Clustering Large Datasets and Visualizations of Large Hierarchies and Pyramids Symbolic Data Analysis Approach

1 citation
Conference unknown

Overview of some clustering methods:

1) Clustering large datasets of mixed (nonhomogeneous) units – units described by variables measured in different types of scales (numerical, ordinal, nominal). The basic idea of the approach is to describe all units and clusters in a uniform way by frequency distributions of values of their variables.

2) The adapted leaders method. The proposed method for clustering very large datasets is a variant of the dynamic clustering method and can be shortly described with the following procedure:

```
-   determine an initial clustering
-   repeat
    determine leaders of the clusters;
    assign each unit to the nearest leader – producing a new clustering
    until the process stabilizes.
```

3) The agglomerative hierarchical clustering method. After reducing, with the leaders method, the dataset to some hundreds of representatives of the obtained clusters we can produce a hierarchical clustering on them.

Visualization of hierarchies and pyramids: hyperbolic display and flags.

Hyperbolic display allows simultaneous view of complete pyramidal or hierarchical clustering and detailed inspection of selected region inside pyramid or hierarchy. It also represents a 'map' structure over clusters.

Flags are essentially a graphical display of the encoded table where each cell is colored with a color(s) (or level(s) of gray) assigned to the cell value(s). They are based on visual comparison of units (table rows). A more informative display can be obtained if we reorder the units according to some clustering. In the case of large datasets such representation is not manageable any more. But we can reduce its size by replacing some clusters with their representatives. Extended flags combine global view with detailed local view. By interactively drilling into the hierarchy we can expose selected units in their contexts.

A Hierarchy Navigation Framework: Supporting Scalable Interactive Exploration over Large Databases

1 citation
Published in:
Database Engineering and Application Symposium, 2005. IDEAS 2005. 9th International

Developed a tree labeling method, called MinMax tree, that allows the movement of the on-line recursive processing of visual user interactions on hierarchical data sets into an off-line pre-computation step. Using MinMax tree we map the recursive processing at the interface level to two dimensional range queries that can be answered efficiently using spatial indexes.
We also employ caching and prefetching at the client side to cope with the real-time response requirements. The techniques have been incorporated into XmdvTool, a free software package for multi-variate data visualization and exploration. We propose a caching strategy that buffers the recently used data items. The cache also uses a fast look up mechanism using a memory resident spatial index. Moreover, the idle time between user operations can be effectively utilized for predicting and prefetching the data for future user requests.

The main contributions of this paper are:
• A hierarchy encoding technique that maps a tree into a 2D space and visual navigation operations into spatial range queries.
• A framework that exploits the encoding technique and characteristics of the visual navigation environment for efficient retrieval of online data. This includes: 1. A caching strategy to reduce fetch latencies. 2. Index structures that exploit hierarchy encoding for efficient searches on cache and database. 3. A direction based prefetching strategy that exploits properties of visual interactive tools to predict future user requests.

XmdvTool is a visualization tool designed for exploration and analysis of multivariate data sets, offering four distinct yet interlinked visualization techniques (other paper). Here we focus on structure space techniques, i.e., selection based on structural relationships between data points. By structure, we mean that we recursively partition data into related groups and identify suitable summarizations for each cluster.

Implementation:
From MinMax tree to 2D Hierarchy Maps to using the 2-D Hierarchy Maps to implement structure-based brushes, to translating structure-based brushes into SQL.

Dual Multi-resolution HyperSlice for Multivariate Data Visualization

18 Citations
Published in:
Information Visualization '96, Proceedings IEEE Symposium on

We present a new multiresolution visualization design which allows a user to control the physical data resolution as well as the logical display resolution of multivariate data. A system prototype is described which uses the Hyperslice representation. The notion of space projection in multivariate data is introduced. This process is coupled with wavelets to form a powerful tool for very large data visualization.

