Formation of probabilistic concepts through observations containing discrete and continuous attributes

Ricardo Batista Rebouças, João José Vasco Furtado
Mestrado em Informática Aplicada (MIA) – Universidade de Fortaleza (UNIFOR)
Av. Washington Soares, 1321 – 60811-905 – Fortaleza – Ce – Brasil.
ricardo@erija.com.br, vasco@unifor.br

Abstract

The probabilistic concept formation general problem in dealing with mixed-data scale environments is due to the use of different evaluation function for each attribute type. We claim that different behaviors for discrete and continuous evaluation functions are due to an unbalanced contribution for each attribute-type evaluation function inside the main evaluation function. This paper describes an approach based on the difference between the predictability gain for each attribute type. Our approach presents a way to work around for the unbalanced contribution for each attribute-type evaluation functions. Experiments using our approach have shown higher quality in terms of inference ability.

Introduction

Incremental algorithms for concept formation perform the building process of concept hierarchies based on a set of observations (normally a list of attribute/value pairs), which characterize an observed entity. In general, these algorithms perform a heuristic search considering every concept feasible of being generated.

In the real world, most of the observed entities are characterized by a combination of attributes with values with varied types. This article concerns specifically the concept formation based on entities characterized by discrete and continuous attributes.

The main problem in concept formation systems using mixed attributes is to define the heuristic function to measure the quality of concepts. Basically, problems in this area derive from the use of different evaluation functions and the combination of its results. We verified that this kind of approach generates an unbalance due to contribution difference (in terms of value scales) during the calculus of the general evaluation function. This unbalance directly affects the inference ability of the generated hierarchy.

In this article, we analyze the main approaches in probabilistic concept formation systems (PCFS) with mixed data-type attributes and its related problems. Based on these analyses, we propose an approach to get around the identified problems.

Incremental probabilistic concept formation

Incremental concept formation systems perform a process to create a concept hierarchy that generalize the observations represented in a hierarchy node in terms of the conditional probability of the observations' characteristics.

Most recent works in the area of concept formation are based on Fisher's (1987) COBWEB, which forms probabilistic concepts. These concepts have a set of attributes and its possible values. Each concept has also the probability of an observation being classified in it and each attribute value has a predictability associated with. Predictability is the conditional probability of an observation X having the value V to the attribute A, given that X is represented by the concept C or *P(A=V|C).*

COBWEB uses an evaluation function called *Category Utility*. This metric favors the creation of concepts that maximize the inference ability. *Category Utility* for a partition of concepts (*CU({C1,C2,..., Cn})*) is

$$CU\ (C) = \frac{1}{n}\sum_{k=1}^{n} P(C_k)\left[\sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2\right]$$

Equation 1: Traditional Category Utility.

Gluck and Corter (Gluck and Corter 1985) made an association among values computed by the expression $\sum_i\sum_j P(A_i=V_{ij}|C_k)^2$ and the quantity of values that can be inferred to an attribute A, given that this attribute belongs to a certain category *C*. Therefore, this expression indicates the inference ability of an attribute in a category, which we will treat as *predictability (C,A).*

Concept formation using mixed attributes

Many alternatives were proposed to solve the COBWEB problem of not dealing with continuous attributes.

CLASSIT (Gennari, Langley, and Fisher 1989), for instance, assumes that values of continuous attributes are distributed normally and it uses the Gauss curve to determine the occurrence probability of a certain value. In this case, the square sum of the probabilities of a discrete attribute becomes the square of the integral of the continuous attribute normal distribution. *Category Utility* transformed for continuous attribute treatment becomes the following:

$$CU\ numeric\ = \frac{1}{n}\sum_{k=1}^{n} P(C_k)\left[\sum_{i}\frac{1}{2\sqrt{\pi}\sigma_{ik}} - \sum_{i}\frac{1}{2\sqrt{\pi}\sigma_{ip}}\right]$$

Equation 2: Category Utility in CLASSIT.

Where k is the number of partition classes, i is the number of attributes, σ_{ik} is the attribute i standard deviation in the class k, and σ_{ip} is the attribute i standard deviation on the hierarchy root. When the standard deviation becomes 0, CLASSIT uses a parameter called *acuity*, which represents the smallest noticeable difference between two numeric values. Experiments (Li 1995), (Reich and Fenves 1991), and (Yoo and Yoo 1995) showed that the hierarchy quality strongly depends on the acuity value choice. Particularly, COBWEB/3 (McKusick and Thompson 1990), COBIT (Bond and Hine 1993) and CLASSITALL (Moller 1997) systems are part of the CLASSIT family because they are implementations of Gennari's approach, with few modifications.

