Categorization and Evaluation of Data Mining Techniques

Claudio J. Meneses[1,2], Georges G. Grinstein[1]

[1]*Institute for Visualization and Perception Research*
Department of Computer Science
University of Massachusetts at Lowell
One University Avenue, Lowell, MA 01854, USA
{cmeneses, grinstei}@cs.uml.edu
[2]Departamento de Ingenieria de Sistemas y Computacion
Universidad Catolica del Norte
Av. Angamos 0610, Antofagasta, Chile

Abstract

A fundamental issue in the application of data mining algorithms to solve problems of real life is to know ahead of the time the usability of the algorithm for the class of problems being considered. In other words, we would like to know, before starting the KDD process for a particular problem P, with its features belonging to a type C_j of problems or tasks, how well a specific data mining algorithm A_i would perform in solving P. In this paper, we survey the main approaches to categorize and evaluate data mining techniques. This will help to clarify the relationship that can exist between a particular data mining algorithm and the type of tasks or problems for which it is best suited. Perhaps the most important conclusion we show is that no single technique provides the best performance for all types of tasks, and that a multi-strategy approach is needed to deal with real complex problems. Categorizing data mining techniques will guide the user, prior the start of the KDD process or during the data mining phase, in the selection of the best subset of techniques to resolve a problem or data mining task.

1 Background

Although data mining is becoming an important field in both the academic and business community, mainly because of some results gotten (i.e. novel discoveries, savings, increment of sales, etc.), the current KDD process presents several problems. Firstly, the pre-processing stage is time and budget consuming, because commonly the input data set owns features (noisy, null values), which require a transformation and cleaning step, in order to match the input format and assumptions of the data mining algorithms being considered. Secondly, the data mining step involves typically the use of one or more inductive learning algorithms, requiring that the user iterates over several steps, especially when the results are not good enough, either in terms of performance (accuracy) or understanding of the rules generated for the model.

An additional problem is the application of data mining techniques. Given the inter-disciplinary nature of the data mining field, there are a wide variety of techniques and software tools available to explore or mine a data set. However, which of these approaches or techniques are better for which type of tasks and under what conditions that relationship remains valid, is a research question, whose answer currently is not well understood. In this context, we want to clarify the relationship that would exist between a particular algorithm and the type of task or domain for which it would perform better than others do. In order words, we would like to explore the issues involved in the answer to the question: which types of functions are best data mined with which techniques? The answer to this question should conduct to develop more intelligent KDD environments, which permit to the user manipulates parameters in order to get better analysis, or perform the knowledge discovery process with the best subset of techniques, regarding the features of the problem and the type of task being sought.

In this paper, we survey and discuss the issues involved in the categorization and evaluation of data mining algorithms. The paper is organized as follows. In sections 2 and 3, we establish the background and the state of the art in data mining approaches and types of data mining tasks, respectively. In section 4, we present and discuss the main results of a survey to determine the relationship between different types of problems and certain data mining (machine learning) algorithms. In section 5, we present a set of criteria defined to assess the quality of inductive learning algorithms, and a ranking of five well know data mining algorithms based on this assessment. In section 6, we summary two approaches to evaluate the performance of classification algorithms: the STATLOG project, which uses only one property to evaluate the performance of data mining algorithms, and the DEA-model, which is multi-criteria based metric for the evaluation of data mining algorithms. Finally, we summary and discuss the main points of each method of categorization and evaluation presented, and our current research related to this topic.

2 Approaches for Data Mining

The variety of methods and techniques used currently for extracting knowledge from large databases comes from diverse disciplines such as statistics, machine learning (AI), databases, visualization, and knowledge representation for expert systems. Thus, according to the methodology used for researching and applications, data mining approaches can be classified in the following categories: Statistical, Machine Learning, Database-oriented, Visualization, and Knowledge representation approaches.

2.1 Statistical Approaches

In the past decades, statisticians have developed methods used to evaluate hypotheses and determine whether differences can be ascribed to random chance. Formal statistical theory supports models of data and methods for prediction. However, classical statistical models, typically linear models, assume small and clean data, and they break down when the data set becomes massive. In this context, Bayesian inference is the most used statistical method for knowledge discovery. Three Bayesian methods used for data mining problems are *Naïve Bayes classifier* (Mitchell [26]), *Autoclass* (Cheeseman & Stutz [6]), and *Bayesian networks* (Heckerman [15]).

2.2 Machine Learning Approaches

Under this category, we include the many prediction methods developed by the computer science community, to specify some interesting model, and then enumerate and search through the possibilities. Some of the most common machine learning algorithms used to mine knowledge from data sets, are *k-nearest neighbor* (Dasarathy [10]), *Decision Trees* (Quinlan [34], [35]), *Neural Networks* (Hertz [16]), and *Genetic Algorithms* (Golberg [13]).

Transactions on Information and Communications Technologies vol 19 © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

2.3 Database-oriented Approaches

Databases are the natural repository of the massive numbers of transactions involved in the business and scientific community. The development of data warehousing and OLAP tools for the analysis of big databases, have encouraged the arising of other methods to mine knowledge from different types of databases. Some of these techniques, are *Attribute-Oriented Induction* (Han & Fu [14]) and *Association Rules* (Agrawal et al. [2]).

2.4 Visualization Approaches

Data visualization techniques are becoming very useful methods to discover patterns in data sets, because they impact directly the human visual system, currently the most powerful pattern recognizer and discoverer. There is a wide variety of techniques, which may be used in the several stages of the KDD process: in the pre-processing stage, to get a rough feeling of the features of the data set; in the data mining stage, to discover patterns, such as clusters of items, correlations or dependencies among attributes, or to visualize the model produced by the data mining algorithm, in order to have a better understanding how the responses are generated by the model. Among the most common visualization techniques to mine knowledge from data, are:

- *2D and 3D scatter-plots*, and *Scatter-Plot Matrix* (Cleveland [9]), available in some visualization tools, such as XmdvTool (Ward [42]).
- *Multi-dimensional visualization techniques* are able to display a larger number of dimensions (attributes) simultaneously using some mapping from an N-dimensional space (N>3) to a 2-dimensional space. Some of the techniques in this category are: Chernoff faces (Chernoff [7]), Stick Figures (Pickett & Grinstein [30]), Parallel Coordinates (Inselberg [18), Recursive Patterns (Keim et al. [19]), Circle Segments (Ankerst et al. [4]), RadViz (Hoffman et al. [17]).
- *Hierarchical techniques* subdivide the N-dimensional space and present the subspaces in a hierarchical fashion. Well-known representatives of this category are: Dimensional Stacking (Leblanc et al. [21]), Worlds within Worlds (Beshers & Feiner [5]), Treemaps (Shneiderman [41]), Cone Trees (Robertson et al. [39]), and InfoCube (Rekimoto & Green [36]).

