Learning Different Types of New Attributes by Combining the Neural Network and Iterative Attribute Construction

Yuh-Jyh Hu

Information and Computer Science Department
University of California, Irvine

Abstract. Most of the current constructive induction algorithms degrade performance as the target concept becomes larger and more complex in terms of Boolean combinations. Most are only capable of constructing relatively smaller new attributes. Though it is impossible to build a learner to learn any arbitrarily large and complex concept, there are some large and complex concepts that could be represented in a simple relation such as prototypical concepts, e.g., m-of-n, majority, etc. In this paper, we propose a new approach that combines the neural net and iterative attribute construction to learn relatively short but complex Boolean combinations and prototypical structures. We also carried a series of systematic experiments to characterize our approach.

Keywords: classification, constructive induction, neural networks.

1 Introduction

Poor representation limits the performance of concept learners. One approach to mitigate the limitation is to construct new features. The need for useful new features has been suggested by many researchers (Matheus, 1991; Aha, 1991; Kadie, 1991; Ragavan *et. al.*, 1993). Constructing new features by hand is often difficult (Quinlan, 1983). The goal of constructive induction is to automatically transform the original representation space into a new one where the regularity is more apparent (Dietterich & Michalski, 1981; Mehra *et. al.*, 1989), thus yielding improved classification accuracy.

There are currently many constructive induction algorithms based on the strategy of constructing new attributes, including FRINGE (Pagallo, 1989), GREEDY3 (Pagallo & Haussler, 1990), DCFringe (Yang *et. al.*, 1991), CITRE (Matheus & Rendell, 1989), LFC (Ragavan & Rendell, 1993; Ragavan *et. al.*, 1993), MRP (Perez & Rendell, 1995), GALA (Hu & Kibler, 1996), etc. Unfortunately, most of the current constructive induction algorithms degrade performance as the target concept becomes larger and more complex in terms of Boolean combinations such as prototypical concepts (Perez & Rendell, 1996). Though MRP is demonstrated to learn several complex relations, the meaning of its extensional representation is implicit in the data, and usually difficult to interpret.

Most of the efforts of constructive induction have been directed to improved classification accuracy. However, in addition to accuracy the comprehensibility of new attributes could also be important, especially when the new attributes represent intermediate concepts, which may further understanding of the domain of interest. Without understandable new attributes, the contribution of constructive induction is limited. Because the new attributes that reflect the intermediate concepts are likely to be useful, we currently concentrate on two types of intermediate concepts, i.e., (1) complex but relatively short Boolean combinations and (2) prototypical structures. Combinations of these intermediate concepts into one target concept can easily produce DNF expressions with tens or hundreds of terms, which are difficult to learn. Our goal is to construct these two types of new attributes that represent the intermediate concepts, but unlike MRP's extensional representation, we describe our new attributes in a human-understandable form.

In this paper, we introduce a multi-strategy approach that combines the neural network with GALA. It successfully constructs Boolean combinations and prototypical structures. The Boolean combinations are explicitly represented, and the prototypical structures are clearly described by the weights and thresholds of the neural network. This combination approach, like GALA, is a preprocessor approach. It could be applied to other standard learning algorithms.

2 Combining Neural Network with Iterative Attribute Construction

2.1 Motivation

One important issue in constructive induction is the types of new attributes constructed (Hu & Kibler, 1996). If we represent the new attributes in disjunctive normal form, we could build a spectrum of the attributes based on the number of terms involved. At one end are the ones with relatively smaller numbers of terms; at the other, those with many terms. There is no uniform correlation between the number of terms and the capability of the algorithms, and there is no universal algorithm to cover the whole spectrum. One possible approach to covering more of the spectrum is to combine different construction strategies.

GALA, a preprocessor approach to constructive induction, which applies relative measures and iterative attribute combination techniques is capable of generating complex but relatively smaller Boolean combinations as new attributes (Hu & Kibler, 1996), thus it could be used to construct those new attributes that belong to one end of the spectrum. The general control flow of GALA is described in fig 1.

