An Interactive-Graphic Environment for Discovering and Using Conceptual Knowledge

Tu Bao Ho, Trong Dung Nguyen, Hiroshi Shimodaira, Masayuki Kimura

Japan Advanced Institute of Science and Technology
Tatsunokuchi, Ishikawa, 923-12 JAPAN

Abstract. In this paper we describe our ongoing project whose objective is to develop an interactive-graphic environment of related tools for discovering and using conceptual knowledge in relation to influential aspects of the environment and representation. These aspects include the amount of supervision, the manner of data presentation, the regularity of the domain, the representation of data and knowledge. The core of this project is two supervised and unsupervised learning methods that induce knowledge in the form of concept hierarchies. The project aims at improving the performance of these methods and at integrating their implementation in the X Window with the direct manipulation style of interaction. The ultimate goal of the project is to provide an environment in which the user can find and use knowledge from data with low-cost and high-quality.

1 Motivation and Background

How to acquire knowledge for knowledge-based systems (KBS) remains as the main difficult and crucial problem of KBS technology. In addition to this difficulty, the explosive growth in the quantity of data stored in databases leads to a common situation of "data rich and knowledge poor". This situation creates a need of techniques and tools for understanding and extracting useful knowledge from data. Knowledge discovery in databases (KDD), the rapidly growing interdisciplinary field of computing which merges together databases, statistics and machine learning techniques, aims at achieving these goals [7]. Thus, the goals of KDD are essentially similar to those of traditional knowledge acquisition (KA) for the KBS development.

In this paper we describe our ongoing project whose objective is to develop an interactive-graphic environment of related tools for discovering and using conceptual knowledge in relation to influential aspects of the environment and representation. The core of our project is the supervised learning method CABRO [12] and the unsupervised learning method OSHAM [14], [15], [16] which induce effectively knowledge in the form of concept hierarchies. The project aims at improving the performance of these methods and integrating their implementation in the X Window with the direct manipulation style of interaction [11], that allows the user to find and use knowledge from data with low-cost and high-quality.

In this section we discuss the three main motivations of our project: "Why do we choose the hierarchical model of conceptual knowledge", "Why do we need to deal with different aspects of the environment and representation?", and "Why do we need to build an interactive-graphic system for modelling?". In sections 2 and 3 we will present the basic ideas and the evaluation of CABRO and OSHAM. Sections 4 and 5 address some issues in the project implementation and conclusions.

1.1 Hierarchical Models of Conceptual Knowledge

Conceptual modelling is a process of forming and collecting conceptual knowledge about the universe of discourse, and documenting the results in the form of a conceptual schema [20]. Conceptual modelling is a widely recognized activity in the process of KBS development. It is important to notice that in the knowledge acquisition community "expertise transfer" paradigms have been replaced in recent years by "knowledge modelling" paradigms [8]. Clancey, the defender of the *modelling view* of knowledge acquisition, argues that "the primary concern of knowledge engineering is modelling systems in the world, not replicating how people think" [4].

The process of forming conceptual knowledge relates closely to the *organization* of knowledge. Among three main alternative schemes of decision lists, inference networks, and concept hierarchies for organizing knowledge descriptions, we are particularly interested in the last ones which are fundamental for the basic modeling scheme of information processing [26] and are widely used in AI products such as KBS tools KEE, Kappa, Nexpert Object, etc. The hierarchical model for conceptual modelling also shares the common structured models on Object-Oriented Modelling [29].

A *concept hierarchy* is a structure composed of nodes and links. Each node represents a concept with its associated intensional description. The links connecting a node to its children specify an "IS-A" or "subset" relation, indicating that the parent's extension is a superset of each child's extension. Typically, a node covers all of the instances covered by the union of its descendents, making the concept hierarchy a subgraph of the partial ordering by generality. More abstract or general nodes occur higher in the hierarchy, whereas more specific ones occur at lower levels [23].

The most widely used forms of concept hierarchies are decision trees and discrimination networks. A *decision tree* is a classifier in the form of a tree structure whose node is either a leaf (a class of instances) or a decision node that specifies some test to be carried out, with one branch for each possible outcome of the test. A *discrimination network* has a similar structure of trees but concepts at a branch do not have to be mutually exclusive, so multiple father nodes can arise. A decision tree/discrimination network can be easily converted into a set of decision rules [27].

Being modelled as a concept hierarchy, the exploitation of knowledge for a classification process involves sorting instances downward through the hierarchy. In general, the modelling task with this structure can be stated as follows:

- *Given*: A set of instances with/without class information;
- *Find*: A concept hierarchy that, to the extent possible, makes accurate predictions about unknown instances.

1.2 Influential Aspects of Environment and Representation

Techniques for acquiring conceptual knowledge are deeply influenced by different aspects of the environment and of the representation of data and knowledge. Similarly to those in [23], one can distinguish the following influential aspects:

- The conceptual knowledge involves *one-step* classification and prediction or *multi-step* inference or problem solving;
- The application domain is *supervised* or *unsupervised*. The degree of supervision concerns whether the class information in data or a domain expert is available. In case of a supervised domain, there is a feedback about the appropriateness of the discovered results and the discovering process is essentially error-driven. Without this feedback in case of unsupervised domain, the discovering process is essentially the search for regularities in data. In our opinion, the degree of supervision is the most important aspect based on it we can choose techniques.
- The manner in which data are provided. One distinguishes a *nonincremental* task when data are presented simultaneously (offline) and an *incremental* task when data are presented serially (online).
- The *regularity* of the environment (e.g., complexity of the target knowledge, number of irrelevant attributes, the amount of noise and missing values, etc.).
- The representation nature of data and knowledge affects the discovering process. One distinguishes data which contain *symbolic attributes* that are nominal or ordinal ones, *numeric attributes* that take on real values, and *relational literals*.
- The three alternative schemes for *organizing the knowledge* descriptions of decision lists, inference networks, and concept hierarchies [23]. As mentioned in subsection 1.1, we are particularly interested in the concept hierarchy structure that is widely used in KBS.

All of these aspects may occur in realistic situations and influence the conceptual modelling process. Often, each discovering method can deal with one or some mentioned aspects but not all. For example, most data analysis or optimization systems are developed for numeric data and most machine learning systems are developed for symbolic data. In the practical use of a discovery system, it is expected that it can function in different situations.

