Pre-publication draft of paper accepted for publication in the Proceedings of the Eighth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE '95), pages 587-592. Gordon and Breach Science Publishers

Polygonal Inductive Generalisation System

D.A.Newlands and G.I.Webb
Deakin University
Geelong
Victoria 3217
Australia
doug@deakin.edu.au, webb@deakin.edu.au
Ph. (052) 811 757
Fax (052) 811 851

February 24, 2005

Abstract

Classification learning has been dominated by the induction of axis-orthogonal decision surfaces. While induction of alternate forms of decision surface has received some attention in the context of decision trees, this issue has received little attention in the context of decision rules. An inductive learning algorithm has been developed which creates arbitrarily shaped concepts. Results from a prototype implementation demonstrate that the approach performs well on target concepts that are not readily represented by long, flat decision surfaces.

Keywords : classification learning, oblique decision surfaces, non-axis-orthogonal decision surfaces, covering algorithms, hyper-polygonal decision regions.

1 Introduction

The complexity of the representation of an hypothesis produced by a given computational learning system is a function of both the hypothesis language and the target concept. If the concept being learned and the hypothesis language are geometrically similar, the concept will be concisely represented. Otherwise the representation of the concept will be by a large number of inappropriately shaped decision surfaces. The typical axis-orthogonal decision tree representation of the concept shown in Figure 1 is an example of the problem.

Systems which represent concepts in disjunctive-normal-form propositional calculus or similar form and which use the natural attributes of the learning situation (decision trees [4], covering algorithms [10, 21]), are implicitly using hyper-rectangles to represent the concept in the instance space. If the concepts to be learned have axis-parallel boundaries, the representation will be concise, using a small number of large decision surfaces. However, concepts with straight but non-axis-parallel or curved boundaries will only be approximately represented by a large number of small decision surfaces. Systems for the induction of concepts with oblique boundaries have been described including oblique decision trees [13] and synthesised attributes [24, 15, 17, 2, 3, 7, 8, 10, 14, 20, 23] but most have limitations when concept boundaries are not linear. Statistical methods of concept formation can be regarded as representing concepts by hyperspheres or similar shapes e.g.CLUSTER [9], UNIMEM [6] and COBWEB [5]. Instance-based learning, derived from nearest neighbour classifiers [1], does not maintain a generalisation of instances but stores instances and examines them to make classifications. Salzberg [18] describes Nested Generalized Exemplar theory which is currently implemented using hyper-rectangles but, in principle, can be implemented using other shaped decision regions.

All of the above methods avoid the n-dimensional geometry of the learning area: the hyper-rectangle based methods by being able to examine each axis separately since surfaces are orthogonal to axes and the statistical methods

Figure 1: A Complex Representation of a Simple Concept.

by imposing symmetry on the situation using Pythagorean measures on the instances. This article examines the use of arbitrarily-shaped n-dimensional solids to represent decision regions. The quality of the representation of hypotheses in this framework should not depend on how well the geometry of the actual concept fits the hypothesis language of the inductive system and thus one would expect good quality hypotheses and representations over a wider range of tasks than methods with an implicit reliance on a regular polygon structure.

2 Polygonal Inductive Generalisation System (PIGS)

The proposed system will construct hyper-polygons, each representing (part of) a concept, using a successive generalisation algorithm such as employed by GOLEM [11], DLG [22] and Multiple Convergence [12]. Each hyper-polygon will use instances as vertices, but no internal vertices or edges will be represented. New instances which are internal to an appropriate (i.e. belonging to the correct concept) hyper-polygon will be regarded as being covered. New instances which are not so covered will be attached to the nearest polygon of the correct concept by performing a minimal generalisation which does not lead to any negative instances being covered. If no generalisation is possible, a new polygon will be started.

Generalisation of a polygon and point will be effected by inserting edges between the new instance and all *visible* vertices which are one edge from the nearest point on the polygon to the new instance. The algorithm for generalisation of a hyper-polygon is shown below.

generalise-polygon(POLYGON p, POINT new-instance)
    FIND nearest vertex, V_0, of polygon p
    FIND all vertices, $V_1 \ldots V_n$, connected to V_0
    IF all $V_1 \ldots V_n$ are visible from new-instance
        REPLACE coordinates of V_0 with those of new-instance
    ELSE
        REPLACE invisible vertices with V_0
        CREATE new vertex, V_{n+1}, containing new-instance
        CONNECT new vertex, V_{n+1}, to $V_1, \ldots, V_n$
        DELETE connections from V_0 to $V_1, \ldots, V_n$
    ENDIF
END.