2.5 Knowledge Representation Approaches

The representation of the knowledge extracted by a data mining system typically involves a tradeoff between expressive power and computational complexity. Among the most common forms of knowledge representation in a data mining system, are:

- *Propositional-like representations*, for example, production rules (Quinlan [31]), decision lists(Rivest 38]), (Ringland & Duce [37]), and ripple-down rule sets (Kivinen et al. [20]).
- *First Order Logic* (FOL). An example of an inductive logic programming system that uses FOL to represent the knowledge, is the FOIL system (Quinlan [33]).
- *Structured Representations* provide a more comprehensible representation (although not more powerful) than FOL, by explicitly stating subtype relationships among objects. Two examples of structured representations are Semantic Networks (Minsky [25]), and Frames and Schemata (Lenat [22]).

3 Types of Data Mining Problems

Although data mining has been used for a wide variety of tasks, from a "high level" point of view, we can say that the goal of a data mining task can be either *prediction* or *description*. Prediction involves using some attributes of a database to predict the (unknown) future values of another variable. Description, in turn, focuses on finding human-interpretable patterns, which describe the data, in order to get insight of the data, before trying to predict anything. Both goals, prediction and description, are really complementary and they use some of the following primary data mining tasks (Fayyad et al. [12]): classification, regression, clustering, summarization, dependency modeling, link analysis, and sequence analysis. In summary, according to the high-level data mining goal, the specific data mining tasks that the user wants to perform, and the characteristics of the data set being mined, we can formulate a decision-making process to determine the specific data mining method to be used which match the goal of the data mining task, as illustrated in figure 1.

Weiss & Indurkhya [43] propose a similar categorization of types of data mining problems. They establish two general categories: (a) prediction, and (b) knowledge discovery, where knowledge discovery problems usually describe a stage prior to prediction, where information is insufficient for prediction. They include *classification*, *regression*, and *time series* (measurements are taken over time for the same features) as prediction problems, and *deviation detection*, *database segmentation*, *clustering*, *association rules*, *summarization*, *visualization*, and *text mining* as knowledge discovery problems.

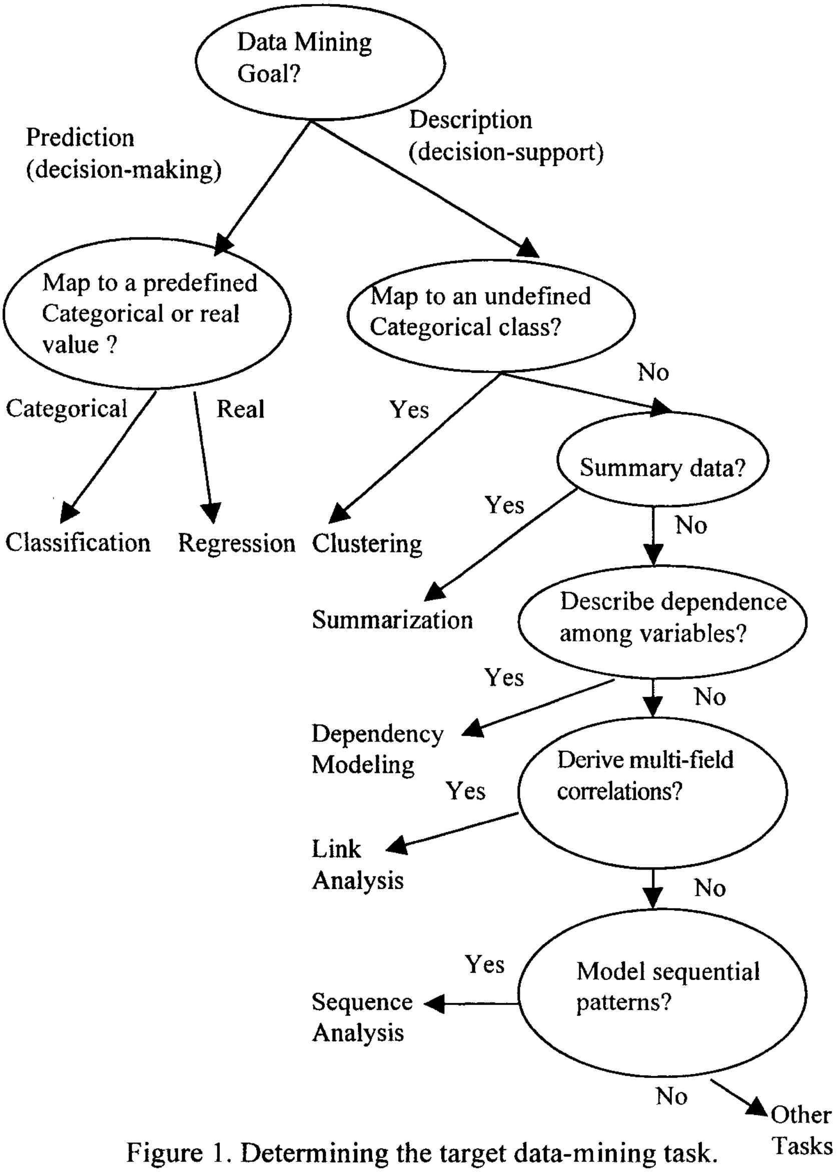
Data Mining Goal?
Prediction (decision-making)
Description (decision-support)
Map to a predefined Categorical or real value ?
Map to an undefined Categorical class?
Categorical
Real
Yes
No
Classification
Regression
Clustering
Summary data?
Yes
No
Summarization
Describe dependence among variables?
Yes
No
Dependency Modeling
Derive multi-field correlations?
Yes
No
Link Analysis
Model sequential patterns?
Yes
No
Sequence Analysis
Other Tasks

Figure 1. Determining the target data-mining task.

4 Learning Algorithms versus Types of Problems

A fundamental issue in the application of data mining algorithms to solve real problems, is to know, beforehand, the usefulness of the algorithm for the class of problems being considered. In other words, we would like to know, before starting the KDD process using a specific data mining algorithm A_i, how well it may perform in solving a specific problem P, which, given its features, belong to the type C_i of problems or tasks.

Moustakis et al. [27] performed a survey among specialists of the machine learning community, about the usefulness of certain machine learning techniques to solve different types of problems. They considered the set $A=\{A_1, A_2, A_3, A_4, A_5, A_6\}$ of machine learning techniques, and the set $C=\{C_1, C_2, C_3\}$ of types of tasks, where:

A_1: k-nearest neighbor
A_2: Decision trees
A_3: Association rules
A_4: Neural networks
A_5: Genetic algorithms
A_6: Inductive logic programming

and

C_1: Classification
C_2: Problem-solving
C_3: Knowledge engineering

Inductive Logic Programming (A_6) is referred to the approach that uses First Order Logic (FOL) to represent the learned knowledge. The aim is to construct an FOL-program that, together with the domain knowledge, has the training set as its logical consequence. Inductive Logic Programming (ILP) algorithms learn a set of rules containing variables, called first-order Horn clauses. Two well-known approaches for ILP are the Sequential Covering algorithms, for example the CN2 program (Clark & Niblett [8]), and the FOIL program (Quinlan [34]).

