Pre-publication draft of a paper which appeared in the Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence (AI'94), pages 29-36. Singapore: World Scientific

Empirical function attribute construction in classification learning

Simon P. Yip
School of Computer Science & Software Engineering, Swinburne University of Tech., Hawthorn, Vic., 3122, Australia
and
Geoffrey I. Webb
School of Computing and Mathematics, Deakin University, Geelong, Vic., 3217, Australia

ABSTRACT

The merits of incorporating feature construction to assist selective induction in learning hard concepts are well documented. This paper introduces the notion of function attributes and reports a method of incorporating functional regularities in classifiers. Training sets are pre-processed with this method before submission to a selective induction classification learning system. The method, referred to as FAFA (function attribute finding), is characterised by finding bivariate functions that contribute to the discrimination between classes and then transforming them to function attributes as additional attributes of the data set. The value of each function attribute equals the deviation of each example from the value obtained by applying that function to the example. The expanded data set is then submitted to classification learning. Evaluation with published and artificial data shows that this method can improve classifiers in terms of predictive accuracy and complexity.

1. Introduction:

Selective-induction or attribute-based classification learning techniques perform poorly when the attributes are inappropriate for the target classifiers. One solution is to have the learning system construct features (higher level attributes) from existing attributes automatically. Past research includes Rendell & Seshu [1990], Pagallo & Haussler [1990], Wnek & Michalski [1994] and Yip & Webb [1992a, 1992b, 1994]. Unlike most previous research on constructive induction, our techniques are designed for use in preprocessing training data for subsequent use by any standard selective induction system.

2. Towards a function attribute construction algorithm:

2.1 Theoretical perspectives:

This paper describes a technique which differs from previous approaches to constructive induction by seeking to identify functional regularities between attributes in training examples and incorporate them as more concise components of classifiers. Suppose we have an instance space where X and Y are the attributes of examples classified as either positive '+' or negative '-'.

Suppose the positive instances fit well on a linear function: $Y=aX+b$. The classifiers one would like are:

If $(Y=aX+b)$ *then class = positive;* *If* $(Y\neq aX+b)$ *then class = negative.*

In real situations, we rarely have such perfect fit. A more realistic situation is in Figure 1. The problem is to find a classifier to describe the decision surfaces as indicated by the dotted lines.

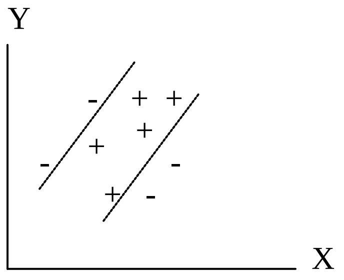

Figure 1: An instance space with two attributes

In finding functional regularities for incorporation in classification learning, the following problem issues arise: (1) We need methods which have the flexibility to find and represent specific functions able to exclude negative instances, as well as functional descriptions general enough to cover positive instances which do not closely fit a function. (2) The next issue is the familiar search problem. The number of candidate functions increases rapidly as the number of attributes and function types increases. (3) Another issue is finding functional regularities within subsets of the data. In building decision trees, functional regularities can be searched when each node is created. However, this will further exacerbate the search problem.

Among machine learning function finding literature, BACON [Langley, Simon, Bradshaw & Zytkow, 1987], is a data driven system to discover function regularities. While such a method can tackle the model-driven large search space problem, it can only find specific functional regularities which are often not observed in real data. Methods to incorporate BACON in classification learning [Yip and Webb, 1992a] performed well on contrived artificial data sets but remained unfruitful on real data sets.

2.2 Converting a function to a function attribute:

Consider Figure 1 again. Suppose we fit a linear function to the positive examples as illustrated in Figure 2.

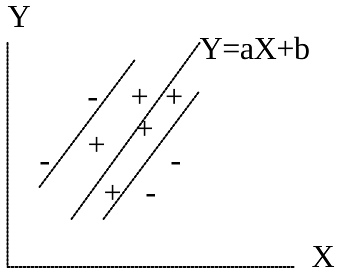

Figure 2: An instance space to illustrate function attribute

We can convert the linear function into an attribute D with values equal to the deviation of each example i (with attributes: X_i Y_i), from the linear function, i.e.

