Use of Machine Learning to Generate Rules

D Hutber and P F Sims
Sowerby Research Centre
British Aerospace plc
FPC 267 P.O. Box 5
Filton
Bristol BS12 7QW

Abstract

The use of Machine Learning techniques applied to visual data is described, within the context of an Alvey exemplar to detect cars in outdoor scenes. A Similarity–Based Learning scheme is employed, that uses segmented images containing cars, to produce rules which are subsequently able to label unknown images. The method is shown to be useful in developing a suitable Knowledge Representation for this vision problem.

1 Introduction

Machine Learning is a technique for automating either the acquisition of knowledge or the improvement of performance of a system. It has been used in a variety of problem domains within Artificial Intelligence, including plant disease diagnosis, checkers playing, and logical circuit design. However, its use within computer vision has been limited to date [1], primarily because of the large quantity, complexity, and generally noisy nature of data involved.

This paper describes the use of a branch of learning called Similarity Based Learning, which looks at similarities and differences between sets of examples and counter–examples of a concept in order to produce rules which are satisfied by the examples, but not by the counter–examples. First the use of this technique in the exemplar will be described, as a bridge between the low–level image processing and the high–level image understanding. This will be followed by sections on the preprocessing of the inputs, a brief description of the learning algorithm, and the form and assessment of the output rules.

1.1 The Alvey Exemplar

The overall aim of the exemplar is to produce a prototype system which can recognise cars in a $2D$ scene. That is, given an input image, a set of $3D$ locations of any cars in the image is inferred by a combination of bottom–up image processing and top–down model–matching. The role of this work in the exemplar has been to generate rules that produce labels that are associated with regions or entire images. The input to the learning stage consists of a segmentation of an image, together with data such as the area, grey level, colour level, and profile based statistics for each region, and an adjacency graph. Segmentation programs [2,3] have been run on images containing cars (examples) and images not containing cars (counter–examples). The learning algorithm is then run on this data, to produce rules of the form 'assign label "car" to regions $R1$ to Rn if observable–property1 and observable–property2 etc'. These labels can then be used to cue in on specific areas in the image. The learning algorithm is run on a training set of data, typically a subset of the database of exemplar images. The rules produced can then be applied to unseen images to produce initial candidate regions of interest. Each region has an associated value to indicate the degree of belief in the assigned label. These values are in the form of Support Logic Pairs (SLOP) [4], and may be incorporated in a schema–based reasoning scheme [5], which may increase or decrease the support for a region having the assigned label. Within the exemplar work, the output from the image processing and reasoning stages is passed to a model–matching scheme [6] which verifies the existence of a car in the image, and infers its 3D location.

2 Preprocessing of Segmentation Data

Given a particular input image (Figure 1), the segmentation data is usually in the form of a region map (Figure 2), with an associated profile map (Figure 3). A set of statistics is provided for each region. These fall into three main categories : grey and colour level statistics, area based statistics, and shape and profile based statistics. Some examples are :

- Mean Grey Level
- Spread of Grey Levels
- Homogeneity

AVC 1987 doi:10.5244/C.1.4

- Area
- Centroid of Region
- Convexity
- Degree of Linearity

There is also information pertaining to the relationships between regions, for example :