Problem solving is referred to a class of problems where, given a *goal* and a set of means for achieving the goal, there is an exploratory process (or searching process) to reach the goal. Within of this type of problems are Toy Problems (e.g. the 8-puzzle, the 8-queens, crypt-arithmetic, the vacuum world) and Real Problems (e.g. route finding, robot navigation, VLSI layout, touring and travelling salesperson problem), (Russell & Norvig [40]).

Knowledge Engineering problems have to deal with the process of building a *knowledge base*, which typically involves to investigate a particular domain, determine what concepts are important in that domain, and create a formal representation of the objects and relations in the domain.

The results of this survey are summarized in the figure 2, which shows the grade of usefulness of each technique in the set A in performing a type of task included in the set C.

According to this survey, neural networks algorithms (A_4) perform better for classification task than genetic algorithms (A_5), which in turn seem to be more appropriate for problem solving tasks. Also, inductive logic programming (A_6) clearly shows advantages to perform knowledge engineering tasks over the other algorithms considered. In addition, decision trees (A_2), k-nearest neighbor (A_1), and association rules (A_3) algorithms, seem, in general, to perform better in classification problems than in problem-solving and knowledge engineering tasks.

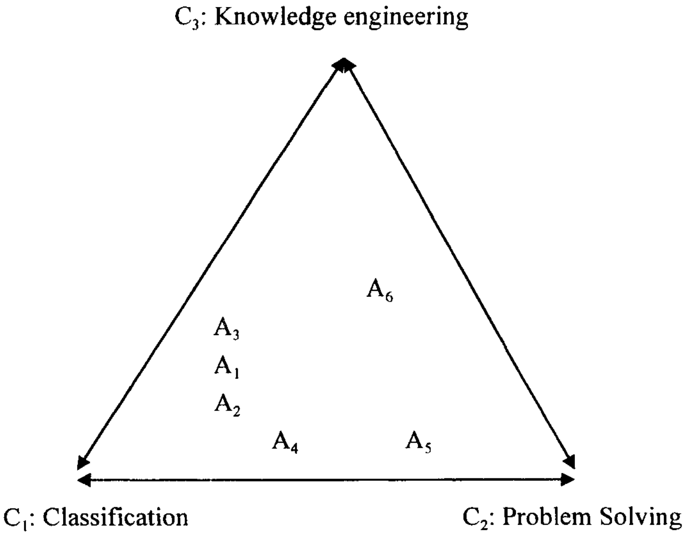

Figure 2. Machine learning algorithms versus types of task.

Two important conclusions that can be derived from this empirical study are:

1. There is not a unique technique, which can have the best performance for all types of tasks that usually are involved in solving a real life problem. In some specific problems, some algorithm A_i can perform better than other algorithm A_j, but in other specific problems, A_j may give better analysis than A_i , even for the same data set.
2. A general-purpose, useful and robust KDD environment should support a variety of data mining techniques (hybrid approach), which are essential to deal with problems of real life, which usually require different types of tasks in the several stages of the data analysis process.

5 Assessing the Quality of Inductive Learning Algorithms

Given the variety of data mining algorithms, an important factor is to establish a set of key features for which a data-mining algorithm should be assessed, in order to select the best possible algorithm for a particular type of problem. For example, some data mining algorithms are able to handle larger input data set than others; other data mining algorithms build models which are easier to understand and derive rules from them; other data mining algorithms demands less computational resources (CPU time, memory space) than others, etc. What and which are a good set of features to assess the quality of a particular data mining algorithm in solving a particular class of problem or task?, and what is the relative importance (in numerical terminology, the weight) of each feature in the selection of a data mining algorithm to resolve a class of problem ?

Adriaans & Zantinge [1] define a set of features $F=\{f_1,...,f_{11}\}$, to evaluate the quality of a data mining algorithm. They logically ordered these features in four groups: D_1, D_2, D_3, and D_4, where

$D_1=\{f_1, f_2, f_3, f_4\}$: Characteristics of the input
$D_2=\{f_5, f_6, f_7\}$: Characteristics of the output
$D_3=\{f_8, f_9\}$: Efficiency (performance) for learning
$D_4=\{f_{10}, f_{11}\}$: Efficiency for applying the model

f_1: Ability to handle large number of records
f_2: Ability to handle large number of attributes
f_3: Ability to handle numeric attributes
f_4: Ability to handle strings
f_5: Ability to learn transparent rules
f_6: Ability to learn incrementally
f_7: Ability to estimate statistical significance
f_8: Disk load in the learning phase
f_9: CPU load in the learning phase
f_{10}: Disk load in the application phase
f_{11}: CPU load in the application phase

They assessed the quality of each feature defined in the set F, for each machine learning algorithm in the set A'={A_1, A_2, A_3, A_4, A_5} defined in section 4. In this case, the quality of each feature f_i can assume one categorical value in the set R={r_1, r_2, r_3}, where

r_1: Poor quality,
r_2: Average quality,
r_3: Good quality

The assessment made by these authors is summarized in figures 3a, 3b, and 3c. They considered the set A'={A_1, A_2, A_3, A_4, A_5} of machine learning algorithms, where

A_1: k-nearest neighbor,
A_2: Decision trees,
A_3: Association rules,
A_4: Neural networks, and
A_5: Genetic algorithms.

Each machine learning algorithm in A' is evaluated by its quality in each feature f_i. In figures 3a, 3b, and 3c, each feature f_i is located in a corner of rectangular area. The quality categories (*good*, *average*, *poor*) are translate to geometric distance to the corresponding feature f_i being considered, such that:

- If an algorithm A_i performs *good* in the feature f_i, then it is located close to the corner of f_i,

- If an algorithm A_i performs *average* in the feature f_i, then it is located on the diagonal (dashed line) of the rectangular area for f_i, and
- If an algorithm A_i performs *poor* in the feature f_i, then it is located far to the corner of f_i.

f_1: Ability to handle large number of records
f_2: Ability to handle large number of attributes
f_3: Ability to handle numeric attributes
f_4: Ability to handle strings