For more complex data, our project shares some common tasks with the ESPRIT European project SODAS [10] of 18 partners on a software for *symbolic data analysis*. Symbolic data analysis [5] is a new attempt that aims at extending problems, methods and algorithms used in standard data analysis to more complex data such as a set of values, intervals of values, or a probabilistic distribution, etc.

1.3 An Interactive-Graphic Environment for Discovering

We perceive that conceptual modelling is an iterative cycle of knowledge refinement in which the system provides and receives interactively feedback to and from the user.

Current discovery systems do not always equal the human ability in identifying useful concepts, and as the search problem in such a complex process requires much background knowledge and heuristics, the human factor in discovery process is always necessary [13].

A strong interaction between the discovery system and the user is expected to be a common feature of discovery systems as the discovery systems cannot produce maximally useful results when operating alone. Recently, there are much efforts in the development of *interactive-graphic environments* in order to improve the performance of discovery systems. In [22] the authors develop an interactive-graphic environment for constructing decision trees. In [24] the author develop system WinViz that integrates multidimensional visualization with the program C4.5 for learning decision trees [27].

To support the knowledge acquisition modelling process, we use a Visual Interactive Model through a rich graphical environment. A Visual Interactive Model (VIM) aims at combining "meaningful pictures and easy interactions to stimulate creativity and insight; promoting a process of 'generate and test', it facilitates a rapid cycle of learning" [1]. Concretely, in our environment VIM offers the user two main benefits: (1) better understanding the induction process and generated decision trees/concept hierarchies, especially by the Tree Visualizer; and (2) a more active role in the modelling process with a interactive mode of operation.

2 Supervised Discovery of Knowledge

2.1 R-measure

The basic task of supervised discovery of conceptual knowledge is from a given set of labelled instances to find a classifier that correctly predicts classes of unseen instances. Among approaches to this problem the *decision tree induction* is probably the most active and applicable one.

Table 1 gives a brief description of the common scheme for decision tree induction. Decision tree induction systems differ from each other in their way to deal with two crucial problems of *attribute selection* (choosing the "best" attribute to split a decision node in terms of a measure for "goodness of split") and *pruning* (avoiding overfitting and obtaining statistical reliability). We have developed a decision tree induction method called CABRO which uses R-measure for the attribute selection, a new measure stemmed from the theory of rough sets [28].

Table 1. Framework of decision tree induction

1. Choose the "best" attribute by an attribute selection measure.
2. Extend tree by adding new branch for each attribute value.
3. Sort training examples to leaf nodes.
4. If examples unambiguously classified then stop else repeat steps 1–4 for leaf nodes.
5. Prune the obtained tree.

Rough set theory is a mathematical tool to deal with vagueness and uncertainty. The basic idea in this theory is to "view" approximately each subset X of an object

set O by its *lower* and *upper* approximations w.r.t. an equivalence relation $E \subseteq O \times O$. These approximations of X are defined, respectively, by $E_*(X) = \{o \in O : [o]_E \subseteq X\}$ and $E^*(X) = \{o \in O : [o]_E \cap X \neq \emptyset\}$, where $[o]_E$ denotes the equivalence class of an object o in E. A key concept in the rough set theory is the *degree of dependency* of a set of attributes Q on a set of attributes P, denoted by $\mu_P(Q)$ $(0 \leq \mu_P(Q) \leq 1)$, defined as

$$\mu_P(Q) = \frac{card(\bigcup_{[o]_Q} P_*([o]_Q))}{card(O)} = \frac{card(\{o \in O : [o]_P \subseteq [o]_Q\})}{card(O)} \quad (1)$$

If $\mu_P(Q) = 1$ then Q totally depends on P; if $0 < \mu_P(Q) < 1$ then Q partially depends on P; if $\mu_P(Q) = 0$ then Q is independent of P.

The measure $\mu_P(Q)$ can be used directly in decision tree induction for the attribute selection with Q stands for the class attribute and P stands for a descriptive attribute. In [12], our analysis and experiments have shown that $\mu_Q(P)$ is not robust with noisy data and not enough sensitive when partitions of O generated by P and Q are nearly identified. From this analysis, we have generalized and formulated a measure for degree of dependency of an attribute set Q on an attribute set P

$$\mu'_P(Q) = \frac{1}{card(O)} \sum_{[o]_P} max_{[o]_Q} card([o]_Q \bigcap [o]_P) \quad (2)$$

The main difference between $\mu_P(Q)$ and $\mu'_P(Q)$ is the latter measures the dependency of Q on P in maximizing the predicted membership of an instance in the family of equivalence classes generated by Q given its membership in the family of equivalence classes generated by P.

Theorem. *For every attribute set P and Q we have*

$$\frac{max_{[o]_Q} card([o]_Q)}{card(O)} \leq \mu'_P(Q) \leq 1$$

This property allows us to define that Q totally depends on P iff $\mu'_P(Q) = 1$, Q partially depends on P iff $max_{[o]_Q} card([o]_Q)/card(O) < \mu'_P(Q) < 1$, and Q is independent of P iff $\mu'_P(Q) = max_{[o]_Q} card([o]_Q)/card(O)$. In practice, to emphasize rules those have the higher generalities we use the following formula, and call it R-measure

$$\tilde{\mu}_P(Q) = \frac{1}{card(O)} \sum_{[o]_P} max_{[o]_Q} \frac{card([o]_Q \bigcap [o]_P)^2}{card([o]_P)} \quad (3)$$

2.2 Evaluation

Three criteria on the size, prediction accuracy and understandability mentioned in [25] for evaluating decision trees are common used, among them the prediction accuracy of pruned trees is widely considered to be of fundamental importance.

We have carried out carefully experimental comparative studies of R-measure with some widely used measures as gain-ratio in C4.5 [27], gini-index in CART [2], χ^2 in statistics [25], by k-fold stratified cross validation.

In k-fold stratified cross validation, the dataset is randomly stratified and divided into k mutually exclusive subsets (folds) of approximately equal size and the same proportions of labels as in the original dataset. One subset is used as testing data and the union of the rest ones is used as training data. One run of k-fold cross validation is the repeat k times of this process each with a new testing subset, and the accuracy is estimated as the average of the accuracies of k runs. The experiments were designed as follows

- use a large number of datasets;
- use 10-fold stratified cross-validation with a random shuffle of data;
- implement studied techniques in a system based on the scheme of CLS;
- to evaluate selection measures we run the system with different attribute selection measures while fixing a pruning and a discretization technique.

