Pre-publication draft of paper accepted for publication in the Proceedings of AI'92, pp 374-379

DISCRIMINANT ATTRIBUTE FINDING IN CLASSIFICATION LEARNING

Simon P. Yip
Department of Computer. Science, Swinburne University of Technology, Hawthorn, 3122, Australia
Geoffrey I. Webb
Department of Computer. Science., Deakin University, Geelong 3217, Australia

Abstract

This paper describes a method for extending domain models in classification learning by deriving new attributes from existing ones. The process starts by examining examples of different classes which have overlapping ranges in all of their numeric attribute values. Based on existing attributes, new attributes which enhance the distinguishability of a class are created. These additional attributes are then used in the subsequent classification learning process. The research revealed that this method can enable relationships between attributes to be incorporated in the classification procedures and, depending on the nature of data, significantly increase the coverage of class descriptions, improve the accuracy of classifying novel instances and reduce the number of clauses in class description when compared to classification learning alone. Evaluation with the data on iris flower classification showed that the classification accuracy is slightly improved and the number of clauses in the class description is significantly reduced.

1 Introduction

One knowledge acquisition technique involves an inductive learning algorithm capable of extracting from a set of positive and negative training instances general rules which can be incorporated in knowledge-based system to classify novel instances. Attribute-value classification learning algorithms, such as AQ (Michalski, 1980, 1984) or 1D3 (Quinlan, 1986), aim to derive classification procedures capable of defining one class of instances as different from other classes. Most such systems are limited in that they are restricted to the domain model with which they are supplied. The condition parts of the classification rules are based on the range of values of each attribute. These algorithms have not in general supported the derivation of conditions based on relationships between attributes.

It is obvious that if the class description is outside the description space that is defined by the domain model which is stated in terms of available attributes or features, then it can only be learnt by extending that space. Indeed, it is possible that the relevant attributes or best features that could be used in the class description may not be explicit or included in the examples (Elio & Watanabe, 1991). Recent effort (Yip & Webb, 1992) on extending the domain model involves incorporating functional relationships between attributes in classification learning. This paper reports an attempt to extend the domain model by creating new attributes in classification learning, with the objective of inducing better classification procedures. Let us use a simple example to illustrate the idea.

Consider the following observations on a sample of male adults:

Weight (wt) (kg)	**Height (ht) (m)**	**Class**
100	1.8	obese
115	1.7	obese
76	1.7	overweight
112.5	1.97	overweight
55	1.63	normal
86	1.9	normal
55	1.7	underweight
58	1.8	underweight

A typical attribute-value classification learning method would report the following:

IF ((100.00<=wt<=115.00) & (1.70<=ht<=l.80)) THEN class = obese
IF ((112.50<=wt<=112.50) & (1.97<=ht <=1.97)) V (76.00<=wt<=76.00) THEN class = overweight
IF ((86.00<=wt<=86.00) & (1.90<=ht<=1.90)) V (1.63<= ht<=1.63) THEN class = normal
IF ((55.00<=wt<=58.00) & (1.70<=ht<=1.80) THEN class = underweight

Suppose we create a new attribute: wt/ht^2 in the learning process, we can arrive at better rules:

IF ($30.86<=wt/ht^2<=39.79$) THEN class = obese

IF ($26.30<=wt/ht^2<=28.99$) THEN class = overwe ight

IF ($20.70<=wt/ht^2<=23.82$) THEN class = normal

IF ($17.90<=wt/ht^2<=19.03$) THEN class = underweight

The attribute: wt/ht^2 is in fact, the Body-mass-index, which medical practitioners use as a guide line to classify male adults according to the rules:

IF ($wt/ht^2>30$) then class = obese; IF ($25< wt/ht^2<=30$) then class = overweight;

IF ($20<=wt/ht^2<=25$) then class = normal; IF ($wt/ht^2<20$) then class = underweight.

The research reported herein seeks to improve upon existing classification learning systems by extending the domain model to include new attributes that can enhance the distinctiveness of a class from other classes. In the following sections, we discuss a classification learning algorithm and then introduce a method of finding and incorporating new attributes in classification learning.