Hyperslice [3] is designed for visualization of multidimensional scalar functions. The main strength of the Hyperslice representation is its interactive environment in which only part of the data is displayed while the rest can be accessedvia direct manipulation. Two approaches are applied in our design to reduce the size of the data and create a fine to coarse data hierarchy - space projection and wavelet transform. Our system combines these two processes and providcs a dual multiresolution visualization environment to improve the browsing capability as well as the data navigation of the original HyperSlice. Without loss of generality, we describe a wavelet as a filter matrix that accepts a data stream with n items, and generates n/2 items of approximations and n f 2 items of details. Our system also provides the error information generated by the wavelet decompositions.

Multiscale visualization of relational databases using layered zoom trees and partial data cubes.

1 citation

IMAGAPP 2010-Proceedings of the International Conference on Imaging Theory and Applications, IVAPP 2010-Proc. Int. Conf. Information Visualization Theory and Applications. 2010.

Video!!! https://www.youtube.com/watch?v=8dfIke95xCM

Me like this☺

Zoom trees to represent the entire history of a zooming process that reveals multiscale details. Every path in a zoom tree represents a zoom path and every node in the tree can have an arbitrary number of subtrees to support arbitrary branching and backtracking. Zoom trees are seamlessly integrated with a table-based overview using ”hyperlinks” embedded in the table. Instead of predefined zoom paths, the interface should be able to support dynamically formed zoom paths. Furthermore, the history of a zooming process should have a tree structure where any node can have an arbitrary number of branches for zooming into different local regions of the dataset. Zoom trees support arbitrary branching and backtracking in a zooming process.

We further propose to use graphics processors (GPUs) to perform real-time query processing based on a partial data cube. We develop an efficient GPU-based parallel algorithm for online cubing and a CPU-based algorithm for grid-based data clustering to support such query processing.

Schema Based Subcube Selection: Instead of analyzing the entire data cube at once, users usually would like to focus on a subset of the dimensions every time. A subcube is defined by a subset of the dimensions. Each of the remaining dimensions is fixed to a specific value. In a data cube, a subcube can be specified with slice/dice operations. Unlike Polaris, at most two nested database dimensions (measures) can be mapped along the horizontal or vertical direction of the table to achieve simplicity and clarity.

We present a simple grid-based algorithm to cluster hundreds of thousands of points into a desired number of clusters. In doing do, we can not only reduce the overhead for transferring a large amount of data but also can reduce screen space clutter.

The described algorithms have been implemented and tested on an Intel Core 2 Duo quad-core 2.4GHz processor with an NVidia GeForce 8800 GTX GPU. To cluster 1 millon randomly generated data points into 10x10 clusters, our grid-based clustering algorithm only takes 22.96ms on a single core.

Information Visualization and Visual Data Mining
1063 citations

Journal IEEE Transactions on Visualization and Computer Graphics 2002

This is a great paper that refers to many implementations – can be a main paper

In this paper, we propose a classification of information visualization and visual data mining techniques which is based on the data type to be visualized, the visualization technique and the interaction and distortion technique. We exemplify the classification using a few examples, most of them referring to techniques and systems presented in this special issue.

Stacked Displays

Stacked display techniques are tailored to present data partitioned in a hierarchical fashion. In case of multidimensional data, the data dimensions to be used for partitioning the data and building the hierarchy have to be selected appropriately. An example of a stacked display technique is Dimensional Stacking. The basic idea is to embed one coordinate systems inside an other coordinate system, i.e. two attributes form the outer coordinate system, two other attributes are embedded into the outer coordinate system, and so on. The display is generated by dividing the outmost level coordinate systems into rectangular cells and within the cells the next two attributes are used to span the second level coordinate system. This process may be repeated one more time. The usefulness of the resulting visualization largely depends on the data distribution of the outer coordinates and therefore the dimensions which are used for defining the outer coordinate system have to be selected carefully. A rule of thumb is to choose the most important dimensions first.

Interactive Filtering

The basic idea of Magic Lenses is to use a tool like a magnifying glasses to support filtering the data directly in the visualization. The data under the magnifying glass is processed by the filter, and the result is displayed differently than the remaining data set. Magic Lenses show a modified view of the selected region, while the rest of the visualization remains unaffected.

