Decision Support Systems, Elsevier Science, Vol. 21 (1997), No. 1, 27-41

Discovering and Using Knowledge From Unsupervised Data

Tu Bao Ho

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

Abstract: Though most knowledge discovery methods have been developed for supervised data, the task of finding knowledge from unsupervised data often arises in real-world situations. Without feedback about the appropriateness of discovered knowledge as in supervised systems, techniques for unsupervised knowledge discovery are essentially different and still much less developed than those for supervised discovery. In this paper we present a method for discovering and using classificatory knowledge from unsupervised data. We first extend the classical view on concepts commonly used in the framework of the Galois lattice by combining it with the prototype and exemplar views, and develop an algorithm for inducing concept hierarchies. We then introduce a procedure that combines matching approaches in inductive learning with case-based reasoning in order to classify unknown cases using discovered knowledge. We present the implementation of the method as an interactive system. An experimental comparative study of some knowledge discovery systems in terms of knowledge description and prediction shows advantages and the application potential of the method in decision making.

Keywords: knowledge discovery, unsupervised data, views on concepts, concept hierarchy, matching approaches, case-based reasoning.

1 Introduction

As knowledge is of central importance in intelligent systems, to acquire useful knowledge is one crucial problem in the development of knowledge-based decision support systems [7], [30]. Recently, knowledge discovery in databases (KDD) has emerged as a rapidly growing interdisciplinary field that merges together techniques of databases, statistics and machine learning in order to find useful knowledge from databases, and KDD has shown its application potential in various domains [8], [10].

Knowledge discovery techniques depend entirely on the degree of supervision in data. Most early work in knowledge discovery focused on the *supervised* task that aimed at finding descriptive knowledge of concepts given their labelled instances.

Supervised discovery methods are always driven by feedback about the appropriateness of discovered knowledge obtained from classified data. Moreover, knowledge discovered from supervised data is often represented in flat structures such as conjunctions of conditions [26], production rules [6], or in the decision tree structure [3], [28], that lead to relatively simple ways of using knowledge (final rules or leaf nodes in decision trees) in predicting classes of unknown cases. However, in real-world situations the *unsupervised* discovery task can also arise quite often that aims at finding in databases "natural" classes with their descriptions where no feedback about the appropriateness of discovered knowledge can be obtained from data. Essentially, unsupervised knowledge discovery concerns two mutual problems: (1) *hierarchical clustering* (i.e., finding a hierarchy of useful subsets of unlabelled instances), and (2) *characterization* (i.e., finding an intensional definition for each of these instance subsets). In consequence, knowledge from unsupervised data seems suitably to be discovered in hierarchical structures such as concept hierarchies that influence the ways of predicting unknown cases, as in which many concepts can be detected and used at intermediate levels [17].

Techniques of unsupervised discovery are essentially characterized by the search for regularities in data and are still much less developed than those of supervised discovery [22]. Basically, unsupervised discovery methods depend strongly on how concepts are understood and represented. Among *views on concepts* in cognitive science and machine learning, the classical, prototype and exemplar ones are widely known and used [19], [31]. Moreover, unsupervised discovery systems compose solutions to the problem by employing one or more of three main *categorization constraints* based on similarity, feature correlation [12], and syntactical structure of the concept hierarchy [24]. The properties of concepts, main strengths and limitations of views on concepts and categorization constraints are given in Table 1 which are summarized from [19], [31], [33].

COBWEB [9] and AUTOCLASS [5] are often referred to as typical unsupervised learning methods which employ the probabilistic (prototype) representation for concepts. COBWEB organizes data so as to maximize inference ability, and AUTOCLASS uses a Bayesian method for determining the optimal classes. CLUSTER/2 [26] is one early influential conceptual clustering method that employs the classical view on concepts to form categories with 'good' conjunctions of common features to all category members. Recently, using the classical view on concepts in the Galois lattice structure, several concept learning systems have been developed. In system GALOIS [4] and in [11], all possible concepts in the Galois lattice are incrementally generated. Although the Galois lattice provides a powerful structure for discovering concepts, this framework has considerable limitations in KDD where the user has to deal often with large databases. To find all possible formal concepts is perhaps not always tractable as in the worse case the number of concepts can be exponential in the size of $\mathcal{O}$ and $\mathcal{A}$. For example, with the commonly used data set Congressional voting of 435 instances described by 16 attributes, the Galois lattice consists of about 150,000 nodes that reaches the space limit of a Sparcstation LX equipped with 32 Mbytes of RAM and the execution time increases dramatically [4]. Moreover, the classical representation of concepts in the Galois lattice does not capture typicality effects and vagueness. In [16], an alternative approach to hierarchical conceptual clustering is proposed which extracts a part of the Galois lattice in the form of a concept hierarchy.

As analysed in [31], each system relying on a single view on concepts has several limitations in capturing the rich variety of conceptual knowledge. Therefore, hybrid systems attempt to improve the concept learning process by combining fairly different theoretical views on concepts and constraints of categorization.

Features listing	
– Nonnecessary features:	features are true only for some/most of concept members.
– Disjunctive concepts:	perhaps no feature shared by all concept members.
– Relational information:	features are about object's function/relation to others.
– Features as concepts:	features are not only atomic units of description.
Internal structure	
– Typicality:	concept members have different typical/representative roles.
– Basic levels:	many concepts are viewed at an intermediate level.
– Superordinate distance:	concept is not always rated most similar to its parents.
Categorization	
– Unclear cases:	concept membership varies in different objects.
– Context effects:	categorization depends on context (available information).
– Multiple categorization:	many categories could apply to an object.
Classical view	*concepts are viewed by necessary and sufficient features*
– strengths	clear semantics, used by standard logic.
– limitations	difficult to specify defining features in many cases, does not capture typicality effects and vagueness.
Prototype view	*concepts are viewed by most common or typical features*
– strengths	generality, flexibility.
– limitations	does not preserve enough information, does not address context effects and explain concept coherence.
Exemplar view	*concepts are viewed by individual examples*
– strengths	conserve information and context sensitivity.
– limitations	ignore generalization, does not explain the concept coherence.
Similarity constraint	categories consist of similar objects
Correlation constraint	maximize intra-correlations and minimize inter-correlations
Structure constraint	syntactical structure of the entire conceptual system

Table 1. Properties, views on concepts and categorization constraints

The two primary goals of discovery systems in practice are *description* and *prediction* [8]. Description focuses on finding human-interpretable patterns describing the data, and relates to the understandability of knowledge. Prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest. The first lesson learned from real databases in [5] is that "discovery of patterns in data is only the beginning of a cycle of interpretation followed by more testing". Though how to use results of knowledge discovery in prediction is an important issue in practice [22], there has been so far little work in KDD associated with procedures for exploiting discovered knowledge, e.g., [1], [34].

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. Without feedback obtained from data about discovered knowledge, unsupervised discovery systems cannot produce maximally useful results when operating alone. The ability to interact with the user allows considerable improvements in the performance of discovery systems [20], [23], [35].

