# Concept Formation and Attention

**John H. Gennari**
Department of Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku-ku; Yokohama, JAPAN 223
⟨gennari@aa.cs.keio.ac.jp⟩

## Abstract

In this paper, I combine the ideas of *attention* from cognitive psychology with *concept formation* in machine learning. My claim is that the use of attention can lead to a more efficient learning system, without sacrificing accuracy. Attention leads to a savings in efficiency because it focuses only on the relevant attributes, retrieves less information from the environment, and is therefore less costly than a system that uses every piece of information available. I present a working algorithm for attention, built onto the CLASSIT concept formation system, and describe results from three domains.[1]

## 1. Motivation

In cognitive psychology, *selective attention* refers to focusing one's cognitive effort (or processing power) on only a fraction of the perceptual input. Intuitively, this is the ability to preferentially 'concentrate' on a single task. This phenomena can be observed with visual input – we can easily attend to part of a scene, allowing unimportant areas and details to be ignored. This is a well-studied phenomena, and it appears in a wide variety of tasks and applications (Treisman, 1969).

Unsupervised *concept formation*, like selective attention, is a basic aspect of human intelligence. Concept formation can be viewed as a general problem that appears in a variety of practical, engineering tasks. In statistics, this problem is known as *cluster analysis* (Anderberg, 1973), and can be defined: given a set of instances or objects, find or impose some classification scheme on those objects. Note that the learning system is not given a set of classified training objects; thus it carrys out *unsupervised* learning. In machine learning, concept formation systems include Lebowitz's (1987) UNIMEM and Fisher's (1987) COBWEB systems; these contrast with Quinlan's (1986) ID3 and Aha's (1989) IBL systems. The former do "learning from internal feedback" (Billman & Heit, 1988), while the latter "learning from example" systems use classified training instances to learn concept definitions.

In this paper, I describe a system that applies selective attention to concept formation. Machine learning is an appropriate place for such a wedding: a discipline with roots in both engineering and psychology. Most previous concept formation research (Lebowitz, 1987; Fisher, 1987) has used all available information to make decisions. In contrast, I will assume that some of the available input should be ignored; the task for the attention mechanism is to find the relevant attributes and allow the system to work with these alone.

My goal here is simply to demonstrate the practical application of attention to concept formation: I expect that attention can provide a tangible reward in terms of *efficiency*. In most applications, there is a cost associated with retrieving information about an instance. This may be negligible in some applications, but in others, such as diagnosis, information retrieval can be very expensive. If an attention mechanism focuses the system on a small percentage of the available input, this can lead to a more efficient, cost-effective system, without sacrificing performance accuracy.

The attention mechanism I introduce here is built on the CLASSIT system (Gennari, Langley & Fisher, 1989; Gennari, 1990). Although this system is not a model of human learning, it is closely related to Billman and Heit's (1988) psychological model of concept formation with attention. Their task is similar to mine: to acquire knowledge about which features are most helpful for classification. However, their model uses a rather different representation for knowledge, and is limited to learning rules about pairs of features. In contrast, CLASSIT uses probabilistic concepts for representation (as in Smith and Medin, 1983), and builds a concept hierarchy to organize its acquired knowledge.

## 2. An overview of the CLASSIT system

For the CLASSIT system (or for COBWEB) an important characteristics of the learning task is that it

[1]This research grew out of work with the ICARUS research group at UCI: Pat Langley, Kevin Thompson, Wayne Iba and John Allen. Mike Pazzani also contributed some key ideas about attention. This research was supported by Contract MDA 903-85-C-0324 from the Army Research Institute, IBM Japan, and by Keio University.

occurs in an *incremental* manner. This means that the set of input instances is treated as a sequence over time, and that the system must learn from each new instance without reprocessing the previously seen instances. This restriction seems intuitive from a psychological perspective: humans do not remember every instance, and are able to learn despite a virtually infinite sequence of instances. This requirement is also reasonable for robotic or real-time applications, where a response may be needed at any point during learning. In this way, CLASSIT differs from statistical methods in cluster analysis and from non-incremental learning systems such as CLUSTER/2 (Michalski & Stepp, 1983).

