Concept Formation taking into account Property Relevance

João José Furtado Vasco

UNIFOR – Universidade de Fortaleza
Av. Washington Soares 1321
Centro de Ciências Tecnológicas – CCT
Fortaleza - CE Brazil
vasco@ufc.br

Abstract

We propose a method to incrementally compute and use, during the concept formation, the relevance of a concept property. This relevance is computed through the account of the property correlation with other ones and it is used by the concept quality function in order to improve predictive accuracy. The proposed approach is analyzed concerning both the prediction power of the generated concepts and the time and space complexity of the concept formation algorithm. Initial results show that, in the task of prediction of values for several attributes, the proposed method has improved the prediction power of the generated concepts.

Introduction

The general aim of concept formation is to construct, based on entity descriptions (observations), a (usually hierarchical) categorization of entities. Each category is provided with a definition, called a concept, which summarizes its elements. Further aim is to use concepts to categorize new entities and to make predictions concerning unknown values of attributes of these entities. Therefore, quality of a concept can be measured in terms of its ability to make *good* predictions about unknown values of attributes (the prediction power). Unlike supervised systems, in which concept quality is measured from the capacity in discovering a value for a single property, in concept formation, the quality of a concept is measured by its capacity in allowing prediction of values for several attributes.

An important aspect that must be considered in concept formation concerns the relevance (sometimes called salience) of particular attributes/values (or properties). Cognitive psychology (Tversky 1978), (Seifert 1989), and machine learning (Gennari, Langley and Fisher 1989), (Stepp and Michalski 1986), (Decaestecker 1991) researchers have pointed out the necessity to determine how much a certain property is relevant within a concept.

In this paper, we propose a method to compute the relevance of a concept property based on the correlation between properties of the entities covered by the concept. We describe our approach using a COBWEB-based algorithm, called FORMVIEW, which can generate several hierarchies of categories describing different perspectives (Vasco 97). In a multi-perspective context, the relevance of a property is crucial because it determines the hierarchical organization of categories. Since FORMVIEW uses a probabilistic representation, correlation between properties is computed from conditional probabilities. However, the proposed approach is generic and can be employed in algorithms, which may use other representations.

The prediction power of the category hierarchies generated by the proposed method is computed and compared with those generated by COBWEB. In particular, we analyze the capacity of the categories generated by these algorithms in prediction of values for several unknown properties of entities. We show that, in such a situation, property relevance based on the correlation between properties improves the prediction power of hierarchies, which are produced by concept formation systems.

Concept Formation Systems

Concept formation (CF, for short) or incremental conceptual clustering systems (Stepp and Michalski 1986), (Fisher 1987) recognize regularities among a set of non-preclassified entities and induce a concept hierarchy that summarizes these entities. A CF algorithm is reduced to a search, in the space of the all possible concept hierarchies, for that one that covers the observed entities and optimizes an evaluation function measuring a quality criterion. In concept formation entities are treated one after another as soon as they are observed and the classification of new entities is made by their adequacy for the existing conceptual categories.

Typically concept representation is probabilistic (Fisher 1987), in which concepts have a set of attributes and all possible values for them. Each concept has the probability that an observation is classified into the concept and each value of a concept attribute has associated a *predictability* and a *predictiveness* (Fisher 1987). The predictability is the conditional probability that an observation x has value v for an attribute a, given that x is a member of a category C, or $P(a=v|C)$. The predictiveness is the conditional probability that x is member of C given that x has value v for a or $P(C|a=v)$.

Frame 1 sketches an algorithm for concept formation.

FUNCTION PRINCIPAL (*Root, Observation*)

1. Incorporate *Observation* in *Root*
2. Choose the best operator to employ on the partition P of the Root's direct sub-concepts, among the following:
   a) Incorporate *Observation* into a concept of P
   b) Create a new concept under *Root* to receive *Observation*
   c) Merge the 2 best concepts of P in a new concept that includes Observation
   d) Split a concept of P in its children, adding Observation to the best of these
3. If the operation 2b , read another observation
   Else return to 1 with Root = the concept of P in which Observation was inserted

