A Cognitive Modeling Approach to Learning of Ill-defined Categories

Mukesh Rohatgi

Information Systems and Decision Sciences Department
Old Dominion University
Norfolk, VA 23529
E-mail: mxr100f@economy.bpa.odu.edu

Abstract

A vast majority of concepts learned by human beings are ill-defined. Such concepts elude precise definitions and are represented by nonverbalizable descriptions. Literature from cognitive psychology provides an insight into processes used by human beings for learning ill-defined categories. This paper describes the implementation of an Adaptive Concept Learning (ACL) system designed to learn ill-defined categories by simulating human learning behavior. The contents of this paper stand in contrast to the extant concept learning systems described in machine learning literature. The extant systems are generally designed to learn well defined concepts that can be represented with the help of verbalizable rules. In contrast, the system described in this paper is designed to learn ill-defined categories. Therefore, the implementation of ACL is a more realistic simulation of concept learning behavior observed in human beings.

Introduction

Concept learning, also referred to as learning by examples or inductive learning, has received much attention within the area of machine learning (Winston 1975, Mitchell 1982, Michalski 1983, Dietterich & Michalski 1983, Quinlan 1986). A typical concept learning system uses generalization to generate a concept description from specific instances that belong to a predefined category. The purpose of the learning system is to induce a maximally-specific or a complete and consistent structural description of the category from the exemplar set. Concept learning systems also use counterexamples to prevent overgeneralization. In such cases, system input consists of positive and negative examples and the goal is to induce a concept description which covers all positive examples and excludes all counterexamples.

As pointed out by Schank et al. (Schank, Collins, & Hunter 1986), the typical learning system, as described above, does not relate to what people do in real life. The main problem highlighted by Schank et al. was the usage of well defined concepts in machine learning systems. A well defined concept is characterized by defining features that form necessary and sufficient conditions for category membership. This view of category membership, referred to as the classical view by cognitive psychologists, is currently being faced with the following problems (for a detailed review of criticisms against the classical view see Smith & Medin 1981):

(a) the failure of experts to specify defining properties for most real-world object concepts;

(b) its inability in explaining the typicality effects exhibited by category members; and

(c) use of nonnecessary features by human beings in the determination of category membership.

Psychological Evidence for Learning Ill-Defined Categories

The failure of classical view in adequately explaining the basis of concept formation has led the way towards empirical demonstrations that natural objects form categories with ill-defined boundaries (Mervis & Rosch 1981). The current thinking on learning concept descriptions for ill-defined categories is dominated by two model types. The first type includes prototype-based models subsumed under the probabilistic viewpoint and the second type includes exemplar-based models conforming to the exemplar viewpoint (Smith & Medin 1981, Barsalou 1990). Comparisons between prototype-based and exemplar-based models have shown that latter fare better in explaining experimental data on categorization (Medin & Florian 1992), but as discussed in Homa, Sterling, & Trepel (1981), the exemplar models may be faring better because of simplistic laboratory conditions being used during evaluation.

The general issue that concerns us is the nature of information acquired by people when they experience several instances of ill-defined categories, i.e., categories that do not reduce to a simple or easily specified rule (Elio & Anderson 1981). Studies conducted by Posner and Keele (Posner & Keele 1968, 1970) have shown that in case of ill-defined categories, human beings tend to abstract or summarize the description of category members in form of a prototype. The category prototype does not represent a set of necessary and sufficient features but captures the central tendency of the properties associated with the experienced exemplars (Reed 1972). Additional empirical evidence obtained by cognitive psychologists (Barsalou 1990, Breen & Schvaneveldt 1986, Homa, Sterling, & Trepel 1981, Homa & Vosburgh 1976, Homa & Chambliss 1975, Homa et al. 1973) also seems to suggest that situations that deal with multiple, highly variable, and large sized ill-defined categories favor a prototype-based model for concept learning. These empirical studies have also shown that an abstracted prototype should be viewed as an evolving, dynamic construct that undergoes repeated transformations with continuing exposure to category instances. In addition to the above, these studies have also shown that the retention of abstracted information is superior to the retention of individual exemplars.

Research Objectives

As pointed out earlier, extant systems designed for concept learning tend to ignore ill-defined categories. Also, the central idea embodied in these systems is to generate a verbalizable "decision rule" as the concept description with the help of generalization heuristics. In our research, we are interested in designing a system that is capable of learning concept descriptions of ill-defined categories as opposed to well defined categories. Furthermore, we want to use category prototypes as the basis for representing concept descriptions. The use of prototypes for concept representation is motivated by the psychological evidence outlined in the previous section.