Eighteen datasets from the UCI repository of machine learning databases were used. Experimental results are reported in Table 2 in which included the following information

- the letters c, g, χ and R stand for gain-ratio, gini-index, χ^2 and R-measure, respectively (first column).
- datasets features: name, number of attributes $\times$ number of instances, numeric, symbolic or mix data;
- the size of trees before and after pruning (even columns);
- the error rates on testing data before and after pruning (odd columns);

Table 2. Experimental results on attribute selection measures

	unpruned		pruned		unpruned		pruned	
	size	errors	size	errors	size	errors	size	errors
	Vote, 16x300, symbolic				Cancer, 9x700, symbolic			
c	22.6 ± 2.5	7.3 ± 3.6	4.0 ± 0.0	**5.0 ± 2.8**	87.9 ± 23.1	7.3 ± 2.2	46.1 ± 19.1	7.4 ± 2.9
g	24.7 ± 2.8	7.0 ± 3.0	7.0 ± 4.2	5.9 ± 2.7	92.3 ± 26.0	6.9 ± 2.3	36.2 ± 11.9	7.4 ± 3.5
χ	24.7 ± 2.8	7.0 ± 3.0	7.0 ± 4.2	5.9 ± 2.7	92.3 ± 26.0	6.9 ± 2.0	36.2 ± 11.9	7.4 ± 3.5
R	25.0 ± 3.0	7.5 ± 3.4	5.8 ± 2.9	5.7 ± 2.7	94.5 ± 26.4	7.0 ± 2.2	37.3 ± 11.2	**7.1 ± 3.4**
	Shuttle, 9x956, symbolic				Promoters, 45x105, symbolic			
c	88.4 ± 10.9	0.2 ± 0.1	53.4 ± 15.8	**0.2 ± 0.1**	18.2 ± 3.3	25.5 ± 9.8	9.8 ± 4.3	24.5 ± 7.5
g	144.8 ± 6.6	0.2 ± 0.1	114.2 ± 12.2	**0.2 ± 0.1**	19.0 ± 4.0	23.6 ± 12.7	9.4 ± 3.7	**22.7 ± 10.0**
χ	199.0 ± 17.2	0.3 ± 0.1	162.7 ± 30.4	0.3 ± 0.1	19.0 ± 4.0	23.6 ± 10.9	9.4 ± 3.7	**22.7 ± 10.0**
R	165.3 ± 15.6	0.2 ± 0.1	135.3 ± 18.3	0.3 ± 0.1	19.0 ± 4.0	23.6 ± 12.7	9.4 ± 3.7	**22.7 ± 10.0**
	Solar Flare, 12x1286, symbolic				Diabetes, 8x768, numeric			
c	104.0 ± 11.6	26.8 ± 2.5	26.8 ± 7.8	**25.3 ± 1.5**	41.2 ± 2.2	24.4 ± 3.1	18.2 ± 9.4	**25.3 ± 2.6**
g	150.8 ± 15.0	28.4 ± 2.9	54.4 ± 15.0	27.8 ± 1.3	53.0 ± 4.4	25.3 ± 2.7	22.0 ± 4.8	25.6 ± 2.5
χ	168.6 ± 24.6	28.3 ± 2.2	45.8 ± 18.1	26.6 ± 2.0	47.6 ± 5.1	25.3 ± 2.7	13.6 ± 6.2	25.5 ± 2.5
R	155.0 ± 19.4	26.9 ± 1.8	44.6 ± 31.1	25.5 ± 1.0	74.2 ± 8.6	24.7 ± 2.6	27.8 ± 19.8	**25.3 ± 2.6**
	Splice, 45x3189, numeric				Glass, 9x214, numeric			
c	529.8 ± 68.0	10.2 ± 1.5	245.8 ± 36.8	**8.0 ± 1.7**	21.0 ± 3.2	33.2 ± 8.6	17.3 ± 5.5	**34.5 ± 8.2**
g	565.8 ± 72.0	10.4 ± 2.4	214.6 ± 39.8	8.4 ± 1.8	35.2 ± 4.4	33.2 ± 7.7	22.3 ± 6.9	36.8 ± 6.8
χ	585.8 ± 76.0	10.5 ± 2.4	253.0 ± 56.8	8.8 ± 1.7	29.6 ± 2.8	35.0 ± 5.6	19.1 ± 7.5	37.3 ± 6.4
R	569.8 ± 77.4	11.0 ± 2.5	207.4 ± 30.4	8.6 ± 1.9	32.2 ± 5.8	34.1 ± 7.3	18.7 ± 6.8	35.9 ± 6.9
	Waveform, 36x3195, symbolic				Heart Disease, 13x270, mixed			
c	1148.3 ± 179.5	28.9 ± 1.7	223.9 ± 72.9	25.7 ± 1.1	13.8 ± 5.4	25.6 ± 4.1	8.8 ± 3.8	25.6 ± 4.1
g	1320.5 ± 193.9	27.8 ± 1.3	244.5 ± 69.5	**24.4 ± 1.6**	33.0 ± 4.0	27.4 ± 4.7	25.8 ± 2.6	25.6 ± 5.6
χ	1355.7 ± 185.0	28.4 ± 1.6	340.6 ± 191.6	26.8 ± 1.3	26.6 ± 9.7	27.4 ± 4.7	9.0 ± 3.2	26.3 ± 4.9
R	1432.3 ± 193.5	29.3 ± 1.5	249.6 ± 78.4	25.1 ± 1.1	38.0 ± 13.4	27.4 ± 4.7	8.2 ± 5.1	**25.2 ± 4.6**
	Vehicle, 18x846, numeric				Hypothyroid, 25x3163, numeric			
c	174.5 ± 35.9	32.4 ± 5.2	131.9 ± 40.7	32.7 ± 5.1	22.6 ± 2.5	1.1 ± 0.4	11.8 ± 1.5	**0.9 ± 0.4**
g	222.8 ± 38.4	31.9 ± 3.2	111.4 ± 37.8	32.0 ± 3.7	49.2 ± 5.4	1.3 ± 0.5	16.8 ± 3.7	**0.9 ± 0.4**
χ	216.2 ± 40.4	30.2 ± 3.9	111.4 ± 47.4	31.9 ± 3.2	54.8 ± 5.0	1.3 ± 0.5	10.6 ± 0.6	**0.9 ± 0.4**
R	218.2 ± 39.6	31.6 ± 3.2	101.5 ± 28.7	**31.8 ± 3.5**	57.8 ± 6.0	1.4 ± 0.4	18.2 ± 4.9	**0.9 ± 0.4**
	Audiology, 70x226, symbolic				Cars, 8x392, numeric			
c	49.8 ± 9.0	29.6 ± 13.7	28.4 ± 13.3	30.9 ± 11.0	32.3 ± 2.0	24.8 ± 4.8	17.1 ± 9.5	26.0 ± 2.0
g	68.2 ± 12.4	29.6 ± 11.5	37.0 ± 16.0	30.9 ± 11.9	44.7 ± 10.3	24.0 ± 4.8	21.4 ± 8.5	26.8 ± 5.2
χ	93.9 ± 22.3	44.3 ± 8.9	66.9 ± 14.9	45.2 ± 8.7	41.8 ± 8.8	23.8 ± 5.0	17.4 ± 9.1	26.5 ± 5.2
R	72.1 ± 10.3	28.3 ± 10.9	41.3 ± 13.4	**29.1 ± 11.7**	44.6 ± 12.5	24.2 ± 5.1	21.8 ± 12.8	**25.2 ± 4.8**
	Horse-colic, 28x368, numeric				Pima-diabetes, 8x768, numeric			
c	48.9 ± 9.1	16.2 ± 3.8	8.2 ± 4.1	**14.3 ± 5.1**	34.3 ± 6.7	24.9 ± 4.5	17.6 ± 5.8	**23.4 ± 3.6**
g	86.4 ± 19.1	17.8 ± 1.9	30.9 ± 20.3	16.8 ± 3.5	45.4 ± 3.5	24.7 ± 4.2	25.4 ± 8.3	23.5 ± 3.5
χ	92.0 ± 25.8	18.1 ± 2.3	22.0 ± 22.4	17.0 ± 3.3	40.0 ± 4.2	24.7 ± 4.2	18.5 ± 9.0	23.5 ± 3.5
R	115.6 ± 22.7	17.0 ± 1.7	15.8 ± 13.7	15.9 ± 4.2	65.1 ± 8.5	24.7 ± 4.2	30.6 ± 17.4	23.9 ± 3.2
	Segmentation, 19x2310, numeric				Iris, 4x150, numeric			
c	327.4 ± 48.2	6.3 ± 1.5	236.4 ± 46.5	6.2 ± 1.6	4.3 ± 0.5	4.0 ± 3.2	4.0 ± 0.0	3.3 ± 3.3
g	341.3 ± 30.9	5.9 ± 1.7	257.5 ± 81.2	**6.1 ± 2.0**	4.3 ± 0.5	3.3 ± 3.3	4.0 ± 0.0	**2.7 ± 3.2**
χ	373.2 ± 25.2	7.3 ± 1.6	310.7 ± 48.3	7.6 ± 2.0	4.3 ± 0.5	4.7 ± 3.7	4.0 ± 0.0	4.0 ± 4.0
R	342.5 ± 33.6	6.1 ± 1.8	272.0 ± 90.4	**6.1 ± 2.1**	4.3 ± 0.5	4.7 ± 3.7	4.0 ± 0.0	4.0 ± 4.0