## Representation

Instances for CLASSIT are described by a simple list of attribute-value pairs. Attributes may be symbolic, with a finite (and usually small) range of values, or continuous, with an infinite set of possible values. Additionally, CLASSIT allows for missing attributes: some instances may not have values for every attribute. This form of representation is reasonably general, but it cannot handle relational or structured information (see Iba & Gennari, in press).

Like COBWEB, CLASSIT uses probabilistic concept descriptions to represent acquired knowledge. Thus, rather than all-or-none conjunctive concept definitions, this approach uses probabilities to build approximate concepts. For symbolic attributes, these probabilities can be computed by counting the number of times each attribute-value appears and the number of member instances. For continuous attributes, where the probability of any single value is zero, the system uses the mean and the standard deviation over member instances.

These concepts are organized in memory in a general-to-specific hierarchy. Toward the top of the tree are general concepts, summarizing many instances. Lower in the tree are more specific concepts, and the leaves may be single instances. A leaf may also summarize a number of (very similar) instances. In this case, the system has *forgotten* those individual instances, and can only retrieve the summarizing leaf concept.

## Algorithm

CLASSIT begins with an empty hierarchy, and adds to its concept hierarchy as it classifies each new instance. The algorithm presented in Table 1 is an overview of how learning (modifying the hierarchy) and performance (classification) occur. The system classifies instances by sorting them through the concept hierarchy from the root node down to the leaves. At each level, CLASSIT can place the node in an existing concept or decide that the instance is sufficiently different to warrant the creation of a new class. The other two choices (operators c and d in Table 1) reorganize the concept

```
1. Incorporate x into the root class.
2. Choose the best of four operators:
      a) incorporate x into a child class.
      b) create a new disjunct based on x.
      c) merge two child classes.
      d) split a class into its children.
3. If the new class has no children, or
   If the match is close enough, end.
   Else, recurse on the chosen class.
```

Table 1: The incremental algorithm used by CLASSIT.

hierarchy; they give the system some ability to recover from a misleading sequence of instances.

In order to avoid storing all instances, and to avoid overfitting in noisy domains (see Gennari, 1990), CLASSIT uses a parameter called the *recognition criterion* as it classifies instances. This tells the system that the new instance matches the current concept well enough to consider that instance as recognized and to halt the classification process (Step 3 in Table 1).

## Evaluation function

In order to choose among the four operators, CLASSIT uses an *evaluation function*. This is an expression that evaluates the quality of a set of concepts and returns a numeric score, allowing the system to choose the operator that leads to the highest score. CLASSIT uses a version of *category utility* for its evaluation function. This function is designed to maximize the predictive ability of classes and was originally developed by Gluck and Corter (1985). For CLASSIT, category utility is:

$$\frac{\sum_{i}^{I}\sum_{j}^{J} P(C_j)Info(C_{ij}) \;-\; Info(C_{ip})}{I \cdot J} \quad , \qquad (1)$$

for $I$ attributes and $J$ classes, where $P(C)$ is the probability of class $C$ and $C_{ip}$ refers to attribute $i$ in the parent class. $Info(C)$ is a function that measures the value or quality of class $C$. For a symbolic attribute $i$ (with $V$ values)

$$Info(C_i) = \sum_{v}^{V} P(x_{iv}|C)^2 \quad ,$$

and for a continuous attribute $i$

$$Info(C_i) = 1/\sigma_{iC} \quad ,$$

where $\sigma_{iC}$ is the standard deviation of an attribute in class $C$.[2] To summarize, this function sums over every

[2] With singleton classes, this standard deviation is zero, leading to an infinite $1/\sigma$. To solve this problem, CLASSIT uses an *acuity* parameter that specifies a minimum (non-zero) standard deviation. This limit corresponds to the notion of a 'just noticeable difference' in psychophysics – the lower limit on our ability to make perceptual discriminations.

1. Select an unseen attribute with probability based on its salience.
2. Update the salience of the selected attribute.
3. Compute the category utility score for the best classification, $X$, based only on observed attributes.
4. Consider all remaining unseen attributes and compute scores for 'worst-case' scenarios: where these attributes might match either
   a) An alternative concept.
   b) A new disjunct.
5. If either of these scores is better than $X$, then go to step 1.
   Else, ignore remaining attributes.

