UNIT II SEARCH METHODS AND VISUALIZATION 9

Search by simulated Annealing – Stochastic, Adaptive search by Evaluation – Evaluation Strategies –Genetic Algorithm – Genetic Programming – Visualization – Classification of Visual Data Analysis Techniques – Data Types – Visualization Techniques – Interaction techniques – Specific Visual data analysis Techniques

VISUALIZATION

Information visualization and visual data analysis can help to deal with the flood of information. The advantage of visual data exploration is that the user is directly involved in the data analysis process. There are a large number of information visualization techniques that have been developed over the last two decades to support the exploration of large data sets.

Benefits of Visual Data Exploration

Visual data mining aims at integrating the human in the data analysis process, applying human perceptual abilities to the analysis of large data sets available in today's computer systems. The basic idea of visual data mining is to present the data in some visual form, allowing the user to gain insight into the data, draw conclusions, and directly interact with the data.

Visual data analysis techniques have proven to be of high value in exploratory data analysis. Visual data mining is especially useful when little is known about the data and the exploration goals are vague. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals can be done in a continuous fashion as needed.

The visualizations of the data allow the user to gain insight into the data and come up with new hypotheses. The verification of the hypotheses can also be done via data visualization, but may also be accomplished by automatic techniques from statistics, pattern recognition, or machine learning. In addition to the direct involvement of the user, the main advantages of visual data exploration over automatic data analysis techniques are:

- Visual data exploration can easily deal with highly non-homogeneous and noisy data.
- Visual data exploration is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters.
- Visualization can provide a qualitative overview of the data, allowing data phenomena to be isolated for further quantitative analysis.

Visual Exploration Paradigm

Visual Data Exploration usually follows a three step process: Overview, zoom and filter, and then details-on-demand. In analyzing large data sets, the user first needs to get an overview of the data.

In the overview, the user identifies interesting patterns or groups in the data and focuses on one or more of them. For analyzing the patterns, the user needs to drill-down and access details of the data. Visualization techniques are useful for showing an overview of the data, allowing the user to identify interesting subsets. In this step, it is important to keep the overview visualization while focusing on the subset using another visualization technique. An alternative is to distort the overview visualization in order to focus on the interesting subsets. This can be performed by dedicating a larger percentage of the display to the interesting subsets while decreasing screen utilization for currently uninteresting data. To further explore the interesting subsets, the user needs a drill-down capability in order to observe the details about the data.

CLASSIFICATION OF VISUAL DATA ANALYSIS TECHNIQUES

There are a number of well known techniques for visualizing such data sets, such as x-y plots, line plots, and histograms. These techniques are useful for data exploration but are limited to relatively small and low dimensional data sets. The techniques can be classified based on three criteria

- Data type
- Visualization Technique
- Interaction Technique

The **data type** to be visualized may be:

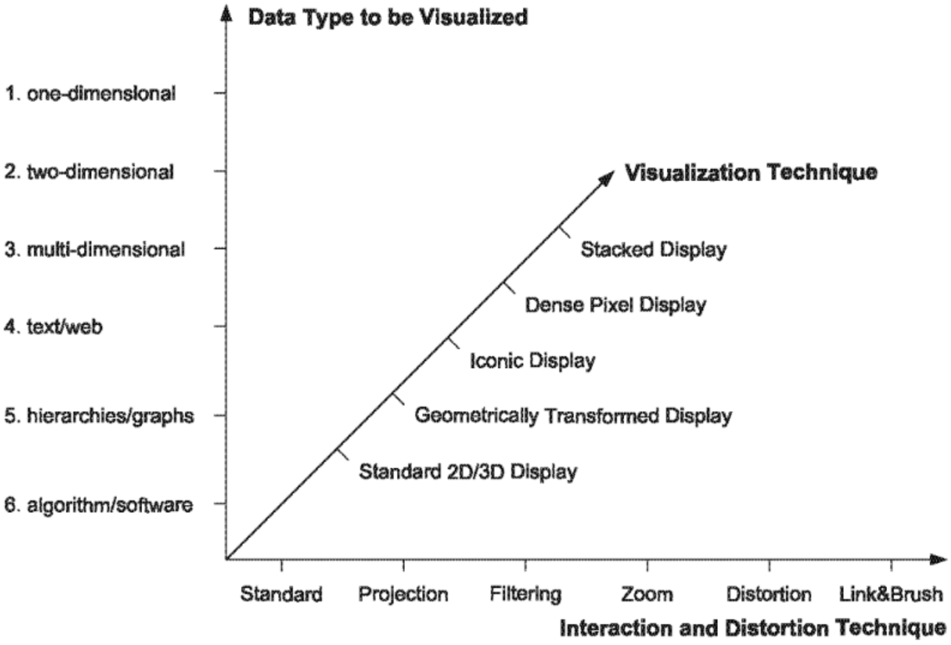
- One-dimensional data, such as temporal (time-series) data
- Two-dimensional data, such as geographical maps
- Multi-dimensional data, such as relational tables
- Text and hypertext, such as news articles and Web documents
- Hierarchies and graphs, such as telephone caUs and Web documents
- Algorithms and software, such as debugging operations