The objective of this paper is to describe the construction and performance of a learning system that has been designed to implement the human learning processes believed to be responsible for learning prototype-based concept descriptions of ill-defined categories. System functions include acquisition and storage of object descriptions (category exemplars); derivation of a prototype from observed exemplars; and dynamic (continuous) modification of prototype as a function of category experience. The following section will describe the organization and implementation of the concept learning system. After describing the system construction, we evaluate its performance with respect to two classification experiments. Finally we present the conclusions along with our agenda for future work.

Description of the Learning System

The learning system, henceforth referred to as the Adaptive Concept Learner (ACL), was implemented under version 2.1 of ART-IM (Automated Reasoning Tool) for MS-DOS. Since a considerable amount of research in cognitive psychology has assumed category prototypes or concept descriptions to be feature-based mental constructs, it was deemed appropriate to use frames as the knowledge representation scheme for the learning system.

As mentioned previously, the overall goal of the learning system is to incrementally learn concept descriptions in form of category prototypes. To achieve its goal, the learning system utilizes multiple learning processes integrated hierarchically in a closed-loop configuration. Each learning process makes use of one or more knowledge transmutation operators (see Michalski 1994) in response to external inputs or to outputs from a lower level learning process. The transmutation operators used by the ACL learning process are shown in Table 1.

Learning Processes	Knowledge Operators	Affected / Resulting Knowledge Structures
Object Learning	Insertion (memorization), and Deletion	Instance Descriptions
Exemplar Learning	Agglomeration	Reduced Exemplar-Based Representations
Prototype Learning	Abstraction, Selection, and Characterization	Category Prototypes
Failure-Based Classification Learning	Characterization	Multiple Category Prototypes

Table 1. Knowledge Transmutation operators used by the learning processes of the ACL System

The learning cycle is initiated with the invocation of object learning process. At this stage, the goal of the learning system is to acquire input knowledge through rote learning. The learning task is accomplished in a supervised learning environment and results in the acquisition of category instances described by a user or a teacher. Besides the acquisition of input knowledge, the object learning process is also responsible for maintaining the background knowledge used by the system to facilitate the user-system interaction.

Descriptions of category instances acquired through the object learning process are then agglomerated into reduced exemplar-based representations (Barsalou 1990) by the exemplar learning process. It also maintains several forms of frequency counts that are used at later stages of the learning cycle for assigning weights to the attribute-value pair included in the category prototype.

The reduced exemplar-based representations serve as inputs to the prototype learning process. The goal of the prototype learning process is to derive a category prototype through abstraction and assign weights to the attribute-value pairs included in the prototype description. A category prototype serves as the characteristic description of an object class during classification. Category prototypes are updated if the learning system is exposed to additional category instances. Updating also results in periodic weeding of idiosyncratic information present in the prototype description. The weeding of idiosyncratic information is equivalent to the selection of relevant attributes for a category prototype.

The prototype learning process generates a single prototype for each category. For highly variable categories, the cognitive economy provided by creating category prototypes may prove to be detrimental. To protect itself from such contingencies, the ACL system relies on a failure-based classification learning process to generate multiple category prototypes for capturing the category variability. Hence, the classification process used by ACL has been designed to perform a dual role. In its first role, it acts as a passive similarity-based classifier which is responsible for the determination of category membership when provided with an object description without the knowledge of its class affiliation. In its second role, the classification process is responsible for generating multiple prototypes for a single category. Prototypes generated through failure-based classification should be considered as abstract representations of subclusters within a highly variable class.

The following sections will describe the details of the learning processes used by the ACL system.

Object Learning Process

The object learning process functions as a surrogate for the perceptual and sensory capabilities present in natural systems. In reality, it is the user-interface which allows a teacher to interact with the learning system. The role of object level processes is to facilitate the teacher-system interaction and acquire object descriptions provided by a teacher through rote learning.

The system starts its learning cycle with zero knowledge about the domain in which it is operating. Knowledge acquisition at object level is concerned with the learning of an object description in terms of attribute-value pairs. The system expects the user to provide it with a label, which it assumes to be the name of an object instance belonging to a basic level category (for the definition of basic level categories see Rosch 1978). After a label has been provided as an input to the system, the user is asked to indicate the object type of the label. The ACL expects the object type to be either a physical object, a conceptual object, a physical action, or a mental action.

To facilitate the user-system interaction, we have provided the learning system with a certain amount of metaknowledge about the typical attributes that can be used to describe a given object type. At this point, it should be understood that the ACL is capable of learning this metaknowledge by itself. The metaknowledge has been preloaded to reduce excessive user-system interaction.

Based on its metaknowledge, the system prompts for a value of an attribute along with a displayed list of known values associated with the attribute. A user can either pick a value from the displayed list or opt to provide a new value for the attribute by choosing the "other" option in the displayed list. The user may also decide not to provide a value for the prompted attribute.