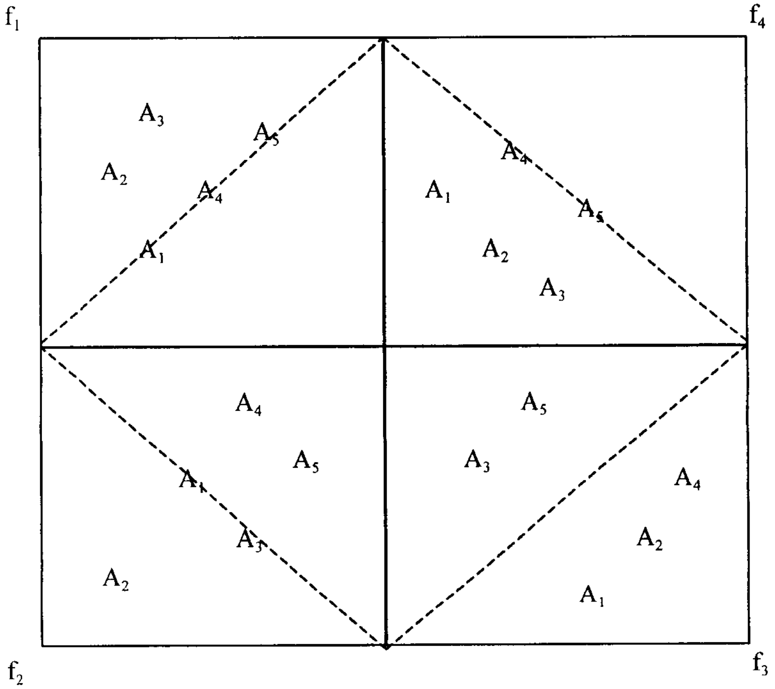

Figure 3a. Quality of DM algorithms based on the characteristics of the input.

The figure 3a shows the judgment of the quality of the algorithms in the set A' in features associated to the input. Thus, decision trees (A_2) and association rules (A_3) perform *good* in handling larger number of records than the others algorithms (A_1, A_4, A_5), which have an *average* performance. Considering the ability to handle large number of attributes (f_2), decision trees (A_2) performs *good*, but neural networks (A_4) and genetic algorithms (A_5) perform *poor*, because their efficiency deteriorates considerably as the number of attributes becomes large in the input data set. Using the type of attributes as criteria to select an algorithm, we observe that k-nearest neighbor (A_1), decision trees (A_2), and neural networks (A_4) perform *good* in handling numeric attributes, and association rules (A_3) and genetic algorithms (A_5) perform *poor* in this aspect; however, when the attributes are strings, a better selection may be genetic algorithms and neural networks, which perform better than the other algorithms considered.

f_5: Ability to learn transparent rules
f_6: Ability to learn incrementally
f_7: Ability to estimate statistical significance

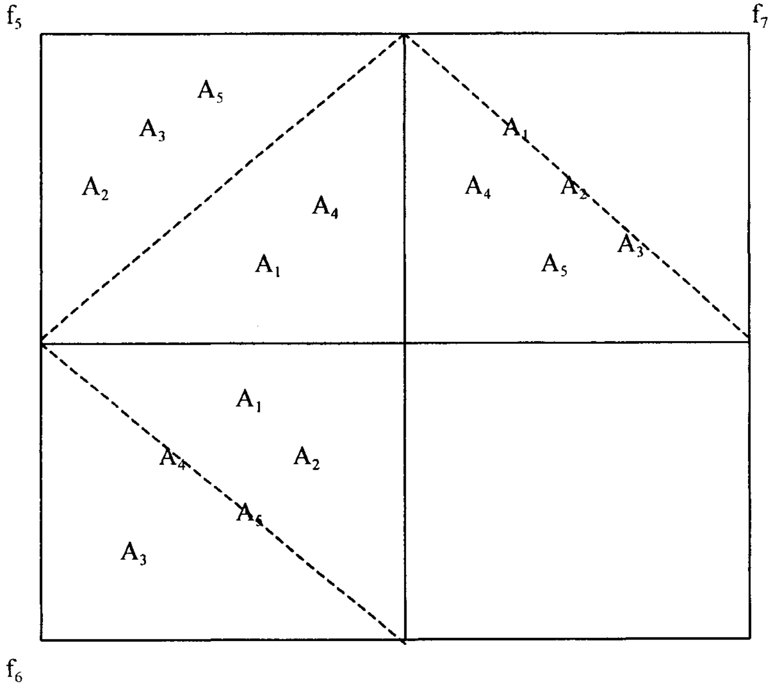

Figure 3b. Quality of DM algorithms based on the Characteristics of the output

In figure 3b, the criteria to select an algorithm are based on the characteristics of the output produced by the algorithm. K-nearest neighbor (A_1) and neural network (A_4) perform *poor* in learning transparent rules, because, although they provide a yes/no answer, no explanation is provided about how the response is reached. Considering the ability to learn incrementally, which is very important with large data sets, because the inductive process does not need to re-start again when new examples are added, association rules (A_3) perform good in this aspect, but k-nearest neighbor (A_1) and decision trees (A_2) are inappropriate when we want to incorporate new cases to the model. If we consider the ability of the algorithm to estimate the statistical significance of the results, neural networks (A4) and genetic algorithms (A5) perform *poor* in this aspect, because it is difficult to evaluate their results from a statistical point of view; a better choice in this aspect is k-nearest neighbor, decision trees, or association rules algorithm.

f_8: Disk load in the learning phase
f_9: CPU load in the learning phase
f_{10}: Disk load in the application phase
f_{11}: CPU load in the application phase

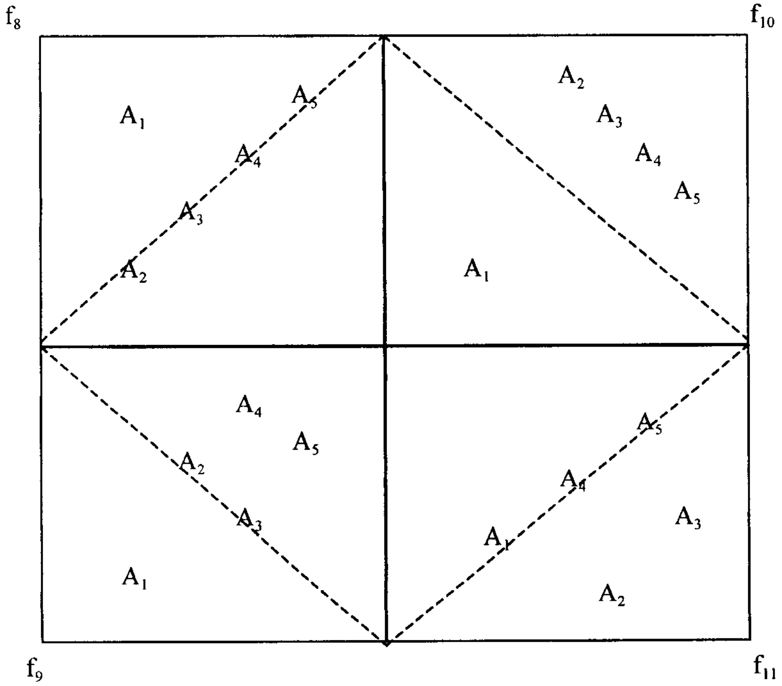

Figure 3c. Quality of DM algorithms based on the performance

Finally, the figure 3c show the quality of the algorithms based on the efficiency in the learning phase and application phase. Although, k-nearest neighbor has a *good* disk/CPU load performance in the learning phase, it performs *poor* in the application phase. In contrast, decision trees (A_2) and association rules (A_3) have a *good* disk/CPU load performance in the application phase, and an *average* rate in the learning phase, which, together with their good scores (in average) to handle input/output, is one of the reasons why they are widely used in data mining applications.

