Multi-Attribute Decision Modeling: Industrial Applications of DEX

Marko Bohanec[1], Vladislav Rajkovič [2,1]
[1] Jožef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia
Phone: +386 61 1773 309, Fax: +386 61 1258 058
[2] University of Maribor, Faculty of Organisational Sciences, Kranj, Slovenia
{marko.bohanec, vladislav.rajkovic}@ijs.si

Keywords: decision support, multi-attribute decision making, qualitative decision models

Edited by: Cene Bavec and Matjaz Gams

Received: October 2, 1999 **Revised:** November 20, 1999 **Accepted:** December 12, 1999

DEX is an expert system shell for qualitative multi-attribute decision modeling and support. During the last decade, it has been applied over fifty times in complex real-world decision problems. In this article we advocate for the applicability and great potential of this approach for industrial decision-making. The approach is illustrated by a typical industrial application in land use planning, and supplemented by an overview of some other completed industrial applications. The learned lessons indicate the suitability of the qualitative DEX methodology particularly for "soft", i.e., less structured and less formalized, decision problems. Practical experience also indicates the importance of methods that facilitate the analysis, simulation, and explanation of decisions.

1 Introduction

In complex decision-making processes, it is often necessary to deal with the problem of choice (Simon, 1977). Given a set of *options* (or alternatives), which typically represent some objects or actions, the goal is

(1) to *choose* an option that best satisfies the aims or goals of decision maker, or
(2) to *rank* the options from the best to the worst one.

One of the approaches to such problems, which is well known and commonly employed within Decision Support Systems (Andriole, 1989), is based on *evaluation models* (Figure 1). The idea is to develop a model that evaluates options giving an estimate of their worthiness (*utility*) for the decision-maker. Based on this estimate, the options are ranked and/or the best one is identified. Usually, a decision model is designed in an interaction between the decision maker and decision analyst.

An important feature of evaluation models is that they can be, in addition to the sole *evaluation* of options, used for various *analyses* and *simulations*, which may contribute to a better justification and explanation of decisions. For example, a *what-if analysis* can provide a better insight into a causal relation between problem parameters and outcomes. Another example is a *sensitivity analysis* that can assess the sensitivity of model with respect to small changes of options.

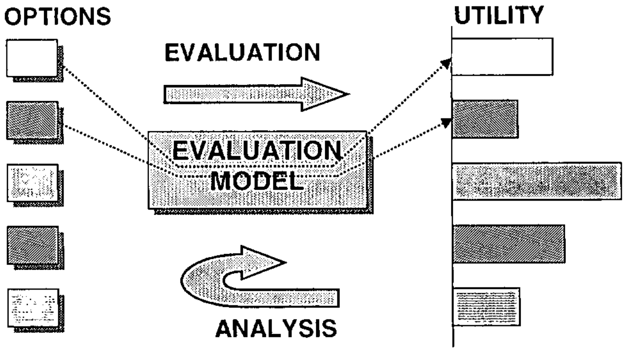


Figure 1: Evaluation-based decision modeling

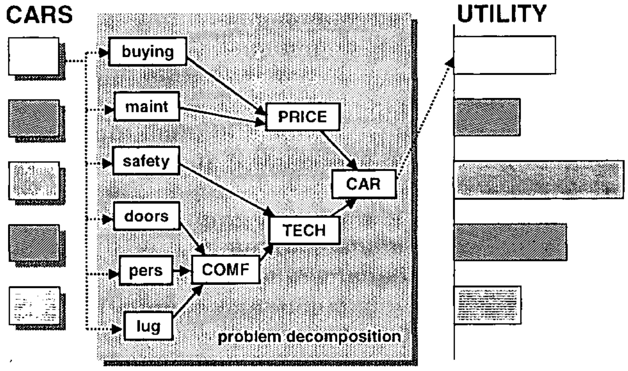


Figure 2: Multi-attribute decision modeling

An evaluation model can be developed in many ways. The approach that prevails in decision practice is based on *multi-attribute decomposition* (Chankong and Haimes, 1983; Saaty, 1993; Buede and Maxwell, 1995): we take a complex decision problem and decompose it into smaller and less complex subproblems. The result of such development is a *decision model* that consists of *attributes*, each of which represents a decision subproblem. Attributes are organized hierarchically and connected by *utility functions* that evaluate them with respect to their immediate descendants in the hierarchy. Figure 2 illustrates this basic principle of multi-attribute modeling by showing a simple hierarchy of attributes for the evaluation of cars.

Real-life applications of multi-attribute methods, which were conducted at Jožef Stefan Institute in Ljubljana, were all based on DEX (Bohanec and Rajkovič, 1990). This is an expert system shell for multi-attribute decision making that combines the "traditional" multi-attribute decision making with some elements of Expert Systems and Machine Learning. The distinguishing characteristic of DEX is its capability to deal with *qualitative* models. Instead of numerical variables, which typically constitute traditional *quantitative* models, DEX uses qualitative variables; their values are usually represented by words rather than numbers, for example "low", "appropriate", "unacceptable", etc. Furthermore, to represent and evaluate utility functions, DEX uses *if-then decision rules*. In contrast, this is traditionally carried out in a numerical way, using weights or similar indicators of attributes' importance.

An important additional feature of DEX is its capability to deal with inaccurate, uncertain or even missing data about options. In such cases, DEX represents options by distributions of qualitative values, and evaluates them by methods based on probabilistic and/or fuzzy propagation of uncertainty.

During the last decade, DEX was used in more than fifty real-life decision problems. The aim of this article is to advocate for the wide applicability of DEX to complex decision problems that occur in industry. In the next section, we first illustrate the approach by a typical industrial application in land use planning. This is followed by an overview of several other completed industrial applications in performance evaluation of companies, evaluation of products, projects and investments, ecology, and loan allocation. Finally, we summarize the lessons learned in these applications, and propose some future directions for the development of underlying methodology.

2 A Real-World Case

One of the most typical applications of DEX occurred with Goriške opekarne, a company located near the Slovenian city of Nova Gorica. The company is engaged in a very traditional business: production of bricks and tiles. Decades ago, they had built a factory near a suitable clay pit that was then providing raw material for their production. Until 1993, however, the clay pit has become almost completely exhausted, so the company was faced with a critical strategic decision of how to survive and continue with this type of production. Their only option was to find a new appropriate clay-pit location.