As error rates of pruned trees are of most importance, we indicate the lowest error rate of pruned trees for each dataset among four measures by bold numbers. The error rates of four measures on eighteen datasets are summarized in Table 3 and Figure 1.

Table 3. Error rates of pruned trees for four measures

datasets	Gain-Ratio	Gini-Index	χ^2	R-measure
Shuttle	**0.2 ± 0.1**	**0.2 ± 0.1**	0.3 ± 0.1	0.3 ± 0.1
Hypothyroid	**0.9 ± 0.4**	**0.9 ± 0.4**	**0.9 ± 0.4**	**0.9 ± 0.4**
Iris	3.3 ± 3.3	**2.7 ± 3.2**	4.0 ± 4.0	4.0 ± 4.0
Vote	**5.0 ± 2.8**	5.9 ± 2.7	5.9 ± 2.7	5.7 ± 2.7
Breast cancer	7.4 ± 2.9	7.4 ± 3.5	7.4 ± 3.5	**7.1 ± 3.4**
Segmentation	6.2 ± 1.6	**6.1 ± 2.0**	7.6 ± 2.0	**6.1 ± 2.1**
Splice	**8.0 ± 1.7**	8.4 ± 1.8	8.8 ± 1.7	8.6 ± 1.9
Horse-colic	**14.3 ± 5.1**	16.8 ± 3.5	17.0 ± 3.3	15.9 ± 4.2
Waveform	25.7 ± 1.1	**24.4 ± 1.6**	26.8 ± 1.3	25.1 ± 1.1
Solar Flare	**25.3 ± 1.5**	27.8 ± 1.3	26.6 ± 2.0	25.5 ± 1.0
Heart-disease	25.6 ± 4.1	25.6 ± 5.6	26.3 ± 4.9	**25.2 ± 4.6**
Diabetes	**25.3 ± 2.6**	25.6 ± 2.5	25.5 ± 2.57	**25.3 ± 2.6**
Promoters	24.5 ± 7.5	**22.7 ± 10.0**	**22.7 ± 10.0**	**22.7 ± 10.0**
Pima-Diabetes	**23.4 ± 3.6**	23.5 ± 3.5	23.5 ± 3.5	23.9 ± 3.2
Vehicle	32.7 ± 5.1	32.0 ± 3.7	31.9 ± 3.2 7	**31.8 ± 3.5**
Audiology	30.9 ± 11.0	30.9 ± 11.9	45.2 ± 8.7	**29.1 ± 11.7**
Glass	**34.5** ± 8.2	36.8 ± 6.8	37.3 ± 6.4	35.9 ± 6.9
Cars	26.0 ± 2.0	26.8 ± 5.2	26.8 ± 5.2	**25.2 ± 4.8**

Other information as the tree sizes, error rates before pruning in Table 2 can be viewed as additional factors for evaluating methods. Some conclusions can be drawn from our various experimental results reported in Table 2, Table 3.