The motivation of this work is to extend and develop the unsupervised discovery method OSHAM introduced in [16], according to the aspects mentioned above. Three themes about discovering knowledge, using knowledge, system implementation and evaluation will be addressed in this paper. In section 2, we enrich the classical view on concepts in the Galois lattice by several features of the prototype and exemplar views, and develop an algorithm for inducing a concept hierarchy from the Galois lattice. In section 3 we propose a procedure for using discovered knowledge to classify unknown cases based on an integration of matching approaches in inductive learning with case-based reasoning. In section 4 we present the implementation of method in the X Window with the direct manipulation style of interaction, and an experimental comparative study of some knowledge discovery methods in terms of knowledge description and prediction accuracy. In section 5 we address the application potential of the method in decision making and conclusions.

2 Discovering knowledge from unsupervised data

2.1 Representing concepts

A *database* is a collection of data organized logically into files or tables of fixed-length *records* (objects), described by a set of *attributes*. Each attribute is defined with a set of potential values known as its *domain*. Information about attributes and their domains is often maintained in a separate *data dictionary*. Each record is an ordered list of values, one value for each field. A *tuple* is a conjunction of attribute-value pairs.

For simplicity of representation, we limit ourselves in considering a single relational database. Denote by $\mathcal{O}$ the set of all objects (records), $\mathcal{A}$ the set of all attributes, and $\mathcal{T}$ the set of all possible tuples in the database. For any object subset $X \subseteq \mathcal{O}$, the largest tuple common to all objects in X is denoted by $\lambda(X)$. For any tuple $S \in \mathcal{T}$, the set of all objects satisfying S is denoted by $\rho(S)$. A tuple S is *closed* if $\lambda(\rho(S)) = S$.

The basis of the most widely understanding of a concept is the function of collecting individuals into a group with certain common properties. One distinguishes these common properties as the *intent* of the concept that determines its *extent* which are objects accepted as members of the concept. Formally, a *concept* C in the classical view is a pair (X, S), $X \subseteq \mathcal{O}$ and $S \subseteq \mathcal{T}$, satisfying $\rho(S) = X$ and $\lambda(X) = S$. X and S are then called extent and intent of C, respectively. Concept (X_2, S_2) is a *subconcept* of concept (X_1, S_1) if $X_2 \subseteq X_1$ which is equivalent to $S_2 \supseteq S_1$, and (X_1, S_1) is then a *superconcept* of (X_2, S_2). For simplicity of representation, we sometimes call the direct superconcepts (father concepts) and the direct subconcepts (son concepts) of a concept by superconcepts and subconcepts, respectively.

It was mathematically shown that the set of all possible concepts associated with the superconcept-subconcept relation has the structure of a complete lattice (called also the Galois lattice), and λ and ρ define a Galois connection between the power sets $\wp(\mathcal{O})$ and $\wp(\mathcal{A})$, i.e., they are two order-reversing one-to-one operators [32]. To overcome the limitations of the classical representation of concepts in the Galois lattice, we enrich this representation by adding several components based on the prototype and exemplar views on concepts that allow dealing better with typical or unclear cases in the region boundaries, and propose a clustering method to extract a concept hierarchy in the Galois lattice of possible concepts.

A *concept hierarchy* $\mathcal{H}$ is a part of the concept lattice satisfying the following properties (1) the root concept $(\mathcal{O}, \lambda(\mathcal{O})) \in \mathcal{H}$; (2) if $C_1 = (X_1, S_1)$, $C_2 = (X_2, S_2) \in \mathcal{H}$ and $X_1 \cap X_2 \neq \emptyset$ then either C_1 is a subconcept of C_2 or C_2 is a subconcept of C_1. Note that the property $(\{o\}, \lambda(o)) \in \mathcal{H}$ for every $o \in \mathcal{O}$ in the usual definition of the hierarchy structure is not necessarily required here as for the generalization purpose many concepts at high levels need to be pruned.

A concept hierarchy is formed by OSHAM in the top-down direction with different levels of generality, from the most general (root concept) to the most specific concept (leaf concept) in each branch. Associated with each concept C_k are the *level* $l(C_k)$, two lists of its direct superconcepts $f(C_k)$ and direct subconcepts $s(C_k)$. The intent $i(C_k)$ is inherited by all subconcepts C_{k_i} of C_k, and thus the intent of each subconcept C_{k_i} is the conjunction of $i(C_k)$ with selected attribute-value pairs. The extent $e(C_k)$ is classified into extent of its subconcepts $C_{k_1}, C_{k_2}, ..., C_{k_n}$ at higher levels in the process of analyzing C_k into subconcepts. In fact, this is a process of searching for regularities among instances of C_k for determining good non-necessary features that correspond to useful subconcepts of C_k. In this process it may happen that some instances of C_k do not satisfy features of any subconcept C_{k_i}, and so will not be classified as instances of any subconcept C_{k_i}. We call these *local* instances of C_k and denote this set by $C_k^r = e(C_k) \setminus \bigcup_{i=1}^n e(C_{k_i})$. In fact, OSHAM splits each concept C_k into subconcepts until the set C_k^r satisfies some unsplittable conditions. The *probability of occurrence* $p(C_k)$ of C_k and the *conditional probability* $p(C_k^r \mid C_k)$ of local instances in C_k are of interest as they will be used later in the prediction of unknown instances.

Without class attribute to drive the search, unsupervised discovery techniques often exploit the relation between intra-similarity and inter-similarity among concepts. The *distance* $\delta(o_p, o_q)$ between two instances o_p and o_q is defined as an extension of Jaccard distance [2], suitably in the Galois lattice

$$\delta(o_p, o_q) = 1 - \frac{\sum_{a \in \lambda(\{o_p, o_q\})} \gamma(a)}{\sum_{a \in \lambda(\{o_p\}) \cup \lambda(\{o_q\})} \gamma(a)} \tag{1}$$

where $\gamma(a) \in \mathbb{Z}^+$ are positive integer weights of attributes a (with value 1 by default). The attribute weights $\gamma(a)$ embody background knowledge about the environment and concepts (importance of attributes, attribute rank by potential relevancy, etc.). The *dispersion* $d(C_k)$ between instances in $e(C_k)$, considered as the inverse of the homogeneity of $e(C_k)$, is defined as the average distance between all pairs of instances in $e(C_k)$

$$d(C_k) = \frac{2 \times \sum_{o_p, o_q \in e(C_k)} \delta(o_p, o_q)}{\mid e(C_k) \mid \times \mid e(C_k) - 1 \mid} \tag{2}$$

If C_k is a non-leaf concept, its local instances in C_k^r may be considered to be more typical and representative than its instances classified into subconcepts C_{k_i}. As an instance o is a member of different concepts along a branch in the concept hierarchy, the concept C_k that $o \in C_k^r$ is of particular interest. If C_k is a leaf concept, we have $e(C_k) = C_k^r$ and its instances are all considered having the same representative role.