$D=(Y_i - Y_i')$ where $Y_i' = aX_i+b$. Let D_L and D_U be the lower and upper bound of the deviation of positive examples from the linear function. Then, we can capture the functional regularities with the following class descriptions:
If $(D_L \leq D \leq D_U)$ *then class=positive; If* $((D>D_U) \vee (D<D_L))$ *then class=negative*
In this way, we can encapsulate the information described by a function in an attribute, referred to as a 'function attribute'. While simple linear functions can be represented by discriminate attributes searched by other methods [Yip & Webb, 1992b, 1994], the notion of function attributes is distinguished by its ability to represent complex nonlinear functions. Thus, the merits of function attributes are that they can represent complex functions and they have the flexibility to encapsulate both specific and less specific functional regularities.

2.3 Incorporating bivariate functions as attributes in classification learning:

In this research, we focus on finding bivariate functions and representing them as function attributes. The advantages of bivariate functions are that they are simple and there is no implication of causal or predictive relations. In this research, we consider the function set: $\{Y=aX+b;\ Y=aX^n;\ Y=aX^n+b\}$. The function attribute finding process starts by examining classes which cannot be distinguished from other classes by existing attributes. The set of bivariate functions are fitted on pairwise combinations of existing numeric attributes. Each of the functions so derived are transformed into a function attribute, the values of which are the deviation of each example from that function. Those function attributes with discriminant performance, as determined by an evaluation function, are selected as additional attributes to the original attribute set. The expanded data set is then subjected to a classification learning system.

3. Function attribute finding algorithm (FAFA):

FAFA, is an algorithm, which finds, among a set of bivariate function candidates, those that can contribute to the distinguishability of the positive class from the negative. First it generates a set of bivariate candidate functions by substituting pairwise combinations of existing attributes into a set of pre-defined bivariate functions. Then, the algorithm checks if any of the existing attributes can discriminate the examples of the class of interest *(POS)* from other classes *(NEG)*. If there is none, the algorithm moves on. A function is selected from the candidate function set and then fitted on *POS*, using the least square criterion, with normal equations for linear functions or progressive hill climbing [Yip and Webb, 1993] for nonlinear functions. The associated R^2, a measure of the fitness[1] is noted. The candidate bivariate function is then transformed into a 'function attribute'. The discriminant performance of the function attribute is determined by an evaluation function. In this research, the evaluation function is as follows: First the cross validation discriminant performance of the target machine learning system is determined. If a function attribute's range can discriminate at a higher level, the new attribute is accepted. If the discriminant performance of the function attribute is perfect and the associated R^2 passes a

[1]R^2 ranges from 0 to 1. Suppose we have N pairs of observations of X and Y. Let *f* be the fitted function and let $Y'=f(X)$: $R^2 = 1-(\Sigma(Y_i-Y_i')^2/\Sigma Y_i^2)$.

predefined fitness value, the attribute is returned, displacing all attributes selected so far and the algorithm stops. The FAFA algorithm 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)
Output: S: a set of function attribute(s) derived from functions selected from the function
       set (FS) or return "No function attribute found" or "No function attribute necessary".
Begin
  FS $\leftarrow$ {$Y=aX+b$; $Y=aX^n$; $Y=aX^n+b$)}
       where X and Y are numeric attributes; a, b, and n are constants;
  PO $\leftarrow$ a set of pairwise combination of original attributes;
  FOUND $\leftarrow$ False;
  Initialise parameter: mf (maximum fitness) (e.g. mf=0.99);
  If POS can be distinguished from NEG by any of the original attributes on its own,
    set FOUND $\leftarrow$ True, return "No function attribute necessary";
  else
    Generate a set of bivariate functions: F $\leftarrow$ $\{Y_i=f_i(X_i): f_i \in FS; (X_i,Y_i) \in PO\}$;
    Initialise all functions of F to unchecked;
    While not FOUND and not all functions checked
    Begin
       Take the next unchecked function FA from F;
       Fit the function FA on POS; note the $R^2$ and derive function attribute (A);
       Derive the range of A (RA) and evaluate its discriminant performance (PN) which
           is the percentage of NEG instances with deviation not within RA;
       If PN = 100 and $R^2$>=mf, set S $\leftarrow$ {A}, FOUND $\leftarrow$ True
       else if PN > Performance criterion (PC), include A in S;
     End {while};
End.
```

Further, to find functional regularities within in subsets, the algorithm searches for functional regularities in subsets partitioned by discrete attributes (if any). The combination of FAFA with selective induction can be expressed as follows:

```
Input: a training set of instances (T)
Output: classifiers
Begin
      function attribute(s) $\leftarrow$ FAFA(T);
      ET(Extended training set) $\leftarrow$ Extend descriptions of instances to include
          function attribute(s) as additional attribute(s);
      classifiers $\leftarrow$ Selective induction classification learning (ET)
End.
```

4. Evaluation:

Two target selective induction classification learning systems are used. C4.5 [Quinlan, 1993] is a decision tree based system. Einstein [Webb, 1992], a variant of Aq [Michalski, 1983] is a decision rule based system. Extensive evaluation has been performed [Yip, 1994]. Two representative studies are presented below. In the following tabulation of results, FAFA+C4.5, for example, represents the method of treating the data set with FAFA before submission to C4.5; "Accuracy" refers to predictive accuracy on unseen instances and "Complexity" refers to the number of nodes of the decision tree or the number of rules of the classifier. A pair-wise t-test is used for comparison.

4.1 Study 1:

In this study, the New-thyroid data from UCI [Murphy & Aha, 1994] with 215 instances is used. With ten-fold cross validation, the predictive accuracy and complexity of classifiers averaged over 10 runs can be presented as follows:

Method	Accuracy(%)	Complexity (nodes/rules)
(1) C4.5 (pruned)	92	16
C4.5 (rules)	92.47	7.1
(2) FAFA+C4.5 (pruned)	94.38	9.8
(compared with (1))	(t=0.89)	(t=8.19; p≤.0005)
FAFA+C4.5 (rules)	94.38	5.2
(compared with (1))	(t=0.74)	(t=6.86; p≤.0005)
(3) Einstein	91.91	7
(4) FAFA+Einstein	93.34	6.1
(compared with (3))	(t=1.96, p≤.05)	(t=3.25; p≤.005)

In the above tabulation, we observe that by applying FAFA, the complexity of induced classifiers is significantly reduced and predictive accuracy of Einstein significantly increased. The effect of FAFA can be further examined by plotting the performance versus training size graph. In this study, 20% of the data set is used as the evaluation set and the training set consists, in turns, of 40%, 60% or 80% of the data set. The performance of the induced trees or rules for each training set size is evaluated over 10 runs. The accuracy performance and complexity comparison can be illustrated in the following graphs:

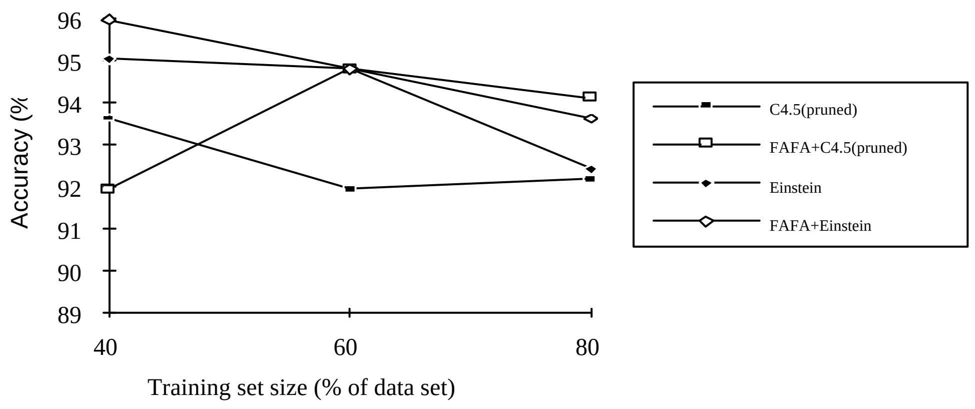

Figure 3: FAFA+C4.5(pruned) vs. C4.5(pruned) and FAFA+Einstein vs. Einstein
Accuracy-Training_size-plot on New-thyroid data set

In Figure 3, we observe that with FAFA, the predictive accuracy of C4.5(pruned) is significantly increased at 60% but decreased at 40% ($t_{40\%}$=-1.97, p≤.05; $t_{60\%}$=3.05, p≤.01; $t_{80\%}$=1.21) where "$t_{40\%}$", for example, represents the t-value at training set size of 40%. The predictive accuracy of Einstein is significantly increased at 40% and 80% ($t_{40\%}$=2.45, p≤.025; $t_{60\%}$=0; $t_{80\%}$=3, p≤.01).

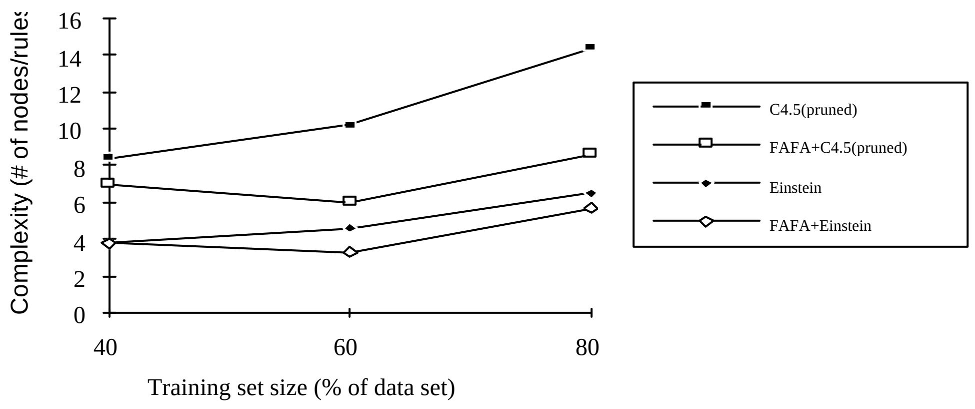

Figure 4: FAFA+C4.5(pruned) vs. C4.5(pruned) and FAFA+Einstein vs. Einstein
Complexity-Training_size-plot on New-thyroid data