However, as any iterative attribute construction algorithm, it has an inherent drawback, i.e., as the concept becomes larger in the number of terms in disjunctive normal form, they fail to find useful new attributes. Larger numbers of attribute combinations incur more attribute interaction that hinders the performance of the algorithms (Hu & Kibler, 1996; Perez & Rendell, 1996). Therefore,

```
Given:  a set of attributes P, training examples E, threshold and
        new attributes NEW
        (NEW is empty when GALA invoked the first time)
Return: a set of new attributes NEW

Procedure GALA(P,E,threshold,NEW)
  If (size(E) > threshold) and (E is not all of same class)
    Then Set Bool to Boolean attributes from Booleanize(P,E)
         Set Pool to attributes from Generate(Bool,E)
         Set Best to attribute in Pool with highest gain ratio
            (if more than one, pick one of smallest size)
         Add Best to NEW
         Split on Best
         N = {}
         For each outcome, Si, of Split on Best
            Ei = examples with outcome Si on split
            NEWi = GALA(P,Ei,threshold,NEW)
            N = N + NEWi
         NEW = NEW + N
         Return NEW
    Else Return {}
```

Fig. 1. GALA

we need another construction strategy to cover the other end of the spectrum. Because it is impossible to build a learner to learn any arbitrarily long and complex concept, we currently concentrate on the prototypical structures, such as the m-of-n rules, majority, etc, which often exist in the real domains like medical diagnoses (Spackman, 1988).

Given a target concept containing prototypical structures and other complex but relatively shorter Boolean combinations, our strategy is to first extract the prototypical structure and represent it as a single new attribute. Second, with the new prototypical attributes added to the primitives, we then apply GALA to extract the remaining Boolean combinations in the target concept. Since the new attributes generated encapsulates the complexity of the prototypical structures, GALA is able to learn the remaining Boolean combinations. The question left is how we extract the prototypical structures.

The multilayer perceptron is probably the most studied neural network technique (Hertz *et. al.*, 1991; Kung, 1993). Each hidden unit draws a simple decision surface in the input space, and then the output units combine these individual regions to form a final region corresponding to the target concept. As the decision region of each hidden unit is formulated in a linear function, some of the hidden units are likely to converge to meaningful linear threshold functions, and could be used as the basis of prototypical structures such as the m-of-n rules,

majority, etc. The linear threshold functions could then be transformed into new attributes to represent prototypical structures. Therefore, the neural network is our answer to the question left above. The general framework of our approach is described in fig 2.

```
Given : a set of training examples E
        a set of primitive attributes P
Return: a set of new attributes NEW

Procedure ANN-GALA(E,P)
   H = hidden units from ANN(E)
   Transform H to a set of new attributes N
   Represent E in P+N as E'
   NEW={}
   NEW = N + GALA(P+N,E',threshold,NEW)
   Return (NEW)
```

Fig. 2. General Framework of Combining ANN with GALA

2.2 How to Apply Neural Network

Since GALA generates new attributes in terms of Boolean combinations, the new attributes generated are human-understandable. However, the new attributes directly derived from the hidden units of the neural network are difficult to interpret.

From the point of view as a prototypical structure, there are two causes of the incomprehensibility. The first is the irrelevant primitive attributes. In the feed-forward neural netwrok architecture, each hidden unit is fully connected to all the input units (i.e., primitive attributes). After the network converges, the links between the hidden units and the irrelevant input units may still carry non-zero weights. These irrelevant non-zero weights, though usually small, make the representation difficult to interpret. It would be more human-interpretable if all the irrelevant weights are zero. The other cause is the weights themselves. The weights are unlikely to be integers after the network converges. It is difficult to infer the prototypical structures from a set of non-integer weights. For example, a 3-of-5 rule is better represented by $X_1 + X_2 + X_3 + X_4 + X_5 \geq 3$ than $1.02X_1 + 0.98X_2 + 1.05X_3 + 0.97X_4 + 0.98X_5 \geq 2.93$.

