Conceptual Clustering, Learning from Examples, and Inference

DOUGLAS H. FISHER (DFISHER@ICS.UCI.EDU)
Department of Information and Computer Science, University of California, Irvine, CA 92717 U.S.A.

Abstract

Conceptual clustering has proved an effective means of summarizing data in an understandable manner. However, the recency of the conceptual clustering paradigm has allowed little exploration of conceptual clustering as a means of improving performance. This paper describes results obtained by COBWEB, a conceptual clustering system that organizes data so as to maximize inference abilities. The performance task for COBWEB (and implied for all conceptual clustering systems) generalizes the performance requirements typically associated with the better known task of *learning from examples*. Furthermore, criteria aimed at improving inference seem compatible with traditional conceptual clustering virtues of conceptual simplicity and comprehensibility.

1. Introduction

Machine learning is concerned with improving performance through automated knowledge acquisition and refinement. This popular view is reflected in Figure 1 (Dietterich, 1982). Learning organizes observations into a knowledge base that facilitates performance with respect to some task. Assumptions about environment, knowledge base, and performance all impact the design of machine learning algorithms and delineate general learning tasks. For instance, *learning from examples* assumes that objects (states, events, etc.) come preclassified with respect to a number of 'teacher' defined classes (e.g., 'positive' vs. 'negative'). Under this environmental assumption a learner induces concepts for each object class. In every learning from examples system, performance reduces to matching previously unseen 'objects' against induced concepts, thus identifying their class membership (e.g., as a 'positive' example).

In contrast to learning from examples, this paper concentrates on the more recently defined task of *conceptual clustering* (Michalski, 1980). Conceptual clustering systems accept object descriptions and produce a classification scheme over the observed objects. These methods do not require a 'teacher' to preclassify objects as in learning from examples, but use an evaluation function to find classes with 'good' concept descriptions. Michalski and Stepp (1983) originally equated 'good' concepts with those that were simple or otherwise (human) comprehensible. However, this paper takes a different tack – 'good' concepts are those that maximize the number of predictions that can be made about objects of the environment. While conceptual clustering has not typically been associated with a performance task that improves with learning, the bias of maximizing inference ability implies such a task.

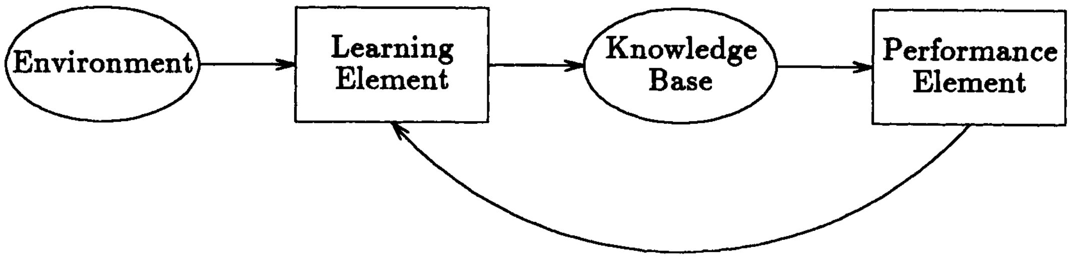

Figure 1. A model of learning and performance

This paper adopts the view that classification schemes produced by conceptual clustering systems are useful in prediction of missing properties of previously unclassified objects – inference is a by-product of classification. This generic performance task clarifies and generalizes the application of clustering (and related) techniques in the context of expert systems (Cheng and Fu, 1985; Fu and Buchanan, 1985), and is consistent with discussions of problem-solving as classification (Clancey, 1984). The performance task associated with conceptual clustering can also be contrasted with that of learning from examples. While learning from examples seeks to maximize correct inference with respect to a single ('teacher' selected) 'attribute' (i.e., class membership), conceptual clustering systems can be viewed as maximizing a probabilistic average of correct prediction across many attributes.