2 Disjunctive Least Generalization algorithm (DLG):

The inductive rule learning algorithm used in this research is disjunctive least generalization (DLG) (Webb, 1991a). It is a variant of the AQ algorithm (Michalski, 1980, 1984) that can process continuous attributes and does not use arbitrary parameters to constrain its search. It is an efficient data driven learning algorithm that can generate disjunctive class descriptions. It differs from other members of the AQ family in the manner in which it develops disjuncts. These are developed by least generalization (Plotkin, 1970, 1971). The DLG algorithm can be expressed as follows:

```
Input:  POS (a training set of instances belonging to the class of interest)
        NEG (a training set of instances NOT belonging to the class of interest)
Output:   R (a disjunction of non-disjunctive descriptions for the class)
Initialize R to False;
While POS is not empty
Begin
        Initialize C to False;
        For X set to each successive instance in POS
                Set L to the least generalization of C that covers X;
                If L does not cover any instances in NEG set C to L;
        Remove from POS all instances covered by C;
        Set R to R v C;
        End.
```

For example, with DLG and given the following training instances:

X	C	Class
3	red	positive
5	yellow	positive
1	brown	positive
1	red	negative
4	brown	negative
6	blue	negative

The class descriptions for each class (expressed in rules) that might be induced are:

IF ((3<=X<=5) & (C∈ {red, yellow})) V ((X = l) & (C∈ {brown})) THEN class = positive
IF ((X=l) & (C∈ {red})) V ((4<=X<=6) & (C∈ {brown, blue}))THEN class = negative

Thus, given the above induced rules and a novel object with attribute values of, for example, X = 4, and C = red, one may classify that object as of class "positive".
To complete the induction process, DLG performs "Conservative Conjunct Deletion" and "Range Generalization". Conservative conjunct deletion is achieved by two scans through the clauses of each rule in opposite directions. For each clause, delete it and then see if the rule covers any cases in other classes. If it does, restore it. Start each scan from the full rule. Finally, delete only those conjuncts that were deleted in both scans. Range generalization (Webb, 1991b) is a step to further generalize the constant range of attributes. Consider the rules:

IF (5<=(y - x) <=10) & (6<=w <=10) THEN class = positive.
IF (-20<=(y – x) <= -5) & (-l<=w<=3) THEN class = negative

Suppose we want to range extend 5 <=(y-x) <=10. Range generalization first deletes this clause and then examines all negative instances mis-classified by this description. Suppose it mis-classifies three negative instances with values of (y - x) equal to -20, -10 and -5. Range generalization finds, among the above values, the maximum value that is below the lower bound of the function to be extended, and the minimum value that is above the upper bound. The values are -5 and none respectively. Thus, for the lower bound of the function, we know that we can extend it to somewhere between 5 and -5 (in this research, we take the mid-point). As to the upper bound, we can generalize that direction to infinity. Using this process, and depending on the examples in the training set, the above rule may be generalized to:

IF ((y – x > 0) & (6<=w<=10) THEN class = positive
IF ((y – x < 0) & (-1<=w<=3) THEN class = negative

Since this range generalization step is quite radical, its execution can be made dependent on the application domain and other user-defined criteria.

3 Discriminant attribute finding algorithm (DAFA):

DAFA, is an algorithm which finds, from among a set of candidate attributes, those that can best discriminate the positive class froth the negative. If a single attribute can successfully distinguish all cases, it is returned; otherwise, the set of all attributes that exceed a pre-defined discriminant performance criteria is returned. It can be expressed as follows:

```
Input:          POS (a set of instances belonging to the class of interest)
                NEG (a set of instances NOT belonging to the class of interest)
                A set of candidate attributes based on dimension analysis and/or domain expert input;
Output:         S (a set of discriminant attribute(s) or No discriminant attribute found)
Initialize a variable DPT (discriminant percentage threshold) to certain user defined value;
Initialize a boolean variable FOUND to False;
Initialize all candidate attributes to unchecked;
While not FOUND and not all candidate attributes checked
Begin
        Take the next unchecked candidate attribute (A);
        Derive a generalized range for the attribute from the POS instances;
        Evaluate the percentage of NEG instances (PN) not covered by the range of the attribute;
        If PN = 100, set S = {A}, FOUND = True
        elseif PN >= DPT, include A in S;
End.
Eliminate redundant discriminant attributes.
```