- Adjacency Graph
- Contact Lengths between Adjacent Regions

Figure 1: Original Image

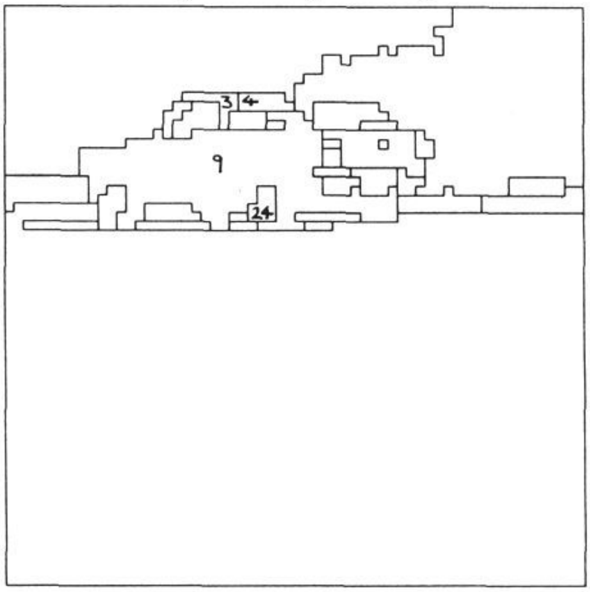


Figure 2: Region Map

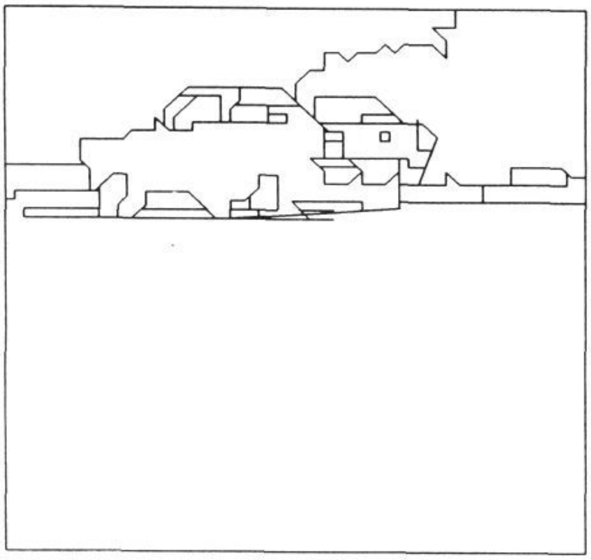


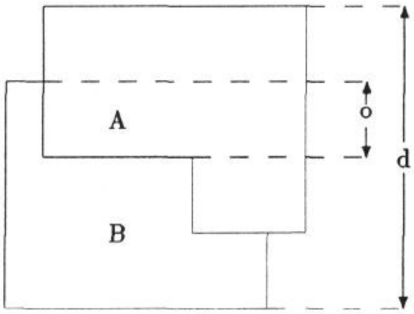
Figure 3: Profile Map

All these features are in a numerical form and are converted into symbolic form for use by the learning algorithm. For this purpose a number of heuristics have been devised. For example, the current definitions of 'above' and 'larger' are as follows :

If

$$\frac{1}{1+\frac{o}{d}} > T$$

Where T is a threshold.



Then Region A is above Region B

If

$$\frac{\text{area of Region A}}{\text{area of Region B}} = R$$

Then

$$1 \leq R \leq 1.5 \Rightarrow \text{A is the same size as B}$$

$$1.33 \leq R \Rightarrow \text{A is larger than B}$$

$$3.33 \leq R \Rightarrow \text{A is much larger than B}$$

Some of the other symbolic features calculated at present are : brighter, ontop, surrounds, aspect–ratio, parallelogram fit, degree of red saturation, etc. Some of the facts generated for Figure 1 are :

Region 4 is the same size as Region 3.
Region 3 is left of Region 4.
Region 4 is above Region 9.
Region 9 is much larger than Region 24.
Region 9 has a high value for red saturation.
Region 24 has a high value for a parallelogram fit.

Note that all the attributes and relations are two-dimensional features of what are usually three-dimensional objects in the image. This means that for the same three-dimensional object, for example the car, there can be a number of values for these features, depending on changes in scale, orientation and illumination. Therefore features have been chosen that will be as 'independent as possible' to these changes, and the main consequence in the exemplar work has been that relations are used in preference to attributes. For example, depending on the overall illumination, the usual shadow beneath a car can either be 'very dim', 'dim' or 'bright', but it is always 'dimmer than' the car body.

3 Use of the Learning Algorithm

The learning algorithm used is one which has been developed over the last two years on the Alvey project. It is based on Michalski's INDUCE algorithm [7], which learns rules from attributes and relations, but has two main differences.

The first of these is the incorporation of uncertainty, which currently uses a fuzzy set model to allocate a single certainty value to each attribute or relation that is presented to the algorithm. The certainty values are used to construct rules that have a good overall fit with the training data without requiring an exact match between a rule and any one example. For the purposes of the exemplar work, the full effects of the uncertainty model have not been considered, but the rules generated allocate Support Logic Pair values (SLOP) to regions based on how well they have performed on the test-run images from the exemplar database.