The **visualization technique** used may be classified as:

- Standard 2D/3D displays, such as bar charts and x-y plots
- Geometrically-transformed displays, such as landscapes and parallel coordinates
- Icon-based displays, such as needle icons and star icons
- Dense pixel displays, such as recursive patterns and circle segments
- Stacked displays, such as treemaps and dimensional stacking

Interaction techniques allow users to directly navigate and modify the visualizations, as well as select subsets of the data for further operations.

- Dynamic Projection, that allows smooth navigations through the data space
- Interactive Filtering, to enable users to isolate subsets of data for focussed analysis
- Zooming, to enlarge data for detailed analysis
- Distortion, to increase the screen space allocated to areas of interest while preserving the context of the entire data set
- Linking and Brushing, to enable users to select data of interest in one view and see it highlighted in other views



DATA TYPE TO BE VISUALIZED

In information visualization, the data usually consists of a large number of records, each consisting of a number of variables or dimensions. Each record corresponds to an observation, measurement, or transaction. Examples are customer properties, e-commerce transactions, and sensor output from physical experiments. The number of attributes can differ from data set to data set; The number of variables can be said as the dimensionality of the data set. Data sets may be one-dimensional, two-dimensional, multi-dimensional or may have more complex data types such as text/hypertext or hierarchies/graphs. Depending on the number of dimensions with arbitrary values the data are sometimes also called univariate, bivariate, or multivariate.

One-dimensional data

One-dimensional data usually have one dense dimension. A typical example of one-dimensional data is temporal data. One or multiple data values may be associated with each point in time. An example are time series of stock prices or time series of news data.

Two-dimensional data

A typical example of two-dimensional data is geographical data, where the two distinct dimensions are longitude and latitude. A standard method for visualizing two-dimensional data are x-y plots and maps are a special type of x-y plots for presenting two-dimensional geographical data. Examples are geographical maps. If the number of records to be visualized is large, temporal axes and maps get quickly cluttered - and may not help to understand the data.

Multi-dimensional data

Many data sets consist of more than three dimensions and therefore do not allow a simple visualization as 2-dimensional or 3-dimensional plots. Examples of multi-dimensional (or multivariate) data are tables from relational databases, which often have tens to hundreds of columns (or dimensions). Since there is no simple mapping of the data dimensions to the two dimensions of the screen, more sophisticated visualization techniques are needed. An example of a technique that allows the visualization of multi-dimensional data is the Parallel Coordinates Technique. Parallel Coordinates display each multidimensional data item as a set of line segments that intersect each of the parallel axes at the position corresponding to the data value for that dimension.

Text and Hypertext

In the age of the World Wide Web, important data types are text and hypertext, as well as multimedia web page contents. These data types differ in that they cannot be easily described by numbers, and therefore most of the standard visualization techniques cannot be applied. In most cases, a transformation of the data into description vectors is necessary before visualization techniques can be used. An example of a simple transformation is word counting which is often combined with principal component analysis or multidimensional scaling to reduce the dimensionality to two or three.

Hierarchies and Graphs

Data records often have some relationship to other pieces of information. These relationships may be ordered, hierarchical, or arbitrary networks of relations. Graphs are widely used to represent such interdependencies. A graph consists of a set of objects, called nodes, and connections between these objects, called edges or links. Examples are the e-mail interrelationships among people, their shopping behavior, the file structure of the hard disk, or the hyperlinks in the World Wide Web. There are a number of specific visualization techniques that deal with hierarchical and graphical data.