To overcome the above problems, we propose a two-stage weight normalization method. The irrelevant weights are usually significantly smaller than others in magnitude and often with opposite sign. In the first stage, we use the mean

of the weight magnitude as the criterion. The weights below the mean and of the opposite sign are set to zero. The reason why we consider the sign is that in case all input units are relevant and of the same sign, we will not discard any relevant weight simply because it is below the mean. Along with the change of weights, we use the difference between the hidden unit value before and after the weight change to adjust the threshold of the hidden unit. In the second stage, we normalize the weights with the mean of the remaining non-zero weights and apply the round-off function to suppress other insignificant weights to zero. More details could be found in fig 3.

```
Given : a set of weights and thresholds W,
        a set of training examples E
Return: a set of normalized weights and thresholds

Procedure normalize(W,E)
  Let W' be W
  Let m be the mean of the absolute values
      of the weights in W'
  Let s be the sign (i.e., + or -) of the weight with
      max absolute value
  For each weight w' in W'
    if (abs(w') < m) and (w' with opposite sign to s)
       w' = 0
  sumdiff = 0
  For each training example e in E
    sumdiff = sumdiff+(eW-eW')
  diff = sumdiff/|E|
  Let m be the mean of the absolute values
      of the non-zero weights in W'
  For each weight w' and threshold t' in W'
    w' = round(w/m);
    t' = round((t'+diff)/m);
  Return (W')
```

Fig. 3. Weights and Thresholds Normalization

The set of weights and thresholds are difficult to interpret, but after we *normalize* them the concept represented by these weights and thresholds is easier to understand. We use $(\sum_{i=1}^{8} x_i \geq 5) + x_1\bar{x}_3\bar{x}_5x_7 + \bar{x}_2\bar{x}_4\bar{x}_6x_8$, a Boolean function with total 12 attributes, as an example to illustrate the idea. Using 5% of the total 4096 examples as the traing set, we trained a neural net with the back-propagation algorithm. The neural net has 12 input units, 6 hidden units and one output unit.

The weights and thresholds before and after normalization are shown in fig. 4. In fig. 4, each hidden unit is described by 12 weights followed by a threshold. Before normalization, the weights and thresholds we learned are not understandable even though the fifth hidden unit approximately converged to $\sum_{i=1}^{8} x_i \geq 5$. However, after the normalization, the fifth hidden unit is easily interpreted as a 5-of-8 rule.

```
The weights of the hidden units before normalization :
Hidden Unit 1:  4.43948,-0.90867,-1.85273,-0.09461,-2.56182,
                0.44131, 3.80673, 0.88315, 1.21998,-0.02205,
                0.03032,-1.60084,       threshold = -7.18849

Hidden Unit 2: -3.00333, 2.49571, 3.30566, 1.22784, 3.31664,
                1.30544,-2.78767, 0.28106,-0.92781, 0.50652,
                0.62082, 1.21651,       threshold =  2.36028

Hidden Unit 3:  1.49163,-0.95159,-0.32819,-0.27491, 1.43506,
               -0.56259, 1.37591, 0.87493, 0.52599,-0.69051,
                1.22448, 2.60315,       threshold = -2.63325

Hidden Unit 4: -1.09608,-3.30526,-1.54907,-2.02781, 0.96142,
               -1.87570,-0.97085, 1.95589,-2.10087, 0.95469,
                0.62773, 1.00338,       threshold = -2.89914

Hidden Unit 5:  3.13984, 3.82628, 3.94562, 3.75279, 3.37203,
                3.75229, 3.07394, 3.34460,-0.24504, 0.06971,
               -0.26181,-0.70992,       threshold = -15.6727

Hidden Unit 6:  0.48870, 1.04775, 0.81234, 0.72570,-0.11177,
                0.78070, 0.52246,-0.42331, 0.66164,-0.36449,
               -0.41729,-0.08977,       threshold =  0.41798

The weights of the hidden units after normalization :
Hidden Unit 1:  2, 0,-1, 0,-1, 0, 1, 0, 0, 0, 0,-1, threshold = -2
Hidden Unit 2: -1, 1, 1, 1, 1, 1,-1, 0, 0, 0, 0, 1, threshold =  1
Hidden Unit 3:  1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 2, threshold = -3
Hidden Unit 4: -1,-2,-1,-1, 0,-1, 0, 1,-1, 0, 0, 0, threshold =  0
Hidden Unit 5:  1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, threshold = -5
Hidden Unit 6:  0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, threshold =  0
```