In Figure 4, we observe that with FAFA, the complexity of C4.5(pruned) is significantly decreased at all three different training set size ($t_{40\%}$=2.69, p≤.025; $t_{60\%}$=5.16, p≤.0005; $t_{80\%}$=6.33, p≤.0005). The complexity of Einstein rules is significantly decreased at 60% and 80% ($t_{40\%}$=0; $t_{60\%}$=1.96, p≤.05; $t_{80\%}$=2.75, p≤.025). This study shows that for this data, in general, with FAFA as the pre-classification learning step, at 80% of the data as training set, the predictive accuracy of classifiers is increased and complexity significantly reduced. At 60%, C4.5 responds positively, while Einstein shows insignificant changes. At 40%, there are significant changes in predictive accuracy, with Einstein responding positively but C4.5 negatively. The result of C4.5 at 40% suggests when the training set size is 'small', where 'small' is data set specific, FAFA should be applied with care (e.g. using a more stringent function attribute selection criterion).

4.2 Study 2:

To further illustrate the merit of FAFA, in this study, an artificial data set is generated. The data set is generated with the following specification. There are three classes: Pos, Norm and Neg; each instance is described by two continuous attributes: X and Y. Let **X** = {random numbers between 1 to 100}. For class=Pos: $X\in\mathbf{X}$, $Y=2X^{1.5}+3000$; for class=Norm: $X\in\mathbf{X}$, $Y=3X^{1.5}+2000$ and for class=Neg: let **Y**={random numbers between lower and Upper bound of Y of Pos and Norm}, $X\in\mathbf{X}$, $Y\in\mathbf{Y}$. The data set consists of 200 instances of each class. The results of ten-fold cross-validation can be presented as follows:

Method	Accuracy(%)	Complexity (nodes/rules)
(1) C4.5 (pruned)	84.33	92.6
C4.5 (rules)	86.33	35.5
(2) FAFA+C4.5		
(pruned)	99.33	9
(compared with (1))	($t=9.58$, $p\leq.0005$)	($t=25.41$, $p\leq.0005$)
(rules)	99.5	5
(compared with (1))	($t=8.55$, $p\leq.0005$)	($t=27.78$, $p\leq.0005$)
(3) Einstein	88.67	65.3
(4) FAFA+Einstein	100	5
(compared with (3))	($t=8.22$, $p\leq.0005$)	($t=134.46$, $p\leq.0005$)

In the above tabulation, we observe that by applying FAFA, the predictive accuracy of induced decision trees or rules are significantly increased and complexity significantly reduced.

5. Discussion:

'Function attribute' is introduced in this paper as a notion to represent a functional relationship between two or more attributes by a single attribute. In this research, only three types of bivariate relations are examined. The techniques are in no way restricted to these relation types. The notion of function attribute can be extended to more complex multivariate functional relations. Function attributes derived by FAFA summarise data regularities that characterise different classes of objects. With *f* bivariate function types, *n* training instances and *p* attributes, the number of bivariate candidate functions is $f.C_2^p$. The algorithm stops when a near perfect function is found. Hence, the time complexity of FAFA, in the worst case, is of the order $O(n.p^2.f)$.

Thus, one limitation of FAFA is that the computational time complexity increases in the order to the square of the number of attributes. Second, FAFA is limited to finding functional relations that characterise all instances of one class as different other classes. Though functional relations that characterise subsets of instances as partitioned by discrete attribute are also searched, other partitions are not addressed. Nevertheless, the counter argument is that functional relations based on small numbers of instances may be misleading.