This paper examines results obtained from COBWEB, a system for conceptual clustering that builds classification trees that facilitate inference. COBWEB's success at prediction is compared to a reconstruction of Quinlan's (1983) learning from examples program, ID3. Finally, it is argued that criteria favoring inference are compatible with criteria traditionally used in conceptual clustering relating to the understandability of derived concepts.

2. An Overview of COBWEB

COBWEB transforms a collection of object descriptions into a classification tree, where objects are described in terms of nominal attribute - value pairs like the animal descriptions of Table 1. Over this data, the tree of Figure 2 was formed. In tree construction, class decomposition at each tree level is guided by a measure of partition quality. COBWEB uses *category utility* (Gluck and Corter, 1985), a measure that favors object set partitions that maximize inference ability. However, category utility was originally developed (and validated) as a means of predicting *basic* (or preferred) category effects observed during human classification.

The definition of category utility can be motivated by appealing to a well-known biological taxonomy. One reason the partitioning of animals into mammals, birds, fish, etc. makes sense is that knowing an 'object' is in one of these subclasses (e.g., a mammal) *raises* the *expected number* of predictions that can be made about that 'object' (e.g., it has hair, is warm-blooded, etc.), an advantage reflected by

Table 1. Animal (object) descriptions

Name	BodyCover	HeartChamber	BodyTemp	Fertilization
'mammal'	hair	4	regulated	internal
'bird'	feathers	4	regulated	internal
'reptile'	cornified-skin	imperfect-4	unregulated	internal
'amphibian'	moist-skin	3	unregulated	external
'fish'	scales	2	unregulated	external

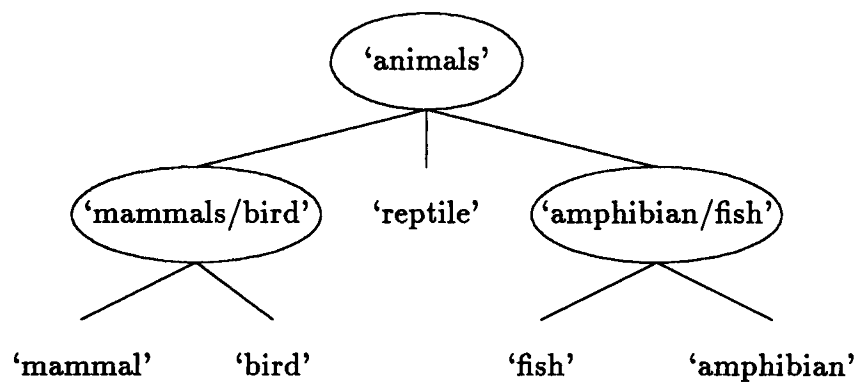

Figure 2. A classification tree over animal descriptions

$$E(\#\text{ of correct predictions} \mid \text{mammal}) - E(\#\text{ of correct predictions}).$$

Gluck and Corter formalize this expression for attribute - value ($A_i = V_{ij}$) representations in terms of conditional probabilities such as $P(A_i = V_{ij}|\text{mammal})$ and base rate probabilities, $P(A_i = V_{ij})$. For an object set partition, $\{C_1, C_2, ..., C_n\}$, category-utility($\{C_1, C_2, ..., C_n\}$) =

$$\frac{\sum_{k=1}^{n} P(C_k)[\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2]}{n}, \quad 2-1$$

where n is the number of categories in a partition. Averaging over categories allows comparison of different size partitions.

Category utility favors information-rich categories and is sensitive to attribute value correlations. Using category utility, COBWEB tends to form trees where the first level is the best partition of the observed objects; this is the case with the tree of Figure 2. However, COBWEB's tree construction algorithm differs significantly from other conceptual clustering algorithms (Michalski and Stepp, 1983; Cheng and Fu, 1985); COBWEB is incremental. Given an existing classification tree and a single object, incorporation is basically a process of classifying the object by descending the tree along a path of 'best' matching classes and subclasses.