Fig. 4. Results of Normalization

3 Experimental Results

There are three purposes of our experiments. One is to demonstrate that the new attributes generated could improve the predictive accuracy of the standard learning algorithms. Another is to verify that combining the neural network and GALA covers more of the attribute spectrum that we introduced earlier. The third is the characterization of the conditions under which our approach is likely to perform better.

We examined our approach across a variety of Boolean functions. This allows us full control of the experiments, and we could exactly verify whether our approach could extract all the intermediate concepts. For example, given a target concept $(\sum_{i=1}^{8} x_i \geq 5) + x_1\bar{x}_3\bar{x}_5x_7 + \bar{x}_2\bar{x}_4\bar{x}_6x_8$, to determine whether the Boolean combination attributes generated are correct we compare them with the Boolean combinations (i.e., $x_1\bar{x}_3\bar{x}_5x_7 + \bar{x}_2\bar{x}_4\bar{x}_6x_8$) in the target concept. As for prototypical structures, the weights and thresholds of the hidden units are compared to the prototypical structure (i.e., $\sum_{i=1}^{8} x_i \geq 5$) in the target concept, and checked if they are exactly presented as the following. Note that irrelevant weights are set to zero.

```
Hidden Unit : 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, threshold = -5
```

Each function is defined on 12 Boolean attributes that produce total 4096 examples. We used 5% and 10% of the total 4096 examples as the training set respectively, and the rest as the testing set.

3.1 Pure Prototypical Concepts

The first part of the experiments is to verify if our neural network strategy is able to extract the prototypical structures when they are the only intermediate concepts in the target concept. We tested several prototypical concepts, including m-of-n, majority (a special case of m-of-n), exact-m-of-n, etc. The summary of the functions are described in table 1. The results are averaged over 20 runs, and reported in table 2.

The second (and sixth) column of table 2 denotes the percentage averaged over 20 runs that the neural network successfully extracted the prototypical structures. The third and fourth (also seventh and eighth) columns denote the accuracy of the neural net and C4.5 (Quinlan 1993) respectively. The new accuracy of C4.5 after adding the new attributes derived from the hidden units is presented in column 5 and 9. Significant difference between the C4.5's accuracy before and after adding the new attributes is marked with "*".

3.2 Concepts composed of prototypical structures and Boolean Combinations

Besides the concepts which contain prototypical structures only, there exist other concepts that are composed of prototypical structures and Boolean combinations

Table 1. Summary of Pure Prototypical Concepts

Concept	Description
P1	$\sum_{i=1}^{12} x_i \geq 5$
P2	$(\sum_{i=1}^{6} x_i \geq 4) + (\sum_{i=7}^{12} x_i \geq 4)$
P3	$(\sum_{i=1}^{6} x_i \geq 4) + (\sum_{i=5}^{10} x_i \geq 4)$
P4	$\sum_{i=1}^{10} x_i = 3$
P5	$\sum_{i=2}^{10} x_i \geq 5$
P6	$\sum_{i=5}^{6} x_i = \sum_{i=7}^{8} x_i$
P7	$\sum_{i=3}^{6} x_i = \sum_{i=7}^{10} x_i$
P8	$\sum_{i=3}^{6} x_i > \sum_{i=7}^{10} x_i$