The extent of all direct subconcepts $C_{k_1}, C_{k_2}, ..., C_{k_n}$ of C_k and the set of local instances C_k^r form a partition P of $e(C_k)$. Denote by $W(C_k)$ the average of all $d(C_{k_i})$ and $d(C_k^r)$. The dissimilarity between subconcepts of C_k, denoted by $B(C_k)$, is defined as the average of distances $\Delta(c(C_{k_i}), c(C_{k_j}))$ between all pairs $(C_{k_i}, C_{k_j}) \in P$, where the distance $\Delta(c(C_{k_i}), c(C_{k_j}))$ is determined as the smallest distance among the distances of all pairs of instances $o_p \in C_{k_i}$ and $o_q \in C_{k_j}$

$$\Delta(c(C_{k_i}), c(C_{k_j})) = Min_{o_p \in C_{k_i}, o_q \in C_{k_j}} \delta(o_p, o_q) \tag{3}$$

The *quality* of splitting a concept C_k into subconcepts in the next level, denoted by $q(C_k)$, is measured by

$$q(C_k) = W(C_k)/B(C_k) \tag{4}$$

Summarily, OSHAM represents each concept C_k in the concept hierarchy by a 10-tuple of the following components

$$< l(C_k), f(C_k), s(C_k), e(C_k), i(C_k), p(C_k), d(C_k), p(C_k^r \mid C_k), d(C_k^r), q(C_k) > \tag{5}$$

2.2 Discovering concept hierarchies

This discovery process is characterized by splitting recursively each existing concept into subconcepts at higher levels without knowing *a priori* the number of subconcepts. OSHAM tends to find a concept hierarchy with sufficiently general and discriminant concepts represented by (5). This tendency lies in the fact that the generality and discrimination of concepts are two dual characteristics, i.e., the more general the less discriminant concepts. There is no way to estimate directly the discrimination of concepts from unsupervised data, however the discrimination ability concerns the partition quality and can be indirectly estimated, such as by the similarity between-class and within-class of a partition. This similarity is estimated based on different measures, as category utility in COBWEB [9], the intercorrelation among variables in WITT [12], or formulas (1)-(4) in OSHAM. For each concept C_k, in general there are many possible tuples derived from $e(C_k)$ that can be used to form subconcepts C_{k_i}, and the quality of the corresponding partition according to (4) differs greatly. OSHAM aims at extracting sequentially general subconcepts C_{k_i} (with large extent), but among the hypothesized tuples which may generate hypothesized C_{k_i} it selects which one that minimizes (4) in order to increase the discrimination.

Input	concept hierarchy $\mathcal{H}$ and an existing splittable concept C_k.
Result	$\mathcal{H}$ formed gradually.
Top-level	call OSHAM(root concept, $\varnothing$).
Variables	α, β, η are given thresholds.

```
Algorithm OSHAM($C_k$,$\mathcal{H}$)

  1. Suppose that $C_{k_1}, ..., C_{k_n}$ are subconcepts of $C_k$ found so far. While $C_k$ is
     still splittable, find a new subconcept $C_{k_{n+1}}$ of $C_k$ that corresponds to the
     hypothesis with minimal $q(C_k)$ among $\eta$ hypotheses $C_{k^1_{n+1}}, ..., C_{k^\eta_{n+1}}$. Each
     hypothesis $C_{k^t_{n+1}}$ is generated by doing the following steps (a)–(d)

       (a) Find an attribute-value pair $(a^*, v^*)$ so that $\bigcup_{i=1}^{n} e(C_{k_i}) \cup \rho(\{(a^*, v^*)\})$
           is the largest cover of $e(C_k)$.
       (b) Find a maximal closed tuple $S$ containing $(a^*, v^*)$.
       (c) Form subconcept $C_{k^t_{n+1}}$ with $i(C_{k^t_{n+1}}) = S$ and $e(C_{k^t_{n+1}}) = \rho(S)$.
       (d) Evaluate $q(C_k)$ with the new $C_{k^t_{n+1}}$.

     Form intersecting subconcepts corresponding to intersections of extent of
     $C_{k_{n+1}}$ with extent of existing concepts on $\mathcal{H}$, excluding its superconcepts.

  2. Update $C_k^r = e(C_k) \setminus \bigcup_{i=1}^{n+1} e(C_{k_i})$. If one of the following conditions holds
     then $C_k$ is considered unsplittable

       (a) There exist not any closed tuple in $C_k^r$.
       (b) $\mid C_k^r \mid \leq \alpha$.
       (c) $d(C_k^r) \leq \beta$.

  3. Apply OSHAM($C_{k_i}$,$\mathcal{H}$) to each $C_{k_i}$ formed in the step 1.
```

Table 2: The OSHAM algorithm

Algorithm OSHAM described in Table 2 forms gradually and recursively a concept hierarchy $\mathcal{H}$, initially with the root concept whose extent is the set of all instances of $\mathcal{O}$ and its intent is $\lambda(\mathcal{O})$ which is often empty. In each recursive application to an existing splittable concept C_k, OSHAM will split C_k sequentially into subconcepts C_{k_i}, until an unsplittable condition holds. Conditions 2(a)-(c) determine whether the concept C_k is possible or worthwhile to split further. In particular, 2(a) ensures that there exists at least one admissible subconcept of C_k, 2(b) guarantees to consider only concepts that cover at least a minimum number α of instances, and 2(c) prevents splitting C_k when its local instances are rather homogeneous.

```
*MaxCoverage*($a^*, v^*, C_k, \mathcal{H}$)
    Find $(a^*, v^*) \in \mathcal{T}_{C_k}$ satisfying
      $| \bigcup_{i=1}^{n} e(C_{k_i}) \cup \rho(\{(a^*, v^*)\}) | = max_{(a,v)\in\mathcal{T}_{C_k}} | \bigcup_{i=1}^{n} e(C_{k_i}) \cup \rho(\{(a, v)\}) |$
    If this maximum holds at several $(a^*, v^*)$ then choose arbitrary one $(a^*, v^*)$ that
    minimizes $| \bigcup_{i=1}^{n} e(C_{k_i}) \cap \rho(\{(a^*, v^*)\}) |$ (referred to as the minimum intersection
    condition).

*MaxClosedTuple*($a^*, v^*, C_k, \mathcal{H}$)
(Find the closed tuple containing a given attribute-value pair $(a^*, v^*)$
    Let $S = \{(a^*, v^*)\}$.
    For every $(a, v) \in \mathcal{T}_{C_k} \setminus (a^*, v^*)$ do if $\rho(\{(a, v)\}) = \rho(\{(a^*, v^*)\})$
    then $S = S \wedge (a, v)$.

*ClosedTuple*($C_k, \mathcal{H}$)
(Verify whether there exists a closed tuple in $C_k^r$)
    1. Determine $(a^*, v^*)$ that satisfies $\varphi(\{(a^*, v^*)\}) = max_{(a,v)\in\mathcal{T}_{C_k}} \varphi(\{(a, v)\})$.
    2. Determine the tuple $S = \{\wedge(a, v) \in \mathcal{T}_{C_k} \mid \rho(\{(a, v)\}) = \rho(\{(a^*, v^*)\})\}$.
    3. If $\varphi(S) < 1$ then return *success* with $S$ else return *failure*.

*IntersectionConcept*($\mathcal{H}$,$S$)
(Form intersecting concepts from a given concept $(\rho(S), S)$)
    1. For every existing concept $(\rho(S'), S')$ on $\mathcal{H}$, excluding superconcepts of
       $(\rho(S), S)$, if $\rho(S) \cap \rho(S') \neq \varnothing$ then create the intersecting concept $C_h =$
       $(\rho(S''), S'')$. The extent $\rho(S'')$ is the intersection of the extent of two con-
       stituent concepts, $\rho(S'') = \rho(S) \cap \rho(S')$. The intent $S''$ is the closed attribute
       set found by procedure MaxClosedTuple $(a^*, v^*, C_h, \mathcal{H})$ where $(a^*, v^*)$ is one
       attribute-value pair chosen arbitrarily from $\lambda\, \rho(S'')$.
    2. Apply recursively IntersectionConcept $(\mathcal{H}, S'')$.
```