Table 2: An algorithm for attention

child concept, $C_j$, and subtracts the information at the parent, $C_p$. Thus, it measures the gain in $Info(C)$ from parent to child levels of the hierarchy.

## 3. Attention applied to CLASSIT

In order to choose only the 'important' attributes, the system must learn the relative *salience* of attributes. Note that this is not given *a priori*, so attention adds a second learning task to the system: CLASSIT must learn both concept descriptions and saliences of attributes. Salience is defined as the per-attribute contribution to category utility (see Equation 1). Hence, for a given attribute $i$,

$$Salience_i = \frac{\sum_j^J P(C_j)Info(C_{ij}) - Info(C_{ip})}{J}$$

These scores produce a dynamic ordering of the attributes from most salient (attributes that should be inspected first) to least salient (attributes that probably need not be inspected). However, attributes are not always inspected in exact order of salience; instead, the system chooses attributes probabilistically as a function of their salience. This allows the system to recover from 'incorrect' scores: even a low-scoring attribute may be occasionally inspected. If such an attribute is 'noticed' in this way, and if that attribute actually is salient, then its score will improve, and it will be more likely to be inspected in the future.

In addition to an ordering, the system must decide how many attributes to inspect before making a clustering decision. CLASSIT resolves this 'stopping condition' problem by imagining a worst-case scenario where the unobserved attributes match some other concept perfectly, and then considering whether this information would change the current clustering decision. If so, then it must continue inspecting attributes; if not, it has inspected sufficient attributes to make a decision.

Table 2 presents the attention algorithm used in CLASSIT. This mechanism is embedded within the basic concept formation algorithm as described in Table 1. Attention is used whenever making a clustering decision: choosing one of the four operators at a level in the hierarchy. If attention chooses to ignore some attributes, these attributes are treated as missing: the system simply makes classification decisions based on partial information. When CLASSIT descends to the next level, attributes that were observed earlier are 'remembered' and added to the list of known attributes. Thus, the system must inspect more attributes in order to make more specific classifications.

With little or no previous information, all attributes are equally salient, and the system must inspect most or all attributes before choosing a clustering operator. However, as more instances are observed, concepts should emerge in which some attributes contribute heavily to the total category utility score, while others contribute less. This means that the salience scores for attributes become more disparate, letting the system inspect only those attributes that have high scores. The attribute learning process is synchronous with the concept learning process: as the system defines concepts, it learns which attributes are more salient.

As stated earlier, the purpose of an attention mechanism is to improve efficiency by looking at fewer attributes. Yet there is no improvement in computational efficiency with this algorithm. In particular, by applying the halting condition after observing each of $n$ attributes (Step 4 in the table), I have added an $O(n^2)$ cost to the algorithm. However, I assume that the cost of observing an attribute is far greater than the time required for internal computation. This is reasonable if one imagines an application to diagnosis or robotics, where considerable work and real time may be needed to observe features (Tan & Schlimmer, 1990).

## 4. Experimentation with attention

The basic claim to be verified by experimentation is that attention can increase efficiency without a loss of accuracy. More specifically, it should decrease the number of attributes the system must inspect without decreasing its predictive accuracy. In addition, I should verify that the attention mechanism behaves as expected: the number of attributes inspected should decrease over time, and the system should focus only on the "relevant" attributes.

In the following experiments, I measure efficiency by counting the total number of attributes observed during classification of each instance. For accuracy, I measure the predictive ability of the concept hierarchy on unseen test instances. Both measures are taken with the system in a 'testing' mode. Learning (the modification of the concept hierarchy) is turned off during testing, and the recognition criterion is set at a low level. The recognition criterion represents a trade-off between careful learning and rapid recognition. Low-