6. Conclusion:

In this paper, we introduce the notion of function attributes in classification learning and illustrate it by incorporating bivariate functional regularities. Functional regularities are searched and if discovered, are converted into additional attributes. The expanded data set is then subjected to classification learning. Evaluation showed that the technique can significantly improve classifier performance when compared to classification learning alone. The technique is most suited to data in which functional regularities exist that are relevant to the classification task. Unfortunately, we, as yet, do not have techniques for identifying in advance such data sets. That FAFA is useful for such data is demonstrated by our first study. The success in the second study suggests that such data exists in real world applications.

7. Appendix:

For the New thyroid data set, (with the attributes referred to as var*n* where *n* refers to the *n*th attribute as in the published data sets), based on 80% of data as training set, examples of bivariate function found are: $var5=0.88*var2^{1.5}+9.45$; $var5=0.0069*var1-0.66$; $var5=-0.0159*var4^{1.5}$; $var5=-0.0042*var2+0.058$; $var5=-0.0087*var3+0.02$.

8. References:

Langley, P., Simon, H.A., Bradshaw, G.L & Zytkow, J.M. (1987) *Scientific Discovery: Computational Exploration of the Creative Process*. MIT press: Cambridge.

Michalski, R.S. (1983) A theory and methodology of inductive learning. In Michalski, R.S., Carbonnel, J,G. & Mitchell, T.M. (Ed.) *Machine Learning: An Artificial Intelligence Approach,* Los Altos, CA: Morgan Kaufmann.

Murphy, P.M. & Aha, D. (1994) UCI repository of machine learning databases. Irvine, CA: University of California, Dept. of information and Computer Science

Pagallo, G. & Haussler, D. (1990) Boolean feature discovery in empirical learning. *Machine Learning*, 5.

Quinlan, R. (1993) *C4.5 Programs for Machine Learning.* Morgan Kaufmann.

Rendell, L. & Seshu, R. (1990) Learning hard concepts through constructive induction: framework and rationale. *Computational Intelligence* 6: p.247-270

Webb, G. I. (1992) Man-machine collaboration for knowledge acquisition. In *Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence.* World Scientific, p.329-334.

Wnek & Michalski, R.S. (1994) Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning*, 14:2, p.139-168.

Yip, S.P. (1994) Empirical attribute space refinement in classification learning. Ph.D thesis. Deakin University. School of Computing and Mathematics.

Geelong, Vic., 3217, Australia. (submitted)

Yip, S.P. & Webb, G.I. (1992a) Function finding in classification learning. In *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence.* KAIST: Seoul, p.555-559.

Yip, S.P. & Webb, G.I. (1992b) Discriminant attribute finding in classification learning. In *Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence.* World scientific, p.374-379.

Yip, S.P. & Webb, G.I. (1993) Nested search and progressive hill climbing in complex spaces. Technical report , SUT-CS-21/93, Dept. Computer Science, Swinburne University of Technology.

Yip, S.P. & Webb, G.I. (1994) Incorporating canonical discriminant attributes in classification learning. In *Proceedings of the Tenth Canadian Conference on Artificial Intelligence.* Morgan Kaufmann, p.63-70.