In ECOBWEB (Reich and Fenves 1991), for continuous attributes, the attribute value arithmetic mean of a category is determined by an interval around the mean value, calculated through the expected number of numeric intervals (n) of an attribute divided by the actual number of intervals for this attribute. The definition of the expected quantity of intervals (n) of an attribute has a significant interference in building the concepts hierarchy, similar to the acuity problem.

ITERATE (Li 1995) is another algorithm that adopts the approach based on the PARZEN WINDOW method (Duda and Hart 1973) to estimate the value for the attribute probability distribution used in the evaluation function. Even though it is an approach that can be used in domains in which numeric values have a distribution different from the normal one, ITERATE keeps using the acuity parameter, initially defined in CLASSIT, and the quality of the created hierarchy continues depending on it as well.

COBWEB95 (Yoo and Yoo 1995) also assumes that continuous values of an attribute are distributed according to a normal curve. The evaluation function for continuous attributes was modified using, as the probability density function, the probability to correctly infer a certain attribute value with some error tolerance. When the standard deviation becomes 0, the probability to correctly infer will be 1, not using the *acuity*. The next section details a problem that was found in the previously mentioned approaches.

Problem characterization

When working with probabilistic concepts, evaluation functions depend on probability distribution of attribute values (probability density functions, PDF for short). For discrete attributes, the probability distribution of its values can be estimated by counting the value occurrences for an attribute. This method is not applicable in continuous attributes because a numeric value is not frequently repeated, which would not be interesting for prediction purposes.

The basic problem in probabilistic concept formation systems is to make PDF for discrete and continuous attributes work together. It is important that their results are equivalent because the predominance of any of the functions can lead to undesirable results, in terms of predicting values.

In traditional Category Utility (COBWEB), the PDF will have results varying according to the number of values of an attribute. Suppose a discrete attribute with 2 values, the PDF result ($\sum_i\sum_j P(A_i=V_{ij}|C_k)^2$) can vary between 0,5 and 1. When the attribute has 3(three) values, similar to the previous example, the limits are within [0,33 ... 1], for four values the limits would be [0,25 ... 1], and so on. The greater the number of values of a discrete attribute, the greater will be the amplitude of PDF possible results.

The evaluation function of continuous attributes has different behavior. Its PDF depends, in general, on some previous knowledge on how the attribute values are distributed. CLASSIT, for instance, according to its PDF for continuous attributes $(2\sqrt{\pi}*1/\sigma)$, will have results that vary according to the standard deviation (σ) of the attribute values. In this case, the smaller the standard deviation of an attribute values is, the greater the PDF's result is, and vice-versa. Suppose the standard deviation of an attribute is 2, for instance, the PDF result would be 0,1410. Notice that the PDF result for continuous attributes has a domain of values wider than the one for discrete attributes.

The amplitude difference of attribute-oriented PDF means that continuous attribute function can have results that the discrete attribute ones would hardly assume. The algorithms of CLASSIT family, COBWEB95 and other approaches share the same problem: they use an evaluation function for continuous attributes with result amplitude different from the one used for discrete attributes.

Besides the amplitude difference between the two PDF, there is another behavior that must be presented: the convergence velocity toward the result limits. Consider a category with 2 distinct values for a discrete attribute. Suppose the insertion of a new observation in this category. The new observation has a third value for the discrete attribute; thus, the mentioned discrete attribute has 3 distinct values for the category. The PDF result, using COBWEB, changes from 0,5 (calculated with 2 values) to 0,33 (calculated with 3 values). Suppose the same scenario for continuous attributes. The continuous attribute has standard deviation equals to 2 for the category. A new observation inserted in the category has a value for the continuous attribute that changes the standard deviation from 2 to 10. In this case, the continuous attribute PDF in CLASSIT, for instance, changes from 0,1410 (standard deviation equals to 2) to 0,0282 (standard deviation equals to 10).

The previous example illustrates that with the insertion of one new observation, the PDF result for discrete attributes decreases in 34%. On the other hand, the function result for continuous attributes decreases, approximately, 80%. This great change in the result of the occurrence probability function for continuous attributes related to discrete attributes ones makes the amplitude difference problem happen whenever a new observation is classified in the hierarchy. Following, we demonstrate an example related to the consequences of this fact in concept formation.