A user is not limited to the attributes that he or she is prompted for. After the system has exhausted the known list of attributes, it displays a menu through which a user can either: (a) add or delete an attribute in the current object description; (b) add or delete a value of an attribute in the current object description; (c) provide multiple values for an attribute; or (d) take no action and quit the menu. The add-delete menu provides the user with an opportunity to define new attributes for the category instance currently being described or change the earlier responses if necessary. For example, if the user wants to specify whether an instance of a ball is solid or hollow, then by using the menu options he or she can define an additional attribute and give it a name of his or her choice, e.g., solid-or-hollow. Once an additional attribute is defined, it becomes a permanent part of the system's metaknowledge. Therefore, the next time around, if a user is describing another instance of a ball, then he or she is also asked to provide a value for the solid-or-hollow attribute. The system also keeps track of the data type (symbolic or numeric) for each attribute. Information about attribute data type enables the system to simultaneously work with both symbolic and numeric attributes.

The above described process of acquiring an object description is linear in nature, i.e., knowledge acquisition proceeds from label to type and then to associated attribute-value pairs. This linear execution process is only possible if no new symbol, either in the form of an attribute or an attribute value, is mentioned to the system.

If during the acquisition of an object description, the user mentions a new symbol to the system, the focus of the inquiry changes and the system attempts to learn more about the new symbol by treating it as a new label. The recursion process is infinite because it is being assumed that the system cannot learn an object description in terms of symbols that have not been learned previously. If the user attempts to use circular descriptions (e.g., if the label currently being described is property, and the value for the context attribute of property is specified to be property), the system suspends the recursion and starts unwinding. The recursion process is also terminated, if the user informs the system that not enough information is known about the new symbol at the present time. In such cases the system memorizes the symbol and avoids the recursion process. Additional information about the memorized symbols can be provided to the system at a later time.

The user-system interaction during knowledge acquisition is further facilitated by providing the user with an online display of brief descriptions that serve as explanatory notes describing the scope and meaning of each symbol being used to name an attribute or a value associated with the attribute. Therefore, when the system is informed of a new attribute or a new value of an attribute, it also asks the user to provide a brief description that can be attached to the new symbol. The word description should not be confused with object descriptions in form of attribute-value pairs. Here the word description refers to a string of characters used as explanatory statements.

The object learning process of the ACL has been designed to adapt its interaction according to the object type being described. For example, the ACL will change the focus of its metaknowledge if a physical action is being described instead of a physical object. In the case of a physical action, the user is prompted for an agent executing the action, object on which action is being performed, effect of the action, context of the action, etc. This implies that the user interface is controlled by the object type that is currently being described.

Exemplar Learning Process

The exemplar learning process is responsible for collecting object descriptions into exemplar-based categories. Object instances sharing the same symbol for their name are grouped into a category. The shared name also becomes the category label. Category attributes at the exemplar level are determined by creating a union of attributes associated with each member object. A category representation similar to the one described above has been termed as the exemplar view of concepts (see Smith & Medin 1981).

Three forms of statistics are computed while organizing the described objects into a category. The first statistic maintains the total number of exemplars in a given category. An attribute count is maintained with the help of the second statistic, while the third statistic is used to maintain the attribute value count. For example, if ten instances of blue balls, five instances of green balls, and five instances of red balls were organized into a category named ball, then the exemplar count is equal to 20. Similarly, attribute value counts for blue, green and red values of the color attribute for category ball are 10, 5, and 5 respectively. The attribute count for color is 20 because the color attribute appears in the description of all 20 exemplars representing the category ball. The attribute value count is maintained only for symbolic attributes. For numeric attributes with continuous values the exemplar learning process maintains the minimum and maximum values of the attribute observed among the category exemplars. The minimum and maximum values are used for the computation of similarity evidence during classification. All statistics are updated with additional exposure to new exemplars belonging to the category.

Two pieces of information are lost by creating a union of object descriptions to form a category representation. First, multiple objects collected in a category lose their individual identities; and second, because of the identity loss, correlation among category attributes is also lost. Loss of identity can be assumed to represent the forgetting of individual stimuli, a common occurrence in human beings. Also, due to the loss of individuality by category exemplars, the exemplar view is not the correct term for the category representation used by the exemplar learning process. As stated by Barsalou (1990), such representations are subsumed under reduced exemplar models for concept formation.

Prototype Learning Process

The prototype learning process is activated immediately after the assimilation of an object description into the reduced exemplar-based representation of a category. This process is responsible for generating a prototype-based concept description for each object label encountered by the system. Category prototypes are derived from exemplar level knowledge structures through abstraction. The abstraction process captures the central tendency of the attributes present in the reduced exemplar structure. A modal value is used for symbolic attributes and a mean value is used for numeric attributes. Symbolic attributes with more than one modal value function as multivalued attributes in the prototype description.