Table 3: The OSHAM algorithm

OSHAM is refined by several auxiliary procedures described in Table 3: 1(a) by *MaxCoverage* $(a^*, v^*, C_k, \mathcal{H})$, 1(b) by *MaxClosedTuple* $(a^*, v^*, C_k, \mathcal{H})$, and 2(a) by *ClosedTuple* $(C_k, \mathcal{H})$. In these procedures, $\mathcal{T}_{C_k}$ stands for the set of attribute-value pairs which are different from those used in the branch from the root concept to the concept C_k being considered, and $\varphi(S) = \mid \rho(S) \mid / \mid \mathcal{O} \mid$ for $S \in \mathcal{T}$. The intersection condition described in *MaxCoverage* and the procedure *IntersectionConcept* allow OSHAM to discover overlapping concepts (Figure 1). By using the constraint $e(C_{k_i}) \cap \rho(\{(a^*, v^*)\}) = \varnothing$ in *MaxCoverage*, OSHAM is able as well to discover disjoint concepts. The difference and benefits of disjoint and overlapping concepts are addressed in more detail, e.g., in [9], [25]. In fact, the interactive-grahic system OSHAM, described in section 4, is capable to discover both disjoint and overlapping concepts according to the user's interest and applications.

Algorithm OSHAM is originally described for discrete attributes with unordered nominal values. In the current version, continuous attributes are discretized before learning process by k-means clustering [13]. 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 similar to (4) with the Euclidean distance is used to choose a value of k that corresponds to the best partition according to this criterion.

3 Using discovered knowledge

Decision making is the process of choosing among alternative courses of actions for the purpose of attaining a goal or goals [30]. In a decision tree obtained from supervised data, all training instances are covered by leaf concepts and only leaf concepts are considered as possible goals. In a concept hierarchy obtained from unsupervised data, training instances are covered by either non-leaf concepts or leaf concepts, and non-leaf concepts may also be considered as possible goals.

There are three broad classes of interpreters for discovered knowledge [22] which can be applied to decision making. The *logical* approach carries out an "all or none" matching process depending on whether the unknown instance satisfies the concept intent. The *threshold* approach carries out a partial matching process and employs some threshold to determine an acceptable degree of match. The *competitive* approach also carries out a partial matching process and selects the best competitor based on estimated degrees of match. It is known that different interpreters can yield different meanings for the same representation of concepts.

As the generality is decreased along branches of $\mathcal{H}$, we say that a concept C_k *matches* an unknown instance e if C_k is the most specific concept in a branch that matches e intensionally (though all superconcepts of C_k match e). Naturally, there are three types of outcomes when matching logically $\mathcal{H}$ with e: only one concept on $\mathcal{H}$ that matches e (*single-match*), many concepts on $\mathcal{H}$ that match e (*multiple-match*), and no concept on $\mathcal{H}$ that matches e (*no-match*). Most KDD works dealing with the cases of no match and multiple-match employ a probabilistic estimation. In particular, the *measure of fit* for no match cases and *estimate of probability* for multiple-match cases in [27] have been widely adopted. Because of the nature of knowledge obtained from unsupervised data, the logical match of the concept intent does not always provide a decision with enough satisfaction. Based on different case studies, we develop a decision procedure that combines matching approaches in inductive learning with the minimum-distance classifier principle in case-based reasoning [21]. In fact, this procedure uses the logical interpretation associated with hierarchical structure information, the probabilistic estimation and the nearest neighbors of unknown instances. The nearest neighbor of e in the object set $\mathcal{O}$ and the concept in $\mathcal{H}$ to which it belongs, denoted by $NN(e)$ and $c[NN(e)]$, provide useful information to be used to reduce the risk of decision in all cases of single-match, multiple-match and no-match.

This decision procedure consists of two stages: (1) find all concepts on $\mathcal{H}$ that match e logically (intensionally), and (2) decide among these concepts which one matches e best. This procedure shares the same stages with the system POSEIDON [1], [27] but functions differently.

In the second stage, we need to determine and compare the degree of match of competitors, then choose the concept that matches e best. The satisficing degree of decision depends on how good the decision is obtained. From various case-studies we found that a concept C_k matches e well (with a low error rate) if it satisfies a majority of the following conditions: $l(C_k)$ is high, C_k is a leaf concept, $p(C_k) \times p(C_k^r)$ is high, $d(C_k)$ is low, $d(C_k^r)$ is low, and generally none of these conditions has a clearly higher priority than the others. Formally, the following functions $\tau_N, \tau_L, \tau_P, \tau_D, \tau_R$, according to the above conditions, can be used to compare two concepts C_k, C_h which match e intensionally:

$$\tau_N(C_k, C_h) = \begin{cases} 1, & \text{if } \; l(C_k) > l(C_h) \\ 0, & \text{if } \; l(C_k) = l(C_h) \\ -1, & \text{if } \; l(C_k) < l(C_h) \end{cases} \tag{6}$$

$$\tau_L(C_k, C_h) = \begin{cases} 1, & \text{if } \; C_k = \text{leaf} \wedge C_h \neq \text{leaf} \\ 0, & \text{if } \; C_k = \text{leaf} \wedge C_h = \text{leaf} \vee C_k \neq \text{leaf} \wedge C_h \neq \text{leaf} \\ -1, & \text{if } \; C_k \neq \text{leaf} \wedge C_h = \text{leaf} \end{cases} \tag{7}$$

$$\tau_P(C_k, C_h) = \begin{cases} 1, & \text{if } \; p(C_k) \times p(C_k^r \mid C_k) > p(C_h) \times p(C_h^r \mid C_h) \\ 0, & \text{if } \; p(C_k) \times p(C_k^r \mid C_k) = p(C_h) \times p(C_h^r \mid C_h) \\ -1, & \text{if } \; p(C_k) \times p(C_k^r \mid C_k) < p(C_h) \times p(C_h^r \mid C_h) \end{cases} \tag{8}$$

$$\tau_D(C_k, C_h) = \begin{cases} 1, & \text{if } \; d(C_k) < d(C_h) \\ 0, & \text{if } \; d(C_k) = d(C_h) \\ -1, & \text{if } \; d(C_k) > d(C_h) \end{cases} \tag{9}$$

$$\tau_R(C_k, C_h) = \begin{cases} 1, & \text{if } \; d(C_k^r) < d(C_h^r) \\ 0, & \text{if } \; d(C_k^r) = d(C_h^r) \\ -1, & \text{if } \; d(C_k^r) > d(C_h^r) \end{cases} \tag{10}$$