Frame 1 a CF's structure of control

Concept Formation taking into account the Relevance of a Property

Using the relevance of a property in concept formation

FORMVIEW's concept formation process is similar to that illustrated in Section 2. It constructs *probabilistic concepts* while privileging their prediction power. The category quality function is, like many of its predecessors, based on the Gluck and Corter's work on cognitive psychology (Gluck and Corter 1985), who have defined a function to discover, within a hierarchical classification tree, the basic level category. Category quality is defined as the increase in the expected number of properties that can be correctly predicted given knowledge of a category over the expected number of correct predictions without such knowledge. FORMVIEW, in addition, takes into account the relevance of the properties. We consider that the increase in the expected number of properties to be predicted from a category depends on the relevance of its properties. Formally, the utility of a category UC_s is defined as below:

$$UCs\ (C) = \left(\sum_{i=1}^{p} \Delta(p_i) P(p_i|C) P(C|p_i) - P(C) P(p_i)^2 \right)$$

Where $\Delta(p_i)$ = the relevance of the category property p_i + $P(p_i)$).

Computing property relevance

To compute property relevance, FORMVIEW uses a strategy that relies on attribute dependence (or attribute correlation) in the way that was defined by Fisher (Fisher 1987). Formally, the dependence of an attribute A_m on other n attributes A_i can be defined as :

$$Mdep(Am,Ai) = \frac{\left(\sum_{i} \sum_{ji} P(Ai=Viji) \sum_{jm} [P(Am=Vm|Ai=Viji)^2 - P(Am=Vmjm)^2] \right)}{n}$$

Where V_{iji} signifies the j_ith value of attribute A_i and $A_i \neq A_m$.

In fact, this function measures the average increase in the ability to guess a value of A_m given the value of a second attribute A_i. We consider that this strategy accounts for the relationship between attribute dependence and the ability to correctly infer an attribute's value using a probabilistic concept hierarchy. We can thus determine those attributes that depend on others and, as a consequence, those that influence the prediction of others. By an *influent attribute*, we mean that, if we know its value, it allows a *good* prediction about the value of others. We have thus defined the total influence *Tinfl* of an attribute A_k on others A_m as the following:

$$Tinfl(Ak) = \frac{\sum_{m=1}^{n} Mdep(Am, Ak)}{n} \quad \text{where } A_k \neq A_m.$$

Our claim is that attribute dependence gives a measure to ponder attribute relevance, which, in this context, means how much an attribute correlates with others.

Computing property relevance for each concept

To compute an attribute dependence, we have defined the probability of predicting a property p given another property p' ; $P(p|p')$. Actually, for each concept C within a hierarchy, we have $P(p|p'$ *and* $C)$. The acquisition of this probability is problematic, since it cannot be computed only from the *predictability* and *predictiveness* stored by FORMVIEW. Instead of keeping all the 2x2-property correlation, which would take too much space, or of computing such a correlation for each new observation, which would be computationally expensive, we have defined a procedure that implements a tradeoff between time and space requirements.

	A1	A2	A3	A4
Obs1	v1	v3	v5	v7
Obs2	v2	v4	v6	v8
Obs3	v1	v3	v6	v7
Obs4	v2	v3	v6	v8

		a1		a2		a3		a4	
		v1	*v2*	*v3*	*v4*	*v5*	*v6*	*v7*	*v8*
a1	*v1*	2		2		1	1	2	
	v2		2	1	1		2		2
a2	*v3*			3		1	2	2	1
	v4				1		1		1
a3	*v5*					1		1	
	v6						3	1	2
a4	*v7*							2	
	v8								2

Frame 2 Example of a triangular array used by FORMVIEW after 4 observations

Our procedure consists of maintaining two triangular arrays which keep the 2x2-property correlation : *T-root* and *T-son*. These arrays keep such a correlation for all the observations already seen. T-root is updated once for each new observation. It allows computing the relevance of the root's properties. T-son accompanies side by side the path followed by the observation during its categorization. It is updated at each hierarchical level until the observation

arrives at the leaves. Frame 2 exemplifies the format of the used arrays after four observations.

Procedures for computing the relevance of a property

FORMVIEW procedures for computing dependence and correlation are shown in Frame 3. They were inserted into the procedure which does the incorporation of an observation (steps 4 and 5 in Frame 3).

Step four concerns the update of the arrays, which maintain the frequency of a property correlation. Two procedures are responsible for that activity: *UpdateArray* and *RefineTson*. In *UpdateArray*, the frequency of a property correlation is computed for all the concept properties. The retrieve of this frequency for a specific property is done in step 1.1.1. Function *Shift* allows it to access the cell which stores the correlation between the properties. Let us suppose the example shown in Frame 3. For the properties (A1=v1) and (A4=v7), *Shift* returns line 1 and shifts 6 columns, which correspond to the sum of all the values of attributes less than (considering the array's order) attribute A4. Thus, we can access cell (1,7) of the array.