Algorithms & Software

Another class of data are algorithms and software. Coping with large software projects is a challenge. The goal of software visualization is to support software development by helping to understand algorithms (e.g., by showing the flow of information in a program), to enhance the understanding of written code (e.g., by representing the structure of thousands of source code lines as graphs), and to support the programmer in debugging the code (e.g., by visualizing errors). There are a large number of tools and systems that support these tasks.

INTERACTION TECHNIQUES

Interaction techniques allow the data analyst to directly interact with the visualizations and dynamically change the visualizations according to the exploration objectives. In addition, they also make it possible to relate and combine multiple independent visualizations.

Interaction techniques can be categorized based on the effects they have on the display. **Navigation techniques** focus on modifying the projection of the data onto the screen, using either manual or automated methods. **View enhancement methods** allow users to adjust the level of detail on part or all of the visualization, or modify the mapping to emphasize some subset of the data. **Selection techniques** provide users with the ability to isolate a subset of the displayed data for operations such as highlighting, filtering, and quantitative analysis. Selection can be done directly on the visualization (**direct manipulation**) or via dialog boxes and other query mechanisms (**indirect manipulation**). Some examples of interaction techniques are described below.

Dynamic Projection

Dynamic projection is an automated navigation operation. The basic idea is to dynamically **change the projections** in order to explore a multi-dimensional dataset. An example is the **GrandTour system** which tries to show properties such as well-separated **clusters -** two-dimensional projections of a multi-dimensional data set as a series of scatterplots. The number of possible projections is exponential to the number of dimensions. The sequence of projections shown can be random, manual, precomputed, or data driven. Systems supporting dynamic projection techniques include XGobi, XLispStat, and ExplorN.

Interactive Filtering

Interactive filtering is a **combination of selection and view enhancement**. In exploring large data sets, it is important to interactively partition the data set into segments and focus on interesting subsets. This can be done by a direct selection of the desired subset (browsing) or by a specification of properties of the desired subset (querying). Browsing is difficult for very large data sets and querying often does not produce the desired results. Therefore, a number of interactive selection techniques have been developed to improve interactive filtering in data exploration. An example of a tool that can be used for interactive filtering is the **Magic Lens**. The basic idea of Magic Lenses is to use a tool similar to a magnifying glass to filter the data directly in the visualization. The data under the magnifying glass is processed by the filter and displayed in a different way than the remaining data set. Magic Lenses show a modified view of the

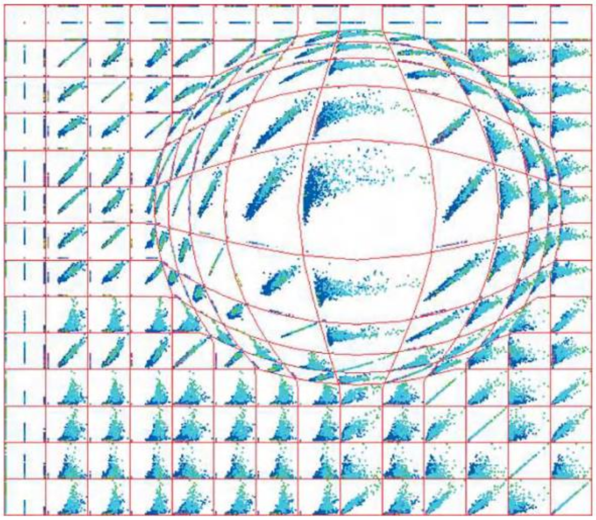
selected region, while the rest of the visualization remains unaffected. Other examples of interactive filtering techniques and tools are InfoCrystal, Dynamic Queries, and Polaris.

Zooming

Zooming is a well known view modification technique that is widely used in a number of applications. In dealing with large amounts of data, it is important to present the data in a highly compressed form to provide an overview of the data. Zooming does not only mean displaying the data objects larger, but also that the data representation may automatically change to present more details on higher zoom levels. The objects may be represented as **single pixels at a low zoom level**, as **icons at an intermediate zoom level**, and as **labeled objects at a high resolution**. An interesting example is TableLens. The basic idea of Table Lens 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 large 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. Other examples of techniques and systems that use interactive zooming include PAD+ +, IVEE/Spotfire, and DataSpace.