The following heuristic function τ compares the degree of match between C_k and C_h. We consider that C_k matches e *better than* C_h if

$$\begin{aligned} \tau(C_k, C_h) = {} & \theta_N \times \tau_N(C_k, C_h) + \theta_L \times \tau_L(C_k, C_h) + \\ & \theta_P \times \tau_P(C_k, C_h) + \theta_D \times \tau_D(C_k, C_h) + \theta_R \times \tau_R(C_k, C_h) \; > \; 0 \end{aligned} \tag{11}$$

where $\theta_N, \theta_L, \theta_P, \theta_D, \theta_R$ are positive weights for the importance of the level, leaf concept, local instance conditional probability, concept dispersion, and local instance dispersion (they are all set to be 1 by default).

Denote by ϕ the *satisficing degree* of decision, and by $c[e]$ the concept that matches e best. The procedure described in Table 4 relies essentially on the comparison, using the function τ, between concepts that match e intensionally and the concept containing the nearest neighbor of e. In this decision procedure, different symbolic values are assigned to the satisficing degree ϕ of decision. They reflect the decreasing rank of decision satisfaction. For example, S_1 may be considered as "best decision", M_1 as "strong decision" while N_1 as "weakly accepted decision" and N_2 as "no decision". The interpretation for different values of ϕ depends on the judgment of the user or domain experts. In order to support the user to make the final decision, all concepts that match e intensionally with their associated information as well as the best matched concept estimated by OSHAM are displayed in both text and graphical forms as described in subsection 4.1.

```
Input       concept hierarchy $\mathcal{H}$ and unknown instance $e$.
Result      best matched concept c[e] and associated satisficing degree $\phi$.
Variables   $\sigma$ is a given threshold.

Procedure Matching($\mathcal{H}, e, c[e], \phi$)

If there is only one concept $C_k \in \mathcal{H}$ that matches $e$ intensionally then

    if $c[NN(e)] = C_k$ then $c[e] \leftarrow C_k, \phi \leftarrow S_1$
    else if $\tau(C_k, c[NN(e)]) \geq 0$ then $c[e] \leftarrow C_k, \phi \leftarrow S_2$
    else $c[e] \leftarrow c[NN(e)], \phi \leftarrow S_3$.

If there are $m$ concepts $C_{i_1}, ..., C_{i_m} \in \mathcal{H}$ that match $e$ intensionally then

    Choose among them $C_{i_K}$ satisfying $\tau(C_{i_K}, C_{i_k}) \geq 0, \forall\, i_k \in \{i_1, ..., i_m\}$,
    if $C_{i_K} = c[NN(e)]$ then $c[e] \leftarrow C_{i_K}, \phi \leftarrow M_1$
    else if $\tau(C_{i_K}, c[NN(e)]) \geq 0$ then $c[e] \leftarrow C_{i_K}, \phi \leftarrow M_2$
    else $c[e] \leftarrow c[NN(e)], \phi \leftarrow M_3$.

If there is not any concept that match $e$ intensionally then

    if $\delta(NN(e), e) \leq \sigma$ then $c[e] \leftarrow c[NN(e)], \phi \leftarrow N_1$
    else $c[e] = \varnothing, \phi \leftarrow N_2$.
```

Table 4: Decision procedure for an unknown instance

4 Implementation and evaluation

In order to support producing maximally useful results, OSHAM has been implemented as an interactive-graphic system that we first describe in this section. We then present a comparative evaluation of OSHAM with other methods in terms of description and prediction. Generally, it is difficult to evaluate unsupervised discovery systems as the class information is not available and so there is less agreement on the evaluation methodology. In [9], a task of flexible prediction was introduced which requires the system to predict the values of one or more arbitrary attributes that have been excised from the test instances. Another way is to employ supervised data but hide the class information in the whole discovering and matching phases and use the class information only to evaluate discovered knowledge [24]. We employ the latter to evaluate OSHAM with the predicted name of each discovered concept C_k is the most frequently occurring name of instances in $e(C_k)$. By this way, it is possible to compare the prediction of supervised and unsupervised discovery systems.

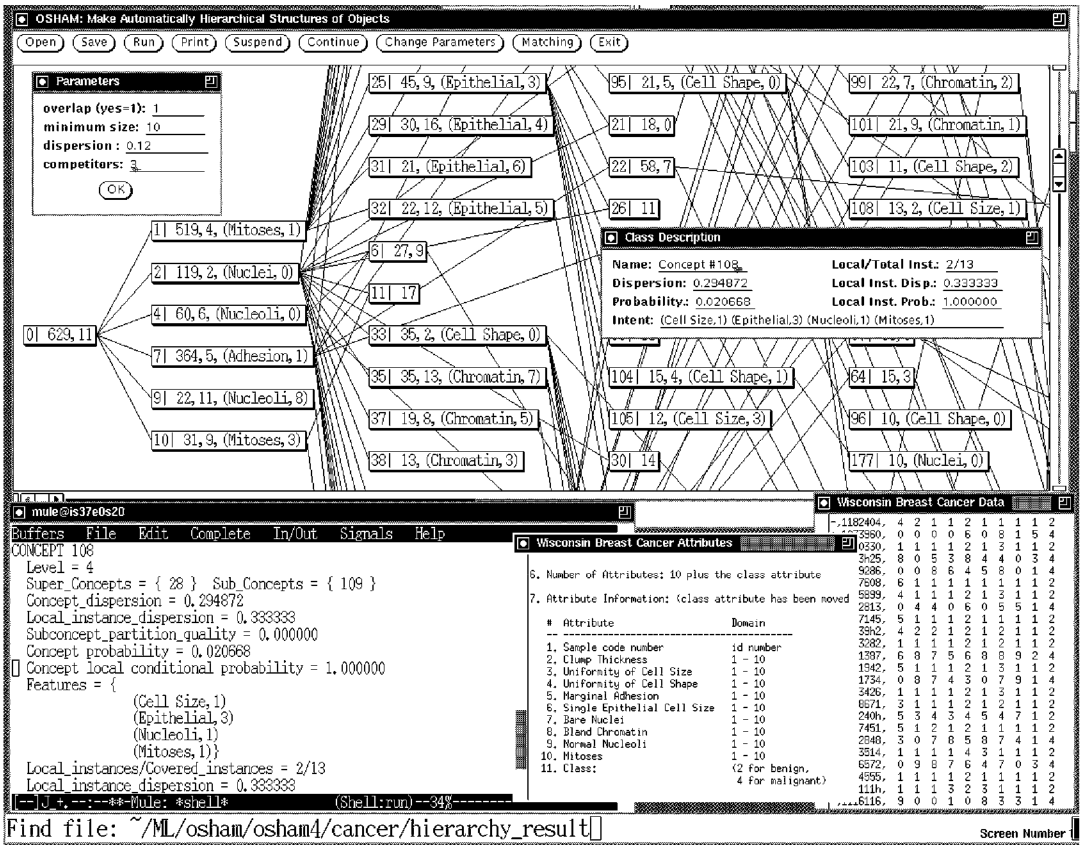
4.1 An interactive knowledge discovery system