Interactive Zooming

The basic idea of TableLens is to represent each numerical value by a small bar. All bars have a one-pixel height and the lengths are determined by the attribute values. This means that the number of rows on the display can be nearly as high as the vertical resolution and the number of columns depends on the maximum width of the bars for each attribute. The initial view allows the user to detect patterns, correlations, and outliers in the data set. In order to explore a region of interest the user can zoom in, with the result that the affected rows (or columns) are displayed in more detail, possibly even in textual form.

Interactive Linking and Brushing

The idea of linking and brushing is to combine different visualization methods to overcome the shortcomings of single techniques. Scatterplots of different projections, for example, may be combined by coloring and linking subsets of points in all projections. In a similar fashion, linking and brushing can be applied to visualizations generated by all visualization techniques described above. As a result, the brushed points are highlighted in all visualizations, making it possible to detect dependencies and correlations. Interactive changes made in one visualization are automatically reflected in the other visualizations.

Interactive Distortion

Interactive distortion techniques support the data exploration process by preserving an overview of the data during drill-down operations. The basic idea is to show portions of the data with a high level of detail while others are shown with a lower level of detail.

Tree-maps: A space-filling approach to the visualization of hierarchical information structures

1326 citations

Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on

A novel method for the visualization of hierarchically structured information. The treemap visualization technique makes 100% use of the available display space, mapping the full hierarchy onto a rectangular region in a space-filling manner. This efficient use of space allows very large hierarchies to be displayed in their entirety and facilitates the presentation of semantic information.

Our interactive approach to drawing directory trees allows users to determine how the tree is displayed. This control is essential, as it allows users to set display properties (colors, borders, etc.) maximizing the utility of the drawing based on their particular task.

Structural information in treemaps is implicitly presented, although it may also be explicitly indicated by nesting child nodes within their parent. Nesting provides for the direct selection of all nodes, both internal and leaf. The space required for nesting reduces the number of nodes which can be drawn in a given display space and hence reduces the size of the trees that can be adequately displayed compared to non-nested drawings (Travers, 1989).

A non-nested display explicitly provides direct selection only for leaf nodes, but a pop-up display can provide path information as well as further selection facilities. Non-nested presentations cannot depict internal nodes in degenerate linear sub-paths, as the bounding boxes of the internal nodes in the sub-path may be exactly equal. Such paths seldom occur and tasks dependent on long chains of single child nodes will require special treatments.

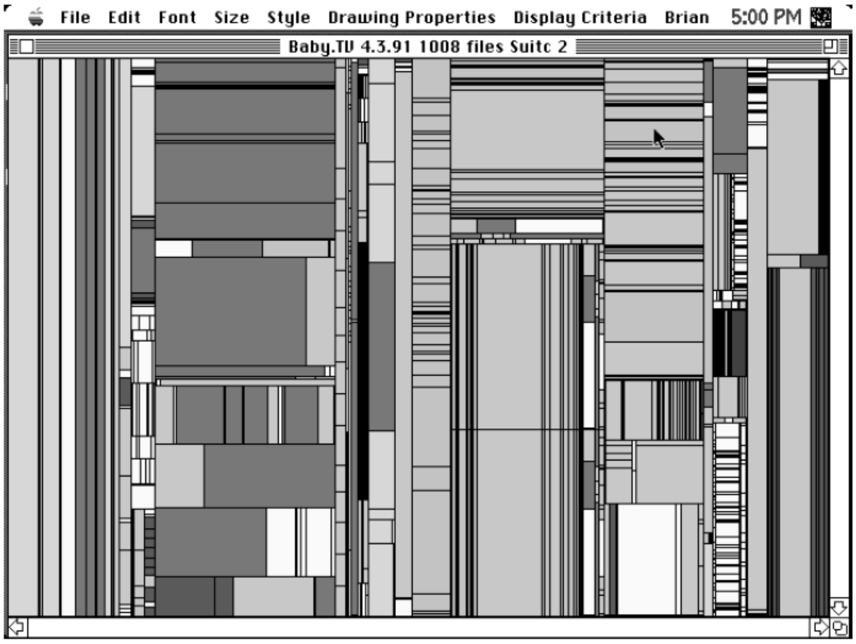

Figure 8. Treemap with 1000 Files