Table 2. Results of Pure Prototypical Concepts

	5%				10%			
Concept	%	NN	C4.5	+NN	%	NN	C4.5	+NN
P1	100	99.9	82.2	100.0*	100	100.0	82.7	100.0*
P2	95	98.8	73.4	98.9*	100	99.9	77.5	100.0*
P3	65	97.5	79.2	96.1*	95	99.9	84.7	99.7*
P4	15	93.4	83.2	94.3*	85	98.6	83.9	99.2*
P5	100	100.0	76.4	100.0*	100	100.0	79.9	100.0*
P6	100	99.4	72.1	100.0*	100	100.0	77.3	100.0*
P7	100	99.5	63.8	100.0*	100	99.9	64.9	100.0*
P8	100	100.0	79.9	100.0*	100	100.0	84.3	100.0*

together, e.g., $(\sum_{i=1}^{8} x_i \geq 5) + x_1\bar{x}_3\bar{x}_5x_7 + \bar{x}_2\bar{x}_4\bar{x}_6x_8$, where $\sum_{i=1}^{8} x_i \geq 5$ is the prototypical structure, and $x_1\bar{x}_3\bar{x}_5x_7 + \bar{x}_2\bar{x}_4\bar{x}_6x_8$ is the Boolean combinations.

The second part of the experiments is to verify if the multi-strategy approach could extract the prototypical structures and Boolean combinations respectively, given a target concept that is composed of prototypical structures and Boolean combinations. Thus, in addition to the percentage that the neural network successfully extracted the prototypical structures, we also examined how many terms in the Boolean combinations GALA successfully generated.

These concepts are further categorized into two categories by comparing the example space covered, i.e., whether the prototypical structure significantly covers more example space than any term in the Boolean combinations. If this condition is true, we call this category of concepts "dominant prototypical concepts"; otherwise, "nondominant prototypical concepts".

Dominant Prototypical Concepts In this category, we tested 9 Boolean concepts summarized in table 3. The literals in the Boolean combinations could be included in, separated from, or overlapped with the prototypical structure.

Table 3. Summary of Dominant Prototypical Concepts

Concept	Description
D1	$(\sum_{i=1}^{8} x_i \geq 5) + x_1\bar{x}_3\bar{x}_5x_7 + \bar{x}_2\bar{x}_4\bar{x}_6x_8$
D2	$(\sum_{i=1}^{8} x_i \geq 5) + x_9\bar{x}_{11}\bar{x}_1x_3 + \bar{x}_{10}x_{12}x_2\bar{x}_4 + x_{11}\bar{x}_1x_3x_5$
D3	$(\sum_{i=1}^{8} x_i \geq 5) + x_9\bar{x}_{10}\bar{x}_{11}x_{12} + \bar{x}_9x_{10}\bar{x}_{11}\bar{x}_{12} + x_9\bar{x}_{10}x_{11}\bar{x}_{12}$
D4	$(\sum_{i=1}^{6} x_i \geq 3) + x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_2x_3\bar{x}_4\bar{x}_5 + x_3\bar{x}_4x_5\bar{x}_6$
D5	$(\sum_{i=1}^{6} x_i \geq 3) + x_1\bar{x}_3\bar{x}_7x_8 + \bar{x}_3x_4\bar{x}_8x_9 + x_2\bar{x}_5x_9x_{10}$
D6	$(\sum_{i=1}^{6} x_i \geq 3) + x_7\bar{x}_8\bar{x}_9x_{10} + \bar{x}_9x_{10}\bar{x}_{11}\bar{x}_{12} + x_7\bar{x}_9x_{10}\bar{x}_{11}$
D7	$(\sum_{i=1}^{8} x_i \geq 5) + x_1\bar{x}_2\bar{x}_3 + \bar{x}_3x_5\bar{x}_8 + \bar{x}_4x_6x_7 + x_7x_8\bar{x}_4$
D8	$(\sum_{i=1}^{8} x_i \geq 5) + x_1\bar{x}_9\bar{x}_{10} + \bar{x}_3x_{11}\bar{x}_{12} + \bar{x}_5x_7\bar{x}_9 + x_6\bar{x}_{10}x_{11}$
D9	$(\sum_{i=1}^{8} x_i \geq 5) + x_9\bar{x}_{10}x_{11} + x_9x_{11}\bar{x}_{12} + x_{10}\bar{x}_{11}\bar{x}_{12} + x_9x_{11}x_{12}$