Distortion

Distortion is a **view modification technique** that supports 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. Popular distortion techniques are **hyperbolic and spherical** These are often used on hierarchies or graphs but may also be applied to any other visualization technique. Examples of distortion techniques include Bifocal Displays, Perspective Wall , Graphical Fisheye Views, Hyperbolic Visualization, and Hyperbox.



The above figure shows the effect of distorting part of a scatterplot matrix to display more detail from one of the plots while preserving context from the rest of the display.

Brushing and Linking

Brushing is an interactive selection process that is often, but not always, combined with **linking**, a process for communicating the selected data to other views of the data set. There are many possibilities to visualize multidimensional data, each with their own strengths and weaknesses. **The idea of linking and brushing** is to **combine** different visualization methods to overcome the shortcomings of individual techniques. **Scatterplots** of different projections, 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. Connecting multiple visualizations through interactive linking and brushing provides more information than considering the component visualizations independently. Typical examples of visualization techniques that have been combined by linking and brushing are **multiple scatterplots, bar charts, parallel coordinates, pixel displays, and maps**. Most interactive data exploration systems allow some form of linking and brushing.

SPECIFIC VISUAL DATA ANALYSIS TECHNIQUES

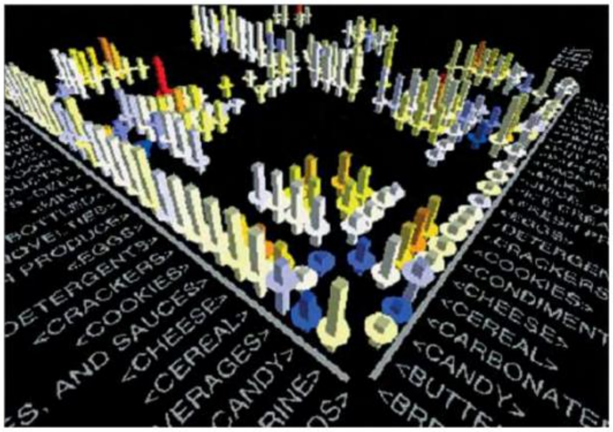
There are a number of visualization techniques that have been developed to support specific data mining tasks such as association rule generation, classification, and clustering.

Association Rule Generation

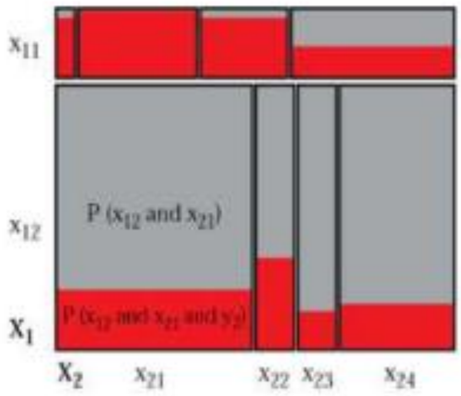
The goal of association rule generation is to find **interesting patterns and trends** in transaction databases. Association rules are statistical relations between two or more items in the data set. In a supermarket basket application, associations express the relations between items that are bought together. It is for example interesting if we find out that in 70% of the cases when people buy bread, they also buy milk. Association rules tell us that the presence of some items in a transaction imply the presence of other items in the same transaction with a certain probability, called **confidence**.

A second important parameter is the **support** of an association rule, which is defined as the **percentage** of transactions in which the items co-occur. Let $I = \{i_1, ...i_n\}$ be a set of items and let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, $X, Y \neq Ø$. The **confidence c** is defined as the **percentage of transactions that contain Y**, given X. The **support** is the **percentage of transactions that contain both X and Y**. For a given support and confidence level, there are efficient algorithms to determine all association rules.