To evaluate which of several classes 'best' matches an object (according to category utility), summary distributional information regarding currently classified objects must be maintained at each node. For example, stored at the 'mammals/bird' node of Figure 2 are probabilities such as $P(\text{BodyCover} = \text{hair} \mid \text{mammals/bird}) = 0.5$ and $P(\text{BodyTemp} = \text{regulated} \mid \text{mammals/bird}) = 1.0$. Probabilistic information stored at tree nodes constitutes a *probabilistic concept* (Smith and Medin, 1981).

While complete distributional information about observed attribute values must be kept at nodes for purposes of evaluation, COBWEB distinguishes certain values as *normative* (Kolodner, 1983) or *predictable* (Lebowitz, 1982). In some systems normative values are specified to be those that are present with a probability (or weight) greater than some constant threshold (e.g., 0.67). COBWEB's designation of normative values generalizes this view. Specifically, an attribute's value becomes 'normative' only at node(s) that render the probability of the value (approximately) independent of other attribute values. This generalizes constant threshold strategies since a value with high probability (i.e., close to 1.0) will tend to approximate independence from other values. The 'norms' of Figure 2's 'mammals/bird' node are {HeartChambers=4, BodyTemp=Regulated, Fertilization=internal}. Taken collectively, normative properties allow compact characterization of object classes and link probabilistic and symbolic representations.

While COBWEB processes observations individually, over a sequence of objects and an initially *empty* tree, a classification tree can be built from 'scratch'. As each new object is incorporated and classified down the tree, probabilistic attribute information is updated. While objects are predominantly classified with respect to existing tree nodes, operators exist for new node (class) creation, node combination (merging), and node division (splitting). Fisher (1987) describes COBWEB as hill-climbing (no backtracking) through the space of classification trees. Node merging and splitting are inverse operators that allow bidirectional movement through hierarchy space and recovery from misguided learning paths. This general control strategy was abstracted from previous work by Lebowitz (1982) and Kolodner (1983). Relevant dimensions of hierarchy quality, incorporation cost, and convergence time are introduced in Schlimmer and Fisher (1986) and demonstrate COBWEB to be economical and robust (Fisher, 1987).

In summary, COBWEB's main conceptual precursors are drawn from cognitive psychology, conceptual clustering, and incremental concept formation. COBWEB draws a measure of concept quality from cognitive psychology (Gluck and Corter, 1985) and is the basis of a second system that accounts for important psychological (basic level and typicality) effects (Fisher, 1987). From a machine learning standpoint this work imposes the framework of conceptual clustering (Michalski and Stepp, 1983; Fisher and Langley, 1985) onto incremental concept formation systems like those developed by Lebowitz (1982) and Kolodner (1983). This combination encourages evaluation along a number of dimensions (e.g., hierarchy quality, convergence time) and opens the way for comparative study, something not traditionally performed in this line of work. Lastly, COBWEB suggests inference of missing attributes as a performance task for conceptual clustering.

3. Classification and Inference

COBWEB forms classifications which tend to maximize the amount of information that can be inferred from category membership. The efficacy of this domain independent heuristic depends on the presence of regularities or 'hidden causes' (Pearl, 1985; Cheng and Fu, 1985) in the environment, and on these regularities being extracted and organized by a conceptual clustering system. For example, a disease class and its associated treatment 'properties' are dependent on inter-correlations among symptoms. Classification of instances based on symptoms can then be used for effective diagnosis and treatment of diseases.

The utility of classification trees for inference was tested in the several domains, including a set of 47 soybean disease cases (data from Stepp, 1984). Each case (object) was described along 35 attributes. Four soybean diseases were represented in the data - Diaporthe stem rot, Charcoal Rot, Rhizoctonia Root Rot, Phytophthora rot. These disease designations were also included in each object description, making a total of 36 attributes (e.g., Precipitation = low, Root-condition = rotted, ..., Diagnostic-condition = Charcoal Rot).[1]

An experiment was conducted in which soybean disease cases were incrementally presented to COBWEB in order to see whether the resultant classification could be used for effective disease diagnosis. After incorporating every 5th instance, the remaining **unseen** cases were classified (but not incorporated) with respect to the classification tree constructed up until that point. Test instances being classified contained no information regarding 'Diagnostic condition', but the value of this attribute was inferred as a byproduct of classification. Specifically, classification terminated when the test object was matched against a leaf of the classification tree. This leaf represented that previously observed object which best matched the test object. The diagnostic condition of the test object was guessed to be the corresponding condition of the leaf. The experiment was terminated after one half of the domain (of 47 cases) had been incorporated.