The second difference is the use of a dynamically changing type lattice for the attributes and relations used. Each feature e.g. 'adjacent', has a number of values it can take 'left adjacent, right adjacent, ontop, beneath'. These can be grouped together by defining a type lattice in the database. When the algorithm is run on a set of data, this type lattice is considered as possible groupings of the values. If however, the existing groupings do not work well on the data, then the algorithm changes the definition of the type lattice so that the new groups perform well on the dataset. This 'group if useful' technique has the advantage that groupings in the examples emerge even if the pre-defined type lattice does not succeed in finding them.

A significant feature of the INDUCE algorithm itself is that the relations between regions are examined before the attributes of the regions. As stated in a previous paper [8] this has the effect of biasing the rules towards those with relation conditions.

The datasets used to generate the rules contain data from between 9 and 20 colour images, with each image contributing 50-300 facts. The cars used in any particular dataset are all either front, side or rear views, at a variety of ranges. This means the 'car' rules produced are further specialised to 'side-on-car', or 'front-on-car' etc.

4 Results from Learning Algorithm

The output from the learning algorithm is a rule, or set of rules that can be used on an unknown image to extract information about that image. This information can be used in several different ways, depending on the training data and the task for which the whole system is being used. Firstly it can be used to label regions as a certain semantic type. A rule generated by the program for the side on view of a car is :

If Region A is left of Region B
Region C is above Region A
Region C is brighter than Region A
Region C is much larger than Region B

then Assert in the database that
Regions A,B,C have label 'side-on-car'

The result of applying the rule to an unknown image is that if there exist three regions in the image that satisfy these criteria, they are all labelled as 'side-on-car'. The algorithm can produce a number of rules from each training session. Thus

If Region A is left of Region B
Region C is much larger than Region A
Region C is much larger than Region B
Region A is a parallelogram

then Assert in the database that
Regions A,B,C have label 'side-on-car'
is another rule which has been used to label Regions A,B,C 'side-on-car'.

Figure 4 shows the result of applying six such

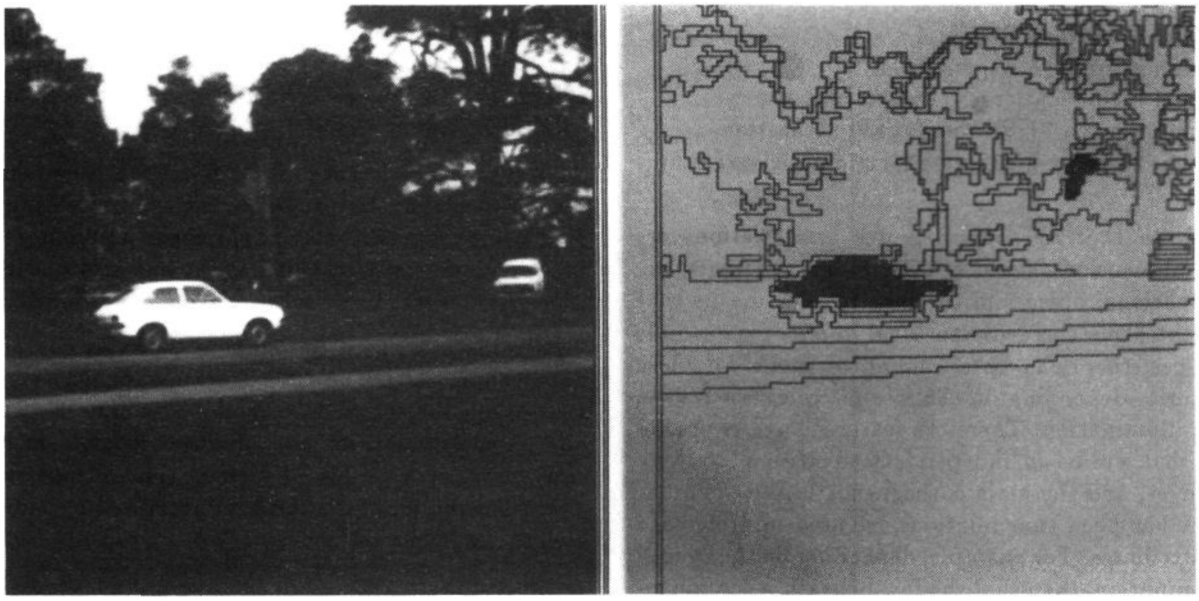


Figure 4: Unknown Image and Selected Regions

rules, including the two above, to an unknown image. The highlighted regions satisfied more than three rules.