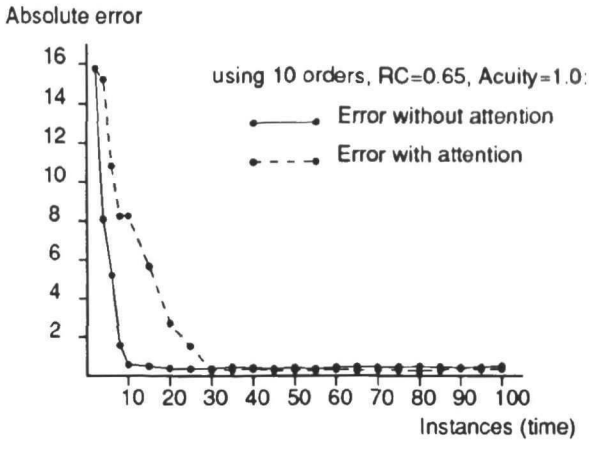


Figure 1: Predictive accuracy – artificial domain

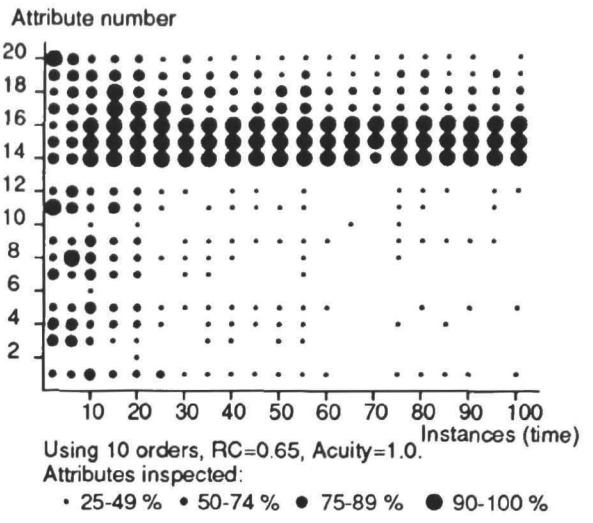


Figure 2: The frequency of attribute inspection in the artificial domain

ering the parameter allows the system to quickly recognize an instance at a high, general level of the tree, allowing the attention mechanism to use only a small number of attributes.[3]

## Results with artificial data

As an initial experiment, it is useful to investigate CLASSIT's performance with an artificial database. With such a domain, one can guarantee that some attributes are truly 'irrelevant' – their values do not depend on class membership. In this experiment, I created a database with 20 continuous attributes, where 12 of these had generating distributions that did not depend on class. Any of the remaining eight attributes can be used to distinguish among the classes, but four are *noisy*: the differences between classes are small, and the standard deviations within a class are high.

Figure 1 shows a learning curve for accuracy with and without the attention mechanism. The predicted attribute is attribute 13, one of the clean, relevant attributes, and accuracy is measured as a simple average of the absolute error of predictions. As expected, both systems reach the same asymptote, although the use of attention seems to result in a slower rate of learning. This may be due to the additional learning task for attention.

Figure 2 characterizes the behavior of the attention algorithm. In this figure, the frequency of attribute inspection is shown over time, for all attributes. Note that the system almost always inspects the three clean, relevant attributes (nos. 14, 15 and 16).[4] CLASSIT next prefers the noisy attributes (nos. 17 through 20) and only rarely does the system inspect the irrelevant attributes (nos. 1 through 12). This figure also shows that the system begins by inspecting many attributes (about 12 to 15), and only after about 30 instances does it 'settle' on the best attributes (inspecting about 7 attributes per instance). It is interesting that this point in time matches the place in Figure 1 where the two systems reach the same level of accuracy.

## Results with real databases

In order demonstrate the practical use of a learning system, one must demonstrate performance of the system in real-world domains. Here, I present results for two databases from the UCI machine learning database repository: [5] the voting database and the heart-disease database. The voting database was collected from 1984 congressional records by Jeff Schlimmer, and consists of 17 symbolic attributes per instance: 16 votes and party affiliation. The heart disease database encodes patient information from the Cleveland Clinic Foundation and was collected by Robert Detrano (see Detrano et al., 1990). This data includes eight numeric attributes and six symbolic ones, including a binary sick/healthy attribute.

Figure 3 shows the accuracy of CLASSIT on these databases with and without the attention mechanism. In both domains, the predicted attribute is binary and symbolic: for the voting database, party affiliation; for the heart-disease database, the sick/healthy attribute. Thus, performance is measured by percentage error – how often the prediction is incorrect. Although the use of attention results in some loss of accuracy in the heart-disease domain, this appears to be a minor effect.

[3] Without the use of the recognition criterion, attention tends to observe all (or almost all) attributes. This is because different attributes are relevant at each level of the tree. By the time classification reaches the leaves of the tree (singleton classes), all or most attributes are 'known'.