The graph of Figure 3 give the results of the experiment. The graph shows that after as few as 5 instances, the classification could be used to correctly diagnose disease over the remaining 42 cases 88% of the time. After 10 instances, 100% correct diagnosis was achieved and maintained. To put these results into perspective, Figure 3 also graphs the results of a simpler, but reasonable inferencing strategy. This 'frequency-based' method dictates that one always guess the most frequently occurring value (Phytophthora Rot) of the unknown Diagnostic-condition attribute and gives a 36% correct prediction rate. Thus, the COBWEB classification tree facilitates a 64% increase in correct prediction.

[1]While Diagnostic condition was included in each object description, it was simply treated as another attribute. Diagnostic condition was not treated as a teacher imposed class designation as in learning from examples.

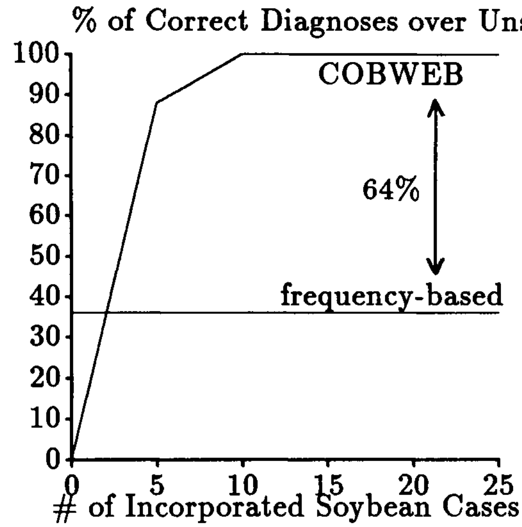

Figure 3. Success at inferring 'Diagnostic condition'

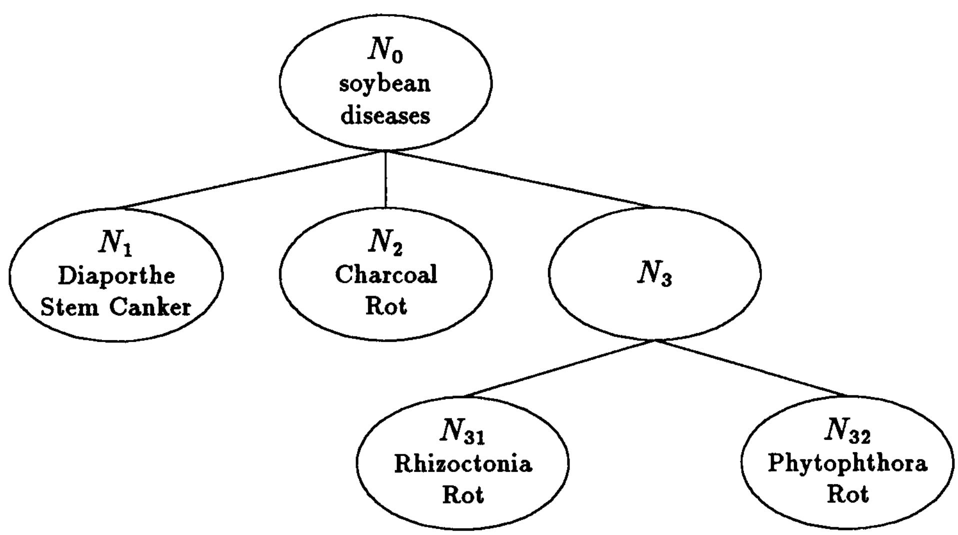

Figure 4. A partial tree over soybean cases

While the results of this experiment seem impressive, they follow from the regularity of this particular domain. In fact, when COBWEB was run on the data with no information of Diagnostic condition at all, the four classes were 'rediscovered' as nodes in the resultant tree. This indicates that Diagnostic condition participates in a network of attribute correlations that is rewarded by category utility. In organizing classes around the correlated network of attributes, classes corresponding to the various Diagnostic conditions are generated (Figure 4).