Recently, several interactive-graphic supervised discovery systems have been developed, e.g., [20], [23], [35]. We have implemented OSHAM in the X Window on a Sparcstation with the direct manipulation style of interaction [14], that allows the user to interact with OSHAM during the discovery process. As mentioned above, OSHAM forms gradually concepts at different levels of generality in the top-down direction, and the size and form of concept hierarchies depend on the parameters α, β and η. In contrast to discovery systems with implicit parameters, such as the fixed pruning threshold in C4.5 [29], we share the view in [1] about the role of parameters in a discovery system that allows the user to explicitly modify them in the discovery process.

With a non-interactive unsupervised discovery system, the user has to run it independently at different times with various parameters, store all generated results, then compare them and choose the most suitable one. Interactive OSHAM allows the user to participate actively in the discovery process. The user can initialize parameters to cluster data, visualize the concept hierarchy gradually, observe the results and the quality evaluation, manually modify the parameters when necessary before the system continues to go further to cluster subsequent data or backtrack to regrow the concept hierarchy with respect to the categorization scheme [18].

Though a full comprehensive investigation on the sensitivity of the trade-off between prediction quality and simplicity of the concept hierarchies regarding the various parameters goes beyond the scope of this paper, the main effects of parameters can be viewed. In principle, the smaller α and/or β the larger size of $\mathcal{H}$, and the larger η the higher quality of generated concepts on $\mathcal{H}$.

Figure 1 shows a main screen of the interactive OSHAM in discovering overlapping concepts from the Wisconsin breast cancer data. The database, obtained from the University of Wisconsin Hospitals, consists of observations of Benign and Malignant cases (shown in the window "Wisconsin Breast Cancer Data") on 9 symbolic symptoms: Clump Thickness, Uniformity of Cell Size, ..., Mitoses, each has 10 possible values (shown in the window "Wisconsin Breast Cancer Attributes"). The window "Parameters" shows the initialized parameters in this run. The main browser window shows a part of the generated concept hierarchy. Each discovered concept corresponds to a small rectangle with some information and links to its superconcepts and subconcepts. For example, information in the small rectangle $\boxed{\text{108 \quad 13, 2 (Cell Size,1)}}$ indicates that its concept identifier in the generated order is 108, the number of instances it covers is 13, the number of local instances is 2 (omitted for leaf concepts), and the local tuple is (Cell Size,1). All tuples in the concept intent can be obtained by aggregating local tuples along the branch from the considered concept to the root, for example $i(C_{108})$ = (Cell Size,1) $\wedge$ (Epithelial,3) $\wedge$ (Nucleoli,1) $\wedge$ (Mitoses,1). The description of each concept can be seen in the window "Class Description" by clicking on its sensitive rectangle, e.g., the description of the concept number 108. OSHAM yields discovered knowledge in a concept hierarchy or transform them into a rule base.



OSHAM: Make Automatically Hierarchical Structures of Objects
Open
Save
Run
Print
Suspend
Continue
Change Parameters
Matching
Exit
Parameters
OK
0| 629, 11
1| 519, 4, (Mitoses, 1)
2| 119, 2, (Nuclei, 0)
4| 60, 6, (Nucleoli, 0)
7| 364, 5, (Adhesion, 1)
9| 22, 11, (Nucleoli, 8)
10| 31, 9, (Mitoses, 3)
25| 45, 9, (Epithelial, 3)
29| 30, 16, (Epithelial, 4)
31| 21, (Epithelial, 6)
32| 22, 12, (Epithelial, 5)
6| 27, 9
11| 17
33| 35, 2, (Cell Shape, 0)
35| 35, 13, (Chromatin, 7)
37| 19, 8, (Chromatin, 5)
38| 13, (Chromatin, 3)
95| 21, 5, (Cell Shape, 0)
21| 18, 0
22| 58, 7
26| 11
104| 15, 4, (Cell Shape, 1)
105| 12, (Cell Size, 3)
30| 14
99| 22, 7, (Chromatin, 2)
101| 21, 9, (Chromatin, 1)
103| 11, (Cell Shape, 2)
108| 13, 2, (Cell Size, 1)
64| 15, 3
96| 10, (Cell Shape, 0)
177| 10, (Nuclei, 0)
Class Description
mule@is37e0s20
Buffers File Edit Complete In/Out Signals Help
Wisconsin Breast Cancer Attributes
Wisconsin Breast Cancer Data
Screen Number