Besides creating a prototype-based concept description, the prototype creation process also computes the attribute value salience and attribute relevancy for each symbolic attribute present in the category prototype. Only attribute relevancy is computed for numeric attributes. Attribute value salience is an estimate of the probability of occurrence of an attribute value with respect to a given category. Similarly, attribute relevancy is an estimate of the probability of occurrence of an attribute for a given category. For example, assume that 20 instances of category ball were experienced by the ACL system, and the attribute color was used to describe 10 of them. If the modal value of the attribute color happens to be red, which was observed in 5 instances, then the salience of the value red for the color attribute is estimated to be 0.50 (10 divided by 5). Based on the same data attribute relevancy for the color attribute will be estimated as 0.5 (20 divided by 10). Hence, the attribute value salience and attribute relevancy are computed by using the exemplar count, attribute count and attribute value count statistics maintained at the exemplar level. The salience and relevancy values are used for similarity computations during classification. Attribute relevancy is also used to eliminate idiosyncratic information from a category prototype.

The category prototype description is modified as and when necessary. With the acquisition of additional information at the object level, it is possible to experience a shift in the central tendency of the category attributes. Therefore, the prototype description along with the salience and relevancy values is updated continuously. The update process propagates in a bottom up fashion, i.e., additional information at the object level results in an update at the exemplar level which in turn affects the prototype level.

As mentioned earlier, the category prototype inherits all attributes from the exemplar level. The exemplar level in turn inherits all attributes used in the description of member instances at the object level. This attribute preserving process leads to the inclusion of two types of attributes in the category prototype. The first type represents the set of characteristic attributes, (i.e., attributes often used to describe category instances) and the second set represents the idiosyncratic information or information associated with only a few exemplars of the category. In order to facilitate the evolution of category prototypes towards a predominance of characteristic attributes, the prototype learning process has been designed to eliminate any idiosyncratic information from the prototype description.

The idiosyncratic attributes are eliminated (or deleted) on the basis of attribute relevancy. As explained earlier, an estimate of attribute relevancy is computed by dividing the attribute count with the exemplar count. In order to eliminate attributes with low relevancy, a system defined parameter representing a threshold value (currently set at 0.1) is used. If the attribute relevancy is greater than the threshold parameter, then the attribute is retained as a part of the prototype description; otherwise, it is eliminated. The elimination process is executed whenever exemplar count mod 20 is equal to zero.

Categorization Process

A plausible way to evaluate the quality of concept descriptions generated by a learning system is to evaluate their usefulness in a classification task. The ability of a concept description to produce correct classifications (defined as accuracy by Bergadano et al. 1988) can be used as a surrogate measure of concept quality. Although the form and content of category descriptions is critical for accurate classification, the role of decision rule used in the categorization task should also be taken into account.

A two stage categorization process is used by the ACL system for the purpose of classification. In the first stage, the categorization process attempts to generate multiple prototypes for each object class currently represented in the knowledge base. Multiple category prototypes generated at this stage can be considered as abstract representations of possible subclusters in a highly variable class. The subclusters are identified through failure-based classification of training instances. Classification of test instances comprises the second stage of the categorization process.

Since the categorization process uses multiple prototypes to represent a category, it is also referred to as the Multiple Prototype Classification Model. The following steps illustrate the categorization process used by the ACL system (see Rohatgi 1994, for a complete account of Multiple Prototype Classification model):

1. Before invoking the categorization process, use the mechanisms of object learning, exemplar learning, and prototype learning processes to generate a single prototype for each object class present in the training data set.
2. Use the following procedure to classify instances in the training set:
   a. Create a frame-based representation of the instance that needs to be classified. Therefore, I (instance to be classified) = $\{(a_1, v_1), (a_2, v_2),..., (a_n, v_n)\}$, where a_i is the ith attribute and v_i is the value for the ith attribute in the instance description.
   b. Compute $S_i = \sum_j EVD(I_j, P_{ij})$ for each i, where i = 1...K, K being the number of categories in the knowledge base, I being the instance, P_i being the prototype for ith category, j being the sum of common and distinct attributes present in the prototype and the instance descriptions, EVD being the evidence for similarity, and S_i being the cumulative similarity evidence for the ith category.

The Following rules should be used to compute similarity evidence along each feature dimension:

Similarity evidence for symbolic attributes:

= (attribute value salience x attribute relevancy), if the attribute and attribute values match.

= -(attribute value salience x attribute relevancy), if the attributes match but there is a value mismatch.

= -1.0, if the attribute is present in the instance and is absent in the prototype.

= -(attribute relevancy), if the attribute is present in the prototype and absent in the instance.