Consider the following:

X(kg)	Y(kg)	Class
4	20	positive
9	10	positive
4	10	negative
8	6	negative

Suppose we consider the operator set {-, +, *, /}. Without input from domain experts, a possible set of candidate attributes, with compatible dimension, for example, is: {(X-Y), (X+Y), (X/Y)}. Assume that we set DPT = 80, i.e. the algorithm may accept an attribute as a discriminant attribute if its value range, based on POS instances, does not cover at least 80% of the NEG instances. On examining the first candidate attribute, the algorithm derives the following from positive instances: $-16 <= (X- Y) <= -l$. Since this range covers one out of two negative instances, (i.e. discriminant percentage = 50), the algorithm ignores it. On examining the next possible attribute, the derived range for positive instances is: $19 <= (X+Y) <= 24$. Since it does not cover any NEG instances, (i.e. discriminant percentage=100), it is accepted as a new attribute capable of discriminating the two classes and the algorithm terminates.
If the discriminant percentage of the second candidate attribute were below 100 but above the DPT, it would be recorded as a potential candidate, and the algorithm moves on to examine the discriminant percentage of the next variable. Attributes with discriminant percentage above the threshold, DPT, are recorded as discriminant attributes. For discriminant attributes involving the same component attributes, e.g. {(X-Y), (X/Y)}, only the one with the highest discriminant percentage is retained. If an attribute with discriminant percentage of 100 is found, only that attribute is retained and the rest are discarded, since it alone, is sufficient to discriminate the two classes under focus.

4 Discriminant attributes in classification learning:

Though DAFA works for simple data, it requires further sophistication.
Consider the following data:

Siren	Speed	Speed limit	Class
No	71	70	speeding
No	80	60	speeding
No	110	100	speeding
No	80	110	not_speeding
No	60	70	not_speeding
No	100	100	not_speeding
Yes	110	100	not_speeding
Yes	105	100	not_speeding
Yes	90	110	not_speeding

With DPT = 80, DAFA cannot find any simple discriminant attribute. If the algorithm is applied only to the data for which Siren = “No”, the discriminant attribute: (Speed - Speed_limit) is found. To this end, if discriminant attributes are not found for a data set, it is useful to apply DAFA to subsets of that data partitioned on a less generalized non-numeric condition. Including the discriminant attribute: (Speed - Speed_limit) in the data set and applying DLG on the data, the class descriptions induced is:

IF (Siren $\in$ {No}) & (1<= (Speed - Speed_limit)<=20) THEN class = speeding
IF (-30<= (Speed -Speed_limit)<= 0) V (Siren $\in$ {Yes}) THEN class = not_speeding

Depending on the data, range generalization can further refine the rules to:

IF (Siren $\in$ {No}) & ((Speed - Speed_limit) >0) THEN class = speeding.
IF ((Speed - Speed_limit) <= 0) V (Siren $\in$ {Yes}) THEN class = not_speeding.

Thus, by extending the domain model to include the new variable: (Speed - Speed_limit), the result is much more general in that it is able to correctly classify any new case even if it involves a Speed and Speed_limit that did not appear in the training set.