Table 1 summaries the results shown in figures 3a, 3b, and 3c, for algorithms A_1-A_5 and features f_1-f_{11}, using the categorical values *poor*, *average*, and *good* to assess the quality of each data mining algorithm in performing each feature considered.

Table 1. Summary of the quality of data mining algorithms.

Ability to handle:	A_1: k-NN	A_2:Decision Trees	A_3: Association Rules	A_4: Neural Networks	A_5: Genetic Algorithms
f_1: large number Of records	Average	Good	Good	Average	Average
f_2: large number Of attributes	Average	Good	Average	Poor	Poor
f_3: numeric attributes	Good	Good	Poor	Good	Poor
f_4: string attributes	Poor	Poor	Poor	Average	Average
f_5: transparent rules	Poor	Good	Good	Poor	Good
f_6: incremental learning	Poor	Poor	Good	Average	Average
f_7: statistical significance	Average	Average	Average	Poor	Poor
f_8: disk load learning	Good	Average	Average	Average	Average
f_9: CPU load learning	Good	Average	Average	Poor	Average
f_{10}: disk load application	Poor	Good	Good	Good	Good
f_{11}: CPU load application	Average	Good	Good	Average	Average

Table 2 shows several rankings of the algorithms A_1-A_5, based on the evaluation made in table 1. Columns 2-5 show rankings based in subsets of features. Column 6 shows the ranking based on all features defined in table 1.

Table 2: Ranking of the algorithms A1-A5 based on the assessment showed in Table 1.

Machine learning algorithms	f_1-f_4: Characteristics of the input	f_5-f_7: Characteristics of the output	f_8-f_9: Performance in the learning phase	f_{10}-f_{11}: Performance in the application phase	f_1-f_{11}: Global Characteristics
A_1: k-NN	2	4	1	5	3
A_2: D-Trees	1	2	2	1	1
A_3: A-Rules	4	1	2	1	2
A_4: N-Nets	2	4	5	3	5
A_5: Gen-Alg	5	2	2	3	3

6 Evaluating Data Mining Algorithms

Establishing an evaluation of data mining algorithms has the main obstacle of defining objective criteria to assess them in a fair form. Fayyad et al. [11] define that the patterns identified by a data mining algorithm should be "valid, novel, potentially useful, and understandable", which are features that are difficult of quantifying and using to rank data mining algorithms. For example, "novel and useful" are subjective qualities (not measurable); "understandable" is a quality which is difficult to measure and clearly depends of the domain-specific background knowledge available; "valid" is the most measurable of these qualities, because there are some criteria that can be used to measure the *validity* of the result of a data mining algorithm, such as the predictive accuracy rate. In this section, we summary two approaches and their results, to evaluate the performance of inductive learning algorithms.

6.1 The STATLOG project

The STATLOG project (Michie et al [24]) was concerned with a comparative study of the performance of different machine learning, neural, and statistical classification algorithms, on a wide range of data sets. They evaluated 23 algorithms, classified in the following categories:

- *Decision Trees and Rule Based Classifiers*: NewID, AC^2, Cal5, CN2, C4.5, CART, IndCART, Bayes Tree, and ITrule.

- *Statistical Classifiers*: ALLOC80, Bayes Tree, CASTLE, k-NN, Linear Discriminant, Quadratic Discriminant, Logistic Discriminant, Naïve Bayes, and SMART.
- *Neural Network Classifiers*: Back-propagation, Cascade, DIPOL92, LVQ, Radial Basis Function (RBF), and Kohonen.

The above algorithms were evaluated using 22 different data sets, which may be classified as:

- *Credit Data Sets*: Credit Management (Cred.Man) and Australian Credit (Cr.Aust).
- *Image Data Sets*: Handwritten digits (Dig44), Karhunen-Loeve digits (KL), Vehicle silhouettes (Vehicle), Letter Recognition (Letter), Chromosomes (Chrom), Landsat satellite image (SatIm), Image Segmentation (Segm), and Cut.
- *Data Sets with Costs*: Head Injury (Head), Heart Disease (Heart), and German Credit (Cr.Ger).
- *Other Data Sets*: Shuttle Control (Shuttle), Diabetes (Diab), DNA, Belgium power (Belg), Belgium power II (BelgII), Machine faults (Faults), and Tsetse fly distribution (Tsetse).

The table 3 summaries the main characteristics of the data sets mentioned above, and lists the first five algorithms with the best performance, based on the error rate on the test data set as the only criterion to measure efficiency, although other measurements were taken, such as the maximum storage used, the time during the training and testing phase, and the error rate for the training set.

An analysis of the results by subject areas shows that:

1. *For credit data sets* (Cr.man, Cr.Aus), the problem is to predict the creditworthiness of applicants for credit, which usually is defined by a human. Therefore, the aim of the decision rule is to devise a procedure that mimics the human decision process as closely as possible. Machine learning (decision trees) procedures are very good at this, and this probably reflects a natural tendency for human decisions to be made in a sequential manner. Then, it is easy for a human to understand the decision tree methods as this best reflects the human decision process.

2. *For data image data sets*, the nine image data sets are categorized as being one of Segmentation or Object Recognition.

- For the Object Recognition data sets (KL, Digits, Vehic, Chrom, Letter), standard statistical procedures (Quadisc, k-NN, Alloc80) and neural networks (Dipol92, LVQ) perform well overall.
- For the Segmentation data sets (Satim, Segm, Cut20, Cut50), machine learning procedures perform fairly well in segmentation data sets, and traditional statistical methods (Discrim, Quadisc, Logdisc) perform very badly.
- An important fact to arise here, is the performance of the k-NN method on image data sets. In four of the nine cases, k-NN is the algorithm with the best performance, and in one case (Cut20), it is in the second place. That means that the best results in image data sets are obtained by the k-NN algorithm (i.e., k-NN is best for images).

3. *For data sets with costs* (Head, Heart, Cr.Ger), i.e. applications areas where costs are important, such as medical data sets (Head, Heart), and credit data sets (Cr.Ger), standard statistical methods (Discrim, Logdisc, CASTLE, Quadisc) perform well; however, machine learning and neural network procedures perform worse than the default (of granting credit to everyone, or declaring everyone to be seriously ill).