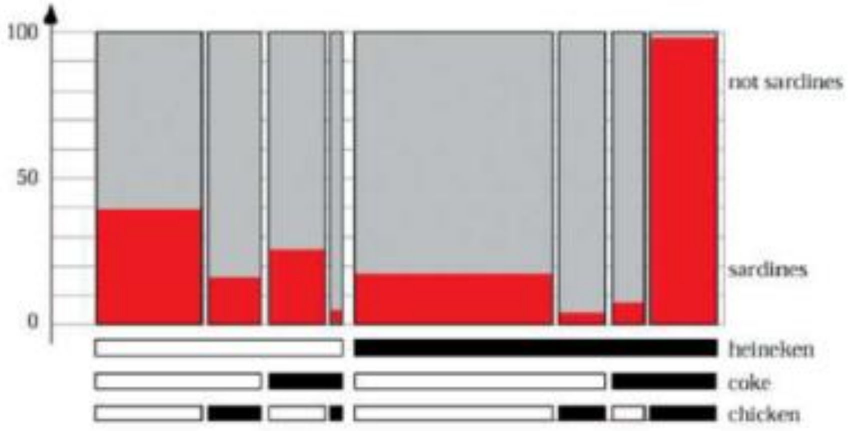
The resulting set of association rules is usually very large, especially for low support and confidence levels.. Visualization techniques have been used to allow an interactive selection of good support and confidence levels.



The above figure shows SGI MineSets Rule Visualizer which maps the left and right hand sides of the rules to the x- and y-axes of the plot, respectively, and shows the confidence as the height of the bars and the support as the height of the discs. The color of the bars shows the interestingness of the rule. Using the visualization, the user is able to see groups of related rules and the impact of different confidence and support levels. The number of rules that can be visualized is limited and the visualization does not support combinations of items on the left or right hand side of the association rules.



(a) Mosaic Plot



(b) Double Decker Plot

The above figure shows two alternative visualizations called **mosaic** and **double decker plots**.

Mosaic plots use the height of the bars instead of their width to show the parameter value. Then each resulting area is split in the same way according to a second attribute. The coloring reflects the percentage of data items that fulfill a third attribute. The visualization shows the support and confidence values of all rules of the form X_1, $X_2 \Rightarrow Y$. Mosaic plots are restricted to **two attributes** on the left side of the association rule.

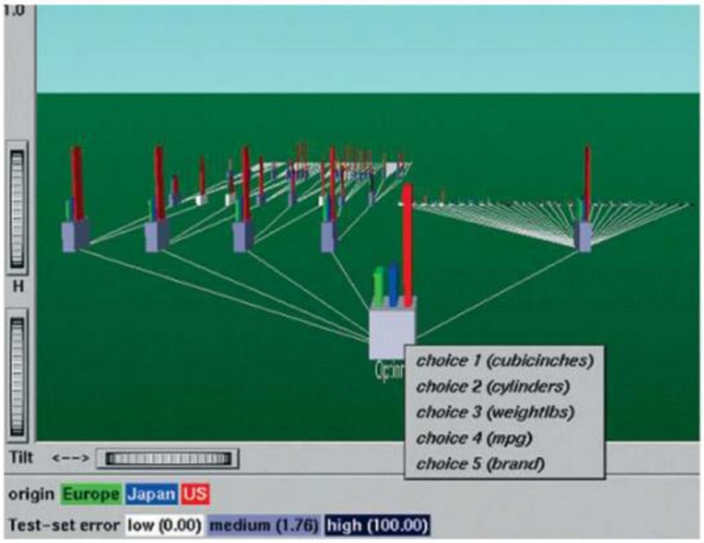
Double decker plots can be used to show more than two attributes on the left side. The idea is to display a **hierarchy of attributes on the bottom** corresponding to the left hand side of the association rules. The **bars on the top correspond to the number of items** in the considered subset of the database and therefore visualize the support of the rule. The **colored areas** in the bars correspond to the **percentage of data transactions** that contain an additional item and therefore represent the support. Other approaches to association rule visualization include graphs with nodes corresponding to items and arrows corresponding to implications and association matrix visualizations to cluster related rules.

Classification

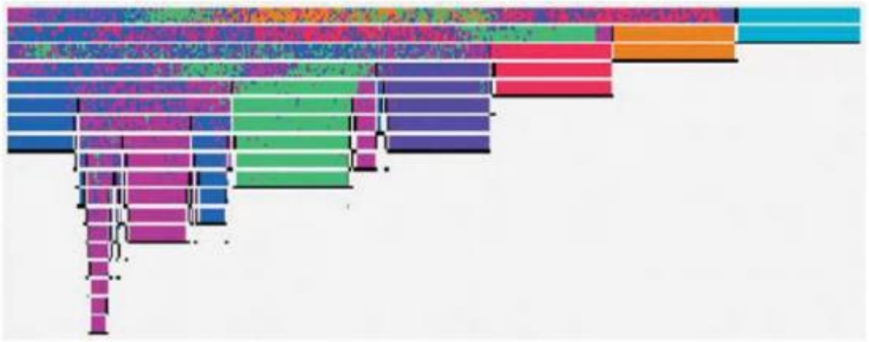
Classification is the process of developing a **classification model based** on a **training data set** with known **class labels.** To construct the classification model, the **attributes** of the training data set are analyzed and an **accurate description** or model of the classes based on the attributes available in the data set is developed. The class descriptions are used to classify data for which the class labels are unknown.