Example

Consider the concept hierarchy (Figure 1a) generated based on the data from table 1. In the mentioned hierarchy, in the level following the top node, two categories were created, C1 and C2. The “C1” category represents the concept of herbivorous animal and the “C2” category represents the concept of carnivorous.

Animal	Offspring	Food	Height
Cow	1	Vegetable	1,65 m
Lion	4-5	Meat	1 m
Buffalo	1	Vegetable	1,7 m
Jaguar	4-5	Meat	0,9 m
Antelope	1	Vegetable	1,68 m
Tiger	4-5	Meat	1,1 m

Table 1: Animals characteristics.

Suppose a new observation with the following characteristics: “OFFSPRING = 1” “FOOD = Vegetable”, and “HEIGHT = 0,95 m”. CLASSIT, for instance, would reorganize the concept hierarchy in figure 1a and it would form the hierarchy depicted in figure 1b, that is, inserting the new observation in category C2. Analyzing figure 1a, intuitively, one can notice that the new observation would be better represented if inserted in category C1 (herbivore, fig. 1c), because it has the same values for discrete attributes, offspring and food, as the values of these attributes in the category. However, CLASSIT inserts the new observation under the category C2 (carnivore), where only one value equals the value in the category, which is the height attribute because it is “closer” to the attribute values mean.

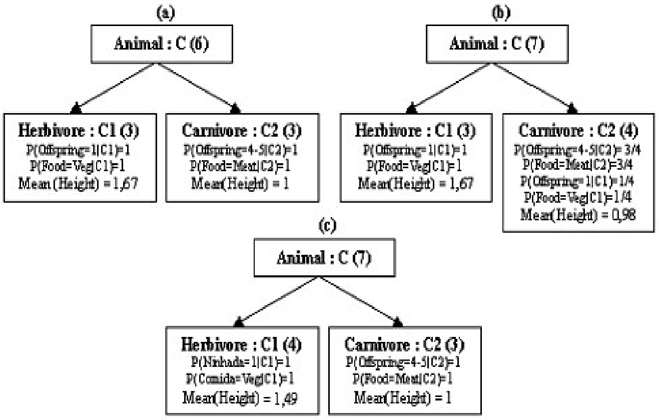

Figure 1: Hierarchies that can be generated based on the data on Table 1.

We can affirm that the hierarchy illustrated in figure 1(b) is not the best option for future prediction because it had two discrete attributes with the occurrence probability decreased. Figure 1(c) exemplifies the situation in which only one attribute, the continuous attribute, would have its inference ability decreased. This behavior is the amplitude difference consequence of the different evaluation functions results. Experiments showed (Rebouças and Furtado 2003) that this behavior happens in all algorithms presented in the previous section. Next section presents an approach for this problem.

FORMVIEW2

This section proposes an evaluation function related to both discrete and continuous attributes. This proposal was implemented in FORMVIEW (Furtado 1997).

FORMVIEW is a concept formation system that finds relations between different concept hierarchies, since these hierarchies were built upon the same entity but with characteristics from different point of view. This approach constitutes a way of linking different expertise. This was the key reason of using FORMVIEW.

FORMVIEW2 proposal, in dealing with mix data-type attributes, is based on the attributes inference ability gain, considering the number of discrete and continuous attributes in the domain.

Variation of Inference Ability

In the classification process on new entities, PCFS typically consider the execution of four operations (insertion, merging, splitting, and creation) to build a concept hierarchy. Among these operations, only the best (The best operation is determined according to the evaluation function) one will be chosen and effectively applied.

The application of the best operation means, besides the change in the hierarchy structure, an update on the information for attributes conditional probabilities calculation. In other words, this information update promotes a variation that directly affect attributes predictability in categories.

Cognitively, this variation defines a factor to measure the increase in attributes inference ability. Even though its cognitive appeal, this variation has been ignored until now by PCFS.

The approach presented here to measure the attribute inference ability variation is based on the predictability percent increase promoted by categories update during the classification process of a new observation. In other words, it is the quotient between the attribute predictability for a category after the insertion of the new observation, and the same predictability before the insertion, as demonstrated in equation 3.

$$\Omega\ (C,A) = \frac{predictability(C, A)}{predictability(C_A, A)}$$

Equation 3: Attribute inference ability variation of a category.

Where *C* represents a category after the insertion of the new observation, while C_A represents the same category before the insertion. Therefore, *Ω(C,A)* is the attribute *A* predictability variation in category *C*. Based on this metric, we propose a solution to the problem caused by the predominance of continuous attributes evaluation function over discrete attributes one's in PCFS, as we present as follows.