Step five in Frame 3 concerns the computation of the relevance of each concept attribute of the hierarchy. The procedure *ComputeRelevance* computes the relevance of a property using the conditional probability that an entity has a property given that another property is known. These properties are computed from the frequency of a property correlation represented in T-son.

The procedure *RefineTson* refines *T-son* each time FORMVIEM descends the concept hierarchy. This refinement consists of updating *T-son* in order to let it only with the account of the existing correlation between the properties of observations covered by the current root concept. For each concept, T-son's actualization is based on the following strategy: if the quantity of observations covered by a concept is greater than the total of its *brothers* (children of the concept's father), T-son is updated from those observations which are covered by these later. Otherwise, T-son only stores the property correlation from observations covered by a concept.

Performance Task

In our process of evaluation of FORMVIEW, we pay attention to the prediction power of a hierarchy generated by it. The basic idea is to submit a set of «questions» (normally, incomplete observations) to the system, whose answer is based on the generated representation. The quality of the representation is measured according to its capacity to give «good» answers (i.e. to infer values for attributes).

We have used three test domains: two animal classification domains (ZOO domains) and the Pittsburgh's bridges domain.

```
Incorporate(C, O, Origin)
1. Update C's conditional probabilities
2. Include O in list of observations covered by C
3. Update the number of observations covered by
the node
4. If C is root (father(C)=« nil »)
       4.1 UpdateArray (T-root, O)
       4.2 T-son = T-root
Else   4.3 RefineTsons (T-son, C, C brothers, O)
       4.4 UpdateArray (T-son, O)
5. Computerelevance (T-son)
UpdateArray (Array, O)
1. For each property pi of O
1.1 For each property pz of O (z ≠ i)
1.1.1.Add      1      to      Array((Sub(pi's
attribute)+Sub(pi's     value))-1),(shift(Sub(pz's
attribute))+ Sub(pz's value)))
 /* Sub(x) = retrieves the array subscribe of an
attribute or a value
 /* shift(Sub(x)) = account the number values for
the attributes with sub less than Sub(x)
RefineTson (Array, C, FC, O)
1. if |C's observations| > |FC's observations|
1.1 Subtract in Array the correlations from
observations covered by FC
   Else
1.2 Create Array with account of correlations
from observations covered by FC
ComputeRelevance (Array)
1. For each property pi of a concept
 1.1 For the other properties pz of this same
concept (pz ≠ pi)
 1.1.1.SubAtmin   =   min(Sub(pz's   attribute),
Sub(pi's attribute))
 1.1.2.SubValmin = Sub of the attribute value
having SubAtmin
 1.1.3.SubAtmax   =   max(Sub(pz's   attribute),
Sub(pi's attribute))
 1.1.4.SubValmax = Sub of the attribute value
that has SubAtmax
 1.1.5.P(pi|pz)=
Array((shift(SubAtmin)                          +
SubValmin),(shift(SubAtmax)+SubValmax))/
Array((shift(Sub(pi's          attribute))+Sub(pi's
value)),
(shift(Sub(pi's attribute))+Sub(pi's value)))
 1.1.6.Pertpi = Pertpi + (P(pi|pz)² - P(pz)²)
1.2 Pertpi = Pertpi/Nb properties-1
```

Frame 3 Procedures for computing a property relevance

The ZOO domains

The *zoo* domain was taken from the UCI machine-learning dataset consisting of 53 observations with 18 attributes describing animals. We have divided this dataset into two sets of observations described by 10 and 11 properties, respectively. In fact, we have adapted this domain through

the insertion of attributes in order to represent two perspectives: the *pet* and the *physiologic* perspectives.

Prediction of several properties in this domain was useful to show more clearly the contribution of the use of predictive influence as a heuristic to compute the relevance of a property. Indeed, hierarchies generated from this heuristic have a better prediction power than those generated by other systems. It is due to the fact that concepts are organized around properties having a strong predictive influence. Thus, when one infers the value of a property, he/she increases the probability to infer values for other properties influenced by the first one. Figure 1 and Figure 2 illustrate tests done in two ZOO domains.