4. *For the other data sets* (Belg, BelgII, Tset, Diab, DNA, Faults, Shuttle, Tech), it is perhaps inappropriate to derive general conclusions for data sets with diverse features, but it seems, from the performance of the algorithms, that these data sets are best dealt with machine learning or neural network procedures.

Table 3: Characteristics of the 22 data sets and their top five algorithms

Data Set	rec.	atttrib.	class	train	1	2	3	4	5
KL	18,000	40	10	(9000, 9000)	k-NN	Alloc80	Quadisc	LVQ	Dipol92
Dig44	18,000	16	10	(9000, 9000)	k-NN	Quadisc	LVQ	Cascade	Alloc80
Satim	6,435	36	6	(4435, 2000)	k-NN	LVQ	Dipol92	RBF	Alloc80
Vehic	846	18	4	9-fold cross-val	Quadisc	Dipol92	Alloc80	Logdisc	Bprop
Head	900	6	3	9-fold cross-val	Logdisc	Cascade	Discrim	Quadisc	CART
Heart	270	13	2	9-fold cross-val	NaiveB	Discrim	Logdisc	Alloc80	Quadisc
Belg	2,500	28	2	(1250, 1250)	Smart	Logdisc	Bprop	Dipol92	Discrim
Segm	2310	11	7	10-fold cross-val	Alloc80	AC2	Baytree	NewID	Dipol92
Diab	768	8	2	12-fold cross-val	Logdisc	Dipol92	Discrim	Smart	RBF
Cr.Ger	1,000	24	2	10-fold cross-val	Discrim	Logdisc	Castle	Alloc80	Dipol92
Chrom	40,000	16	24	(20000, 20000)	Quadisc	Dipol92	Discrim	LVQ	k-NN
Cr.Au	690	14	2	10-fold cross-val	CAL5	Itrule	Discrim	Logdisc	Dipol92
Shutt	58,000	9	7	(43500, 14500)	NewID	Baytree	CN2	CAL5	CART
DNA	3,186	60/180/ 240	3	(2000, 1186)	RBF	Dipol92	Alloc80	Discrim	Quadisc
Tech	7,080	56	91	(4500, 2580)	NewID	IndCart	AC2	C4.5	CN2
BelgII	3,000	57	2	(2000, 1000)	Smart	IndCart	Baytree	NewID	C4.5
Faults	570	45	3	10-fold cross-val	AC2	Dipol92	Discrim	Logdisc	Bprop
Tsetse	4,999	14	2	(3500, 1499)	CN2	Baytree	IndCart	NewID	CART
Cut20	18,700	20	2	(11220, 7480)	Baytree	k-NN	C4.5	Alloc80	NewID
Cut50	18,700	50	2	(11220, 7480)	k-NN	CN2	Alloc80	Baytree	C4.5
Cr.man	20,000	7	2	(15000, 5000)	Smart	Dipol92	C4.5	CAL5	Bprop
Letter	20,000	16	26	(15000, 5000)	Alloc80	k-NN	LVQ	Quadisc	CN2

In table 4, we summary the top five algorithms for the 22 data sets, by type, where

- *ML* means Machine Learning
- *Stat* means Statistical
- *NN* means Neural Network

Table 4: Top five algorithms for the 22 data sets classified by type.

Type/Name Data Set	First	Second	Third	Fourth	Fifth
Image - KL	Stat	Stat	Stat	NN	NN
Image - Dig44	Stat	Stat	NN	NN	Stat
Image - Satim	Stat	NN	NN	NN	Stat
Image - Vehic	Stat	NN	Stat	Stat	NN
Cost - Head	Stat	NN	Stat	Stat	ML
Cost - Heart	Stat	Stat	Stat	Stat	Stat
Other - Belg	Stat	Stat	NN	NN	Stat
Image - Segm	Stat	ML	ML	ML	NN
Other - Diab	Stat	NN	Stat	Stat	NN
Cost - Cr.Ger	Stat	Stat	Stat	Stat	NN
Image - Chrom	Stat	NN	Stat	NN	Stat
Credit - Cr.Aus	ML	ML	Stat	Stat	NN
Other - Shuttle	ML	ML	ML	ML	ML
Other - DNA	NN	NN	Stat	Stat	Stat
Other - Tech	ML	ML	ML	ML	ML
Other - BelgII	Stat	ML	ML	ML	ML
Other - Faults	ML	NN	Stat	Stat	NN
Other - Tset	ML	ML	ML	ML	ML
Image - Cut20	ML	Stat	ML	Stat	ML
Image - Cut50	Stat	ML	Stat	ML	ML
Credit - Cr.Man	Stat	NN	ML	ML	NN
Image - Letter	Stat	Stat	NN	Stat	ML

6.2 The DEA-model

Nakhaeizadeh & Schnabl [28] define a multi-criteria based metric for the evaluation of data mining algorithms, based in the DEA concept developed originally by Operations Research Community to measure efficiency. They define the *efficiency* of a data-mining algorithm as

$$\textit{efficiency} = \sum \text{weighted output components} \,/\, \sum \text{weighted input components}$$

Positive properties are called *output components* and negative properties are called *input components*. Efficiency, as defined above, is a more general notion than *interestingness*, proposed by Fayyad et al. [11], which considers only positive properties, and therefore, *efficiency* can be used as multi-criteria based metric for the evaluation of data mining algorithms. The output and input components are assumed to be known, and the weights, instead of being assigned subjectively by the opinion of decision makers, are determined for each data mining algorithm during the computation, such that they maximize the efficiency of the algorithm.

In this form, computing the efficiency of a data-mining algorithm is formulated as a linear programming problem, in which the weights of inputs and output components are computed such that they maximize the efficiency. After solving this linear programming problem (using the Simplex method), and determining the weights, the algorithms with *efficiency*=1 (100%) are *efficient* algorithms and form the *efficiency frontier* or *envelope*.

For ranking the algorithms, the AP-model (Andersen & Petersen [3]) is used, in which, for the case of input-oriented models, the AP-value measures how much an efficient algorithms can radially enlarge its input-levels while remaining still efficient

Nakhaeizadeh & Schnabl [28] evaluated 19 DM-algorithms, using the DEA-model, and compared their results with the evaluation made by Michie, Spiegelhalter, and Taylor (MST [24]), who evaluated the performance of 23 classification algorithms on 22 different domains, using only one property (the accuracy rate for the test data set). Table 5 presents the results reported by MST for the Credit Management Data set, where '*' means missing information, and FD denotes that the corresponding algorithm failed on this data set.