The success at inferring Diagnostic condition implies a relationship between an attribute's dependence on other attributes and the utility of COBWEB classification

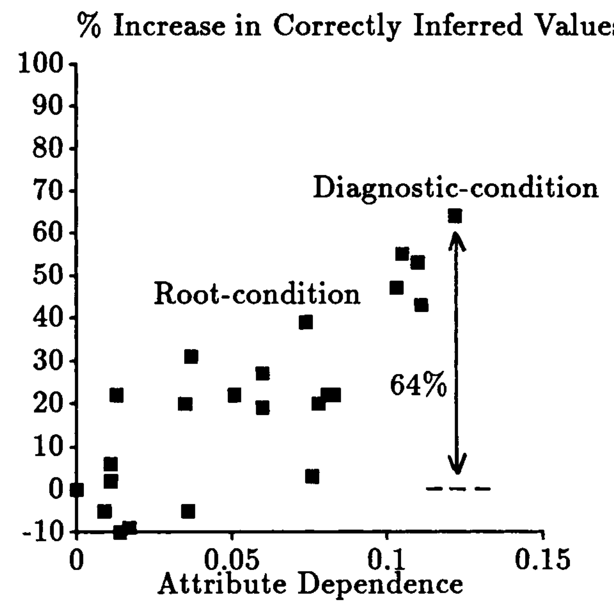

Figure 5. Correct attribute inference as a function of attribute dependence

trees for induction over that attribute. To characterize this relationship it is necessary to introduce a measure of attribute dependence. Function 3-1 gives one such measure. The dependence of an attribute A_M on other attributes A_i is given as

$$\frac{\sum_{i \neq M} \sum_{j_i} P(A_i = V_{ij_i}) \sum_{j_M} [P(A_M = V_{Mj_M} | A_i = V_{ij_i})^2 - P(A_M = V_{Mj_M})^2]}{|\{i | A_i \neq A_M\}|} \qquad 3-1$$

This function is derived in much the same way as category utility, but instead measures the average increase in the ability to guess the value of A_M given one knows the value of a second attribute. If A_M is independent of all other attributes, A_i, then 3-1 equals 0 since $P(A_M = V_{Mj_M} | A_i = V_{ij_i}) = P(A_M = V_{Mj_M})$ for all A_i, and thus $P(A_M = V_{Mj_M} | A_i = V_{ij_i})^2 - P(A_M = V_{Mj_M})^2 = 0$.

The induction test conducted for the Diagnostic condition attribute was repeated for each of the remaining 35 attributes. Averaged over all attributes, correct induction of attributes values for unseen objects levels off at 88% using the COBWEB classification tree as compared with 72% for the frequency-based method. A breakdown of the results in terms of individual attributes is given in Figure 5, where the increase in correct inferences afforded by the COBWEB classification tree over the frequency-based method is shown as a function of attribute dependence (3-1). Each point on the scatter graph represents one of the 36 attributes used to describe soybean cases. The graph indicates a significant positive correlation between an attribute's dependence on other attributes and the degree that COBWEB trees facilitate correct inference. For example, Diagnostic condition participates in dependencies with many other attributes and is also the most predictable attribute.

The soybean data strongly suggests that COBWEB captures the important inter-correlations between attributes, and summarizes these correlations at classification tree nodes. In doing so, COBWEB promotes inference of attributes in proportion to the degree that they participate in correlations – COBWEB tends to maximize inference across all attributes according to some (as yet unformalized) 'weighted' average. This is in contrast to learning from examples systems that seek to maximize correct prediction with respect to a single 'teacher' selected attribute. This observation suggests a way of obtaining an upper bound on COBWEB's inference ability, just as the frequency-based method supplied a lower bound.