Category Inference Ability Gain

It is important to point that the greatest predominance damage is caused by the fact that: even though a new observation has two values for discrete attributes equals to these attributes values in a certain category, the algorithms evaluated in this paper choose the other category as the best choice, in which only one continuous attribute is best suited. This behavior decreases the system performance in terms of future attribute value prediction.

Applying the idea defined by inference ability variation, we can see the predominance issue by another perspective. The coincidence between two discrete values in a new observation and the category values can be seen as a positive variation of the predictability for these two discrete attributes in this category. Similarly, in the other category, there was a positive predictability variation of only one continuous attribute.

One can notice a relation between the continuous attributes predominance and the inference ability variation of these types of attributes.

The approach proposed in this paper indicates that systems should consider the inference ability variation for discrete and continuous attributes besides the quality measured through the traditional evaluation function. That is, the insertion preferred category should be the one with greatest positive inference ability variation beside only the evaluation function.

For this reason, we define a function to measure the inference ability gain of a category in terms of quantity of each attribute-type. This function is calculated through the addition of the inference ability variation of all attributes in the domain, even considering the proportion of each attribute-type in the domain. The proportion of each attribute (*ϕ*) is calculated by the number of attributes of a certain type divided by the total number of attributes in the domain. The equation 4 exemplifies this function.

$$\Psi(C) = P(C) \times \sum_i \phi(A_i) \times \Omega(C, A_i)$$

Equation 4: Category inference ability gain.

Where *P(C)* is the probability of an observation belonging to the category *C*. *ϕ*(A_i) represents the proportion of the attribute-type A_i in the domain and *Ω*(C,A_i) is the inference ability variation of the attribute A_i in the category *C*.

Evaluation Function

The FORMVIEW2 approach implemented in this work was based on COBWEB95. That is, it uses an evaluation function for discrete attributes and another one for continuous attributes. For a better understanding, we remind you of the traditional evaluation function used by COBWEB95.

$$CU(C) = \frac{1}{n}\sum_{k=1}^{n} P(C_k)\begin{cases}\sum_i\sum_j P(A_i = V_{ij} \mid C_k)^2 - \sum_i\sum_j P(A_i = V_{ij})^2 \rightarrow \textit{discrete} \\ \sum_i P[\mid Xi - \mu_{ik} \mid < \delta] - \sum_i P[\mid Xi - \mu_{ik} \mid < \delta] \rightarrow \textit{continuous}\end{cases}$$

Equation 5: Evaluation function used by COBWEB95.

Where *C* represents the top node of a partition formed by the categories C_k. $P(C_k)$ represents the probability of an observation being represented by the category C_k. The top part of the equation is the same one defined in COBWEB, used for discrete attributes. The bottom part is used for continuous attributes and constitutes the COBWEB95's approach. Where $P[|X_{ik} - \mu_{ik}| < \delta]$ represents the probability to correctly infer a value in a normal distribution with an error tolerance δ, since this value belongs to the category C_k. While the equation identified by $P[|X_i - \mu_i| < \delta]$ represents the same probability without the knowledge of C_k. Both are defined in equation 6.

$$\int_{-\delta}^{\delta} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\mu^2}} dx$$

Equation 6: Integral to calculate the occurrence probability of continuous values in COBWEB95.

Where δ indicates the tolerance used, and σ and μ are, respectively, the standard deviation and the attribute value mean in C_k. Without the knowledge of C_k, these values, σ and μ, are taken from the root category.

Even though FORMVIEW practically uses the same approach as COBWEB95, it does not neglect the fact that the different evaluation functions have distinct behaviors, as shown in the previous section concerning the problem characterization.

To apply the category inference ability gain approach defined in this work, we identified the parts that represent the attribute predictability of an attribute A in a category C from the respective evaluation functions. These parts are illustrated on equation 7.

$$\textit{predictabi lity}(C, A) = \begin{cases}\sum_j P(A = V_j \mid C)^2 \quad \rightarrow \quad \textit{Discrete} \\ \int_{-\delta}^{\delta} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\mu^2}} dx \rightarrow \quad \textit{Continuous}\end{cases}$$

Equation 7: Equation to measure the attribute inference ability of "A" attribute in "C" category.

The top part of the equation refers to the discrete attributes while the bottom part refers to continuous attributes. The following section presents the results of the FORMVIEW2 application.