An exploratory study revealed three possible candidate locations. Unfortunatelly, none of them was really appropriate as numerous difficult problems were foreseen, ranging from technological, transportational and financial to environmental and socio-psychological. The latter two problems seemed particularly important as the project was inevitably going to affect the environment, leading to a possible rejection of local inhabitants. For these reasons, a group of experts was formed to thorougly analyze the problem and propose alternative solutions (Bohanec, et al., 1993).

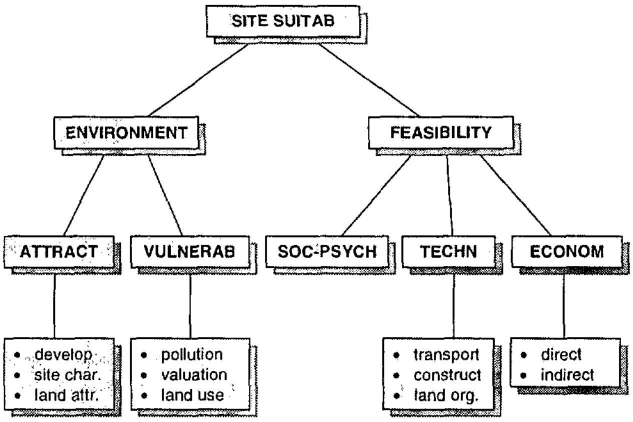


Figure 3: Topmost levels of clay-pit evaluation model

In the first stage, the experts developed the structure of multi-attribute model for the evaluation of clay-pit locations.. Two primary evaluation dimensions were taken into account: Environmental impact and Feasibility of the project. For each of these, the most relevant attributes were identified and organized into a hierarchical structure (Figure 3). Note that only topmost levels of the model are shown in the figure. In total, the model contained 49 attributes: 29 basic (terminal nodes) and 20 aggregate (internal nodes).

Table 1: Decision rules for Site suitability

	ENVIRONMENT	FEASIBILITY	SITE
1	*	unacc	unacc
2	unacc	*	unacc
3	less-acc	less-acc	marg-acc
4	≥ acc	less-acc	less-acc
5	less-acc	acc	less-acc
6	acc	acc	acc
7	good	acc	good

The second stage involved the definition of decision rules. Basically, these are simple *if-then* rules that for each of the 20 internal nodes in the model determine its evaluation with respect to its lower-level descendants in the hierarchy. Usually, they are represented in a tabular form. For example, Table 1 shows decision rules that were defined by the experts for the topmost node Site suitability. In the table, an asterisk '*' represents any value, and '≥' means 'better or equal'.

In the third stage of the decision-making process, the options are identified and described by the values of basic attributes. In our case, there were three clay-pit locations, each of which was represented by 29 data items that corresponded to basic attributes of the model. Furthermore, as some of these items, such as Social-psychological feasibility, were inherently inaccurate or difficult to obtain, several variations of the descriptions were formed, anticipating either an "optimistic" or "pessimistic" development of the project. Effectively, this increased the number of considered options to eight (Figure 4) and provided a foundation for subsequent what-if analysis.

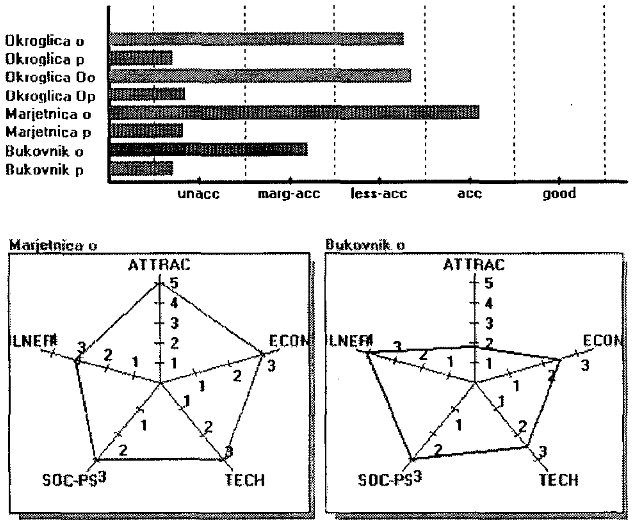


Figure 4: Visualization of clay-pit evaluation results

In the last stage, the model was utilized to evaluate the clay-pit locations. As shown in Figure 4, the best location was the one called Marjetnica, which was evaluated as "acceptable", but only in its "optimistic" instance. On the other hand, all the "pessimistic" instances were unacceptable, indicating the great sensitivity of decision. Therefore, thorough what-if and sensitivity analyses were performed for each location. The most important result was achieved by comparing "optimistic" and "pessimistic" options with respect to basic attributes. The outcome of this comparison was a comprehensive list of possible problems that could occur with each location. On this basis, the expert team not only was able to find the best location, but also to foresee potential pitfalls and suggest how to avoid them.

3 Other Applications

In about ten years time, DEX was used in more than fifty real-life decision problems in various areas. About one half of the problems can be classified as industrial, while the remaining were conducted in the fields such as education or medicine and health care (Bohanec, et al., 1999). Some of the industrial problems were very difficult and involved substantial financial and other risks for decision-making organizations. In what follows we briefly outline five representative application areas, which clearly indicate the wide applicability of DEX for a variety of decision problems. The description of some other early industrial applications can also be found in (Urbančič, et al., 1991).

3.1 Performance Evaluation of Companies

Here, the general task is that a company or agency develops an evaluation model that assesses the performance of some other companies. The aim is, for example, to find a suitable business partner. The work with DEX in this area began in 1987, where a number of such models were developed in collaboration with the International Center for Public Enterprises (Bohanec and Rajkovič, 1990). An example hierarchy of attributes that was used to assess the performance of 54 public enterprises in Pakistan, is shown in Figure 5. This work culminated in 1989 with the development of models that were used in the privatization of Peruvian public enterprises.

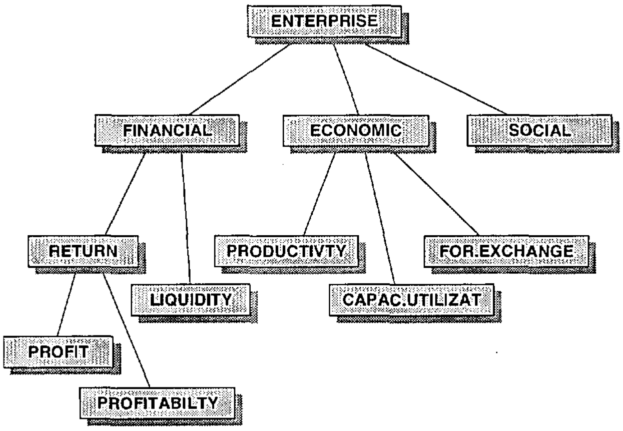


Figure 5: Topmost levels of the model for performance evaluation of public enterprises

3.2 Product portfolio evaluation

The problem is to assess the quality of products made by a company or production unit. This assessment is vital for the formation of strategies. The approach with DEX was based on the so-called portfolio method (Krisper, et al., 1991), which evaluates products using two primary evaluation dimensions: market attractiveness and competitive ability. Several practical cases were analyzed in this way, including the products of some well-known Slovenian companies Fructal, Radenska, SRC, and DZS.