If there is a subsequent problem with covering of negative instances, the generalisation can be rolled back. This form of generalisation, when there are no invisible vertices, does not lead to any increase in storage size since no new vertex is created; only the location of a pre-existing one is altered.

In a continuous instance space, the measure of minimality of generalisation should be in terms of the volume of instance space enclosed since the representation is purely geometric and not restrained to be at particular angles to the axes. In methods where the representation is at a symbolic level, least generalisation, as discussed by Plotkin [16], in terms of the hypothesis language is reasonable but subsumes larger, rectangular volumes of instance space. The actual generalisation method is, therefore, considerably more conservative than hyper-rectangle based methods and would lead one to expect fewer false positives than with other generalisation techniques. Post-processing on the decision regions (hyper-polygons) should permit extraction of higher level hypotheses by fitting large regular shapes to the regions, using mathematical techniques to select among possible large shapes and using regularities in one area to complete other areas.

2.1 The Prototype

This initial implementation will be in 2-dimensions since all situations are readily visualisable in 2-dimensions. In particular then, a concept, in the prototype, will be a set of surfaces; each surface will be a set of lines and each line will be a pair of vertices. An instance of a concept is a point within one of the surfaces representing that concept and classifying an instance requires identification of which surface(s) it lies within.

The learning task is to construct concepts when presented with attribute vectors consisting of pairs of continuous numerical values. Non-numeric attributes and missing values are not considered here.

2.2 Cover and Generalisation

An instance will be **covered** by a concept if it lies within one of the surfaces of the concept. A simple approach to cover and a desire to minimise storage requirements require a little care in generalisation so that the polygon has no internal edges. Consider the left part of figure 2 where points a,b,c have already been generalised to form a surface, P is a new positive instance of the same concept and N is a negative instance. If the generalisation is done as at the right in figure 2, the internal lines xa and xc will cause the cover algorithm to malfunction. (Generalisation bpc is not permitted because it would cover the negative instance!) The algorithm for generalisation of a concept is

```
generalise(POINT new-point, CONCEPT concept)
    FOR all polygons in concept
        FIND nearest vertex to new-point
        PUT polygon, vertex and distance in possible-list
    ENDFOR
```

Figure 2: Faulty Generalisation

```
    ORDER possible-list on distance
    PUT an empty polygon last in possible-list to
        guarantee generalisability
    FOR each polygon-vertex-distance-tuple in the possible-list
        generalise-polygon (polygon, new-point)
        IF polygon now covers any negative point
            roll back generalisation
        ELSE
            RETURN from successful generalisation
        ENDIF
    ENDFOR
END.
```

2.3 Spiking

The reason for sorting the polygons before generalisation is to avoid the situation where an instance is generalised onto an inappropriate, distant region. Such effects almost always create very narrow 'spikes' as, to be viable, they must not cover any negative instances. However spiking has been observed to occur in two other situations. Early in the induction phase when there are few, or no, valid (i.e. having 3 or more points in the 2-d case being studied) regions, two points which are from different regions of the same concept may get joined and if a third point near one of these is processed then a spike from one region to another will result. This can be valid if no negative instance in the training set contraindicates this generalisation. It is undesirable as, when applying a classifier to classify previously unsighted objects, if an instance falls within the spike, it will be classified positive to both the concept represented by the spike and to the concept of the area through which the spike passes. This problem can be side-stepped by a constraint on the length of the new sides of the newly generalised area either relative to the pre-existing edges of the newly generalised area or in absolute terms. Clearly the optimal length limit is less than an actual concept width but the geometry of concepts is not available before the induction process so some heuristic scheme has to be used. Preprocessing the data, which will be seen later to have some attractions, would allow estimation of the average inter-instance distance and this could be used to form an absolute length limit.

The other reason for needing a length limit is in the case of the actual concept being quantised and there not being any negative instances e.g. two bands with a "forbidden" region between them. Now, if the data starts with 2 instances in one band and one in the other, generalisation would occur across the "forbidden" area between them. While it is true that the classifier could never get any instance wrong because none could occur in the "forbidden" area, there would be no possibility of extracting correct higher level formulations of the actual concept.

3 Evaluation

The Conservation Law of Generalisation Performance [19] states that no learning algorithm can, in general, obtain higher generalisation performance than any other. In this context, it is incumbent upon the researcher presenting a learning algorithm to identify the types of learning problems for which it might be expected to obtain high generalisation performance. By escaping the constraints of axis-orthogonal decision surfaces, PIGS should enjoy an advantage over systems restricted to axis-orthogonal decision surfaces when learning concepts that cannot be readily represented by such surfaces. With respect to oblique decision trees, the relatively short lines developed by PIGS should give it an advantage when the target concept cannot be well approximated by long, straight decision surfaces. PIGS, however, is unsuited to learning tasks where closeness in the instance space is not predictive of class. With this in mind, it is expected that PIGS will perform well on a wide variety of concept shapes since there is no bias towards a particular geometry.