Table 4. Results of Dominant Prototypical Concepts (5%)

		5%				
Concept	%	DNF	NN	C4.5	+NN	+NN+GALA
D1	85	1.1,0.0	94.4	78.4	94.7*	95.5#
D2	55	0.8,0.3	90.5	77.7	87.3*	91.0#
D3	50	1.0,0.3	92.7	68.6	86.2*	91.8#
D4	50	0.0,1.1	96.9	91.5	97.2*	97.4#
D5	70	0.1,0.6	94.5	85.1	93.1*	93.8#
D6	100	0.0,1.0	97.1	84.3	96.1*	97.3#
D7	15	0.4,0.1	88.8	82.2	86.1*	89.3#
D8	15	0.3,0.3	85.1	75.1	82.2*	84.0#
D9	80	0.1,0.8	95.5	78.8	95.9*	97.5#

The results are reported in table 4 and 5. The second column has the same meaning as column 2 (and column 6) in table 2. The first number in the third column presents the average number of terms constructed by GALA that exactly correspond to those in the Boolean combinations in the target concept, and the second number denotes the average number of terms that approximate. By approximate we mean that the term is part of the Boolean conjuncts (i.e., overly

general). For example, given a target concept $(\sum_{i=1}^{8} x_i \geq 5)+x_1x_2x_3+x_3x_4x_5+x_1x_5x_7$, we may have the following terms such as x_1x_2 and x_3x_4 that approximate $x_1x_2x_3$ and $x_3x_4x_5$ respectively.

The fourth and fifth columns denotes the accuracy of the neural net and C4.5 respectively. The new accuracy of C4.5 after adding the new attributes generated by the neural net alone is presented in column 6. Column 7 denotes the new accuracy of C4.5 after adding all the new attributes constructed by the neural net and GALA together. Significant difference between the C4.5's accuracy before and after adding the new attributes is marked with "*" and "#" respectively.

Table 5. Results of Dominant Prototypical Concepts (10%)

10%						
Concept	%	DNF	NN	C4.5	+NN	+NN+GALA
D1	100	1.2,0.2	99.3	82.2	99.5*	99.7#
D2	90	2.0,0.4	96.3	80.7	96.1*	98.2#
D3	85	1.0,0.8	99.7	72.7	97.3*	97.9#
D4	40	0.0,1.0	99.6	97.1	99.1*	99.2#
D5	85	0.7,0.6	96.5	91.5	95.8*	97.8#
D6	100	0.0,1.7	98.5	89.7	98.1*	99.2#
D7	35	0.9,0.1	95.7	88.6	93.8*	94.2#
D8	70	1.8,0.2	92.9	81.1	93.4*	95.5#
D9	100	0.4,0.7	99.8	82.2	99.9*	99.9#

Nondominant Prototypical Concepts For the second category, we tested 3 concepts. These concepts are summarized in table 6, and table 7 and 8 present these results. Note that in the first test concept (i.e., ND1), the percentage (i.e., column 2) is 0, and the average number of terms generated by GALA that correspond to the Boolean combinations is also low (i.e., column 3). The reason is that the neural net strategy failed to extract the prototypical structure, and the literals of the Boolean combinations (i.e., $x_1..x_5$) are included in the prototypical structure; therefore, most of the new attributes generated by GALA are part of the prototypical structure instead of the Boolean combinations.