3.3 Evaluation of projects and investments

The evaluation of projects or investment strategies is an industrial application context in which DEX has got the largest number of applications. The most typical investments included various software, hardware and technology, such as data base management systems, production control software, meteorological radar equipment, or a production line. The decision problems were often related to various investment proposals and tenders. An example of such applications, which is documented quite in detail, is a model for the evaluation of research and development projects (Bohanec, et al., 1995).

3.4 Remediation of dumpsites

This is a recent application in the field of environmental care. In order to alleviate the problem of illegal dumpsites in Slovenia, an expert system was developed that assesses the environmental impact of dumpsites and suggests activities for their remediation (Špendl, 1998). The environmental impact of dumpsites is assessed by a qualitative DEX model (Figure 6), which is embedded in the expert system.

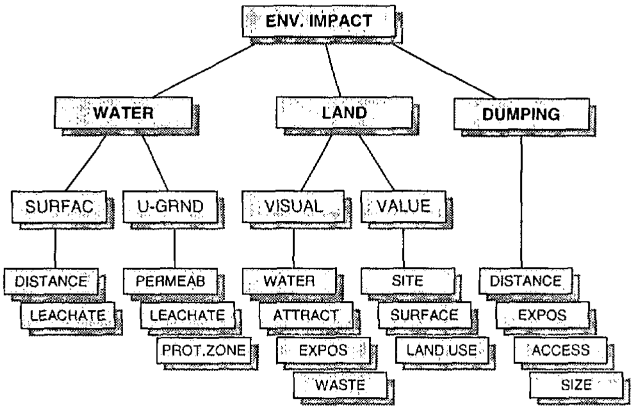


Figure 6: Model for the assessment of dumpsite's environmental impact (topmost levels only)

3.5 Housing loan allocation

This is an example of a repetitive decision-making task being supported by a DEX model. The model is a part of a management decision support system that is used since 1991 by the Housing Fund of the Republic of Slovenia for the allocation of housing loans with favorable terms to citizens (Bohanec, et al., 1996). Until 1999, the Fund has issued 16 floats of loans, i.e., about two per year, and approved almost 20 thousand loans.

The amount requested by applicants in a float typically far exceeds the available funds. Thus, the applicants must be ranked into a priority order. The procedure is required to be fast, reliable, transparent, and fair for all applicants. The request for transparency asks for effective explanations of loan priority order, which have to be provided to both the decision-making committee and a large number (usually, several thousands) of applicants. In the Fund's system, these requirements were fulfilled by a qualitative model that ranks the applications into five priority classes and provides a foundation for various explanations, which are obtained by analyses and simulations of application data and the model itself.

4 Experience

Some important lessons have been learned in the applications of DEX. Here, we present some findings related to the duration of model development processes, difficulty of development stages, and categories of decision problems that seem to be particularly well suited for the application of DEX.

The time needed to develop a DEX model turns out to be extremely problem-dependent: it may take from few hours to several months. Most typically, however, the development requires about two working days for the development of model structure, from one to two days to define decision rules, and from one to several days to collect data about options, to evaluate, and analyze them. Therefore, the process most typically lasts from two to ten working days.

The most difficult stage of the process is its first one, in which the relevant attributes must be identified and appropriately organized into a hierarchical structure. This stage heavily relies on knowledge and experience of decision-makers and experts, and requires a deep understanding of the decision problem. It can still be considered more art than science. The remaining stages have been found much less problematic. Therefore, an appropriate identification of model structure mostly determines the success of the decision-making process.

DEX with its qualitative modeling and ability to handle inaccurate and/or incomplete data about options appears particularly well suited for decision problems that involve qualitative concepts and a great deal of expert judgement. Also, it seems that the usefulness of DEX increases with the increasing difficulty, or "complexity", of the decision problem. So far, the best results were achieved in problems that required large models, consisting of at least 15 attributes, and/or involving a large number of options, i.e., from about 10 to several hundreds of options. On the other hand, DEX turned out to be unsuitable for problems that require exact formal modeling, numerical simulation and/or optimization.

5 Further Work

Currently, there are three limitations of the DEX approach that, we believe, can be greatly improved by appropriate extensions of the methodology. First, the difficult stage of model structure development could be additionally supported by a machine learning method that would develop (or at least suggest) model structure using decision examples taken either from an existing database of past decisions, or provided explicitly by the decision-maker. A considerable progress in this direction has already been made by the development of a learning method called HINT (Zupan, et al., 1999). Given training examples, HINT develops a hierarchical multi-attribute evaluation model that explains and possibly generalizes the examples. The structure of the models developed by HINT is essentially the same as the structure of models developed "manually" using DEX. The HINT's model development is based on function decomposition, an approach that was originally developed for the design of digital circuits.

Another limitation of DEX is that it is strictly limited to qualitative decision models; it cannot use numerical variables nor analytically represented utility functions that are commonly used in traditional quantitative models. This is sometimes advantageous in comparison with other decision modeling systems, which exclusively rely on quantitative models. However, many real-life decision problems require both qualitative and quantitative attributes, so the integration of these two may have a great practical impact: it may increase the flexibility of the method and extend the range of decision problems that can be successfully approached. Methodologically, such integration appears quite difficult and requires more research. In the context of DEX, we consider it a long-term goal.

Last but not least, the major part of DEX software has been developed about ten years ago and currently appears quite outdated. Therefore, an overall redesign and renewal of software is planned for the near future. Currently, we are developing a program called DEXi, an educational subset of DEX to be used by students and teachers in secondary schools and faculties. We plan to follow this by the development of a functionally complete state-of-the-art DEX system.