Evaluating FORMVIEW2

The learning model proposed by Dietterich and Michalski (Dietterich and Michalski 1983) motivated the experiments performed in this research work to evaluate FORMVIEW2. This model, in general, concerns learning under three aspects: knowledge base, learning ability and the environment. This article will focus on the learning ability aspect that evaluates systems in terms of prediction power.

The methodology used to measure this ability consists in modifying a test observation by ignoring an attribute and classifying this observation in a previously built concept hierarchy. From the concept found, the algorithm must suggest a value for the attribute based on the attribute value with higher predictability. This process is performed for each attribute of each test observation. The higher the number of correct suggestions, the better the concept hierarchy is, in terms of prediction.

The experiment results for FORMVIEW2 were compared with COBWEB95, COBWEB/3, and COBIT. In FORMVIEW2 and COBWEB95, the error tolerance used was 20%. The *acuity* value used in COBWEB/3 was 1, as suggested in (Gennari, Langley and Fisher 1989).

In a first evaluation, we used 35 artificial mixed-attribute datasets with 100 observations each. The number of attributes in each dataset ranged from 6 to 10, in such a way that each dataset have a different quantity of discrete and numeric attributes. The discrete attributes have from 2 to 4 values. For continuous attributes, its values were randomly selected from a normal distribution.

For the results analysis, we contrast (a) the percent of correct attribute values predictions of the algorithms, (b) the percent increase in correct predictions of FORMVIEW2 related to the other systems, and (c) the number of datasets in which FORMVIEW2 had better, worse or equal performance related to the others.

Table 2 presents the averaged results on the 35 datasets. For the first two comparisons (a) (b), we also highlighted the minimum and the maximum values. The S.T. term in the first comparison (a) represents the correct prediction standard deviation.

System	(a)				(b)			(c)		
	% Correct Prediction				% Increase			Score (%)		
	Avg	Min	Max	S T	Avg	Min	Max	Better	Worse	Equal
COBIT	91,8	81,2	97,2	4,5	4	-2	17	27	5	3
COBWEB/3	65,2	56,6	79,3	4,52	47	28	71	35	0,0	0,0
COBWEB95	92,8	86,6	96,11	2,6	3	-2	8	32	3	0,0
FORMVIEW2	95,5	90	97,9	1,46	-	-	-	-	-	-

Table 2: Averaged results in 35 artificial dataset.

Analyzing the data in table 2, one can notice that (a) FORMVIEW2 had better performance in terms of correct prediction than all other algorithms. The lowest correct prediction standard deviation means that FORMVIEW2 also behaves more constant.

Contrasting FORMVIEW2 performance with COBIT, one can notice that FORMVIEW2 had a prediction increase of 4% in average. The better cases in relation to COBIT had an increase of 17%. On the other hand, in the worst cases COBIT out performs FORMVIEW2 in, at most, 2%. However, FORMVIEW2 had better performance in 27 cases against only 5 worse cases. The same analysis can be done with COBWEB/3 and COBWEB95, and FORMVIEW2 keeps having superior performance as well.

FORMVIEW2 also presented better results in other analysis where we tested the inference ability of systems also in terms of the number of training observations needed to get the maximum prediction performance. It was also analyzed the performance of the systems using different number of discrete and continuous attributes, different number of relevant continuous attributes (Furtado 1998), and different error tolerance for continuous attributes. In all of them FORMVIEW2 out performed the others. These experiments are detailed in (Rebouças and Furtado 2003).

For an analysis using public domain datasets, we used 3 datasets from the UCI ML Repository to demonstrate FORMVIEW2 performance. For each dataset, 80% of the observations were used to train the algorithm, while the other 20% were used for testing. According to Quinlan (Quinlan 1983), the learning ability of a system can be verified through the evaluation of the correct inferences number with different training set size. On top of that, the training observations were divided in subsets with respectively, 25%, 50%, 75%, and 100% of the training observations. Each subset was used to build a hierarchy where the test observations set will be evaluated.

Table 3 shows the characteristics of these databases in terms of the number of discrete and continuous attributes, and the size of the training and test sets.

Base	Discrete	Numeric	Training	Test
BRIDGES	8	3	88	20
AUTO-MPG	3	5	336	70
HEART DISEASE	9	5	252	51

Table 3: Data base information from the UCI ML Repository.

The results from running the algorithms on these databases are depicted in figure 2 as graphics, where the vertical axis on the graphic indicates the correct inferences percent value, while the horizontal axis represents the quantity of observations used on the training set.