To evaluate these assumptions, comparisons between PIGS, OC1 (oblique decision trees [13]) and C4.5 (axis-orthogonal decision trees [15]) were performed on a range of artificial data sets ranging from "squares" where one would expect a decision tree to perform best since it will automatically produce straight edges of the correct orientation, to "POL"[13] (parallel oblique lines) where an axis orthogonal decision tree would do less well than, say, an oblique decision tree [13], to various curved concepts where all decision trees should do less well and PIGS should be superior. The test concepts shown in figure 3 were used for experimentation. Each dimension is in the range [0..16]. Training sets consisted of 1600 randomly generated points and test sets of 400 not drawn from the training set. Fifty training and test set pairs were generated for each of the test concepts (except POL, only 30) and presented to PIGS, OC1 and C4.5. When developing rules, the maximum edge length allowed was the default for PIGS, 6. When applying the rules developed by PIGS, if no rule applied to an instance, the instance was inferred to belong to the nearest concept using a simple nearest neighbour technique. In no case did two or more contradictory

Figure 3: Test Concepts

rules cover an instance. The accuracy of PIGS was compared to OC1 and C4.5 using a 1-tailed, matched-pairs t-test. The results are shown in table 1 and table 2.

Table 1.

Concept	PIGS		OC1		Statistics	
	Mean	St. Dev.	Mean	St. Dev.	t value	Probability
squares	98.90	1.107	99.02	0.848	0.8915	0.3770
quad	99.55	0.467	99.38	0.429	-3.0645	0.0035
circle	99.09	0.834	98.17	1.080	-8.0501	0.0000
discs	98.84	1.002	98.22	1.134	-4.9240	0.0000
polo	98.15	1.838	97.43	1.672	-3.5298	0.0009
POL	98.22	0.684	99.18	0.636	6.8737	0.0000

The mean and standard deviation of the accuracy of each system is shown together with the test statistic and the probability that the outcome is by chance.

Table 2.

Concept	PIGS		C4.5		Statistics	
	Mean	St. Dev.	Mean	St. Dev.	t value	Probability
squares	98.81	1.167	99.69	0.354	5.8082	0.0000
quad	99.52	0.414	98.82	0.758	-8.3494	0.0000
circle	99.07	0.813	98.31	1.193	-6.8948	0.0000
discs	98.84	1.002	98.16	1.093	-6.4078	0.0000
polo	98.59	1.245	97.27	1.883	-8.3143	0.0000
POL	98.22	0.684	94.40	1.063	-16.572	0.0000

It can be seen that PIGS provided significantly better performance than C4.5 and OC1 on all concepts with curved geometry. On the "squares" data set, C4.5 does very well as one would expect but OC1 (not set to prefer axis-orthogonal surfaces) does not do significantly better than PIGS. On the "POL" data set, OC1 does significantly better than PIGS, as expected, but C4.5 does significantly worse as it is badly biased for this type of concept where there are no axis-orthogonal components. While testing PIGS it was observed that:-

- the number of surfaces per concept was typically 2 to 5 and this variation seems to be a function of the order in which instances are seen.

- in more than 93% of runs, no instance was classified as belonging to two classes. This suggests that the simple, absolute value restriction on the size of new sides is reasonably successful in stopping spiking.

- a number of items, on average 6%, lay outside the decision regions produced by PIGS. Clearly, the edges of concepts are over-specialised and there will always be interstices between concepts. Consequent upon this, we note that it is errors in the nearest neighbour technique for unclassified points which produces the majority of the false positive and negative outcomes, *not* PIGS itself.

4 Conclusions

PIGS obtains significantly better predictive accuracy than C4.5 on every domain except "squares"where it was expected to be inferior. PIGS also obtains better predictive accuracy than OC1 on all curved concepts and is not significantly worse for "squares". OC1 does perform better on "POL" but this is not unexpected given the learning bias of OC1. The results give a clear practical demonstration of the consequences of conservation of generalisation performance. This performance from the prototype justifies continuing with future plans.

As well as extending PIGS to n-dimensions, other areas to be investigated include

- preprocessing the data with the objective of having clumped data at the front to enable seeding of good concepts in the induction process which should minimise spiking without any ad hoc constraint.

- preprocessing the data with the objective of having well separated points at the front to form large concepts early and minimise the amount of induction to be done by having more points covered early in the process.

- postprocessing the concepts to aggregate overlapping polygons to reduce the amount of computation in subsequent classification of instances.