6 Conclusion

The DEX system effectively integrates two methodologies: multi-attribute decision making and expert systems. To a limited extent, it also includes some elements of machine learning and fuzzy logic. By this, it facilitates a structured and systematic approach to complex decision problems. So far, DEX has been successfully used in over fifty real-life decision problems in industry, medicine, health care and education, which all speak in favor for its wide applicability and flexibility. From the practical viewpoint, the most important characteristics of DEX are:

1. Qualitative (symbolic) decision modeling, which is particularly well suited for "soft" decision problems, i.e., less structured and less formalized problems, which involve a great deal of expert judgement.
2. Focus on the explanation and analysis of options, which lead to better-understood and justified decisions.
3. Active support of the decision-maker in the acquisition of decision rules, which speeds up model development and reduces the number of errors.

The goals of further research and development related to DEX are twofold. First, we wish to improve the support in the difficult stage of model structure development, and propose to use machine learning methods, such as HINT, for that purpose. To further improve the flexibility and general applicability of the approach, we suggest further research towards an integration of qualitative and quantitative decision models.

7 References

[1] S.J. Andriole: Handbook of Decision Support Systems. TAB Books, 1989.

[2] M. Bohanec, V. Rajkovič, V.: DEX: An expert system shell for decision support, *Sistemica* 1, 145–157, 1990.

[3] M. Bohanec, B. Kontić, D. Kos, J. Marušič, S., Polič, J. Rakovec, B. Sedej, et al.: Comparison of clay-pit locations Okroglica, Bukovnik, Marjetnica with respect to environmental protection (in Slovenian). Ljubljana: Jožef Stefan Institute, Report DP-6742, 1993.

[4] M. Bohanec, V. Rajkovič, B. Semolič, A. Pogačnik: Knowledge-based portfolio analysis for project evaluation. *Information & Management* 28, 293–302, 1995.

[5] M. Bohanec, B. Cestnik, V. Rajkovič: A management decision support system for allocating housing loans. *Implementing Systems for Supporting Management Decision* (eds. P. Humphreys, L. Bannon, A. McCosh, Migliarese, J.-C. Pomerol). Chapman & Hall, 1996.

[6] M. Bohanec, B. Zupan, V. Rajkovič: Hierarchical multi-attribute decision models and their application in health care. *Proc. Medical Informatics Europe 99* (eds. P. Kokol, B. Zupan, J. Stare, M. Premik, R. Engelbrecht), Amsterdam: IOS Press, 670–675, 1999.

[7] D.M. Buede, D.T. Maxwell: Rank disagreement: A comparison of multi-criteria methodologies. *Journal of Multi-Criteria Decision Analysis* 4, 1–21, 1995.

[8] V. Chankong, Y.Y. Haimes: *Multiobjective Decision Making: Theory and Methodology*. North-Holland, 1983.

[9] M. Krisper, V. Bukvič, V. Rajkovič, T. Sagadin: Strategic planning with expert system based portfolio analysis. *EXPERSYS-91: Expert system applications* (eds. J. Hasemi, J.G. Gouardères, J.P. Marciano). IITT International, 1991.

[10] T.L. Saaty: *Multicriteria Decision Making: The Analytic Hierarchy Process*. RWS Publications, 1993.

[11] A.H. Simon: *The New Science of Management Decision*. Prentice-Hall, 1977.

[12] R. Špendl: *An expert system for the evaluation of environmental impact and remediation of illegal dumpsites* (in Slovenian). M.Sc. Thesis. University of Ljubljana, Faculty of Information and Computer Science, 1998.

[13] T. Urbančič, I. Kononenko, V. Križman: *Review of Applications by Ljubljana Artificial Intelligence Laboratories*. Ljubljana: Jožef Stefan Institute, Report DP-6218, 1991.

[14] B. Zupan, M. Bohanec, J. Demšar, I. Bratko, I.: Learning by discovering concept hierarchies. *Artificial Intelligence* 109, 211–242, 1999.

Perception-Based Classification

Mihael Ankerst, Christian Elsen, Martin Ester, Hans-Peter Kriegel
Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 München, Germany
{ankerst I ester I kriegel}@dbs.informatik.uni-muenchen.de, c.elsen@elsen.net

Keywords: classification, decision tree, data mining, visualization

Edited by: Cene Bavec and Matjaz Gams

Received: October 2, 1999 **Revised:** December 2, 1999 **Accepted:** December 19, 1999

Classification is an important problem in the emerging field of data mining. Given a training database of records, each tagged with a class label, the goal of classification is to build a concise model that can be used to predict the class label of future, unlabeled records. A very popular class of classifiers are decision trees because they satisfy the basic requirements of accuracy and understandability. Instead of constructing the decision tree by a sophisticated algorithm, we introduce a fully interactive method based on a multidimensional visualization technique and appropriate interaction capabilities. Thus, domain knowledge of an expert can be profitably included in the tree construction phase. Furthermore, after the interactive construction of a decision tree, the user has a much deeper understanding of the data than just knowing the decision tree generated by an arbitrary algorithm. The interactive approach also overcomes the limitation of most decision trees which are fixed to binary splits for numeric attributes and which do not allow to backtrack in the tree construction phase. Our performance evaluation with several well-known datasets demonstrates that even users with no a priori knowledge of the data construct a decision tree with an accuracy similar to the tree generated by state of the art algorithms. Additionally, visual interactive classification significantly reduces the tree size and improves the understandibility of the resulting decision tree.

1 Introduction

The success of computerized data management has resulted in the accumulation of huge amounts of data in several organizations. There is a growing perception that analyses of these large databases can turn this "passive" data into useful information. The term *Data Mining* refers to the discovery of non-trivial, previously unknown, and potentially useful patterns embedded in databases.

Classification is one of the major tasks of data mining. The goal of *classification* is to assign a new object to a class from a given set of classes based on the attribute values of this object. Different methods [12] have been proposed for the task of classification, for instance *decision tree classifiers* which have become very popular. Decision tree classifiers are primarily aimed at attributes with a *categorical* domain, that is a small set of discrete values. *Numeric* attributes, however, play a dominant role in application domains such as astronomy, earth sciences and molecular biology where the attribute values are obtained by automatic equipment such as radio telescopes, earth observation satellites and X-ray cristallographs. [6] discusses an approach that splits numeric attributes into multiple intervals rather than just two intervals. The well-known algorithms, however, perform a binary split of the form for a numeric attribute *a* and a real number v. The SPRINT decision tree classifier [3] processes numeric attributes as follows. There are $n - 1$ possible splits for n distinct values of a. The gini index is calculated at each of these n - 1 points and the attribute value yielding the minimum gini index is chosen as the split point. CLOUDS [4] draws a sample from the set of all attribute values and evaluates the gini index only for this sample thus improving the efficiency.