Similarity evidence for numeric attributes (e.g. diameter):

$$= 1 - \frac{|\text{instance value - prototype value}|}{(\text{Max - Min + Delta}) / 2.0}$$

Delta = arbitrary small number (0.05). It is being used as a mathematical convenience to avoid dividing by zero.

Note: Min and Max values are identified at the exemplar level, while the attribute value salience and attribute relevancy are computed at the prototype level during the training phase.

c. Compute C = Max{ S_i }.
d. Classify the instance as belonging to the category i for which $\sum_j EVD(I_j, P_{ij}) = C$.

3. A misclassification in step 2 highlights the inadequacy of a single category prototype for accurate classification. The misclassified instance is used as the basis for creating a new subcluster. A prototype for the new subcluster is also generated. At this point, the prototype is based on a single member of the subcluster.

4. Repeat steps 2 and 3 for all training instances (prototypes generated in step 3 also participate in the categorization task after their initial creation). If a subcluster gets assigned as the possible category by the classification procedure in step 2 then the prototype representing the subcluster and the subcluster exemplar count are updated. The exemplar count of the subcluster (number of instances in the reference set of the subcluster) divided by the category exemplar count tabulated in step 1 (number of instances in the entire category) estimates the span of a subcluster in terms of fractional representation. No action is taken if a category prototype generated in step 1 is ruled as most similar to the instance under consideration. It should be noted that the presentation order of training instances may affect the number and quality of subclusters identified in step 3.

5. After exhausting all training instances, the categorization process uses a truncation parameter to drop certain subclusters. The truncation parameter represents a threshold value that ranges between 0.0 and 1.0. Any subcluster with a fractional representation smaller than the truncation parameter is deleted.

6. Classify test instances by using the classification rule outlined in step 2. The object classes present in the training set are assumed to be represented by the multiple prototypes retained after truncation in step 5. If the classification rule picks a subcluster as the possible category then it is assumed that the test instance belongs to the parent class of the subcluster. A match between the class predicted by the classification rule and the true class of the test instance is considered a hit.

To achieve computational economy, the ACL system has been designed to accept a user-defined truncation parameter. If the user is interested in an optimal value of the truncation parameter, then the ACL system can be asked to progressively drop each subcluster starting from the one having the least value in terms of fractional representation. The classification procedure in step 6 is repeated after the elimination of each subcluster. The truncation parameter value which results in most accurate classification is considered as the optimal value of the parameter.

The optimality of the truncation parameter is confined to a given domain and its optimal value will change from one domain to another. Furthermore, the truncation parameter can be understood as an estimate of the abstraction bias in the overlapping area between classes. The significance of the truncation parameter value is discussed in the next section along with the analysis of classification experiments performed on the ACL system.

Execution Summary. The execution sequence of the above described learning processes can be summarized in the following algorithmic form:

```
/* Use Training Data Set */

While Object Instances are Available to be Described

  Symbol = Get_Object_Label

  If not a Previously Known Symbol Then
        Get_Object_type (Symbol);
  End-If

  Object_Description =
  Get_Object_Description (Symbol)

  /* If a previously unknown symbol is used while
  describing an object instance then use recursion and
  invoke Get_Object_Type (New_Symbol) */

  Object_Description =
  Add_Delete_Menu (Object_Description)

  /* If a previously unknown symbol is used while
  modifying an object description then use recursion and
  invoke Get_Object_Type (New_Symbol) */

  If Exemplar Representation does not Exist Then
     Reduced_Exemplar_Structure =
     Create_New_Exemplar_Representation
     (Object_Description)
  Else
     Reduced_Exemplar_Structure =
     Update_Exemplar_Representation
     (Object_Description)
  End-If

  If Prototype Representation does not Exist Then
     Prototype = Create_New_Prototype_Representation
     (Reduced_Exemplar_Structure)
  Else
     Prototype = Update_Prototype_Representation
     (Reduced_Exemplar_Structure)
  End-If

End While loop

/* Invoke Classification Process */

Generate_Multiple_Prototypes (Training_Set)

/* Use test data set */

Perform_Classification (Test_Data)
Print_Classification_Statistics

End Program
```

Evaluation

As mentioned previously, classification accuracy can be used as a surrogate measure for concept quality. Therefore, to measure the quality of concept descriptions generated by the ACL, the prototype system was used to perform two classification tasks.

Description of Data Sets

A summarized description of data sets used for the classification tasks is shown in Table 2. The detailed description is available on the internet node (ics.uci.edu) which functions as the secondary source for data sets used in machine learning experimentation. We chose these previously used data sets to compare the concept learning performance of the ACL prototype system with the classification results reported by other studies (See Weiss & Kulikowski 1991 and Michalski 1989). The breast cancer data provides an opportunity for the ACL system to work with symbolic attributes while the Iris data set tests the capability of ACL to work with numeric attributes. Furthermore, while working with the cancer data, the ACL system is also challenged with classes exhibiting a high degree of overlap.