overlap (yes=1): 1
minimum size: 10
dispersion : 0.12
competitors: 3

Name: Concept #108 Local/Total Inst.: 2/13
Dispersion: 0.294872 Local Inst. Disp.: 0.333333
Probability.: 0.020668 Local Inst. Prob.: 1.000000
Intent: (Cell Size,1) (Epithelial,3) (Nucleoli,1) (Mitoses,1)

```
CONCEPT 108
  Level = 4
  Super_Concepts = { 28 }  Sub_Concepts = { 109 }
  Concept_dispersion = 0.294872
  Local_instance_dispersion = 0.333333
  Subconcept_partition_quality = 0.000000
  Concept probability = 0.020668
  Concept local conditional probability = 1.000000
  Features = {
              (Cell Size,1)
              (Epithelial,3)
              (Nucleoli,1)
              (Mitoses,1)}
  Local_instances/Covered_instances = 2/13
  Local_instance_dispersion = 0.333333
[--]J_+.--:--**-Mule: *shell*            (Shell:run)--34%--------
Find file: ~/ML/osham/osham4/cancer/hierarchy_result
```

```
6. Number of Attributes: 10 plus the class attribute

7. Attribute Information: (class attribute has been moved

   #  Attribute                     Domain
   -- -----------------------------------------
   1. Sample code number            id number
   2. Clump Thickness               1 - 10
   3. Uniformity of Cell Size       1 - 10
   4. Uniformity of Cell Shape      1 - 10
   5. Marginal Adhesion             1 - 10
   6. Single Epithelial Cell Size   1 - 10
   7. Bare Nuclei                   1 - 10
   8. Bland Chromatin               1 - 10
   9. Normal Nucleoli               1 - 10
  10. Mitoses                       1 - 10
  11. Class:                        (2 for benign,
                                     4 for malignant)
```

```
-,1182404,  4  2  1  1  2  1  1  1  1  2
  3960,     0  0  0  0  6  0  8  1  5  4
  0330,     1  1  1  1  2  1  3  1  1  2
  3h25,     8  0  5  3  8  4  4  0  3  4
  9286,     0  0  8  6  4  5  8  0  1  4
  7608,     6  1  1  1  1  1  1  1  1  2
  5899,     4  1  1  1  2  1  3  1  1  2
  2813,     0  4  4  0  6  0  5  5  1  4
  7145,     5  1  1  1  2  1  1  1  1  2
  39h2,     4  2  2  1  2  1  2  1  1  2
  3282,     1  1  1  1  2  1  2  1  1  2
  1387,     6  8  7  5  6  8  8  9  2  4
  1942,     5  1  1  1  2  1  3  1  1  2
  1734,     0  8  7  4  3  0  7  9  1  4
  3426,     1  1  1  1  2  1  3  1  1  2
  8671,     3  1  1  1  2  1  2  1  1  2
  240h,     5  3  4  3  4  5  4  7  1  2
  7451,     5  1  2  1  2  1  1  1  1  2
  2848,     3  0  7  8  5  8  7  4  1  4
  3514,     1  1  1  1  4  3  1  1  1  2
  6572,     0  9  8  7  6  4  7  0  3  4
  4555,     1  1  1  1  2  1  1  1  1  2
  111h,     1  1  1  3  2  3  1  1  1  2
  6116,     9  0  0  1  0  8  3  3  1  4
```

Figure 1: Discovering overlapping concepts by the interactive OSHAM

There is a considerable distinction between concepts found by OSHAM in contrast to those from others, such as supervised system C4.5 [29], unsupervised systems COBWEB [9] and AUTOCLASS [5]. C4.5 induces decision trees in which concepts are represented only by their intent associated with a predicted error rate, and it also need not to maintain intermediate concepts. Without the class information, OSHAM needs to induce concepts with additional information as described in subsection 2.1. Concepts obtained by OSHAM also differ from those of COBWEB and

AUTOCLASS. COBWEB represents each concept C_k as a set of attributes a_i associated with a set of their possible values v_{ij}, the occurrence probability of concept $P(C_k)$, and the conditional probability $P(a_i = v_{ij} \mid C_k)$ associated with each value v_{ij}. In AUTOCLASS, a concept (referred to as a class) is defined as a particular set of parameter values and their associated model. A *classification* is defined as a set of classes, the probability of each class, and two additional probabilities for each hypothesized model: the model probability $P(H)$ and the conditional parameter probability distribution $P(p \mid H)$. Below is an example of a concept in the output file

```
CONCEPT 22
  Level = 3
  Super_Concepts  = { 7 20 }  Subconcepts = { 78 97 }
  Concept_dispersion = 0.414614
  Local_instance_dispersion = 0.582011
  Subconcept_partition_quality = 4.063252
  Concept probability = 0.092210
  Features = {  (Thickness,5)
                (Adhesion,1)
                (Mitoses,1)}
  Local_instances(7) = { 63 163 234 269 477 480 612 }
  Local_instance_conditional_probability = 0.120690
```

4.2 Evaluation of the predictive accuracy

As mentioned above we use the benchmark introduced in [24] for evaluating the predictive accuracy of unsupervised discovery systems. It is worth mentioning that multiple train-and-test experiments are much computationally expensive but give more reliable evaluation than single train-and-test experiments. A *k-fold cross-validation* is the process of doing k times a single train-and-test experiment then average the results as the final evaluation. The data is divided into mutually exclusive subsets of approximately equal size. In each experiment, one subset is taken as testing data and the others as training data.

We carried out experiments on several databases from the UCI repository of machine learning databases, including the Wisconsin Brest cancer, Congressional voting, Mushroom, Tic-tac-toe with 10-fold cross validation and Monks with single train-and-test experiments. Table 5 gives information about the size of these databases in terms of number of discrete and continuous attributes, number of instances, and number of natural classes in the original data.

All experiments are carried out for four systems C4.5, CART, AUTOCLASS and OSHAM in the same conditions, i.e., the same randomly divided data sets. The predictive accuracies of C4.5 and CART are estimated directly by these programs with fixed thresholds for the post-pruning. For AUTOCLASS, we use the public version AUTOCLASS-C implemented recently in C language and run three steps of *search, report* and *predict* with the default parameters. The predicted name and predictive accuracy of AUTOCLASS are obtained by the same way of those in

OSHAM, i.e., the predictive accuracy is the ratio of the number of testing instances correctly predicted regarding the predicted name of concepts over the total number of testing instances. In order to have an unbiased evaluation of OSHAM, although in each concrete database the user can adjust OSHAM's parameters to obtain the most suitable concept hierarchy, we fixed values $\alpha = 1\%$ of the size of the training set, $\beta = 15\%$, and $\sigma = 10\%$ of the number of attributes, and $\eta = 3$ in all experiments of OSHAM.

Data sets	discrete	continuous	instances	classes
Breast cancer	9	–	699	2
Vote	16	–	435	2
Mushroom	23	–	8125	2
Tic-tac-toe	9	–	862	9
Glass	–	9	214	6
Ionosphere	–	35	351	2
Waveform	–	21	300	3
Pima Diabetes	–	8	768	2
Thyroid (new)	–	6	215	3
Heart Disease	8	5	303	2

Table 5. Description of databases

Table 6 reports the results of predictive accuracy (%) of C4.5, CART, OSHAM and AUTOCLASS (AUTOC), the average number of concepts in hierarchies and CPU time of OSHAM. Some remarks can be drawn from the experimental results:

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

- The complexity of OSHAM is $\mathcal{O}(|\ \mathcal{O}\ ||\ \mathcal{A}\ |)$, and the concept hierarchy is constructed by OSHAM in linear time in the number of instances and the total number of attribute-value pairs. The average number of concepts in concept hierarchies and CPU times (in second) of induction obtained by 10-fold cross validation on the Sparcstation are also reported in two last columns of Table 6. In comparison with the huge number of nodes and execution time of the Galois lattice (e.g., Congressional voting), OSHAM provides a much simpler solution with the reasonable prediction accuracy.

- Using two different representations of concepts, the predictive accuracy of OSHAM and AUTOCLASS in these experiments are only slightly different. AUTOCLASS is better than OSHAM in Breast cancer data, but vice-versa in Tic-tac-toe data. The main advantage of OSHAM in representing concepts over the probabilistic representation is it maintains the concept coherence and so its concept hierarchies can be easily understood.