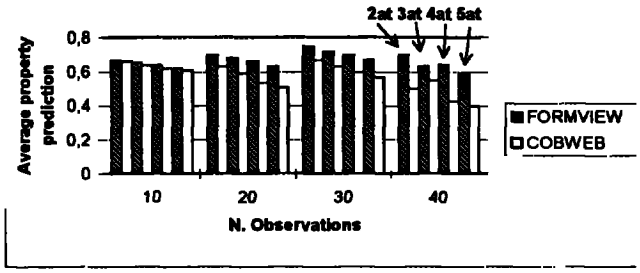

Fig. 1 Prediction of several attributes in FORMVIEW and COBWEB : ZOO Physiologic

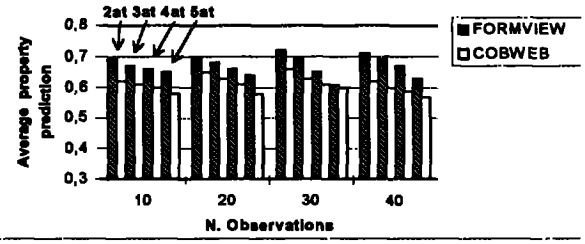

Fig. 2 Prediction of several attributes in FORMVIEW and COBWEB : ZOO Pet

The Pittsburgh Bridges domain

The prediction of several properties often appears in problems of conception. Therefore, we decided to use the domain of Pittsburgh's bridges in our tests. Design domains are characterized by having a set of specification properties which define the user's needs and a set of design properties which describe the artifact's characteristics. The bridge domain contains descriptions of 108 bridges built in Pittsburgh since 1818. Each observation is described by 12 properties of which seven are specification properties and five are design properties. This domain was largely explored by (Reich 1994 and Reich and Fenves 1991), where they show the suitability of COBWEB-like systems to design domains.

We have again defined attribute relevance as a function of the attribute dependence. The task consists of inferring values for all the product properties. We suppose that the specification properties are less susceptible of noises since they are informed by the user guiding the process of construction of the artifact.

Figure 3 shows the predictive accuracy of FORMVIEW against that of COBWEB. The results of these tests indicated us the adequacy of FORMVIEW in constructing hierarchies for the tasks of design. Actually, specification properties are much correlated and, consequently, very *influent*. This causes good inferences on properties of the product.

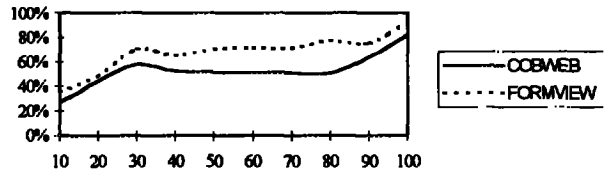

Fig. 3 Prediction power from Pittsburgh domain

Complexity

To examine the time and space complexity required by FORMVIEW's approach, we consider:

n the number of categorized entities

L the average branch factor of the concept tree

The cost of the procedure for computing property relevance is bound to the cost of updating *T-root* and *T-son*. It should be reminded that, FORMVIEW stores, in these arrays, the 2x2-correlation for each concept property. First, we determine the necessary space to stock these correlations. Let *cell* be the unit where the frequency of correlation between two properties will be kept.

In each array that maintains the frequency of correlation between properties, for *n* observations, which have in average *nbAT* attributes with in average *nbV* values, we need $\sum_{i=1}^{nbATxnbV} i$ cells, that is to say, $O\left(nbAT \times nbV\right)^2$ cells.

As for the time complexity, we have the cost to update *T-root* and *T-son*. when a new observation is categorized. The cost of *T-root's* update is the same as that required in space. *T-son's* updating is more expensive than that of *T-root* because it must be done to each level of the hierarchy (on average time $\log_L^n$). For each level, only the frequencies of correlation between observation properties covered by the current node must be represented. Thus, *T-son* must be actualized m times ($m<n$), where *m* represents the minimum between the number of observations which are not covered by the current node and the number of observations covered by the current node. In the worst case, we have $m = n/2$, which would make the geometric progression ($n/2$, $n/4$, $n/8$,..., 1) for all the depth of the tree. The cost of *T-son* actualization is thus the order:

$$O\left(2^{n/2} \times \left(nbAT \times nbV\right)^2\right).$$

Finally, it is necessary to mention that the cost of computing the predictive influence for every concept also

requires time effort of the order $O\left(nbAT \times nbV\right)^2$.