Experiments with Breast Cancer Data

The data set for evaluating the prognosis of breast cancer reoccurrence was used for the first classification task. 70% of the test cases were randomly selected for learning concept descriptions and the remaining 30% were used for testing. This process was repeated four times to create four data sets. The technique used to create the training and test data sets is the same as described in Michalski (1989). This was done to replicate the experimental conditions of the studies performed by Michalski. Since other researchers have only reported the average results of the four data sets, the following discussion will also refer to the average performance of the multiple prototype classification model.

As shown in Table 3, the multiple prototype model has a better classification accuracy--0.73 compared to 0.68 for AQ15 and 0.72 for ASSISTANT tree--when truncation was used for all three methods. On the other hand, without truncation, the multiple prototype model performs very poorly. In fact, the system performance goes down with the increase in the number of prototypes used for representing a fixed number of classes. Similarly, performance also deteriorates if only one prototype is used to represent each class. This is because a single prototype represents too much abstraction, while too many prototypes for overlapping classes (like the two classes in the breast data) confounds the classification process.

Table 4 (source: Weiss & Kulikowski 1991) shows classification error results for several different classification methods. In comparison to other methods, the multiple prototype model ranks third behind the PVM rule and the CART tree. It is also interesting to note that the classification accuracy of human experts for the breast cancer data is only 0.64 and that most classification methods, including the Multiple Prototype model, can deliver a better performance.

Experiments with the Iris Data

The Iris data set was used for the second classification task. As opposed to the train and test procedure used for breast cancer data, the cross-validation method was used for arriving at the classification accuracy. This was done to be consistent with other classification studies that have used the Iris data set.

A classification accuracy of 0.94 or an error rate of 0.06 places the multiple prototype model as sixth among the classification models shown in Table 4. From the results shown in Table 4, it is clear that the performance of multiple prototype classification model is clearly not among the best.

Analysis

The iris and breast cancer data sets were chosen to study the classification performance of ACL prototype on numeric and symbolic attribute types. The comparative performance data shown in Tables 4 indicate that the classification performance of ACL is better for symbolic data, and average for numeric data. This implies that compared to numeric concept descriptions, the quality of symbolic concept descriptions learned by the ACL prototype system is higher.

For best classification results, a suitable value for the truncation parameter used by multiple prototype model must be found. Higher values of the truncate parameter indicate that it may be possible to use abstraction for maintaining information about exemplars belonging to category areas that overlap with another category in a multidimensional space. On the other hand, lower values of the truncate parameter imply that category instances in the overlap area may have to be remembered as individual exemplars with little scope of abstraction.

Classification performance of ACL for a set of user specified truncation parameter values is shown in Table 5. On the basis of these results it may be concluded that a classification model using a hybrid of prototype and exemplar representations may eventually prove to be the best. This viewpoint is similar to the decentralized view of concept learning proposed by Brooks (1987).

It is worth mentioning that attributes associated with each decision class in the cancer or the iris data set are the same. Under these circumstances, a categorization process is limited to the exploitation of similarities and differences among attribute values for predicting a decision class. This is generally true for objects at the subordinate level in an object hierarchy consisting of superordinate, basic, and subordinate levels (see Rosch 1978). On the contrary, at the basic and superordinate levels, attributes associated with each object class form the basis for classification rather than attribute values. Unfortunately, most data sets for which published classification results are available tend to use attribute values as discriminants for decision classes. Due to lack of comparative data, the classification behavior of the multiple prototype model for basic level objects was not investigated.

Comparison of ACL with Related Research

The following salient characteristics of ACL stand in contrast with respect to existing work in the area of concept learning and classification:

1. The concept learning process used by ACL is based on abstraction instead of the traditional inductive generalization techniques (Michalski 1983) used by most of the existing systems (e.g., see Dietterich and Michalski 1983).
2. ACL subscribes to the probabilistic view of concept learning instead of the classical view adopted by concept learning systems based on inductive generalization techniques.
3. ACL uses prototype-based concept representations as opposed to exemplar-based representations used in machine learning systems like PROTOS (Bareiss 1989) and CYRUS (Kolodner 1984), which follow the case-based reasoning paradigm.
4. The technique used for generating multiple prototypes through failure-based classification is quite similar in spirit to the growth algorithm presented in Kibler and Aha (1987). Although similar, the two approaches differ in their output because the growth algorithm retains its classification failures as exemplars whereas ACL attempts to create prototype-based abstractions from its classification failures.
5. Clustering programs like COBWEB (Fisher 1987) and UNIMEM (Lebowitz 1987) are capable of generating a concept hierarchy while ACL at present is limited to a single level of clustering. ACL uses a truncation parameter in conjunction with classification accuracy to optimize the number of resulting clusters. Unfortunately, ACL like COBWEB does not use any operators to mitigate the effect of instance ordering.
6. The ACL system is not limited to either symbolic or numeric attributes but can function with both attribute types simultaneously.