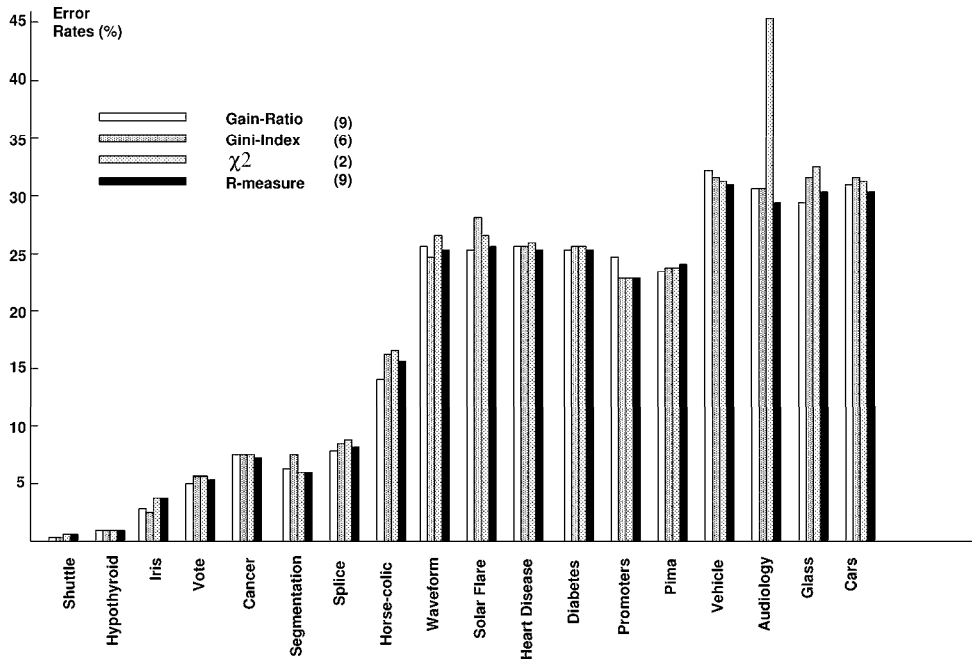

Figure 1. Graphical representation of error rates of pruned trees

- For pruned trees, the numbers of datasets on which each measure attains the lowest error rate are 9 (gain-ratio), 6 (gini-index), 2 (χ^2) and 9 (R-measure), and the smallest size are 11 (gain-ratio), 2 (gini-index), 5 (χ^2) and 5 (R-measure). These results verified that though certain methods are very good but they are not always the best, and it may be necessary to select the most suitable technique in certain applications of decision tree induction. Our easy-to-use system described in next section aims at supporting this selection.
- Careful experimental results show that R-measure is a good one. Evaluating together with the most widely used and stable measures, R-measure attains reasonably comparable error rates in various datasets. This allow us to believe in the high performance of R-measure and its application potential.

3 Unsupervised Discovery of Knowledge

3.1 Concept Representation and Clustering

The basic task of acquiring knowledge in this situation is that from a given set of unlabelled instances to find simultaneously a hierarchical clustering that determines useful object subsets and intensional definitions for these subsets of objects. Essentially, unsupervised concept learning methods differ from each other in two factors of *views on concepts* and *constraints of categorization*. Among views on concepts, the classical, prototype and exemplar ones are widely known and used. Among categorization constraints, the similarity, feature correlation, and structure of the concept hierarchy are widely known and used. The learning system OSHAM proposed in [14], which employs the classical view on concepts, and is able to form effectively a concept hierarchy from unlabelled data. Essentially, OSHAM searches to extract a good concept hierarchy by exploiting the structure of Galois lattice of concepts as the hypothesis space. OSHAM has been extended to a hybrid system that allows obtaining a higher performance by combining its original view on concepts with the prototype and exemplar views [15].

Instead of characterizing a concept only by its intent and extent, OSHAM represents each concept C_k in a concept hierarchy $\mathcal{H}$ by a 10-tuple

$$< l(C_k), f(C_k), s(C_k), i(C_k), e(C_k), d(C_k), p(C_k), d(C_k^r), p(C_k^r|C_k), q(C_k) > \qquad (4)$$

where

- $l(C_k)$ is the level of C_k in $\mathcal{H}$;
- $f(C_k)$ is the list of direct superconcepts of C_k;
- $s(C_k)$ is the list of direct subconcepts of C_k;
- $i(C_k)$ is the intent of C_k (set of all common properties of instances of C_k);
- $e(C_k)$ is the extent of C_k (set of all instances satisfying properties of $i(C_k)$);
- $d(C_k)$ is the dispersion between instances of C_k;
- $p(C_k)$ is the occurrence probability of C_k;
- $d(C_k^r)$ is the dispersion of local instances of C_k which are not classified into subconcepts of C_k;
- $p(C_k^r|C_k)$ is the conditional probability of these unclassified instances of C_k;
- $q(C_k)$ is the quality estimation of splitting C_k into subconcepts C_{k_i}.

Explanation and analysis of this hybrid representation can be found in [15]. Below is an example of concepts discovered by OSHAM

CONCEPT 43
Level = 5, Super_Concepts = {29}, Sub_Concepts = {52, 53}
Features = (Uniformity of Cell Size, 1) ∧ (Bare Nuclei, 1) ∧ (Bland Chromatin, 1) ∧
(Uniformity of Cell Shape, 2)
Local_instances/Covered_instances = 6/25
Local_instances = {8, 127, 221, 236, 415, 661}
Concept_probability = 0.041666
Local_instance_conditional_probability = 0.240000
Concept_dispersion = 0.258848
Local_instance_dispersion = 0.055556
Subconcept_partition_quality = 0.519719

Table 5 presents the essential ideas of the main algorithm in OSHAM which allows to discovering both disjoint and overlapping concepts depending on the user's interests by refining the condition 1.(a) and the intersection operation. In [16] we corrected and improved the interpretation procedure for OSHAM introduced in [15] that combines the concept intent, hierarchical structure information, probabilistic estimations and the nearest neighbors of unknown instances.