3.3 Analysis

In the first part of the experiments, all the Boolean functions are either the combination of linearly separable concepts or simply linearly separable concepts themselves. As expected, the neural network strategy successfully extracted the prototypical structures in almost every Boolean function. This demonstrates

Table 6. Summary of Nondominant Prototypical Concepts

Concept	Description
ND1	$(\sum_{i=1}^{5} x_i \geq 4) + x_1x_3\bar{x}_4 + x_2\bar{x}_3x_5 + x_1\bar{x}_2\bar{x}_4 + x_2x_4\bar{x}_5$
ND2	$(\sum_{i=1}^{5} x_i \geq 4) + x_1\bar{x}_9\bar{x}_{10} + \bar{x}_3x_{11}\bar{x}_{12} + \bar{x}_5x_7\bar{x}_9 + x_6\bar{x}_{10}x_{11}$
ND3	$(\sum_{i=1}^{5} x_i \geq 4) + x_6\bar{x}_8\bar{x}_9 + x_8\bar{x}_{10}x_{11} + x_7\bar{x}_9x_{12} + x_7x_{11}\bar{x}_{12}$

Table 7. Results of Nondominant Prototypical Concepts (5%)

5%						
Concept	%	DNF	NN	C4.5	+NN	+NN+GALA
ND1	0	0.3,1.2	97.3	98.4	97.8	99.1
ND2	15	2.1,0.2	84.3	80.2	82.6*	90.5#
ND3	0	2.8,0.0	82.3	82.5	81.1	92.7#

that when the target concept is a pure prototypical concept, the hidden units are likely to converge to the prototypical structures. After the normalization of the weights and thresholds, the prototypical structures are explicitly represented by the hidden units. By adding the prototypical structures as new attributes, we significantly improve the accuracy of C4.5.

When the target concept is composed of prototypical structures and Boolean combinations, our multi-strategy approach performed differently on the two categories of the concepts (i.e., dominant and nondominant). If the prototypical structures in the target concept significantly cover more example space (i.e., the dominant prototypical concepts as defined earlier), the neural net strategy is able to separate the prototypical structures from the Boolean combinations, and consequently the hidden units could converge to the prototypical structures as in the pure prototypical concepts. After we transform the prototypical structures into new attributes, GALA could avoid the complexity of the prototypical structures that has been encapsulated in the new prototypical attributes, and thus extract the remaining Boolean combinations.

Table 8. Results of Nondominant Prototypical Concepts (10%)

10%						
Concept	%	DNF	NN	C4.5	+NN	+NN+GALA
ND1	0	0.0,0.1	99.7	99.5	99.8	100.0
ND2	45	2.5,0.1	94.2	86.6	94.8*	98.5#
ND3	25	2.0,0.5	92.5	88.5	93.2*	96.1#

As for the nondominant prototypical concepts, the neural network has difficulties distinguishing the prototypical structures from the Boolean combinations. Therefore, the hidden units are unable to converge to any meaningful prototypical structure. GALA, in this case, treats the whole target concept as a *big* Boolean combination concept, and extracts as many terms as possible. These terms may be either part of the prototypical structures or of the Boolean combinations.

4 Conclusion

When describing constructed attributes in disjunctive normal form, we have an attribute spectrum based on the number of terms involved. Unfortunately, there is no universal learning algorithm that could learn the whole spectrum.

We currently concentrate on relatively short but complex Boolean combinations and prototypical structures. The objective of this paper is to learn these two types of attributes and represent the new attributes in a human-understandable form, unlike other systems that adopt extensional representation. We proposed a new approach that combines the neural network and iterative attribute construction. With different characteristics and advantages, the neural network strategy is used to learn the prototypical structures; the iterative attribute construction algorithm, to learn the complex Boolean combinations. The new attributes generated are either explicitly represented in terms of Boolean combinations or human-understandably described by the weights and thresholds of the hidden units (Section 2.2).