Classification is sometimes also called **supervised learning** because the training set is used to teach the system how to classify the data. There are a large number of algorithms for solving classification tasks. A popular class of approaches are algorithms that inductively construct **decision trees**. Examples are IDS, CART, ID5, C4.5, SLIQ , and SPRINT. In addition, there are approaches such as **neural networks, genetic algorithms, or Bayesian networks** that are used to solve the classification problem. Since most algorithms work as **black box** approaches, it is often difficult to understand and optimize the decision model. Problems such as overfitting or tree pruning are difficult to tackle. Visualization techniques can help to overcome these problems.

The **decision tree visualizer in SGIs MineSet system** shows an overview of the decision tree together with important parameters such as the attribute value distributions The system allows an interactive selection of the attributes shown and helps the user to understand the decision tree.



Visual classification is another sophisticated approach, which also helps in decision tree construction. The basic idea is to show each attribute value by a colored pixel and arrange them in bars - similar to the Dense Pixel Displays. The pixels of each attribute bar are sorted separately and the attribute with the purest value distribution is selected as the split attribute of the decision tree. The procedure is repeated until all leaves correspond to pure classes.



An exemplary decision tree resulting from this process is shown in the above figure. Compared to a standard visualization of a decision tree, additional information is provided that is helpful for explaining and analyzing the decision tree, namely

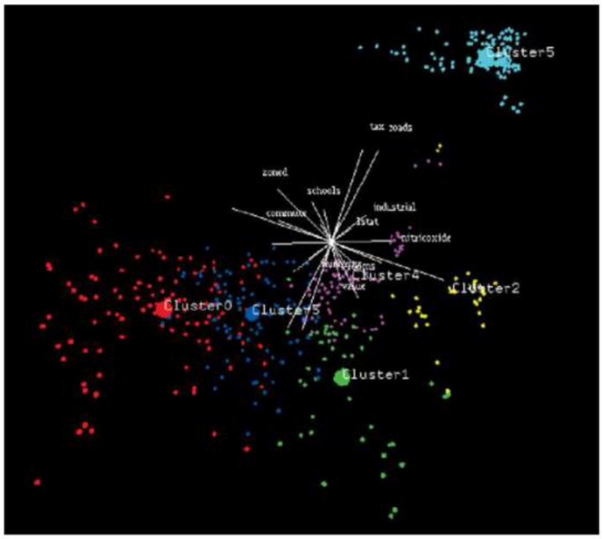
- size of the nodes (number of training records corresponding to the node)
- quality of the split (purity of the resulting partitions)
- class distribution (frequency and location of the training instances of all classes).

In general, visualizations can provide a better understanding of the classification models and they can help to interact more easily with the classification algorithms in order to optimize the model generation and classification process.

Clustering

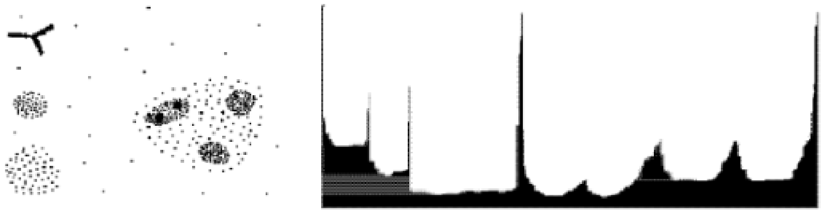
Clustering is the process of finding a **partitioning of the data set** into homogeneous subsets called clusters Unlike classification, clustering is often implemented as a form of **unsupervised learning**. This means that the classes are unknown and no training set with class labels is available. A wide range of clustering algorithms are **density-based methods** such as KDE and **linkage-based methods**. Most algorithms use assumptions about the properties of the clusters that are either used as defaults or have to be given as input parameters. Depending on the parameter values, the user obtains different clustering results.