A commercial system for interactive decision tree construction is SPSS CHAID [15] which - in contrast to our approach - does not visualize the training data but only the decision tree. Furthermore, the interaction happens only before the tree construction yielding user defined values for global parameters such as maximum tree depth or minimum support for a node of the decision tree.

Visual representation of data as a basis for the human-computer interface has evolved rapidly in recent years. [8] gives a comprehensive overview over existing visualization techniques for large amounts of multidimensional data. Recently, several techniques of visual data mining have been introduced. [5] presents the technique of *Independence Diagrams* for visualizing dependencies between two attributes. The brightness of a cell in the two-dimensional grid is set proportional to the density of corresponding data objects. This is one of the few techniques which does not visualize the discovered knowledge but the underlying data. However, the proposed technique is limited to two attributes. [10] presents a decision table classifier and a mechanism to visualize the resulting *decision tables*. It is argued that the visualization is appropriate for business users not familiar with machine learning concepts. In contrast to well-known decision tree classifiers, our novel interactive approach enables arbitrary split points for numeric attributes, the use of domain knowledge in the tree construction phase and backtracking.

In this paper, we introduce a novel interactive decision tree classifier based on a multidimensional visualization of the training data. Our approach allows to integrate the domain knowledge of an expert in the tree construction phase and it overcomes the limitation of binary splits for numeric attributes. The rest of this paper is organized as follows. In section 2 we introduce our technique for visualizing the training data. The support for interactively constructing a decision tree - which we have implemented in the Perception-Based Classification (PBC) system - is discussed in section 3. Section 4 reports the results of an extensive experimental evaluation on several well-known datasets. Section 5 summarizes this paper and outlines several issues for future research.

2 Visualizing the training data

In our approach, we visualize the training data in order to support interactive decision tree construction. We introduce a novel method for visualizing multidimensional data with a class label such that their degree of impurity with respect to class membership can be easily perceived by a user. Our pixel-oriented method maps the classes to colors in an appropriate way. The basic idea of pixel-oriented visualization techniques [8] is to map each attribute value v_i of each data object to one colored pixel and to represent the values belonging to different attributes in separate subwindows. The proposed techniques [9] differ in the arrangement of pixels within a subwindow.

Circle Segments [2] is a recent pixel-oriented technique which was introduced for a more intuitive visualization of high-dimensional data. The Circle Segments technique maps d-dimensional objects to a circle which is partitioned into *d* segments representing one attribute each. Figure 1 illustrates the partitioning of the circle as well as the arrangement. Within each segment, the arrangement starts in the middle of the circle and continues to the outer border of the corresponding segment in a line-by-line fashion. These lines upon which the pixels are arranged are orthogonal to the segment halving lines. An extension of this technique has been applied in the context of cluster analysis [1] .

While most approaches of visual data mining visualize the discovered knowledge, our approach is to visualize the training data in order to support interactive decision tree construction.

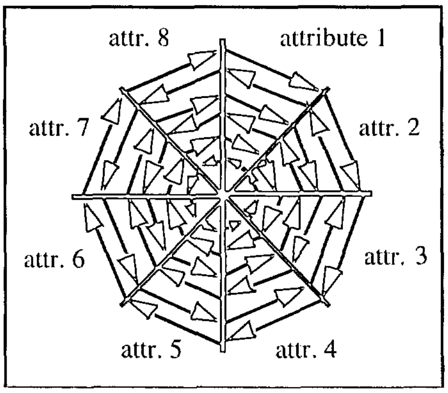


Figure 1.Illustration of the Circle Segments technique for 8-dimensional data objects

We introduce a novel method for visualizing multidimensional data with a class label such that their degree of impurity with respect to class membership can be easily perceived by a user. Our method performs pixel-oriented visualization and maps the classes to colors in an appropriate way.

Let D be a set of data objects consisting of d attributes $A_1, \ldots, A_d$ and having a unique *class label* from the set of $\mathit{Classes} \doteq \{c_1, c_2, \ldots c_k\}$. For each attribute A_i, let a total order $\leq$ be defined, for example the *≤-order* for numeric attributes or the lexicographic order for string attributes.

To map each attribute value of D to a unique pixel, we follow the idea of the Circle Segments technique, i.e. we represent all values of one attribute in a segment of a circle with the proposed arrangement inside a segment. We do not use, however, the overall distance from a query to determine the pixel position of an attribute value. Instead, we sort each attribute separately and use the induced order for the arrangement in the corresponding circle segment. The color of a pixel is determined by the class label of the object to which the attribute value belongs. In the following, we introduce our technique for mapping classes to colors.

Let *Colors* be the set of all different colors which can be represented in a given color model such as the RGB model, denoted as $\mathit{Colors} = \{col_1, col_2, \ldots col_m\}$, $m \leq k$. We are looking for an injective function *visualize*: $\mathit{Classes} \rightarrow \mathit{Colors}$ which, roughly speaking, should map "similar" classes to "similar" colors and "dissimilar" classes to "dissimilar" colors. We use distance functions to define a formal notion of similarity for both the classes and the colors: the smaller the distance, the larger the similarity and vice versa. For example, when we have no additional information about the semantics of the classes c_i, we use the following distance function for classes:

$$dist_{categorical}(c_i, c_j) = \begin{cases} 0 & if\ i=j \\ 1 & else \end{cases}$$

There are, however, many cases where we know more about the semantics of the classes. For example, there may be a class hierarchy defined by a predecessor function *pred* for each class. Then we may use the distance function $dist_h$ defined as follows:

$$dist_h(c_i, c_j) = \begin{cases} 0 \text{ if } i=j \\ 1+dist_h(pred(c_i), pred(c_j)) \text{ else} \end{cases}$$

The indices of the classes c_i are chosen such that classes with a low distance receive neighboring indices implying a total order of *Classes* (which may not be uniquely defined). We define the total class distance $dist_{\text{total-class}}$ as follows:

$$dist_{total-class} = \sum_{i=1}^{k-1} dist(c_i, c_{i+1})$$

To define a distance function and a total order for the *Colors*, we need a suitable color scale with the following properties:

- preservation of the order of the attribute values
  In the color scale, each color col_i should be perceived as "preceeding" any color col_{i+1}.
- uniformity of the perceived distances
  For any pairs (col_i, col_{i+1}) and (col_j, col_{j+1}) the perceived "distance" between col_i and col_{i+1} should be the same as the perceived "distance" between col_j and col_{j+1}.