Data Set Name	Decision Classes	Number of Instances	Attributes per Instance	Values / Attribute	Attribute Type
Breast Cancer	2	286	9	5.8	Symbolic
Iris	3	150	4	n/a	Numeric

Table 2. Description of Data Sets

Domain : Breast Cancer

Method	Complexity	Accuracy
AQ15		
Coverage = Complete	Complexes = 41 Selectors = 160	0.66
Truncation, unique > 1	Complexes = 32 Selectors = 128	0.68
Top Rule	Complexes = 2 Selectors = 7	0.68
ASSISTANT Tree		
No Pruning	Nodes = 120 Leaves = 63	0.67
With Pruning	Nodes = 16 Leaves = 9	0.72
ACL		
Coverage = Maximal[1]	Prototypes = 23	0.38
Truncation parameter = 0.20	Prototypes = 3.5	0.73
Human Experts	-	0.64

Table 3 Comparison of Multiple Prototype Classification Model with AQ15 and ASSISTANT (descendant of ID3)[2]

Notes for Table 3:
1. Maximal is equivalent to no truncation
2. Performance data source for AQ15 and Assistant: Michalski (1989)

Method	Err_{cv}* (Iris Data)	Err_{Test}* (Cancer Data)
Linear	0.020	0.294
Quadratic	0.027	0.344
Nearest Neighbor	0.040	0.347
Bayes Independence	0.067	0.282
Bayes 2nd Order	0.160	0.344
Neural Net (BP)	0.033	0.285
Neural Net (ODE)	0.027	0.276
PVM rule	0.040	0.229
Optimal Rule Size 2	0.020	-
CART Tree	0.047	0.229
ACL (Truncation Parameter)	0.060 (0.025)	0.270 (0.200)

*Represents the error rate or rate of misclassification. Lower values indicate better performance.

Table 4. Comparative Performance of Multiple Prototype Classification Model on Breast Cancer and Iris Data

Truncation Parameter	Classification Accuracy	Average Number of Prototypes
Breast Cancer Data*:		
0.05	0.40	13.75
0.10	0.63	6.00
0.15	0.65	4.25
0.20	0.73	3.50
1.00	0.60	2.00
Maximal**	0.38	23.25
Iris Data:		
0.025	0.94	5.96
1.000	0.94	3.00
Maximal**	0.91	12.92

* Reported classification accuracy is the average performance of Multiple Prototype Classification model on 4 test data sets.

** Maximal is equivalent to no truncation.

Table 5. The Affect of Truncation Parameter on Classification Accuracy

Conclusions and Future Work

A major contribution of this paper is the use of abstraction mechanisms and prototype-based concept representations in concept learning tasks. This is a marked departure from the traditional thinking embodied in the design of extant concept learning systems described in machine learning literature. Furthermore, the system (ACL) described in this paper is not limited to well defined concepts but has been designed to function with ill-defined categories.

The concept learning processes incorporated in the ACL system are nonobservable processes that have been postulated by cognitive psychologists to explain their empirical results obtained from concept learning experiments with human subjects. Experiments conducted with the help of ACL system demonstrate that these learning processes may not be the best option in all circumstances. Results from classification experiments indicate that mathematical formulations may fare better when compared to a simulation of human learning processes. It was also found that concept learning processes used by human beings may be more appropriate in problem domains that are characterized by symbolic attributes.

The conclusions outlined in the previous paragraph are based on a limited set of experiments. Further experimentation with the ACL system is required to substantiate the validity of our results. In the future we plan to conduct few more experiments with ACL to explore its classification behavior in other types of artificial and natural domains. In particular, our plan is to compare the classification performance of ACL with human subjects. We also plan to direct our future effort towards the type of experimentation that will help us in refining the definition of truncation parameter and characterize its role in classification tasks.

References

Bareiss, R. 1989. *Exemplar-Based Knowledge Acquisition.* San Diego, CA: Academic Press.

Barsalou, L. W. 1990. On the Indistinguishability of Exemplar Memory and Abstraction in Category Representation. In T. K. Srull and R. S. Wyer (eds.), *Advances in Social Cognition*, Vol. 3. Hillsdale, New Jersey: Lawrence Erlbaum.

Bergadano, F.; Matwin, S.; Michalski, R. S.; and Zhang, J. 1988. Measuring Quality of Concept Descriptions. In Proceedings of the Third European Working Session on Learning. London: Pitman Publishing.

