visual programming:
  - "Box" represents primitives of program operations and database operations
  - "Arrow" represents the sequencing of the primitives.
- Visual feedback:
  - Visual demonstration of results of each programming steps in real time
  - Example:
    - Visually shows the data queried for the SQL instructions.
    - Focus on the latter part—visual feedback....
User Interface of Tioga-2 (1)
- Menu bar for invoking primitive operations
- Windows for visual programming
- “Canvas” for “painting” results of programming

User Interface of Tioga-2 (2)
- Add Table “Station” that has datasets (relations) of weather stations along with their observations
- Filter the datasets to the stations in Louisiana
- Project out un-needed data fields
- Default visual result of the above sequences of databases operations

User Interface of Tioga-2 (3)
- “Box” (or primitive procedure) will generate “output”, which is the “input” of the successor “Box”.
- “Inputs” or “Outputs” of database primitive procedures actually are datasets (relations or tuples). They are referred as “displayable” in the Tioga-2.
- “Displayable” includes:
  - Extended Relations (R)
  - Composite (C)
  - Group (G)

Miscellaneous of presenting data of Tioga-2 (1)
- Slaving Views: Move or delete “slaved” viewers together
- Magnifying Glasses: Overlap viewer of other data on current viewer

Miscellaneous of presenting data of Tioga-2 (2)
- Replicated Viewer
- Stitched View Stitch two viewers

Detail of presenting data of Tioga-2 (1)
- Composite:
  - Data semantic: Union of different relations
  - Visual semantic: Superimposition of “Canvases” (or visualization) of different relations
- Group:
  - Data semantics: Union of different composites
  - Visual semantics: Juxtaposition of visualizations of different composites.
- Elevation:
  - Data semantics: number of tuples shown on the “Canvas”
  - Visual semantics: degree of zooming (the height you watch the image)

Detail of presenting data of Tioga-2 (2-1)
- Drilling down
  - Refined view of the same data
  - Changed view of different but related data
- Rear View Mirror
- Refined view of the same data
  - Set Range: Set range of data that a view can zoom in/out
  - Overlay: Overlay different displays of the same data.
  - Shuffle: Change drawing order of relations within a composite.

Detail of presenting data of Tioga-2 (2-2)
- Changed view of different but related data
  - Wormholes
    - A viewer mentioned previously
    - A viewer onto another canvas, which visualize datasets relating to the data visualized on the current canvas
  - Defined by parameters of size of the viewer, the destination canvas, the elevation (of datasets) from which the canvas is viewed, etc.

Detail of presenting data of Tioga-2 (2-3)
- Rear View Mirrors
  - A mirror to retain the “canvas scenes” before zooming in/out

Model of presenting data of Tioga-2 (1)
- Extended Relations:
  - Relations in data itself + relations on “Canvas”

Model of presenting data of Tioga-2 (2)
- Each tuple of R
- Each display on “Canvas”

Model of presenting data of Tioga-2 (3)
- Location and display attributes of data
  - Location attributes determines how to position tuples on 2D canvas (x axis, y axis, sliders)
  - Display attributes determines how tuples look like on 2D canvas (point, line, rectangle, circle, polygon, text, viewer (viewer on canvas))
- Default location and display of tuples (default visualization)
- Operations for altering visualization
  - Add attribute of data itself along with location or of display
  - Set attribute of location or display
  - Remove attribute of data itself along with location or of display
  - Swap attribute of data itself along with location or of display
  - Scale, Translate attribute of location
  - Combine attribute of display

Critiques
- Pros
  - Pioneered concept of multi-scale visualization of databases
  - Visualization for aiding programming in real time
- Cons
  - Users are still tasked with being required to be familiar with SQL queries and basic programming primitives—not suitable for general public
  - Users are tasked with configuring visualization—non-visualization expert might not feel the advantage of flexibility