Table 4. Framework for unsupervised induction

1. While C_k is still splittable, find a new subconcept of it that corresponds to the hypothesis minimizing the quality function $q(C_k)$ among η hypotheses generated by the following steps
   (a) Find a "good" attribute-value pair concerning the best cover of C_k.
   (b) Find a closed attribute-value subset S containing this attribute-value pair.
   (c) Form a subconcept C_{k_i} with the intent is S.
   (d) Evaluate the quality function with the new hypothesized subconcept.
   Form intersecting concepts corresponding to intersections of the extent of the new concept with the extent of existing concepts excluding its superconcepts.
2. If one of the following conditions holds then C_k is considered as unsplittable
   (a) There exist not any closed proper feature subset.
   (b) The local instances set C_k^r is too small.
   (c) The local instances set C_k^r is homogeneous enough.
3. Apply recursively the procedure to concepts generated in step 1.

3.2 Evaluation

A way to evaluate unsupervised learning system is to employ supervised data but hide the class information in the whole learning and interpreting phases and use the class information only to estimate the predictive accuracy. We employ this way to evaluate unsupervised learning systems where the predicted name of each learned concept C_k

is determined by the most frequently occurring name of instances in $e(C_k)$. With this predicted name of learned concepts, the error rate of an unsupervised learning system can be estimated as the ratio of the number of testing instances correctly predicted regarding the predicted name over the total number of testing instances [18].

Table 5. Predictive accuracies of AUTOCLASS and OSHAM

datasets	attributes		inst.	class	AUTO-CLASS	OSHAM
	disc	cont				
Wisconsin breast cancer	9	–	699	2	**96.6**	92.6
Congressional voting	17	–	435	2	91.2	**93.7**
Mushroom	23	–	8125	2	86.5	**88.2**
Tic-tac-toe	9	–	862	9	82.3	**92.6**
Glass identification	–	9	214	6	55.7	**65.3**
Ionosphere	–	35	351	2	**91.5**	84.6
Waveform	–	21	300	3	59.2	**73.0**
Pima diabetes	–	8	768	2	68.2	**72.7**
Thyroid (new) disease	–	6	215	3	**89.3**	84.6
Heart disease cleveland	8	5	303	2	49.2	**60.8**

Table 5 report the predictive accuracies of AUTOCLASS [3] and OSHAM, estimating on ten datasets from the UCI repository of machine learning databases. The numbers of attributes (discrete and continuous), instances and "natural" classes of these datasets are given in columns 2–5.

All experiments on these datasets are carried out with 10-fold cross validation. For AUTOCLASS, we use the public version AUTOCLASS-C implemented in C and run three steps of *search, report* and *predict* with the default parameters. The predicted name and predictive accuracy of AUTOCLASS and OSHAM are obtained as mentioned above. Some conclusions can be drawn from these experiments.

- The predicted name obtained in OSHAM and AUTOCLASS by the majority of occurring name of instances in concepts is different from the concept name obtained in supervised learning (e.g., C4.5) using the pruning threshold based on the class information. An unsupervised concept in the worse case may contain nearly equal numbers of instances belonging to different natural classes, and an unsupervised classification may be failed in distinguishing very similar instances. It explains that while the predictive accuracies between these supervised and unsupervised methods look not so different, they are slightly different in nature.

- The predictive accuracies of OSHAM and AUTOCLASS in these experiments are only slightly different. In these first trials, each system is better in several datasets and these two systems can be considered having comparable performance.

- One advantage of OSHAM is its concept hierarchies can be easily understood by its extended classical view on concepts and the graphical support.

4 An Interactive-Graphic Environment

We address the improvement and implementation of CABRO and OSHAM in order to deal with different situations of the practical use mentioned in section 1.

CABRO and OSHAM are originally designed for discrete attributes with unordered nominal values. We choose the discretization of continuous attributes into discrete ones before learning process. For continuous attributes in supervised data, we employ the recursive entropy minimization based on *Minimum Description Length* according to the experimental analysis in [6]. For continuous attributes in unsupervised data, we use the well-known *k-means clustering* [9]. In fact, for each continuous attribute the k-means algorithm is applied to cluster its values into k groups ($k = 1, 2, ..., K$). A criterion based on within-class and between-class similarities with the Euclidean distance is used to choose a value of k that corresponds to the best partition according to this criterion.