The total cost of the procedure for computing the predictive influence of property is:

$$O\left(2^{n/2} \times (nbAT \times nbV)^3 + \log_L^n \times (nbAT \times nbV)^2\right) =$$
$$O\left((nbAT \times nbV)^2\left(2^{n/2} + \log_L^n\right)\right)$$

Related Work

The definition of a property relevance has been treated in early incremental concept formation systems. ADECLU (Decaestecker 1993) uses a statistical measure to quantify the correlation between the property of a concept and the variable "membership of the concept". It maintains a 2x2-contingence table for each property of each concept. However, there is no account of the correlation between the properties.

In ECOBWEB (Reich and Fenves 1991), property relevance is taken into account in the categorization process in the same way we have implemented here. However, the information of which properties are relevant is given by the user. Cluster/CA (Stepp and Michalski 1986) equally uses the information about property relevance defined by the user in the GDN (Goal Dependence Network). Early versions of FORMVIEW also follow this same idea (Vasco, Faucher and Chouraqui 1995), (Vasco, Faucher and Chouraqui 1996).

The non-incremental system WITT (Hanson and Bauer 1989) computes the correlation between properties to create categories. It keeps for each concept and each property pair a contingence table. Such a table contains the frequency of simultaneous occurrence of property pairs. For A attributes, WITT keeps for each concept A(A-1)/2 contingence tables. This can be a very tough requirement in terms of space.

Conclusion and Future Research

We have defined a method to compute and use the relevance of a property in concept formation systems. The first results obtained with the use of property relevance are encouraging. They shown us that, FORMVIEW's approach can provide representations it generates with better prediction power than those generated from systems that do not take into account the relevance of properties. However, additional tests are necessary, mainly with regards to evaluate the performance of FORMVIEW with more data. In order to analyze the generality of the proposed method, future research consists of the application of this method to systems which use different representations.

References

Decaestecker, C. 1991. Description Contrasting in Incremental Concept Formation. In proceedings of European Working Session Learning.

Decaestecker, C. 1993. Apprentissage et outils statistiques en classification conceptuelle incrémentale. *Revue d'Intelligence Artificielle*, 7(1).

Fisher, D.H. 1987. Knowledge Acquisition via Incremental Conceptual Learning. *Machine Learning*, 2(2).

Fisher, D.; Pazzani, M. and Langley, P. eds. 1991. Concept Formation: Knowledge and Experience in Unsupervised Learning. Morgan Kaufmann.

Gennari, J.H.; Langley, P.; and Fisher, D. eds. 1989. Models of Incremental Concept Formation. *Artificial Intelligence*, 40.

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

Hanson, S. and Bauer, M. 1989. Conceptual Clustering, Categorization, and Polymorphy. *Machine Learning*, 3: 343-372.

Michalski, R.; Carbonnel, J. and Mitchell, T. eds. 1986: Machine Learning, An Intelligence Approach. Vol II. Morgan Kaufmann, CA.

Reich, Y. and Fenves, S. 1991. The Formation and Use of Abstract Concepts in Design. In (Fisher 91).

Reich, Y. 1994. Macro and Micro Perspectives of Multistrategy Learning. In Michalski and Tecuci eds, Machine Learning: A Multistrategy Approach. Vol. IV. Morgan Kauffmann.

Seifert, C. 1989. A Retrieval Model Using Feature Selection. Proc. of the Sixth International Workshop on Machine Learning. Morgan Kauffmann.

Smith, E. and Medin, D.L. 1981. *Categories and Concepts.* Library of Congress Cataloging in Publication Data. Cognitive Science series 4.

Stepp, R. and Michalski, R. 1986. Conceptual Clustering: Inventing goal-oriented classifications of structured objects. In Michalski, R., Carbonnel, J., Mitchell, T. eds. Machine Learning, An Intelligence Approach. Vol II. Morgan Kaufmann, CA.

Tversky, A. and Gati, I. 1978. Studies of similarity. In Rosch, E. and Lloyd, B. eds, Cognition and Categorization, 79-98, Erlbaum.

Vasco, J.J.P.F.; Faucher, C. and Chouraqui, E. 1995. Knowledge Acquisition based on Concept Formation using a Multi-Perspective Representation. Florida AI Research Symposium, Melbourne, FL.

Vasco, J.J.F.; Faucher, C. and Chouraqui, E. 1996. A Knowledge Acquisition Tool for Multi-perspective Concept Formation. In Proceedings of European Knowledge Acquisition Workshop - EKAW-96. Shadbolt, Shreiber and O'Hara, eds. Springer Verlag.

Vasco, J.J.F. 1997. Formation de concepts dans le contexte des langages de schémas. PhD. Diss., Université d'Aix Marseille III, IUSPIM/DIAM.