Besides introducing the new approach, we also carried a series of systematic experiments to characterize our approach. We tested and analyzed our multi-strategy approach on different categories of concepts, and conclude that example space coverage plays an important role in the characterization (Section 3.3).

Though there exist more complex concepts than those we studied in this paper, we argue that our test concepts are inspired by the real domains such as medical diagnoses. Those simplifications do not prevent us from analyzing the essential information. Our results illustrate other directions for future research in multi-strategy learning and constructive induction.

References

Aha, D. "Incremental Constructive Induction: An Instanced-Based Approach", in Proceeding of the 8th Machine Learning Workshop, p117-121, 1991.

Dietterich, T. G. & Michalski, R. S. "Inductive Learning of Structural Description : Evaluation Criteria and Comparative Review of Selected Methods", Artificial Intelligence 16 (3), p257-294, 1981.

Hertz, J., Krogh, A. and Palmer, R. G. "Introduction to the Theory of Neural Computation", Addison-Wesley, 1991.

Hu, Y. & Kibler, D. "Generation of Attributes for Learning Algorithms", in Proceeding of the 13th National Conference on Artificial Intelligence, p806-811, 1996.

Kadie, C. M. "Quantifying the Value of Constructive Induction, Knowledge, and Noise Filtering on Inductive Learning", in Proceeding of the 8th Machine Learning Workshop, p153-157, 1991.

Kung, S. Y. "Digital Neural Networks", Prentice Hall, 1993.

Matheus, C. J. & Rendell, L. A. "Constructive Induction on Decision Trees", in Proceeding of the 11th International Joint Conference on Artificial Intelligence, p645-650, 1989.

Matheus, C. J. "The Need for Constructive Induction", in Proceeding of the 8th Machine Learning Workshop, p173-177, 1991.

Mehra, P., Rendell, L. A., Wah, B. W. "Principled Constructive Induction", in Proceeding of the 11th International Joint Conference on Artificial Intelligence, p651-656, 1989.

Pagallo, G. "Learning DNF by Decision Trees", in Proceeding of the 11th International Joint Conference on Artificial Intelligence.

Pagallo, G. & Haussler, D. "Boolean Feature Discovery in Empirical Learning", Machine Learning 5, p71-99, 1990.

Perez, E. & Rendell, L. "Using Multidimensional Projection to Find Relations", in Proceeding of the 12th Machine Learning Conference, p447-455, 1995.

Perez, E. & Rendell, L. "Learning Despite Concept Variation by Finding Structure in Attribute-based Date", in Proceeding of the 13th Machine Learning Conference, p391-399, 1996.

Quinlan, J. R. "Learning efficient classification procedures and their application to chess end games" , in Michalski *et. al.*'s Machine Learning : An artificial intelligence approach. (Eds.) 1983.

Quinlan, J. R. *C4.5 : Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.

Ragavan, H., Rendell, L., Shaw, M., Tessmer, A. "Complex Concept Acquisition through Directed Search and Feature Caching", in Proceeding of the 13th International Joint Conference on Artificial Intelligence, p946-958, 1993.

Ragavan, H. & Rendell, L. "Lookahead Feature Construction for Learning Hard Concepts", in Proceeding of the 10th Machine Learning Conference, p252-259, 1993.

Rendell L. A. & Ragavan, H. "Improving the Design of Induction Methods by Analyzing Algorithm Functionality and Data-Based Concept Complexity", in Proceeding of the 13th International Joint Conference on Artificial Intelligence, p952-958, 1993.

Spackman, K. "Learning Categorical Decision Criteria in Biomedical Domains", in Proceeding of the 5th International Workshop on Machine Learning, p36-46, 1988.

Yang, D-S., Rendell, L. A., Blix, G. "A Scheme for Feature Construction and a Comparison of Empirical Methods", in Proceedingof the 12th International Joint Conference on Artificial Intelligence, p699-704, 1991.