The graphics demonstrate satisfactory results from FORMVIEW2 related to other PCFS.

Conclusion and future works

FORMVIEW2 is an incremental algorithm of inductive learning that performs concept formation in domains with discrete and continuous attributes. The basic problem in approaches in these scenarios is the treatment of different attribute-types together because it is necessary to consider the equivalence of values in different evaluation functions for each attribute.

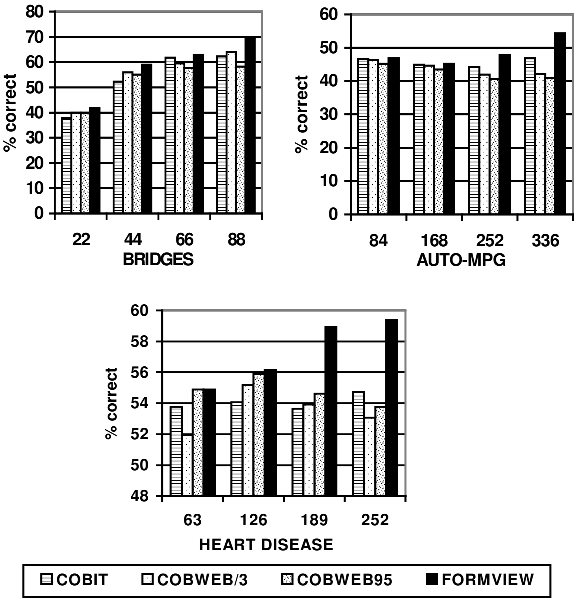

Figure 2: FORMVIEW2 evaluation results.

We showed that the contribution from different evaluation functions in the general evaluation function happens due to the amplitude differences in the attribute-oriented PDF results. The analysis of convergence velocity toward the PDF result limits demonstrated that the contribution difference could happen for each new entity submitted to the algorithms.

This article presents an approach that considers this disparity based on the inference ability gain that is different for each attribute type. Evaluation experiments with FORMVIEW2, showed satisfactory results related to the other approaches, demonstrating to be less prone to factors inherent to environments with mixed attributes.

The study on different evaluation functions, to the general heuristic, presented a different and unbalanced behavior in these functions, which was, up to know, not considered.

Besides the initial satisfactory results, this proposal presents some limitations that must be considered when applied in an unknown domain. FORMVIEW2 assumes that continuous values are distributed according to the normal curve, which can be untrue in some cases.

The use of inference ability gain was possible because the functions are based on probabilistic concepts. The application of the same idea in environments with distinct functions requires a deeper study.

References

Fisher, D. 1987. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*, v.2,n.2,1987.

Furtado, J.J.F, Faucher, C., Chouraqui, E. 1996. Knowledge Acquisition via Multi-perspective Concept Formation. *Journal of Brazilian Computer Society*, v.3.

Furtado, J.J.F. 1998. Determining property relevance in concept formation by computing correlation between properties. In *Proceedings of the Tenth European Conference on Machine Learning, ECML98*, volume 1398 of Lecture Notes in Articial Intelligence, 310-315, Chemnitz, Germany, Springer Verlag. 13.

Gennari, J., Langley, P., and Fisher, D. 1989. Models of Incremental concept formation. *Artificial Intelligence*, 40: 11-62.

Gluck, M. and Corter, J. 1985. Information, uncertainty, and the utility of categories. *Proceedings of the 7th Annual Conference of Cognitive Science Society,* 283-287. Irvine, CA: Lawrence Erlbaum.

Li, C. 1995. Extending ITERATE conceptual clustering scheme in dealing with numeric data. Master's thesis, Vanderbilt University.

McKusick, K. and Thompson, K. 1990. COBWEB/3: A Portable Implementation, Technical Report FIA-90-6-18-2, NASA Ames Research Center.

Rebouças, R.B. and Furtado, J.J.F. 2003. Formação de conceitos probabilísticos através de observações contendo atributos discretos e contínuos. In *Proceedings of IV ENIA Encontro Nacional de Inteligência Artificial*, Campinas, SP.

Reich, Y. and Fenves, S.J. 1991. The Formation and Use of Abstract Concepts in Design. in *Concept Formation: Knowledge and Experience in Unsupervised Learning,* Fisher, D.H. and Pazzani, M.J. and Langley, P. eds., 323-353. Morgan Kaugmann, LosAltos, CA.

Yoo, J., Yoo, S. 1995. Concept Formation in Numeric Domains. ACM 0-89791-737-5, (pp. 36-41).