Breen, T. J. and Schvaneveldt, R. W. 1986. Classification of empirically derived prototypes as a function of Category Experience. *Memory and Cognition* 14(4):313-320.

Brooks, L. R. 1987. Decentralized Control of Categorization: The Role of Prior Processing Episodes. In U. Neisser (ed.), *Concepts and Conceptual Development: Ecological and Intellectual Factors in Categorization.* New York: Cambridge Press.

Dietterich, T. G. and Michalski, R. S. 1983. A Comparative Review of Selected Methods for Learning from Examples. In R. S. Michalski; J. G. Carbonell; and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. 1. Palo Alto, CA: Morgan Kaufmann Publishers.

Elio, R. and Anderson, J. R. 1981. The effects of Category Generalizations and Instance Similarity on Schema Abstraction. *Journal of Experimental Psychology: Human Learning and Memory* 7(6):397-416.

Fisher, D. 1987. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning* 2:139-172.

Homa, D. and Chambliss, D. 1975. The Relative Contributions of Common and Distinctive Information on the Abstraction from Ill-defined Categories. *Journal of Experimental Psychology: Human Learning and Memory* 104(4):351-359.

Homa, D. and Vosburgh, R. 1976. Category Breadth and the Abstraction of Prototypical Information. *Journal of Experimental Psychology: Human Learning and Memory* 2(3):322-330.

Homa, D., Sterling, S. and Trepel, L. 1981. Limitations of Exemplar-Based Generalization and the Abstraction of Categorical Information. *Journal of Experimental Psychology: Human Learning and Memory* 7(6):418-439.

Homa, D.; Cross, J.; Cornell, D.; Goldman, D.; and Shwartz, S. 1973. Prototype Abstraction and Classification of New Instances as a Function of Number of Instances Defining the Prototype. *Journal of Experimental Psychology* 101(1):116-122.

Kibler, D. and Aha, D. W. 1987. Learning Representative Exemplars of Concepts: An Initial Case Study. In Proceedings of the Fifth International Workshop on Machine Learning, 24-30. San Mateo, CA: Morgan Kaufman.

Kolodner, J. L. 1984. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Lebowitz. M. 1987. Experiments with Incremental Concept Formation: UNIMEM. *Machine Learning* 2:103-138.

Medin, D. L. and Florian, J. E. 1992. Abstraction and Selective Coding in Exemplar-Based Models of Categorization. In A. F. Healy; S. M. Kosslyn; and R. M. Shiffrin (eds.), *From Learning Processes to Cognitive Processes*, Vol. 2. Hillsdale, NJ: Lawrence Erlbaum Associates.

Mervis, C. B. and Rosch, E. 1981. Categorization of Natural Objects. *Annual Review of Psychology* 32:89-115.

Michalski, R. S. 1989. Learning Flexible Concepts: Fundamental Ideas and a Method Based on Two-Tiered Representation. In Y. Kodratoff and R. S. Michalski (eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. 3. Palo Alto, CA: Morgan Kaufmann.

Michalski, R. S. 1994. Inferential theory of Learning: Developing Foundations for Multistrategy Learning. In R. S. Michalski and G. Tecuci (eds.), *Machine Learning: A Multistrategy Approach*, Vol. 4. Palo Alto, CA: Morgan Kaufmann.

Mitchell, T. 1982. Generalization as Search. *Artificial Intelligence* 18:203-226.

Posner, M. I. and Keele, S. W. 1970. Retention of abstract ideas. *Journal of Experimental Psychology* 83:304-308.

Posner, M. I. and Keele, S. W. 1968. On the genesis of abstract ideas. *Journal of Experimental Psychology* 77:353-363.

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

Reed, S. 1972. Pattern Recognition and Categorization. *Cognitive Psychology* 3:382-407.

Rohatgi, M. 1994. A human Learning Approach For Designing Adaptive Knowledge-Based Systems. Ph.D. diss., Dept. of Information Systems, Texas Tech University.

Rosch, E. (1978). Principles of Categorization. In E. Rosch and B. B. Lloyd (eds.), *Cognition and Categorization.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R.C.; Collins, G.C.; and Hunter, L.E. 1986. Transcending Inductive category formation in learning. *Behavioral and Brain Sciences* 9:639-686.

Smith, E. E. and Medin, D. L. 1981. *Categories and Concepts*. Cambridge, MA: Harvard University Press.

Weiss, S. M. and Kulikowski, C. A. 1991. *Computer Systems that Learn.* Palo Alto, CA: Morgan Kaufmann.

Winston, P.H. 1975. Learning structural descriptions from examples. In P.H. Winston (ed.), *The Psychology of Computer Vision.* New York: McGraw Hill.