Putting together the ideas we discussed so far, the resulting algorithm, called DLG $_{daf}$ (discriminant attribute finding in disjunctive least generalization) can be expressed as follows:

```
Input:          POS (a training set of instances belonging to the class of interest)
                NEG (a training set of instances NOT belonging to the class of interest)
Output:            R (a disjunction of non-disjunctive descriptions for the class)
        For classes with no discriminating attribute(s)
                While discriminant attribute(s) not found and
                there is a less generalized non-numeric condition, c, which has not been examined:
                        discriminant attribute(s) <-- DAFA (select (POS, c), select (NEG, c));
        Extend the descriptions of cases in POS & NEG to include discriminant attribute(s)
                as additional attribute(s);
        rules <-- DLG(POS, NEG);
        Generalize rules using Conservative Conjunct Deletion;
        Generalize rules using function range generalization (if applicable).
```

Consider the following data:

X(m)	Y(kg)	Z(kg)	Class
1	5	10	A
2	7	8	A
3	5	8	B
4	7	4	B
5	8	10	C
6	7	8	C

On examining the data, the algorithm finds that based on the range of value of X, class A can discriminate itself from the other classes. Next, it notices that class B cannot be discriminated from class C by any the existing attributes. With DAFA, and the possible dimensionally compatible discriminant attribute candidate set of :{(Y - Z), (Y + Z), (Y/Z), (X/Y), (X/Z), (X*Z),(X*Z) }, the discriminant attribute of (Y + Z) is found. The initial rules derived by DLG are:

IF (1 <= X <= 2) & (5 <= Y <= 7) & (8 <= Z <= l0) & ((Y+Z) = 15) THEN class = A
IF (4 <= X <= 5) & (5 <= Y <= 7) & (4 <= Z <= 8) & (11 <= (Y + Z) <= 13) THEN class = B
IF (4 <= X <= 6) & (7 <= Y <= 8) & (8 <= Z <= 10) & (15 <= (Y + Z) <= 18) THEN class = C

After conservative conjunct deletion the step, the rules are:

IF (1 <= X <= 2) THEN class = A; IF (11 <= (Y+Z) <= 13) THEN class = B
IF (4 <= X <= 6) & (l5 <= (Y+Z) <= 18) THEN class = C

We can use the above rules to classify novel instances. To classify uncovered cases, we performed the highest percentage matching (HPM) step which can be expressed as follows:

```
For each uncovered case
        Retrieve all the clauses deleted by the Conservative Conjunct Deletion;
        Assign the case to the class with the highest percentage of matching clauses.
```

Suppose we have two novel instances: **(1)** X = 3, Y = 4, Z = 6 **(2)** X = 3, Y = 5, Z = 10. With the rules after conjunct deletion step, both cases are uncovered by the rules. With the rules before the conjunct deletion step retrieved, we can classify the uncovered cases based on the highest percentage of matching clauses. For case (1), class B has one matching clause: (4 <= Z <= 8), whereas the other two classes have no matching clauses; thus we classify case (1) as class = B. For case (2), class A, has 3 out of 4 (or 75%) matching clauses ; whereas, B and C have only 1 and 2 matching clauses respectively, thus, we classify case (2) as of class A. If there are more than 1 class with highest clause-matching percentage, the case remains as uncovered.

5 Evaluation:

This study uses the much researched Fisher's (1936) data on classifying a set of 150 iris flowers. Each example consists of four integer-valued variables: sepal length (sl), sepal width (sw), petal length (pl) and petal width (pw) in centimetres. There are 3 classes of species: Iris setosa, Iris versicolor, and Iris virginica. To enable comparison with other learning algorithms, this study used the "leaving-one-out" cross-validation method which uses one example as the testing data and the remaining examples for training. The method was repeated for every example in the data set. Thus, for this data set, there were 150 runs. On each run, 149 examples were used as training and one example was tested. The accuracy of the 150 tests was recorded.

With DLG $_{daf}$, initially, it was found that, based on existing attributes, Iris setosa can be distinguished from other classes. Examples of the other two classes: Iris versicolor and Iris virginica were then examined with DAFA. In this study, we examined only discriminant attributes involving two existing attributes. Since the data are all of same unit (cm.), we were contented with the operator set of {+, -, /). With DPT = 80, the discriminant attributes found by DAFA are: (pl + pw), (sw - pl) and (sw/pw). These discriminant attributes were then added to the data set and the expanded data set were then learned by DLG. When compared with the result based on the original attributes only, DLG $_{daf}$, though only slightly improves the accuracy, significantly reduces the total number of disjunctive clauses (D) ($t = 129.49$, $p \leq .0005$) and non-disjunctive clauses (ND) ($t = 96.02$, $p \leq .0005$) in the class descriptions.