	C4.5	CART	OSHAM	AUTOC	Concepts	CPU times
Brest Cancer	93.3	94.1	92.6	96.6	98	484
US Voting	94.5	93.8	93.7	91.2	69	151
Mushroom	100.0	100.0	88.2	86.5	63	1288
Tic-Tac-Toe	88.0	86.2	92.6	82.3	204	475
Glass	66.6	66.0	65.3	55.7	37	6.5
Ionosphere	91.5	88.3	81.2	91.5	110	64
Waveform	72.4	72.5	73.0	59.2	215	278
Pima Diabetes	71.2	72.2	72.7	68.2	39	72
Thyroid (new)	91.1	90.3	84.6	89.3	19	5
Heart Disease	59.5	52.4	58.4	49.2	65	13

Table 6. Predictive accuracies, number of concepts and CPU time of OSHAM

5 Conclusion

We have described the unsupervised discovery system OSHAM that combines different views on concepts and categorization constraints in discovering knowledge. We have also developed a decision procedure by integrating interpreters in inductive learning with case-based reasoning that allows using discovered knowledge to decide the class of unknown instances with the satisficing degree. The main contribution of this work lies in the enrichment of the classical views on concepts, its way of finding and using knowledge which is different from other approaches in the KDD literature. Moreover, it has been implemented as an interactive and highly graphic system that permits the interaction of the user with the system during the discovery process and the interpretation of unknown cases. Careful experiments on different databases show that OSHAM can find classificatory knowledge from data with high predictive accuracy and understandability.

OSHAM has an application potential in business decision making, particularly in the construction of knowledge-based decision support systems. Generated in the hierarchical form and supported by the graphical browser, concepts discovered by OSHAM can be easily understood and used in making decision for unknown cases. They can be rewritten in hierarchical object knowledge bases by tools for knowledge-based systems such as KEE, KAPPA, NEXPERT OBJECT, or used directly by the generator TESOR [15]. Concepts discovered in the hierarchical structure by OSHAM can also be transformed into the usual form of decision rules with some modification. Summarily, OSHAM is amenable to business applications in which the user need to construct knowledge bases from databases with low-cost and high-quality.

References

[1] Bergadano, F., Matwin, S., Michalski, R.S., Zhang, J., Learning two-tiered descriptions of flexible concepts: the POSEIDON system, Machine Learning, Vol. 8 (1992), 5–43.

[2] Boyce, B.R., Meadow, C.T., Kraft, D.H., Measurement in Information Science, Academic Press, 1994.

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

[4] Carpineto, C., Romano G., A lattice conceptual clustering system and its application to browsing retrieval, Machine Learning, Vol. 10 (1996), 95–122.

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

[6] Clark, P., Niblett, T., The CN2 induction algorithm, Machine Learning, Vol.3 (1989), 261–284.

[7] El-Najdawi, M.K., Stylianou, A.C., Expert support systems: Integrating AI technology, Communications of the ACM, Vol.36 (1993), No.12, 55–65.

[8] 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.

[9] Fisher, D., Knowledge acquisition via incremental conceptual clustering, Machine Learning, Vol.2 (1987), 139–172.

[10] Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J., Knowledge discovery in databases: An overview, in Knowledge Discovery in Databases, G. Piatetsky-Shapiro and W.J. Frawley (Eds.), AAAI Press, 1993, 1–27.

[11] Godin, R., Missaoui, R., An incremental concept formation approach for learning from databases, Theoretical Computer Science, 133 (1994), 387–419.

[12] Hanson, S.J., Bauer, M., Conceptual clustering, categorization and polymorphy, Machine Learning, Vol. 3 (1989), 343–372.

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

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

[15] Ho, T.B., Pham, N.K., Bach, H.K., Hoang, T.M., Ngo, C.S., Nguyen, T.D., Tong, T.H., Hoang, Q.T., Development and applications of the expert system generator TESOR, Proceedings of NCSR of Vietnam, Vol.2 (1992), N.2, 3–14.

[16] Ho, T.B., An approach to concept formation based on formal concept analysis, IEICE Trans. Information and Systems, Vol. E78-D (1995), No.5, 553–559.

[17] Ho, T.B., A hybrid model for concept formation, in Information Modelling and Knowledge Bases VII, Y. Tanaka et al. (Eds.), IOS Publisher, 1996, 22–35.

[18] Ho, T.B., Nguyen, T.D., Integrating human factors with a concept formation process, Proceedings 6th International Conference on Human-Computer Interaction, 1995, 74.

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

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

[21] Kolodner, J., Case-Based Reasoning, Morgan Kaufmann, 1993.

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

[23] 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, 1995, 198–203.

[24] McKusick, K.B., Langley, P., Constraint on tree structure in concept formation, Proceedings of International Joint Conference on Artificial Intelligence, 1991, 810–816.

[25] Martin, J.D., Billman, D.O., Acquiring and Combining Overlapping Concepts, Machine Learning, Vol. 10 (1994), 121–155.

[26] Michalski R. S. and Stepp R. E.: Learning from observation: Conceptual learning. In Machine Learning: An Artificial Intelligence Approach, Vol. 1, R. S. Michalski, J. G. Carbonelle, T. M. Michell (Eds.), Morgan Kaufmann, 1983, 331–363.

[27] Michalski, R.S., Learning flexible concepts: Fundamental ideas and a method based on two-tiered representation , in Machine Learning: An Artificial Intelligence Approach, Vol. III, R. S. Michalski and Y. Kodratoff (Eds.), Morgan Kaufmann, 1990.

[28] Quinlan, J. R., Decision trees and decisionmaking, IEEE Trans. Systems, Man, and Cybernetics, Vol. 20 (1990), N.2, 339–346.

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

[30] Turban, E., Decision Support and Expert Systems: Management Support Systems, Printice-Hall, 1995.

[31] Van Mechelen, I., Hampton, J., Michalski, R.S., Theuns, P. (Eds.), Categories and Concepts. Theoretical Views and Inductive Data Analysis, Academic Press, 1993.

[32] Wille, R., Restructuring lattice theory: An approach based on hierarchies of concepts. In Ordered Sets, I. Rival (Ed.), Reidel, 1982, 445–470.

[33] Wrobel, S., Concept Formation and Knowledge Revision, Kluwer Academic Publishers, 1994.

[34] Wu, X., Hybrid interpretation of induction results, in Advanced IT Tools, N. Terashima and E. Altman (Eds.), Chapman & Hall, 1996, 497–506.

[35] Zytkow, J., Baker, J., Interactive mining of regularities in databases, in Knowledge Discovery in Databases, G. Piatetsky-Shapiro and W.J. Frawley (Eds.), AAAI Press, 1993, 31–53.

Tu Bao Ho is currently a visiting associate professor at the Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He received a B.Tech degree in Applied Mathematics from the Hanoi University of Technology, in 1978, and M.S. and Ph.D. degrees in Computer Science from Pierre and Marie Curie University, Paris, in 1984 and 1987. In 1979, he joined the Institute of Information Technology, National Centre for Natural Science and Technology of Vietnam where he is an associate professor. His research interests include knowledge-based systems, machine learning, decision support systems, knowledge discovery and data mining.