COBWEB's accuracy (stemming from a single classification tree) was compared to results obtained from a reconstruction of Quinlan's (1983) learning from examples program, ID3. The program, ID3′, builds *decision trees* to distinguish object classes. Specifically, for *each* of the 36 attributes, the training set (of 25 instances) observed by COBWEB was used to train ID3′; in each case the values of the attribute were treated as 'teacher' imposed classes. Thus, ID3′ built one decision tree to distinguish the various diagnostic conditions, a separate tree to distinguish root conditions, and a distinct tree for each subsequent attribute. These decision trees were used to predict the appropriate attribute values in the remaining unclassified soybean cases.[2]

The *differences* between the percent of correct predictions for each attribute using the corresponding ID3′ decision tree and the COBWEB classification tree are given in Figure 6. For example, COBWEB's tree predicts Root-condition correctly 100% of the time, while the ID3′ decision tree yields 96% correctness, thus giving a difference of -4%. Overall, correctness afforded by the COBWEB classification tree is comparable to, if not slightly better than that afforded by the 36 ID3′ decision trees. However, this statement must be qualified. First, the reconstruction of ID3 does not include protections against 'exceptional' objects (e.g., the chi-square measure) and decision trees can overly specialize. Secondly, comparisons of COBWEB's classification tree and ID3 decision trees may be unfair – ID3 trees discriminate based on a *single* attribute at decision points (i.e., they are *monothetic* classifiers), while the use of category utility as a matching function makes COBWEB trees essentially *polythetic* and therefore more sensitive to attribute inter-correlations. The advantage of polythetic classification may be magnified by the small data set.

Granting the limitations of ID3′ however, the data still suggests the cost effectiveness of the COBWEB approach to improving inference over the use of a learning from examples approach for each attribute. However, better clarification of this relationship requires further experimentation, including tests in more domains and with polythetic learning from examples systems. A current hypothesis is that while learning from examples systems delimit upperbounds on COBWEB's performance, in general COBWEB approximates their abilities. One trait suggested by current data is that performance afforded by COBWEB and a suitable learning from examples system will diverge (i.e., learning from examples will be better) at attributes of intermediate dependence (0.012 - 0.036 in the soybean experiment), while performance will converge

[2]This experiment was suggested independently by Jeff Schlimmer and Nick Littlestone.

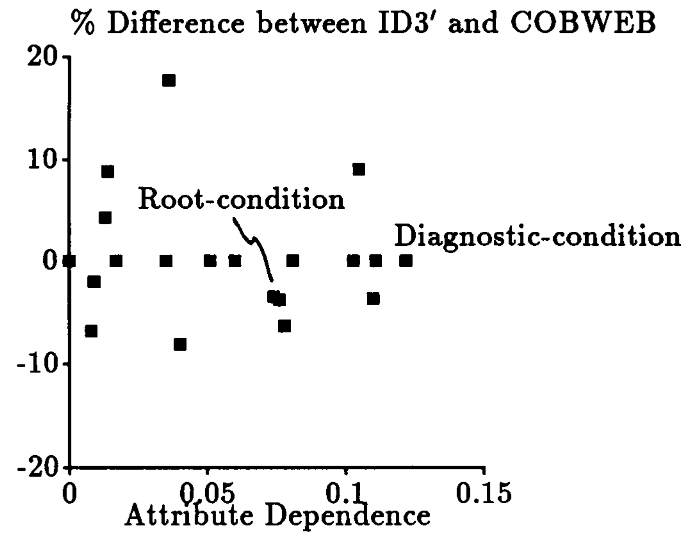

Figure 6. Positive vertical axis values indicate ID3′ outperforms COBWEB.

at attributes exhibiting either significant dependence or independence; highly dependent attributes will be easy to spot by any system (including a conceptual clustering system) and rules concerning independent attributes will be difficult to spot by any method (including learning from examples).