The function $map: \{1,..,k\} \rightarrow \{1,..,m\}$ maps class *indices* to color *indices* as follows:

$$map(i) = \begin{cases} 1 \text{ if } i=1 \\ map(i-1) + \left\lceil \dfrac{dist(c_{i-1}, c_i)}{total-class-dist} \times (m-1) \right\rceil \text{ else} \end{cases}$$

Note that (m-1) is the maximum difference between the indices of two elements from *Colors* and $\lceil x \rceil$denotes the smallest integer i with $i \geq x$. Finally, we define the function *visualize:Classes* → *Colors* mapping classes to colors as follows:

$$visualize(c_i) = col_{map(i)}$$

Several color scales satisfying these requirements have been proposed [11]. These color scales are appropriate when a total or partial order is defined for the classes. For the purpose of comparability of the results, the experiments reported in this paper have been performed on several datasets where no semantics about the classes is known. If no order of the classes is given, we do not need the first requirement to preserve the order of the attribute values. Furthermore, the second requirement is weakened such that each pair of colors col_i and col_j is perceived as being different, i.e.

$$dist(col_i, col_j) = \begin{cases} 0 \text{ if } i=j \\ 1 \text{ else} \end{cases}$$

The amount of training data that can be visualized at one time is approximately determined by the product of the number of attributes and the number of data objects. For example, 2.000 data objects with 50 attributes can be represented in a 374x374 window and 10.000 objects with 20 attributes fit into a 516x516 window.

We have developed a color scale for class labels based on the *HSI color model* [7] , a variation of the HSV model. The HSI model represents each color by a triple (hue, saturation, intensity). In our experiments, we observed the most distinctly perceived colors for the following parameter settings: For *col 1* we set hue = 2.5 and intensity = saturation = 1.0, for *col m* we set hue = 0.5 and intensity = saturation = 1.0, and all other colors were obtained by partitioning the hue scale into equidistant intervals.

Our approach of visualizing the training data also considers attributes having a low number of distinct values. In that case, there are many objects sharing the same attribute value and their relative order is not uniquely defined. Depending on the chosen order, we might create homogeneous (with respect to the class label) areas within the same attribute value. To avoid the creation of artificial homogeneous areas, we use the technique of *shuffling*: for a set of data objects sharing the same attribute value the required order for the arrangement is determined randomly, i.e. their class labels are distributed randomly.

3 Perception-based classification

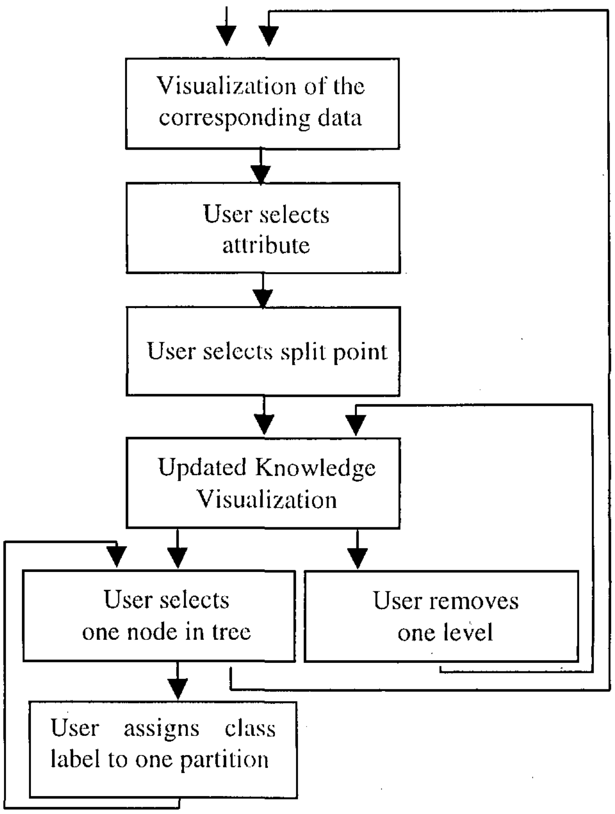
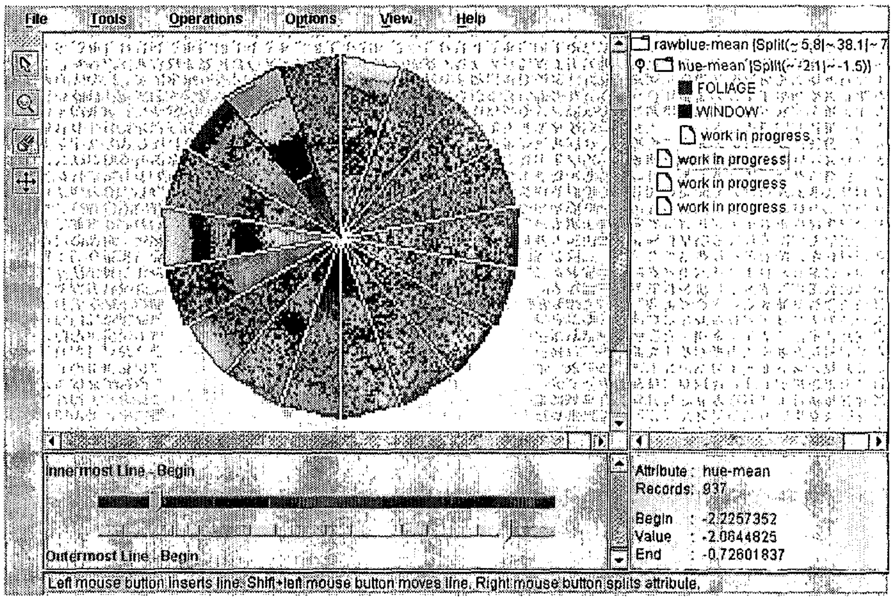

Visualization of the corresponding data
User selects attribute
User selects split point
Updated Knowledge Visualization
User selects one node in tree
User removes one level
User assigns class label to one partition

Figure 2. A model for interactive classification



rawblue-mean [Split(~5,8|~38.1|~7
hue-mean [Split(~-2.1|~-1.5)]
FOLIAGE
WINDOW
work in progress
work in progress
work in progress
work in progress

Innermost Line - Begin
Outermost Line - Begin

Attribute : hue-mean
Records: 937
Begin : -2.2257352
Value : -2.0644825
End : -0.72601837

Left mouse button inserts line. Shift+left mouse button moves line. Right mouse button splits attribute.

Figure 3. A Screen Shot of the PBC system

The described visualization of the data is the basis of our approach of interactive classification. Figure 2 depicts our model for interactive decision tree construction.