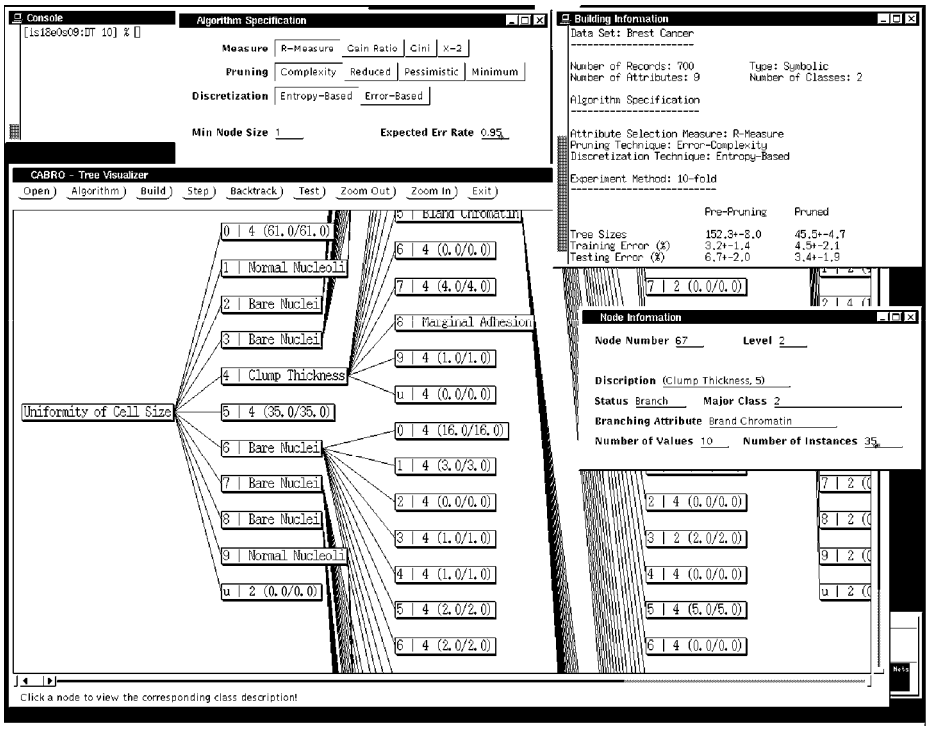

Figure 2. Generating decision trees by CABRO

CABRO and OSHAM are originally designed for a nonincremental environment. As the execution cost of CABRO is low even with large datasets, this program can be applied effectively in an incremental environment by reapplying it for the whole updated dataset. As the execution cost of OSHAM is relatively high, we have started to develop INCOSHAM – an incremental algorithm derived from OSHAM – that extracts a concept hierarchy from the hypothesis space with the Galois lattice structure. INCOSHAM preserves the nonexhaustive search strategy of OSHAM and exploits only the relevant part of the hypothesis space [17].

Recently, by combining common features between the rough set theory and formal concept analysis, a theory of rough concept analysis with the slogan "rough set + formal concept = rough formal concept" was introduced [21]. The rough concept analysis provides a framework for representing and learning *approximate concepts*. In this framework we developed unsupervised conceptual clustering method A-OSHAM, inspired by OSHAM, for inducing concept hierarchies with their approximations [19]. Concept approximations allow us to refine the common outcomes of predicting unknown instances.

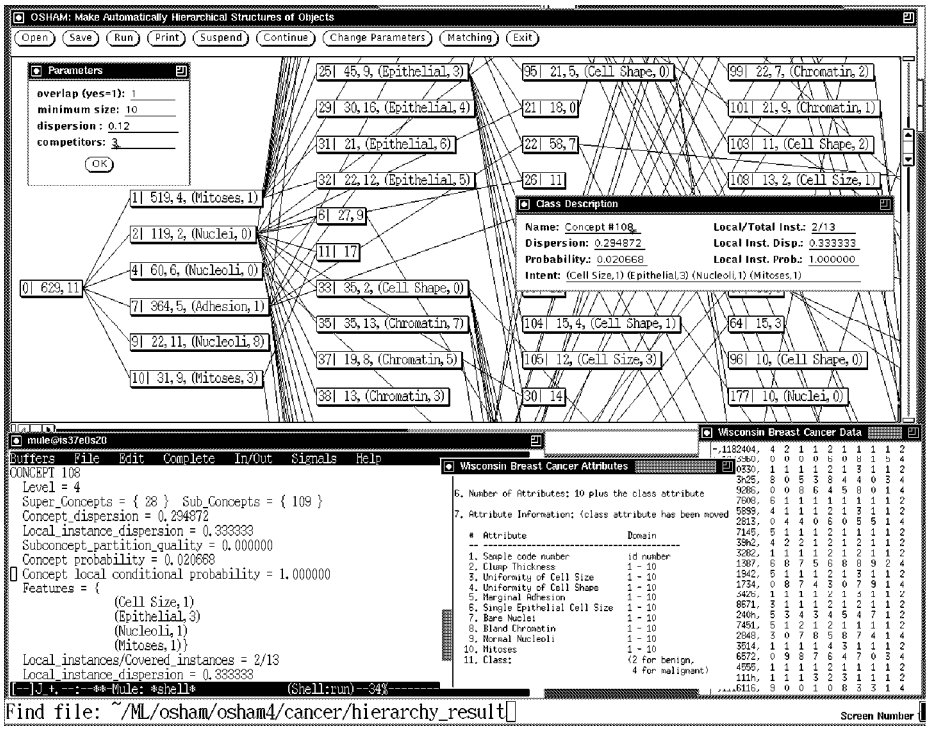

Figure 3. Generating hierarchies with overlapping concepts

Figure 2 shows a generated decision tree by CABRO and Figure 3 shows a main screen of the interactive OSHAM with an overlapping concept hierarchy learned from the Wisconsin breast cancer dataset.

We are investigating feature selection techniques to deal with irrelevant attributes, or techniques to mitigate the noise effect, the missing data in a pre-treatment process before using CABRO and OSHAM.

We are now integrating programs CABRO and OSHAM in a common system implemented in the X Window on the workstation with the direct manipulation style of interaction [11]. The conceptual architecture of the system is shown in Figure 4. This system accepts input in various situations of application domains (e.g., Boolean, symbolic, numeric attributes, nonincremental or incremental data, supervised or unsupervised data) and results as output decision knowledge that can be used for KBSs.

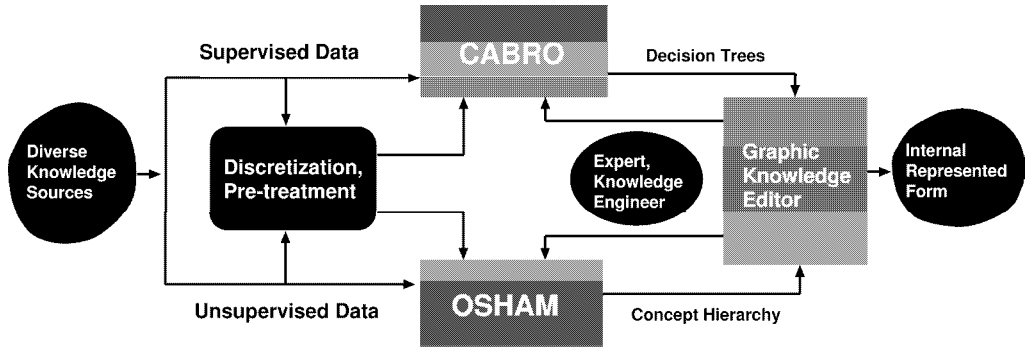

Figure 4. Conceptual architecture of the system

As introduced in subsection 1.3, the Tree Visualizer gives the users a graphical view of both decision tree/concept hierarchy structure and the detail information of each node, in spite of the size of the tree/hierarchy. To gain a full comprehension of the tree/hierarchy, the user can navigate through tree/hierarchy structure, switch among several view modes, or choose alternatively which parts of the tree/hierarchy to be displayed.