Table 6 summaries the ranking of different DEA-models versus the MST ranking for the Credit Management data set for 19 algorithms, where three input components (maximum storage, training time, and testing time) and two output components (accuracy rate for the test data set and accuracy rate for the training data set) were used. Input oriented versions are denoted by 5I (3 input components, 2 output components) and 4I (3 input components, 1 output component); output oriented versions are denoted by 5O and 4O, with analogous meaning.

Table 5: Evaluation of 22 algorithms by the MST method for the Credit Management data set (2 classes, 7 attributes, and 20,000 observations)

Algorithm	Maximum Storage	Training Time (sec)	Testing Time (sec)	Training error rates	Testing error rates	Rank
Discrim	68	32.2	3.8	0.031	0.033	13
Quadisc	71	67.2	12.5	0.051	0.050	21
Logdisc	889	165.6	14.2	0.031	0.030	8
SMART	412	27930.0	5.4	0.021	0.020	1
ALLOC80	220	22069.7	*	0.033	0.031	10
k-NN	108	124187.0	968.0	0.028	0.088	22
CASTLE	48	370.1	81.4	0.051	0.047	19
CART	FD	FD	FD	FD	FD	-
IndCART	1656	423.1	415.7	0.010	0.025	6
NewID	104	3035.0	2.0	0.000	0.033	13
AC^2	7250	5418.0	3607.0	0.000	0.030	8
Baytree	1368	53.1	3.3	0.002	0.028	7
NaiveBay	956	24.3	2.8	0.041	0.043	16
CN2	2100	2638.0	9.5	0.000	0.032	12
C4.5	620	171.0	158.0	0.014	0.022	3
`Itrule	377	4470.0	1.9	0.041	0.046	18
Cal5	167	553.0	7.2	0.018	0.023	4
Kohonen	715	*	*	0.037	0.043	16
DIPOL92	218	2340.0	57.8	0.020	0.020	1
Backprop	148	5950.0	3.0	0.020	0.023	4
RBF	253	435.0	26.0	0.033	0.031	10
LVQ	476	2127.0	52.9	0.024	0.040	15

In general, the MST ranking differs from the DEA-based ranking, mainly because the MST model is based on only one comparison criterion, and the DEA-models on multi-criteria metrics. Algorithms with low accuracy rate get a poor ranking using the MST model, but with DEA-models they may improve their ranking, because the low accuracy rate may be compensated by other properties, such as low training time and testing time (e.g. NaiveBayes, NewID, ITrule). Also, algorithms with high accuracy rate get a good ranking using the MST model, but with DEA-models they may decrease their ranking, because the high accuracy rate may be compensated by high training time (e.g. DIPOL92). In other cases, (e.g. SMART) the MST and DEA rankings do not differ significantly, because the high accuracy rate comes together with good values for the other components (input and output components).

Table 6: Ranking of 19 algorithms using MST and several DEA-models, for the Credit Management data set

Algorithm	MST	5I	4I	5O	4O
Discrim	13	3	3	9	7
Quadisc	21	14	12	18	17
Logdisc	8	16	13	15	12
SMART	1	2	2	5	5
k-NN	22	15	14	19	19
CASTLE	19	6	9	9	7
IndCART	6	17	17	14	13
NewID	13	1	8	9	7
AC^2	8	9	19	7	16
Baytree	7	9	5	1	1
NaiveBay	16	7	10	9	7
CN2	12	9	16	8	15
C4.5	3	9	4	4	4
Itrule	18	8	11	9	7
Cal5	4	4	6	2	2
DIPOL92	1	9	1	6	6
Backprop	4	5	7	3	3
RBF	10	18	15	16	14
LVQ	15	19	18	17	18

7 Summary and Discussion

In this paper we have shown several studies and models to assess the usefulness of data mining algorithms to resolve general and specific domain problems.

The survey among machine learning specialists (103 experts) characterize, for six machine learning algorithms (originally twelve methods) and three types of problems (originally nine task categories), the grade of usefulness of each method in solving intelligent tasks. The main conclusion of this study is that cooperation between learning systems, using either multi-strategy or hybrid learning frameworks, it is necessary to support the solution of intelligent tasks.

The assessment of the quality of inductive learning algorithms is based on general features of the algorithms, which are independent of the any specific problem. Therefore, it may be used to have an approximate idea about how well a specific algorithm would perform in solving a problem, which can be characterized in terms of the importance that each feature has for a good solution of the problem. Some features (i.e. characteristics of the input) can be objectively determined from the input data set; however others (i.e. characteristics of the output, efficiency in the learning and application phase) would require the subjective judgment of the user. Also, selecting a different subset of features (i.e. considering only some specific features) may suggest different algorithm with the best usability, even for the same data set, as shown in table 2.

Regarding the approaches for evaluation of data mining algorithm discussed before, the STATLOG project produces some hints to characterize the performance of different approaches (statistical, decision trees/rule based, and neural network) and methods (23 algorithms) over a wide variety of data sets (22 domains). Analysis of the results shows that for different domains, different approaches perform better than others (i.e. for credit data sets, decision trees/rule based classifiers have the best performance; for data image data set, classical statistical methods, neural networks, and especially k-NN are more appropriate than other algorithms; for data set with costs, classical statistical methods perform well, and machine learning and neural network methods perform poor in this domain). The DEA-model extends the criteria to evaluate data mining algorithms, incorporating the concept of *efficiency* of a data-mining algorithm, which considers both positive and negative properties. Generally, the rankings of the DEA-approach differ from the rankings of the STATLOG project, because the accuracy rate on the test data set may come together with some negative properties of the algorithm (i.e. high training and testing time, poor accuracy on the training test, etc.)

Our current research is involved in formulating a model that combines problem-domain independent aspects with performance issues, which necessarily are problem or domain dependent, in order to formulate a full model. This model would guide the user in the selection of the best subset of techniques, before starting the KDD process or during the data-mining phase. We have formulated a preliminary guide, called the *Usability* model (Meneses [23]), based on features of the problem, domain independent and domain dependent features of the data mining algorithms. This model can be used to assess the usability of a data mining algorithm to solve a specific problem P, and help the user in the selection of the best subset of data mining techniques to solve P. However, further research is required in order to evaluate the practicability of the *Usability* model to effectively guide the user in the selection of the best subset of data mining algorithms to solve real complex problems. Different extensions of the proposed model should be considered, in order to incorporate, for example, automatic sensibility analysis of its main parameters, and extend the set of features that characterize problems and data mining algorithms.

8 Acknowledgment

This research is being supported by a grant from the National Institute for Standards and Technology (NIST, Washington D.C.), and by the Institute for Visualization and Perception Research (IVPR) at University of Massachusetts Lowell.

References

[1] Adriaans, P., Zantinge D., *Data Mining*, Addison-Wesley, 1996.

[2] Agrawal, R., Mannila, H. Skrkant, R., Toivonen H., Verkamo, I., Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining (Fayyad et al., editors), AAAI Press / The MIT Press, pp. 307-328, 1996.