In two- or three-dimensional space, the impact of different algorithms and parameter settings can be explored easily using simple visualizations of the resulting clusters (for example, x-y plots), but in higher dimensional space the impact is much more difficult to understand. Some higher-dimensional techniques try to determine two- or three-dimensional projections of the data that retain the properties of the high-dimensional clusters as much as possible.



The above shows a three-dimensional projection of a data set consisting of five Clusters. While this approach works well with low- to medium-dimensional data sets, it is difficult to apply it to large high-dimensional data sets, especially if the clusters are not clearly separated and the data set also contains noise. In this case, more sophisticated visualization techniques are needed to guide the clustering process, select the right clustering model, and adjust the parameter values appropriately.

The visualization techniques help in high-dimensional clustering is **OPTICS (Ordering Points To Identify the Clustering Structure)**. The idea of OPTICS is to create a **one-dimensional ordering** of the database representing its density-based clustering structure.



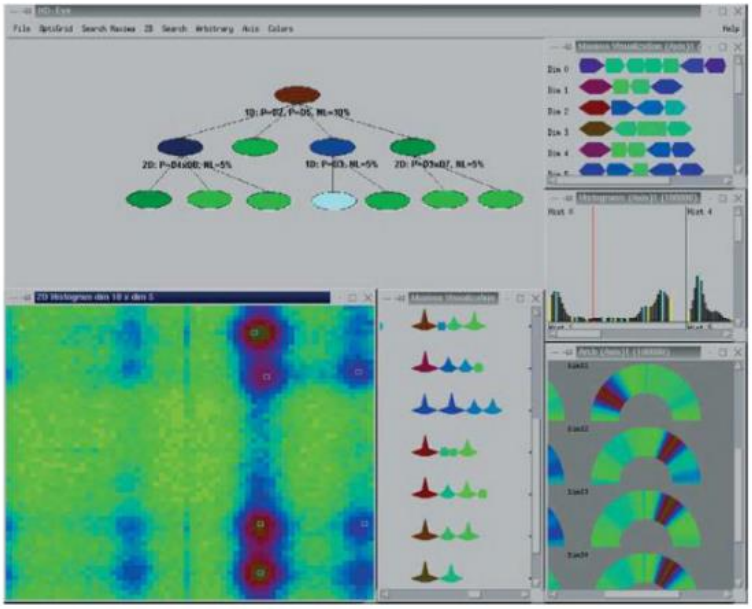
(a) Data Set

(b) Reachability Plot

The figure shows a two-dimensional data set together with its reachability distance plot. Intuitively, points within a cluster are close in the generated one-dimensional ordering and their reachability distance is similar. Jumping to another cluster results in higher reachability distances. The idea works for data of arbitrary dimension. The reachability plot provides a visualization of the inherent clustering structure and is therefore valuable for understanding the clustering and guiding the clustering process.

Another interesting approach is the **HD-Eye system**. The HD-Eye system considers the clustering problem as a partitioning problem and supports a tight integration of advanced clustering algorithms and state-of-the-art visualization techniques, allowing the user to directly interact in the **crucial steps of the clustering process**. The crucial steps are the **selection of dimensions** to be considered, the **selection of the clustering paradigm**, and the **partitioning of the data set**.

Novel visualization techniques are employed to help the user identify the most interesting projections and subsets as well as the best separators for partitioning the data. The figure shows an example of the HD-Eye system with its basic visual components for cluster separation.



The **separator tree** represents the **clustering model** produced so far in the clustering process. The **abstract iconic displays** (top right and bottom middle in figure) visualize the **partitioning potential** of a large number of projections. The **properties** are based on **histogram information** of the point density in the projected space. The **number of icons** corresponds to the **number of peaks in the projection** and their color to the number of data points belonging to the maximum. The **color** follows a given color table ranging from **dark colors for large maxima** to **bright colors for small maxima**. The measure of how well a maximum is separated from the others is reflected by the shape of the icon and the degree of separation varies from sharp spikes for well-separated maxima to blunt spikes for weak-separated maxima. The color- and curve-based point density displays present the density of the data and allow a better understanding of the data distribution, which is crucial for an effective partitioning of the data. The **visualizations** are used to decide which **dimensions are taken for the partitioning**. In addition, the partitioning can be specified interactively directly within the visualizations, allowing the user to define non-linear partitionings.