4. A Note on Inference and Understandability

The use of probabilistic (versus logical) concepts and validation of classification trees with respect to inference (versus understandability) distinguish COBWEB from the work of Michalski and Stepp (1983). However, these representations and performance objectives need not be incompatible (Cheeseman, 1985; Rendell, 1986). Generally, it is not difficult to show that category utility represents a tradeoff between the *predictability* of attribute values (operationalized as $P(A_i = V_{ij}|C_k)$) and the *predictiveness* of values (i.e., $P(C_k|A_i = V_{ij})$). An appropriate tradeoff of predictability and predictiveness is necessary in classification structures useful for inference – predictive values combine to direct the classification of partially described objects. Once classified, predictable values can be asserted to complete partial object descriptions. However, as Medin, Wattenmaker, and Michalski (1986) point out, predictability and predictiveness generalize logical necessity (characteristic) and sufficiency (discriminant), respectively. It is probable that an analogous tradeoff for logical concepts (e.g., Michalski and Stepp's measures of 'simplicity' and 'fit') would result in trees that facilitate inference.

As logical concepts may approximate probabilistic ones in facilitating prediction, understandable symbolic descriptions can be generated from probabilistic ones. The gap between probabilistic and symbolic representations is bridged by normative values. Consider COBWEB's selection of normative values for nodes N_2 and N_3 of

Table 2. Normative values of soybean tree nodes

$N_2 \equiv$ 'Charcoal Rot' $A_i = V_{ij}[P(A_i = V_{ij} \mid N_2), P(N_2 \mid A_i = V_{ij})]$	N_3 $A_i = V_{ij}[P(A_i = V_{ij} \mid N_3), P(N_3 \mid A_i = V_{ij})]$
Precipitation = below-normal [1.0, 1.0]	Plant-stand = less-than-normal [0.93, 1.0]
Temperature = above-normal [0.60, 1.0]	Temperature = below-normal [0.63, 1.0]
Stem-cankers = absent [1.0, 1.0]	Stem-cankers = below-soil [0.67, 1.0]
Fruit-pod-condition = normal [1.0, 0.50]	Fruit-pod-condition = does-not-apply [1.0, 1.0]
Canker-lesion-color = tan [1.0, 1.0]	Severity = severe [0.59, 0.84]
Outer-stem-decay = absent [1.0, 0.48]	Damaged-area = low-areas [0.96, 0.90]
Internal-stem-discoloration = black [1.0, 1.0]	
Sclerotia-internal-external = present [1.0, 1.0]	

Figure 4 (Table 2). N_2 has a total of 8 normative values, 5 of which are individually necessary and sufficient for class membership. N_3 has 6 normative values, one of which is necessary and sufficient. In Michalski and Stepp's terminology, necessary and sufficient values represent concepts that are simple and tightly fit the data. In cases where there are no (simply stated) necessary and sufficient conditions, *polymorphous* rules (Hanson and Bauer, 1986) can be used to symbolically describe classes. Specifically, a list of normative values is regarded as a class prototype and any object with a specified number (e.g., a majority) of these values can be regarded as a class member.

Lastly, while probabilistic concepts are typically justified because they generalize logical (typically conjunctive) representations (Smith and Medin, 1981; Hanson and Bauer, 1985), statistically based representations seem more suitable for incremental systems (Schlimmer, personal communication). Even heuristic measures applied to logical representations may be computed from summary statistics. However, in the context of nonincremental systems these statistics can be computed as needed and there is no need to make them an explicit part of the concept representation. In incremental systems it is advantageous to maintain summary statistics, thus reducing cost when incorporating new objects. Nonincremental methods tend to compute statistics as necessary, whereas an effective incremental strategy is to generate symbolic descriptions as necessary.

5. Concluding Remarks

COBWEB is an incremental conceptual clustering system that builds predictive models of an environment. It takes a 'shotgun' approach to improve inference with respect to all attributes, as opposed to a learning from examples system that seeks to optimize inference with respect to a single teacher selected 'attribute'. Initial experiments indicate that COBWEB's approach compares favorably with respect to a reconstruction of ID3, but further experimentation in other domains and with other learning from examples systems is required to fully understand this relationship. The direction of this work promises to unify the tasks of conceptual clustering and learning from examples in a common processing abstraction.