[3] Andersen, P., Petersen, N.C., *A Procedure for Ranking Efficient Units in Data Envelopment Analysis*. Management Science, Vol. 39, No. 10, pp. 1261-1264, 1993.

[4] Ankerst, M., Keim, D.A., Kriegel, H-P., Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets. Hot Topics, IEEE Visualization 96 Conference, pp. 4-7, 1996.

[5] Beshers, C., Feiner, S., *Visualizing n-Dimensional Virtual Worlds with n-Vision*. Computer Graphics, Vol. 24, No. 2, pp. 37-38, 1990.

[6] Cheeseman, P., Stutz, J., *Bayesian Classification (AutoClass): Theory and Results*. In Advances in Knowledge Discovery and Data Mining (Fayyad et al., editors), AAAI Press / The MIT Press, pp. 153-180, 1996.

[7] Chernoff, H., *The Use of Faces to Represent Points in k-dimensional Space Graphically*. Journal of the American Statistical Association, Vol. 68, No. 342, pp. 361-368, 1973.

[8] Clark, P., Niblett, R., *The CN2 induction algorithm*. Machine Learning, 3, pp. 261-284, 1989.

[9] Cleveland, W.S., *Visualizing Data*. AT&T Bell Laboratories, Murray Hill, NJ, Hobart Press, Summit, NJ, 1993.

[10] Dasarathy, B. V., *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, CA, IEEE Computer Society Press, 1991.

[11] Fayyad, U., Piatestsky-Shapiro, G., Smyth, P., *From Data Mining to Knowledge Discovery: An Overview*. In Advances in Knowledge Discovery and Data Mining (Fayyad et al., editors), AAAI Press / The MIT Press, pp. 1-34, 1996.

[12] Fayyad, U., Simoudis, E., *Data Mining and KDD: An Overview*, Tutorial in the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, California, 1997.

[13] Goldber, D.E., *Genetic and Evolutionary Algorithms Come of Age*. Communications of the ACM, Vol. 37, No. 3, pp. 113-119, 1994.

[14] Han, J., Fu, Y., *Exploration of the Power of Attribute-Oriented Induction in Data Mining*. In Advances in Knowledge Discovery and Data Mining (Fayyad et al., editors), AAAI Press / The MIT Press, pp. 399-421, 1996.

[15] Heckerman, D., *Bayesian Networks for Knowledge Discovery*. In Advances in Knowledge Discovery and Data Mining (Fayyad et al., editors), AAAI Press / The MIT Press, pp. 273-305, 1996.

[16] Hertz, J., *Introduction to the Theory of Neural Computing*, Reading, MA, Addison-Wesley, 1991.

[17] Hoffman, P., Grinstein, G., Marx, K., Grosse, I., Stanley, E., *DNA Visual and Analytical Data Mining*. Proceedings of the IEEE Visualization '97 Conference, Phoenix, AZ, pp. 437-441, 1997.

[18] Inselberg, A., Dimsdale, B., *Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry*. Proceedings of Visualization'90, San Francisco, CA, pp. 361-370, 1990.

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

[20] Kivinen, J., Mannila, H., Ukkonen, E., *Learning Rules with Local Exceptions*. Technical Report, University of Helsinki, 1993.

[21] LeBlanc, J., Ward, M.O., Wittels, N., *Exploring N-dimensional Databases*. Proceedings of Visualization '90, San Francisco, CA, pp. 230-237, 1990.

[22] Lenat, D., EURISKO: *A Program that Learns New Heuristics and Domain Concepts*. The Nature of Heuristics III: Background and Examples. Artificial Intelligence, 21, pp. 61-98, 1983.

[23] Meneses, C., *Categorization and Evaluation of Data Mining Techniques*. Technical Report, Institute for Visualization and Perception Research, Department of Computer Science, University of Massachusetts Lowell, 1998.

[24] Michie, D., Spiegelhalter, D.J., Taylor, C.C., *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, Chichester, 1994.

[25] Minsky, M., *A Framework for Representing Knowledge*. In The Psychology of Computer Vision (Winston, P.H., editor), McGraw-Hill, NY, pp. 211-277, 1975.

[26] Mitchell, T.M., *Machine Learning*. McGraw-Hill, 1997.

[27] Moustakis, V.S., Letho, M., Salvendy, G., *Survey of expert opinion: which machine learning method may be used for which task?*, Special issue on machine learning of International Journal of HCI, 8(3), pp. 221-236, 1996.

[28] Nakhaeizadeh, G., Schnabl, A., *Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms*, in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, California, pp. 37-42, 1997.

[29] Piatetsky-Shapiro, G., *Siftware: Tools for Data Mining and Knowledge Discovery*. http://www.kdnuggets.com/, 1998.

[30] Pickett, R.M., Grinstein, G.G., Iconographic Display for Visualizing Multidimensional Data. In Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, Beijing and Shenyang, China, pp. 514-519, 1988.

[31] Quinlan, J.R., *Induction of Decision Trees*. Machine Learning, 1: pp. 81-106, 1986.

[32] Quinlan, J.R., *Generating Production Rules from Decision Trees*. In Proceedings of the 10th International Joint Conference on Artificial Inteligence (IJCAI '87), Milan, pp. 304-307, 1987.

[33] Quinlan, J.R., *Determining literals in inductive logic programming*. In Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sidney, Australia, pp. 746-750, 1991.

[34] Quinlan, J.R., *Learning Logical Definitions from Relations*. Machine Learning, 5, pp. 239-266, 1990.

[35] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[36] Rekimoto, J., Green, M., *The Information Cube: Using Transparency in 3D Information Visualization.* Proceedings of Third Annual Workshop on Information Technologies and Systems (WITS '93), pp. 125-132, 1993.

[37] Ringland, G.A., Duce, D.A. (editors), (1988). *Approaches to Knowledge Representation: An Introduction.* Research Studies Press Ltd., Letchworth, England, 1988.

[38] Rivest, R.L., *Learning Decision Lists.* Machine Learning, 2: pp. 229-246, 1987.

[39] Robertson, G.G., Mackinlay, J.D., Card, S.K., *Cone Trees: Animated 3D Visualizations of Hierarchical Information.* Proceedings of Human Factors in Computing Systems CHI'91 Conference, New Orleans, LA, pp. 189-194, 1991.

[40] Russell, S.J., Norvig, P., *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Inc., 1995.

[41] Schneiderman, B., *Tree Visualization with Treemaps: A 2D Space-Filling Approach.* ACM Transactions on Graphics, Vol. 11, No. 1, pp. 92-99, 1992.

[42] Ward, M.O., *XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data.* Proceedings of Visualization '94, Washington D.C., pp. 326-336, 1994.

[43] Weiss, S.M., Indurkhya, N., *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann Publishers, Inc., 1998.