Initially, the complete training set is visualized in the *Data Interaction Window* together with an empty decision tree in the *Knowledge Interaction Window*. The user selects a splitting attribute and an arbitrary number of split points. Then the current decision tree in the Knowledge Interaction Window is expanded. If the user does not want to remove a level of the decision tree, he selects a node of the decision tree. Either he assigns a class label to this node (which yields a leaf node) or he requests the visualization of the training data corresponding to this node. As depicted in figure 3, the latter case leads to a new visualization of every attribute except the ones used for splitting criteria on the same path from the root. Thus the user returns to the start of the interaction loop. The interaction is finished when a class has been assigned to each leaf of the decision tree.

Interactive Selection of Split Points

The interactive selection of split points consists of two steps: (1) selecting splitlines and (2) selecting a split point on each of the selected splitlines.

First, by clicking on any pixel in the chosen segment, the user selects a *splitline* which is one of the lines (orthogonal to the segment halving line) upon which the pixels are arranged. Then by the system this splitline is replaced with an animated line on which alternatively black and white strips move along. Since the colors black and white are not used for the mapping of the classes, the brushed splitline is well perceptible. In a separate area, the pixels of the selected splitline are redrawn in a magnified fashion which enables the user to set the exact split point. Note that the separation of two different colors is not the only criteria for determining the exact split point. If not all attribute values on the splitline are distinct, the same attribute values may belong to objects of different classes. In this case, setting a split point between two differently colored pixels would not be reasonable. Hence we provide feedback to the user in both the basis data visualization and the separate splitline area, such that the attribute value of the pixel at the position of the mouse pointer appear in a subwindow. Figure 3 illustrates the visualization support for the selection of a splitline and an exact split point.

Splitting strategy

Our interactive approach overcomes the limitations of binary splits in attributes with a continuous domain. This additional flexibility rises the question about an appropriate splitting strategy. In our experiments, we observed the best results in terms of accuracy and tree size if the choice of the splitting attribute is based on the strategy described below. The strategy has four options and the first of them which is applicable in the current visualization should be chosen. We will use the term *partition* for a coherent region of attribute values in the splitting attribute that the user intends to separate by split points.

1) *Best Pure Partitions (BPP)*. First choose the segment with the largest pure partitions. A partition is called *pure* if the user decides to label this partition with the most frequent class. This decision leads to leaf nodes in the decision tree, thus reducing the size of data which is not classified.

2) *Largest Cluster Partitioning (LCP)*. If no pure partition is perceptible, the segment with the largest cluster clearly dominant in one color should be chosen. In contrast to a pure partition, such a cluster will not be labeled by the most frequent class.

3) *Best Complete Partitioning (BCP).* If a choice upon BPP or LCP fails, the segment should be chosen that contains the most pixels that can be divided into partitions where each has one clearly dominant color.

4) *Different Distribution Partitioning (DDP).* If none of the above options applies, choose the segment where different distributions can be best separated through partitioning.

After an attribute is chosen the split points have to be set. If the choice follows BPP or LCP, additional split points in the remaining partition should be set if it leads to a separation of clusters or of different distributions. Thus, more inherent information of the splitting attribute is used for deriving the decision tree. Note that the splitting attribute will not reappear in lower nodes of the same path.

4 Experimental evaluation

In comparison to algorithmic decision tree classifiers, the process of interactive classification reveals additional insights into the data. To illustrate this advantage, in this section we discuss an example of two consecutive steps in the tree construction phase. Furthermore, we compare our classifier with popular algorithmic classifiers in terms of accuracy and tree size.

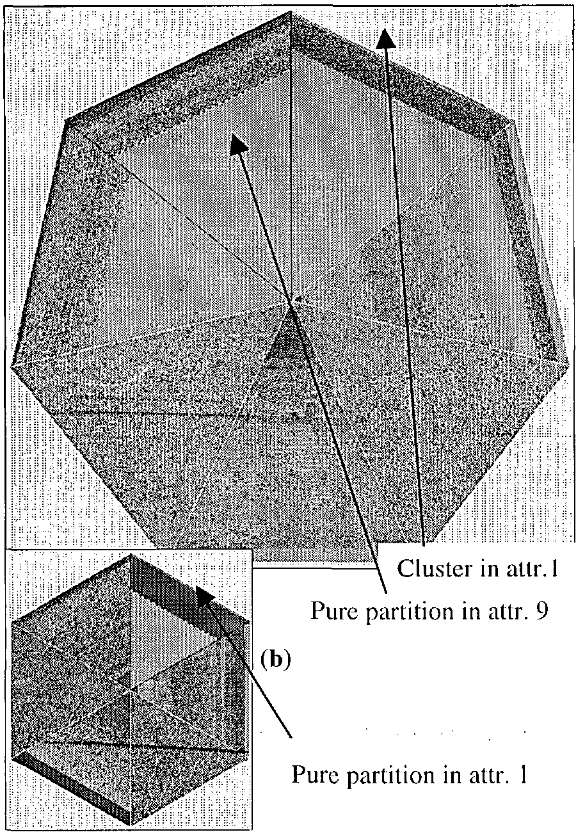


Figure 4. Visualization of the Shuttle data before (a) and after a split (b)

Attributes 1 and 9 are obvious candidates for splitting. According to 'Best Pure Partitions', attribute 9 should be chosen because in contrast to the larger cluster in the segment of attribute 1, the split leads to a pure partition. Note that the non-homogeneity of the cluster in attribute 1 can only be perceived in the color representation. The pure partition can be assigned to the class of its only color. The visualization of the remaining partition has to be examined in a further step. This is shown in figure 4(b) representing the data objects visualized in figure 4(a) except for all objects belonging to the pure partition in attribute 9. Attribute 9 is not visualized any more because it was already used as a splitting attribute on this path of the decision tree. One effect of our visual approach becomes very clear in this example the removal of some training objects from the segment of the splitting attribute may yield the removal of objects from another segment which make a partition of this segment impure. For example, the cluster in attribute 1 (figure 4(a)) becomes a pure partition after the split (figure 4(b)).