A rule generated by using example images of rear views of a car was :

If Region A above Region B
Region C is above Region B
Region B is much larger than Region A
Region D is much brighter than Region A

then Assert in the database that
Regions A,B,C have label 'rear-on-car'

A second way in which the information can be used is more global, and may be thought of as more of a contextual rule. A typical generated rule of this type is :

If Region A is brighter than Region B
Region C is above Region B
Region C is brighter than Region D

then Assert in the database that
Regions B,C have class label 'car'

In this case the resultant information only applies to a subset of the regions considered in the rule, and makes no inference on the labels of the extra regions.

By applying a number of rules to the test-run images, SLOP values are derived that reflect the degree of confidence that a particular region has been correctly labelled. A SLOP pair allocates a degree of support for the asserted label (necessary support), and also a degree of support against the asserted label (possible support). These values are allocated based on the number of rules that fired on the region under consideration.

5 Discussion

In this section, some of the problems encountered when selecting data and running the algorithm are discussed.

5.1 Importance of Training Data

It has already been stated that the information provided by the learning algorithm can be used and interpreted in several different ways. This is determined primarily by the data on which the algorithm is run, and secondarily on the parameters used in the algorithm. An important consequence of this is that one of the uses of a learning algorithm is to test the Knowledge Representation being used. For example, if some hand-crafted feature is developed, which it is thought may be useful in identifying cars, then it can be tested by including it in the initial data presented to the algorithm. If the feature does have good discrimination power, then it will be selected for use in the conditions of the rules, and the converse. Good features can thus be chosen by including any number in the input data, and the 'best' of these will be selected for use in the rules. Since the structure of vision data often differs according to the segmentation parameters and noise, finding invariant features is a difficult task. The features used at present do give some discrimination. For example, as might be intuitively expected, a number of the rules generated contained a condition that a candidate region had a parallelogram shape. However, this condition was not sufficient in itself to discriminate between the examples and counter examples, and was always interlinked with a number of other conditions before all counter-examples could be eliminated.

An important consideration in the selection of training data is the choice of counter-example images. Winston [9] uses the concept of 'near-misses' to help the computer home in on a reasonable rule. A near-miss is defined as data that is identical to an example except in one respect (i.e. only one feature is different). This enables the program to decide that it is this particular feature that is important in classifying the object, and means it can quickly select 'the correct rule'. This is an important simplification of the task, and one that it is not possible to make in the case of the exemplar. The 'near misses' are impossible to select unless the target rule is known in advance, which is not the case. In general it is only possible to use 'far misses' which differ from an example by more than one feature, and consequently the process suffers to some extent from the credit assignment problem (i.e. not knowing which feature or combination of features caused the set of regions to become a counter-example). However, a single counter-example image will generally contain a large number of regions, which are all counter-examples of cars, and so the volume of data processed by the algorithm helps overcome this problem. An interesting addition to the set of features is the use of region labels. By giving the learning algorithm information on the classification of neighbouring regions, truly contextual rules, which relate the classification of an unlabelled region to the known labels of its neighbours, may be generated. For the purposes of the exemplar, only features directly measurable from the images have been used, but it is intended to extend the training data in the near future to incorporate labels.

5.2 Combinatorics of Machine Learning.

The machine learning algorithm used is essentially a search over the space of all possible rules, pruned by a set of heuristics. This space is searched starting with the most general rule, and working towards the most specific. Having found a set of rules that perform reasonably well, the rules are generalised by relaxing the conditions in certain ways. The generalisation part of the algorithm is ignored in this consideration of the combinatorics.

There are two distinct contributions to the computation time of the algorithm. These are :

i) A large number of possible rules to be tried.

ii) A large number of ways in which each rule will match with the data.