[4] Attribute 13 is not available because it is used for the prediction task.

[5] To obtain these databases or information about them, send e-mail to ml-repository@ics.uci.edu, or contact Patrick Murphy, ICS Department, UC Irvine 92717.

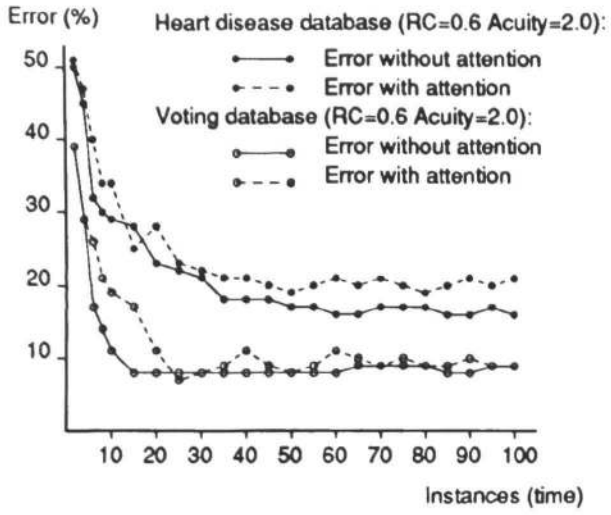


Figure 3: Predictive accuracy – real domains

The attribute frequency graph for the voting database (not shown) provides additional evidence for the attention mechanism. Although not as dramatic as Figure 2, it shows that CLASSIT focuses on partisan votes such as 'aid to El Salvador' and 'aid to the contras', at the expense of non-partisan attributes such as a vote about immigration. In this domain, CLASSIT focuses almost immediately onto about 9 attributes. A simple explanation for this rapid learning is that it is very easy (requires few instances) to distinguish between the two parties in this database.

The heart disease database has an attribute frequency graph that indicates a more even distribution among attributes. This suggests that all or most of the attributes in this domain are relevant for prediction. This is not surprising since the attributes were selected by an expert in the domain. However, even in this domain, CLASSIT learns to use only about 7 attributes out of a total of 13.

## 5. Discussion

One important domain for attention is visual processing. This is an interesting application since real-world images have thousands of attributes and a large amount of irrelevant information. However, before applying attention to vision, I must improve the 'stopping condition' currently used by CLASSIT. Experiments to date have confirmed that the the current method is too costly and too conservative about not risking accuracy.

The results presented here demonstrate that the attention mechanism does work as it should: it increases the efficiency of the system without reducing accuracy. However, experimental results do not "prove" that this is the best approach, nor support this research as a model of human attention and learning. This is simply an effort to use a known phenomena in human psychology to achieve improvement in a learning system. In turn, I hope that as these computer systems evolve and improve, they can lead to advances in developing models of human cognition.

## References


Aha, D. (1989). Incremental, instance-based learning of independant and graded concept descriptions. *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 387–391). Ithaca, NY: Morgan Kaufmann.

Anderberg, M. (1973). *Cluster analysis for applications*. New York: Academic Press.

Billman, D., & Heit, E. (1988). Observational learning from internal feedback: A simulation of an adaptive learning method. *Cognitive Science, 12*, 587–625.

Detrano, R., et al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology, 64*, 304–310.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2*, 139–172.

Gennari, J. H. (1990). *An experimental study of concept formation* (Doctoral dissertation, also Technical Report 90-26). Irvine: University of California, Department of Information and Computer Science.

Gennari, J. H., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence, 40*, 11–61.

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.

Iba, W., & Gennari, J. H. (in press). Learning to recognize movements. In D. Fisher & M. Pazzani (Eds.), *Computational approaches to concept formation*. San Mateo, CA: Morgan Kaufmann.

Lebowitz, Michael. (1987). Experiments with incremental concept formation: UNIMEM. *Machine Learning, 2*, 103–138.

Michalski, R. S., & Stepp, R. (1983). Learning from observation: Conceptual clustering. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

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

Tan, M., & Schlimmer, J. C. (1990). Two case studies in cost-sensitive concept acquisition. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 854–860) Boston, MA: MIT Press.

Treisman, A. M. (1969). Strategies and models of selective attention. *Psychological Review, 76*, 282–299.