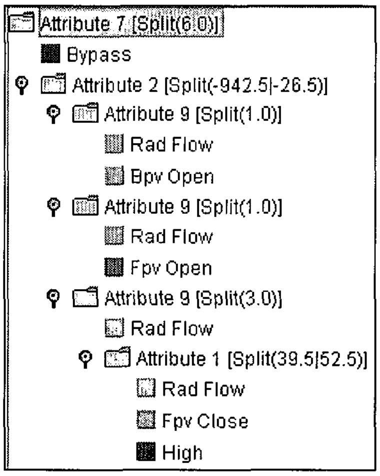
We used the accuracy and the tree size (total number of nodes) as quantitative measures to compare PBC with well-known algorithmic approaches. We used the tree size besides accuracy since small trees are easier to understand and we consider understandability of the discovered knowledge to be a major goal. For the comparison, we used three datasets from the Statlog database [13] for which the accuracy and the tree size of many algorithms is known [4] . The Satimage, Segment and Shuttle datasets were chosen because all of their attributes are numeric. We performed the experiments as suggested in the dataset descriptions. As comparison partners we chose the popular decision tree classifiers CART and C4 from the IND package [14] as well as the recently proposed SPRINT [3] and CLOUDS [4] classifiers. The results of CLOUDS were produced with the SSE/DM method.

Accuracy	CART	C4	SPRINT	CLOUDS	**PBC**
Satimage	85.3	85.2	86.3	85.9	83.5
Segment	94.9	95.9	94.6	94.7	94.8
Shuttle	99.9	99.9	99.9	99.9	99.9

Tree Size	CART	C4	SPRINT	CLOUDS	**PBC**
Satimage	90	563	159	135	60
Segment	52	102	18.6	55.2	39.5
Shuttle	27	57	29	41	14.6

Table 1, Table 2: Accuracy and tree size of PBC and algorithmic approaches

Table 1 depicts the accuracy of PBC and the algorithmic approaches, table 2 their tree sizes. Our performance evaluation demonstrates that the approach of interactive visual classification yields an accuracy similar to the accuracy obtained by well-known algorithms. PBC significantly reduces the tree size and thus obtains decision trees which are much better understandable.



```
Attribute 7 [Split(6.0)]
  Bypass
  Attribute 2 [Split(-942.5|-26.5)]
    Attribute 9 [Split(1.0)]
      Rad Flow
      Bpv Open
    Attribute 9 [Split(1.0)]
      Rad Flow
      Fpv Open
    Attribute 9 [Split(3.0)]
      Rad Flow
      Attribute 1 [Split(39.5|52.5)]
        Rad Flow
        Fpv Close
        High
```

Figure 5. A decision tree for the Shuttle dataset

To illustrate this advantage, figure 5 shows a decision tree for the Shuttle dataset constructed with the PCB system. Attribute 7 represents the root of the tree with one split point at 6.0. The following two nodes are the left (attribute 7 < 6.0) and right son of this root. The left son of the root is already assigned to a class (Bypass). The colored square besides the class label depicts the color representing the class. We observe that the nodes with the splitting attributes 1 and 2 both have two split points yielding a 3-ary decision tree that cannot be generated by the algorithmic approaches.

5 Conclusion

In this paper, we introduced a fully interactive method for decision tree construction based on a multidimensional visualization technique and appropriate interaction capabilities. Thus knowledge can be transfered in both directions. On one hand, domain knowledge of an expert can be profitably included in the tree construction phase. On the other hand, after going through the interactive construction of a decision tree, the user has a much deeper understanding of the data than just knowing the decision tree generated by an arbitrary algorithm. Our approach has several additional advantages compared to algorithmic approaches. First, the user may set an arbitrary number of split points which can reduce the tree size in comparison to binary decision trees that are generated by most state of the art algorithms. Furthermore, in contrast to the greedy search performed by algorithmic approaches, the user can backtrack to any node of the tree when a subtree turns out to be suboptimal. We conducted an experimental evaluation on several popular datasets. We found that even users with no a priori knowledge of the training data construct a decision tree that has a similar accuracy and a significantly smaller tree size compared to algorithmic approaches.

In our future work, we will improve the scalability with respect to the maximum amount of data that can be processed. Furthermore, we plan to extend our PBC system by features of algorithmic approaches and we want to explore methods of integrating PBC with a database management system.

6 References

[1] Ankerst, M., Breunig M, Kriegel H.-P. and Sander J.:"OPTICS: Ordering Points To Identify the Clustering Structure", in *Proc. ACM SIGMOD* '99, Int. Conf. on Management of Data, Philadelphia, PA, 1999.

[2] Ankerst M., Keim D. A. and Kriegel H.-P.: "Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets", *Proc. Visualization '96*, Hot Topic Session, San Francisco, CA, 1996.

[3] Agrawal R., Mehta M. and Shafer J.C.: "SPRINT: A Scalable Parallel Classifier for Data Mining", in *Proc. VLDB '96*, 22nd Intl. Conf. on Very Large Databases, Bombay, India, 1996, pp. 544-555.

[4] Alsabti K., Ranka S. and Singh V.: "CLOUDS: A Decision Tree Classifier for Large Datasets", in *Proc. KDD '98*, 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York City, 1998, pp. 2-8.

[5] Berchtold S., Jagadish H.V. and Ross K.A.: "Independence Diagrams: A Technique for Visual Data Mining", in *Proc. KDD '98*, 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York City, 1998, pp. 139-143..

[6] Fayyad U.M. and Irani K.: "Multi-interval Discretization of Continuous-Valued Attributes for Classification Learning", in *Proc. IJCAI '93*, Int. Joint Conf. on Artificial Intelligence, 1993.

[7] Keim D.A.: "Visual Support for Query Specification and Data Mining", PhD Thesis, University of Munich, Germany, 1994.

[8] Keim D. A.: "Visual Database Exploration Techniques", Proc. Tutorial Int. Conf. on Knowledge Discovery & Data Mining, Newport Beach, CA, 1997. http://www.informatik.uni-halle.de/~keim/PS/KDD97.pdf

[9] Keim D. A., Kriegel H.-P. and Ankerst M.: "Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data", in *Proc. Visualization '95*, Atlanta, GA, 1995, pp. 279-286.

[10] Kohavi R. and Sommerfield D.: "Targeting Business Users with Decision Table Classifiers", in *Proc. KDD '98*, 4th Intl. Conf. on Knowledge Discovery and Data Mining, New York City, 1998, pp. 249-253.

[11] Levkowitz H.: "Perceptual Steps Along Pseudo-Color Scales", International Journal of Imaging Systems and Technology, 7:97-101, 1996.

[12] Mitchell T.M.: "Machine Learning", McGraw Hill, 1997.

[13] Michie D., Spiegelhalter D.J. and Taylor C.C.: "Machine Learning, Neural and Statistical Classification", Ellis Horwood, 1994.

[14] NASA Ames Research Center: "Introduction to IND Version 2.1", 1992.

[15] http://www.spss.com/.