Finally, it is hoped that this work casts conceptual clustering as a useful tool for problem-solving, by assigning it the performance task of predicting unknown attribute values. This view need not oppose an earlier view of conceptual clustering as summarizing data in an understandable manner – important criteria of each view appear compatible. Elaborating the connection between these views, as well as extending conceptual clustering systems to handle more complex representation languages (Stepp, 1984), promises to yield an interpretation of conceptual clustering as a model of evolving expertise (Kolodner, 1983).

Acknowledgements

Discussion with Jeff Schlimmer, Pat Langley, Dennis Kibler, Rick Granger, Rogers Hall, Dave Ruby, and other members of the UCI machine learning group have influenced this paper. This work was supported by Hughes Aircraft Company.

References

Cheeseman, P. (1985). In defense of probability. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 1002-1009). Los Angeles, CA: Morgan Kaufmann.

Cheng, Y., & Fu, King-sun (1985). Conceptual clustering in knowledge organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence,7*, 592-598.

Clancey, W. J. (1984). Classification problem solving. *Proceedings of the National Conference on Artificial Intelligence* (pp. 49-55). Austin, TX: William Kaufmann, Inc.

Dietterich, T. (1982). Chapter 14: Learning and inductive inference. P. Cohen & E. Feigenbaum (Eds.), *The handbook of artificial intelligence.* Los Altos, CA: William Kaufmann, Inc.

Fisher, D. (1987). *Knowledge acquisition via incremental conceptual clustering* (Doctoral Dissertation - in preparation). Irvine, CA: Department of Information and Computer Science.

Fisher, D. and Langley, P. (1985). Approaches to conceptual clustering. *Proceedings of the Ninth International Conference on Artificial Intelligence* (pp. 691-697). Los Angeles, CA: Morgan Kaufmann.

Fu, L., & Buchanan, B. (1985). Learning intermediate concepts in constructing a hierarchical knowledge base. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 659-666). Los Angeles, CA: Morgan Kaufmann.

Gluck, M., & Corter, J. (1985). Information, uncertainty, and the utility of categories.

Proceedings of the Seventh Annual Conference of the Cognitive Science Society (pp. 283-287). Irvine, CA: Lawrence Erlbaum Associates.

Hanson, S., & Bauer, M. (1986). Machine learning, clustering, and polymorphy. L. Kanal & Lemmer (Eds.), *Uncertainty and artificial intelligence*, Amsterdam: North-Holland.

Kolodner, J. (1983). Reconstructive memory: A computer model. *Cognitive Science, 7*, 281-328.

Lebowitz, M. (1982). Correcting erroneous generalizations. *Cognition and Brain Theory, 5*, 367-381.

Medin, D., Wattenmaker, W., & Michalski, R. (1986). *Constraints and preferences in inductive learning: An experimental study comparing human and machine performance* (Technical Report ISG 86-1). Urbana, IL: University of Illinois, Department of Computer Science.

Michalski, R. (1980). Knowledge acquisition through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems, 4*, 219-243.

Michalski, R., & Stepp, R. (1983). Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 5*, 396-409.

Pearl, J. (1985). Learning hidden causes from empirical data. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 567-572). Los Angeles, CA: Morgan Kaufmann.

Quinlan, J. (1983). Learning efficient classification procedures and application to chess end games. R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine learning: An artificial intelligence approach.* Palo Alto, CA: Tioga.

Rendell, L. (1986). A general framework for induction and a study of selective induction. *Machine Learning, 1*, 177-226.

Schlimmer, J., & Fisher, D. (1986). A case study of incremental concept induction. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 496-501). Philadelphia, PA: Morgan Kaufmann.

Smith, E., & Medin, D. (1981). *Categories and concepts.* Cambridge, MA: Harvard University Press.

Stepp, R. (1984). *Conjunctive conceptual clustering: A methodology and experimentation* (Technical Report UIUCDCS-R-83-1189), Doctoral Dissertation, Urbana, IL: University of Illinois, Department of Computer Science.