In the interactive mode of operation, the users can easily make a concrete decision tree/concept hierarchy building algorithm just by selecting a combination among various techniques for attribute selection, pruning and discretization problems, etc. The cycle of changing parameters, generating tree/hierarchy, testing and comparing can also be faster and more effective. Moreover, the users can take more control in the building process by run it step by step, examine intermediate tree/hierarchy, backtrack or go-forward in order to find a high potential trees/hierarchies with respect to the categorization scheme.

5 Conclusion

We have briefly presented the main ideas of our current project for knowledge discovering in databases which is based on two methods CABRO and OSHAM. The relevant domains for CABRO and OSHAM probably are those with one-step classification and prediction tasks, or with some form of multi-step inference or problem solving. With the high prediction accuracy of CABRO and OSHAM and the effectiveness of the interactive-graphic environment as illustrated in this paper, we expect that the project will achieve its ultimate goals and provide an environment for discovering high-quality knowledge in data with low-cost.

Acknowledgements

This work is supported by Kokusai Electric Co., Ltd., Monbusho (Ministry of Education, Science, Sports and Culture) and JAIST (Japan Advanced Institute of Science and Technology, Hokuriku). Thanks are also given to the donors and maintainers of the UCI Repository for providing access to the databases.

References

[1] Belton, V. and Elder, M.D., "Decision Support Systems: Learning from Visual Interactive Modelling", *Decision Support Systems*, Vol. 12, 1994, 355–364.

[2] Breiman, L., Friedman, J., Olshen, R., Stone, C., *Classification and Regression Trees*, Belmont, CA: Wadsworth, 1984.

[3] Cheeseman, P., Stutz, J., "Bayesian classification (AutoClass): Theory and results ", *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad *et al.* (Eds.). AAAI Press/MIT Press, 1996, 153–180.

[4] Clancey, W.J., "The knowledge level reinterpreted: Modeling Socio-technical systems", *International Journal of Intelligent Systems*, Vol. 8 (1), 1993, 33–49.

[5] Diday, E., "Des objets de l'analyse de données à ceux de l'analyse de connaissance", in *Induction Symbolique et Numérique à partir de Données*, Y. Kodratoff and E. Diday (Eds.), Cepadue Editions, 1991, 9–75.

[6] Dougherty, J., Kohavi, R., Sahami, M., "Supervised and unsupervised discretization of continuous features", in *Proceedings 12th International Conference on Machine Learning*, San Francisco, 1995, 194–202.

[7] Fayyad, U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., "From data mining to knowledge discovery: An overview", in *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad et al. (Eds.), AAAI Press/MIT Press, 1996, 1–36.

[8] Gaines, B.R., "Transforming rules and trees into comprehensible knowledge structures", in *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad et al. (Eds.), AAAI Press/MIT Press, 1996, 205–226.

[9] Hartigan, J.A., *Clustering Algorithms*, Wiley, New York, 1975.

[10] Hebral, G., "The SODAS project: A software for symbolic data analysis", *Proceedings Data Science, Classification and Related Methods*, 1996, 175–178.

[11] Helander, M., *Handbook of Human-Computer Interaction* (Ed.), Elsevier Science Publisher, 1991.

[12] Ho, T.B., Nguyen, T.D., Kimura, M., "Induction of Decision Trees Based on the Rough Set Theory", in *Data Science, Classification and Related Methods*, C. Hayashi et al. (Eds.), Springer-Verlag Tokyo, June 1997 (in press).

[13] Ho, T.B., Nguyen, T.D., "Integrating Human Factors With An Concept Formation Process", *6th International Conference on Human-Computer Interaction*, Yokohama, July 1995, 74.

[14] Ho, T.B., "An Approach to Concept Formation Based on Formal Concept Analysis ", *Journal IEICE Trans. Information and Systems*, E78-D, 1995, 553–559.

[15] Ho, T.B., "A Hybrid Model for Concept Formation", in *Information Modelling and Knowledge Bases VII*, Y. Tanaka et al. (Eds.), IOS Press, 1996, 22-35.

[16] Ho, T.B., "Discovering and Using Knowledge From Unsupervised Data", to appear in *Decision Support Systems*, Elsevier Science, June 1997.

[17] Ho, T.B., "Incremental Conceptual Clustering in the Framework of Galois Lattice", in *KDD: Techniques and Applications*, H. Lu, H. Motoda and H. Luu (Eds.), World Scientific, 1997, 49–64.

[18] Ho, T.B., Luong, C.M., "Using Case-Based Reasoning in Interpreting Unsupervised Inductive Learning Results", *International Joint Conference on Artificial Intelligence IJCAI'97*, Nagoya, August 1997, 258-263.

[19] Ho, T.B., "Acquiring Concept Approximations in the Framework of Rough Concept Analysis", *7th European-Japanese Conference on Information Modelling and Knowledge Bases*, Toulouse, May 1997, 186–195.

[20] Kangassalo, H., "On the concept of concept for conceptual modelling and concept detection", in *Information Modelling and Knowledge Bases III*, S. Ohsuga et al. (Eds.), IOS Press, 1992, 17–58.

[21] Kent, R.E., "Rough concept analysis", in *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, 1994, 248–255.

[22] Kervahut, T. and Potvin, J.Y., "An interactive-graphic environment for automatic generation of decision trees", *Decision Support Systems*, Vol. 18, 1996, 117–134.

[23] Langley, P., *Elements of Machine Learning*, Morgan Kaufmann, 1996.

[24] Lee, H.Y., Ong, H.L., Quek, L.H., "Exploiting visualization in knowledge discovery ", *Proceedings of First International Conference on Knowledge Discovery and Data Mining*, Montreal, 1995, 198–203.

[25] Mingers, J., "An Empirical Comparison of Selection Measures for Decision Tree Induction", *Machine Learning*, 3, 1989, pp. 319–342.

[26] Ohsuga, S., "Aspects of conceptual modelling - As kernel of new information technology", in *Information Modelling and Knowledge Bases VII*, Y. Tanaka et al. (Eds.), IOS Press, 1996, 1-21.

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

[28] Pawlak, Z. (1991) *Rough Sets – Theoretical Aspects of Reasoning About Data*, Kluwer, 1991.

[29] Rumbaugh,J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., *Object-Oriented Modelling and Design*, Prentice Hall, 1991.