- postprocessing the concepts to smooth their surfaces to reduce their over-specialisation and to minimise the interstices between concepts.

- postprocessing the concepts to construct higher level hypotheses from regularities in the polygons and their placement in instance space, e.g. having three equally spaced concepts, two of which are spherical and one of which is poorly represented, one might induce that the odd one should also be spherical; having four identically shaped, regularly spaced polygons, one might induce some kind of repetitive law.

PIGS has demonstrated the feasibility of induction of oblique decision surfaces within a covering algorithm. While this prototype implementation is restricted to two-attribute domains, the approach is, in principle, extensible to any number of dimensions. The excellent results obtained by this prototype demonstrate great potential.

References

[1] Aha,D.W. and Kibler,D. and Albert,M.K.: Instance-Based Learning Algorithms. Machine Learning **6** (1991) 37–66

[2] Bloedorn,E. and Michalski,R.S.: Data-Driven Constructive Induction in AQ17-PRE. Proceedings of the Third International Conference on Tools for AI (1991) San Jose, CA, 30–37

[3] Breiman,L. and Friedman,J.H. and Olsen,R.A. and Stone,C.J.: Classification and Regression Trees. Wadsworth statistic/probability series (1994).

[4] Cestnik,B. and Kononenko,I. and Bratko,I.: ASSISTANT 86: A Knowledge-Elicitation Tool for Sophisticated Users. Progress in Machine Learning;Proc.of EWSL 87 (1987)

[5] Fisher,D.: Knowledge Acquisition Via Incremental Conceptual Clustering. Machine Learning **2** 139-172

[6] Lebowitz,M.: Experiments with Incremental Concept Formation:UNIMEM. Machine Learning **2,2** 103–138

[7] Matheus,C. and Rendell,L.A.: Constructive Induction on Decision Trees. Proceedings of the International Joint Conference on Artificial Intelligence. (1989), 645–650

[8] Mehra,P. and Rendell,L.A. and Wah,B.W.: Principled Constructive Induction. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 651–656

[9] Michalski,R.S.: Learning from Observation:Conceptual Clustering. In Machine learning: an Artificail Intelligence Approach (1983) Springer Verlag

[10] Michalski,R.S. and Mozetic,I. and Hong,J. and Lavrac,N.: The Multipurpose Incremental Learning System AQ15 and its Testing and Application to three Medical Domains. Proceedings of the Fifth National Conference on Artificial Intelligence (1986), 1041–1045

[11] Muggleton,S. and Feng,S.: Efficient Induction of Logic Programs. Proc. First Conference on Algorithmic Learning Theory (1990), Tokyo, 1–14

[12] Murray,K.S.: Multiple Convergence: An Approach to Disjunctive Concept Acquisition. Proc. Tenth International Joint Conference on Artificial Intelligence (1987), Morgan Kaufman, Los Altos, 297–300

[13] Murthy,S.K. and Kasif,S. and Salzberg,S.: A System for Induction of Oblique Decision Trees. Jour. of Artificial Intelligence **2** (1994) 1–32

[14] Pagallo,G. and Haussler,D.: Boolean Feature Discovery in Empirical Learning. Machine Learning **5**

[15] Quinlan,J.R.: C4.5 Programs for Machine Learning, Morgan Kaufman (1993) San Mateo, California.

[16] Plotkin,G.D.: A Note on Inductive Generalisation. In B.Melzer & D.Michie (Eds.), Machine Intelligence **5** (1970) Edinburgh University Press, Edinburgh, 153–163

[17] Rendell,L.A. and Seshu,R.: Learning Hard Concepts through Constructive Induction:Framework and Rationale (1990) Computational Intelligence

[18] Salzberg,S.: A Nearest Hyperrectangle Learning Method. Machine Learning **6** 251–276

[19] Schaffer,C.: A Conservation Law for Generalisation Performance. Machine Learning Conference (1994)

[20] Utgoff,P.E. and Brodley,C.E.: Linear Machine Decision Trees:COINS. Tech. Report 91-10, Univ. of Massachussetts, Amherst,MA.

[21] Webb,G.I.: Man-Machine Collaboration for Knowledge Acquisition. Proc. Fifth Australian Joint Conference on Artificial Intelligence (1992) World Scientific, 329–334

[22] Webb,G.I.: Learning Disjunctive Class Descriptions by Least Generalisation, Tech. Report C92/9, Deakin University, Geelong, Victoria 3217, Australia

[23] Wnek,J. and Michalski,R.S.: Hypothesis-Driven Constructive Induction in AQ17-HCI. Machine Learning **14**,2 , 139–168

[24] Yip,S.P. and Webb,G.I.: Empirical Function Attribute Construction in Classification Learning. AI (1994) (to be published)