DataJewel: Tightly Integrating Visualization with Temporal Data Mining.


What is Data Mining?

- **Data mining**, also known as **knowledge-discovery in databases (KDD)**, is the practice of automatically searching large stores of data for patterns.
- Data mining uses computational techniques from statistics and pattern recognition.

Temporal Data Mining

- Each record has a timestamp
- Databases evolve as a consequence of organizational need
- Linking together two databases with respect to time can give us a powerful tool to explore the union of attributes

User-centric data mining

1. User selects data source/ attributes
2. Data is compressed and loaded
3. Data is visualized
4. User invokes algorithm
5. User interacts with visualization
6. User selects date range
7. User selects visualization technique
8. How data is shown

The Visualization Component

- Calendar View
  - Visual metaphor: Calendar.
  - Structure of data is represented along the event dates is the frequency of events.
  - Designed for domain experts – intuitive and versatile design.
  - If there are few events the visualization is powerful since human’s pre-attentive perception is very efficient in looking for variety of patterns.

The Temporal Mining Component

- Have algorithms that discover patterns
- Determine which events are involved in the patterns
- Automatically select colors based on the patterns
- Visualize not just data but also patterns
- Use of the same color assignment interface by user and algorithm.

The Database component

- Each event is stored in one record
- Data resides in tables in one or more relational databases
- Aggregate database events according to event date (using select count(*) ... group by ...)
- Access the raw data of all attributes

The Temporal Mining Component

- Discover one event of one event attribute
  - For example - highest variance, most interesting trend - give the event a unique color
- Discover multiple events of one event attribute
  - Set of events that together represent a pattern (for example - discovery of similar events) - each event that is part of the pattern receives a distinct color
- Discover one event for each event attribute
  - Look for patterns relating event attributes to each other instead of analyzing them separately. (for example – finding similar events across different event attributes) – update the color assignments of each event attribute accordingly.
Press here for running mining algorithm
Critique (+)

- Combine data mining algorithms with visualization
- Can work with several databases
- Scalable – handles large databases
- Intuitive and easy to use – don’t need a data mining expert

DEVise: Integrated Querying and Visual Exploration of Large Datasets


Critique (-)

- Hard to see patterns over weeks or months or within a single day
- Only one event attribute for each calendar presentation
- Not easily transferable to other domains like author claims.
- Only for categorical attributes
- Does not handle other types of databases other than relational
- No user studies
What is DEVise?
- A data exploration system that allows users to develop, browse, and share visual representations of datasets from several sources.
- A framework which describes a set of querying and visualization primitives that is combined to develop a visual presentation.

Basic concepts
- Mapping each source data record to a visual symbol on screen

**TData** (Textual Data) – a collection of records with one or more attributes (along with a schema).

**GData** (Graphical Data) – high level representation of the screen (x, y, size, color, pattern, orientation, shape)

**Mapping** – a function that is applied to the TData record to produce a GData record.

Visualization model

**View** – basic display unit
- TData
- mapping
- Background (title, axes)
- data display
- cursor display – additional data independent information
- visual filter - set of selection (a query) on the GData of a view

**Window** – collection of views

**Visual presentation** – collection of windows

Overall_sales (date, Did, totRev)
Sales (date, itemid, custid, number)

Record link example

**Cursors** – allows the visual filter of one view to be seen as a highlight in another view

**Links** – constraints that allows the contents of two views to be coordinated.
- Visual – associate visual filters of two views
- Record – the projection of the data in one view (on the linked attributes) will act as a filter on the TData of the other view
- Operator
- aggregate
Semantics of a visual display

A mapping function is applied from the \( TData \) record to produce a \( Gdata \) record:

\[
\begin{align*}
< t_1, t_2, \ldots, t_n > &\rightarrow \delta_{t_1} \\
< t_1, t_2, \ldots, t_n > &\rightarrow \delta_{t_2} \\
&\quad \vdots \\
< t_1, t_2, \ldots, t_n > &\rightarrow \delta_{t_n}
\end{align*}
\]

A view can then be represented as: \((B, \sigma^G, \mu, T, C)\)

- \(B\) – Background
- \(\sigma\) – visual filter
- \(\mu\) – mapping
- \(T\) – \(TData\)
- \(C\) – cursor layer

Achievements

- Visual presentation capabilities – users can render their data. Simple mapping between data and presentation
- Ability to handle large distributed databases (not limited to available memory)
- Collaborative data analysis
- Support for interactively exploring the data visually at any level of detail

Visual Queries and SQL

- Visual queries – user selection on visual attributes of a view. (zoom in/out, scroll, point selection)
- Can save and transfer a visual query
- Enables users to generate sophisticated SQL queries through intuitive graphical operations
- Can be used as an SQL front-end (but not only!)

Another Example:

- Input data: has information about deposits into various accounts at 2 different banks:
  - Account (bankNum, SSN, accNum, pic, …)
  - Deposit (accNum, date, amount)
- Problem: We want to analyze the transactions to find out who has a suspiciously large number of transactions within a short period of time.
critique

+ 
  - Very thorough well-defined framework
  - Many examples of implementations in real application

- 
  - Leaves the visualization decisions to the user (but that’s the idea…)
  - Some visualizations are very hard or impossible to do

Questions?