One of the principle reasons for the decision not to use the uncertainty model built into the learning algorithm for the exemplar work was to reduce the number of possible rules to be tried. By allowing multiple (uncertain) representations for an image, the search space is greatly increased. Instead the inclusion of an uncertainty model is delayed until after the learning process.

A number of heuristics are used to cut down this search time, some of which affect i) or ii), and some of which affect both. The most important heuristics are set out below.

In the course of the training phase, each rule is given a performance measure after it has been tested against the examples/counter-examples. At each level of the search tree, every node corresponds to a rule, and after all the rules at that level of the tree have been evaluated, the tree is beam-searched on the basis of the performance measure. The performance measure is based on how well the rule is satisfied by the examples/counter-examples, and consequently only a few of the (best) nodes are expanded down at the next level of the tree. This restricts an exponentially growing set of rules to a roughly constant number at each level of the tree.

Another heuristic used is the restriction of relations to those between adjacent regions, and to the largest regions in the image. This corresponds to the assumption that small regions are unaffected by distant small regions. For example, consider the relation 'is brighter than'. For an image with N regions, there are N(N+1)/2 relations of this form between the regions. By restricting the relations considered to those between adjacent regions and the largest M regions, the number of relations is substantially reduced. If each region has A adjacent regions, this reduces the number of relations by a factor of $(N+1)/((M+A)\times 2)$. Hence for an image with N=50, M=3, and A=5, the number of relations would be reduced by a factor of approximately 3. Since the time for the algorithm to run is proportional to (number of relations)^2, this heuristic reduces the run-time by about an order of magnitude.

The last heuristic considered here is the restriction that rules must have a repeated variable in the conditions. For example, the rule

If Region A is brighter than Region B
Region C is above Region B

then Assert in the database that Regions B, C have label 'car'.

is allowed, but the rule

If Region A is brighter than Region B
Region C is brighter than Region D

then Assert in the database that Regions B, C have label 'car'.

is not, because the two conditions have variables that are unrelated to each other. This has the effect of restricting the number of matches between any given example and the rule, because in the first rule there are fewer regions involved, and consequently more conditions per region are imposed, which means that fewer matches are found.

6 Conclusions

An implementation of a similarity-based learning technique as applied to image analysis has been described. The process provides a systematic way of evaluating the features being used, and has been used iteratively to improve the Knowledge Representation for the exemplar. It bridges the gap between observable features and labels by looking for the best combinations of discriminant features, and the rules generated are useful as a cueing process by generating evidence for and against labels for regions.

Since the form of the data in the vision problem is so complex, and the nature of induction is uncertain, the rules produced are used as an intermediate representation, i.e. to restrict the input for more complex reasoning processes. The variable nature of the data also means that in order to find sets of features which are invariant under different noise, viewpoint and general illumination conditions, the relationships between different regions provide a more fruitful source of information than absolute values.

7 References

1. J.H. Connell and M.Brady 'Learning Shape Descriptions', IJCAI 1985 pgs 922–925.

2. A.Page 'A Segmentation Algorithm for Colour Images', B.Ae.S.R., Alvey Contract 'The Identification of Objects in 2D Images' in preparation.

3. J.Hyde 'Image Segmentation and Attribute Generation' M.C.C.S., Alvey Contract 'The Identification of Objects in 2D Images' , Proc. AVC 1987.

4. J.Baldwin 'Support Logic Programming', University of Bristol, ITRC 65

5. S.Morton 'Object Hypothesis by Evidential Reasoning', University of Bristol, Alvey Contract 'The Identification of Objects in 2D Images' , Proc. AVC 1987.

6. K.Brisdon 'Instance Verification and Refinement', University of Reading, Alvey Contract 'The Identification of Objects in 2D Images' , Proc. AVC 1987.

7. R.S.Michalski, J.G.Carbonell, T.M.Mitchell eds. 'Machine Learning - an A.I. approach' Tioga Publishing Co., Palo Alto, California 1983.

8. D.Hutber, P.Sims 'Learning Rules from Uncertain Data' Alvey Contract 'The Identification of Objects in 2D Images' Alvey Ref. No. AOI/TR/BAIT/860217 1986.

9. P.H.Winston 'Artificial Intelligence' Addison–Wesley 1984.