Algorithm	Accuracy (%)	Mean D	Mean ND
DLG	94.2	8.95	24.85
DLG $_{daf(DPT=80)}$	95.3	6.00	18.91

DLG $_{daf}$ can be compared with other learning techniques using the same cross-validation method:

Algorithm	Accuracy (%)
DLG $_{daf(DPT=80)}$	95.3
EACH	95.3
CART	93.0
PVM rule	96.0
Neural net	96.7

EACH (Salzberg, 1991) is a program based on nested generalized exemplar theory. CART comes from a study by Crawford (1989). The PVM rule and neural net results are from the study by Weiss and Kapouleas (1989). In the above comparison, though DLG $_{daf}$ offers no improvement on the accuracy rate, it is a simple method to achieve comparable accuracy.

6 Conclusion:

In this paper, we have presented an algorithm (DLG $_{daf}$) based on deriving and then including discriminant attributes in classification learning. Most existing classification learning algorithms derive classification procedures based on values of given attributes. Intuitively, deriving new attributes from existing ones and incorporating in the data set those new attributes which enhance class distinctiveness, can improve classification procedures derived by data-driven methods, by reducing the number of disjunctive sets in the descriptions, classifying cases which would otherwise be uncovered and improving the accuracy of classifying novel instances.

Evaluation of DLG $_{daf}$ showed that the objectives are met for data of different classes with overlapping ranges in their attribute values. Given such data, the individual attributes cannot be used effectively as the basis to derive classification procedures. New variables characteristic of each class, if any, will then be crucial in deriving classification procedures. Thus, extension of domain model by finding discriminant attributes in classification learning can improve upon existing attribute-value classification learning methods.

References:

Crawford. S. (1989) **Extensions to the CART algorithm**. *The International Journal of Man Machine Studies*. 31, 197-217.

Elio, R & Watanabe, L. (1991) **An incremental deductive strategy for controlling constructive induction in learning from examples**. *Machine Learning.* 7, 7-44.

Fisher, R.A. (1936) **The use of multiple measurements in taxonomic problems**. *Annals of Eugenics,* 7, 179-188.

Michalski, R.S. (1980) **Pattern recognition as rule-guided inductive inference**. *IEEE Transactions on Pattern Recognition and Machine Intelligence,* 2, 349-361.

Michalski, R.S. (1984) **A theory and methodology of inductive learning**. In R.S. Michals ki, J.G. Carbonell & T.M.Mitchell (eds.) *Machine Learning: An Artificial Intelligence Approach* Springer-Verlag, Berlin, 83-129.

Plotkin, G.D. (1970) **A note on inductive generalization**. In B. Meltzer & D. Mitchie (eds.) *Machine Intelligence* 5, Edinburgh University Press, Edinburgh, 153-163.

Plotkin, G.D. (1971) **A further note on inductive generalization**. In B.Meltzer & D. Mitchie (eds.) *Machine Intelligence* 6, Edinburgh University Press, Edinburgh, 101-124.

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

Salzberg, S. (1991) **A nearest hyper rectangle learning method**. *Machine Learning* 6: 251-276

Webb, G.I. (1991a) **Learning disjunctive characteristic descriptions by least generalization**, *Technical report 2/91*, Deakin University, Geelong

Webb, G.I. (1991b) **Einstein: An interactive inductive knowledge acquisition tool**. *Proceedings of the 1991 Knowledge Acquisition for Knowledge-Based System Workshop,.Banff.*

Weiss, S. & Kapouleas, I. (1989) **An empirical comparison of pattern recognition, neural nets and machine learning classification methods**, *Proceedings of IJCAI-89* Detroit, MI, Morgan Kaufmann 781-787.

Yip, S.P. & Webb, G.I. (1992) **Function finding in classification learning**. *Proceedings of the 2nd Pacific Rim International Conference on Artificial Intelligence.* Seoul.