Combining Evolutionary, Connectionist, and Fuzzy Classification Algorithms for Shape Analysis

Paul L. Rosin
Department of Computer Science
Cardiff University
UK
Paul.Rosin@cs.cf.ac.uk
and
Henry O. Nyongesa
School of Computing
Sheffield Hallam University
UK
H.Nyongesa@shu.ac.uk

Abstract

This paper presents an investigation into the classification of a difficult data set containing large intra-class variability but low inter-class variability. Standard classifiers are weak and fail to achieve satisfactory results however, it is proposed that a combination of such weak classifiers can improve overall performance. The paper also introduces a novel evolutionary approach to fuzzy rule generation for classification problems.

1 Introduction

This paper describes a series of experiments in tackling a difficult classification problem. The data consists of various beans and seeds, examples of which are shown in figure 1. Although some of the objects are larger than others (e.g. almonds compared to lentils) we are interested in classifying them based on their shape alone without using information about their size. This corresponds to the situation where the distance between the objects and the camera is not fixed, and so their apparent imaged sizes would vary. The difficulty of the task lies in the relatively small inter-class difference in shape and the high intra-class differences. In other words, all the objects look similar, appearing roughly elliptical. Although the shapes of some objects (e.g. almonds) are fairly consistent others vary considerably (e.g. corn kernels).

The basis for classifying the objects will be a set of shape properties measured from their silhouettes. Since size information is to be discarded the properties need to be invariant to scaling. Likewise, invariance to position and orientation changes is necessary. Furthermore, it may be useful to include invariance to additional transformations of the shape. For instance, if the determining shape factor of a class is its similarity to an ellipse then the aspect ratio may be irrelevant. The computer vision literature provides a variety of shape measures [1]. A selection of these, in combination with some new shape properties developed by Rosin [2], have been applied to generate a set of 17 measurements of each sample. They can be divided into subgroups according to their properties and/or algorithms:

- *moments #1* – four attributes invariant to rotation, translation, and scaling (invariant under similarity transformations).
- *moments #2* – three attributes invariant to rotation, translation, scaling, and skew (invariant under affine transformations).
- *standard* – four standard attributes – eccentricity, circularity, compactness, and convexity (invariant under similarity transformations).
- *geometric #1* – three measurements of ellipticity, rectangularity, and triangularity (invariant under affine transformations, similarity transformations and stretching along the axes, and affine transformations respectively).
- *geometric #2* – three alternative measurements of ellipticity, rectangularity, and triangularity.

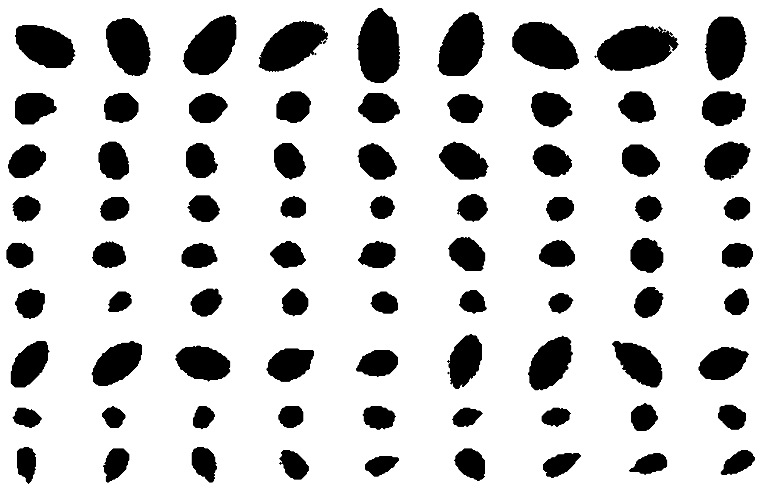

Figure 1: Examples of data; rows contain: (1) almonds, (2) chickpeas, (3) coffee beans, (4) lentils, (5) peanuts, (6) corn kernels, (7) pumpkin seeds, (8) raisins, (9) sunflower seeds.

In this paper we have investigated four different classification techniques, and combined them in an attempt to improve overall classification. Two implementations of decision trees are used. The first method is the well known C4.5 machine learning method developed by Quinlan [3]. C4.5 induces trees that partition feature space into equivalence classes using axis-parallel hyperplanes (e.g. in 2D this would consist of horizontal and vertical lines). The second approach is OC1 [4] which is a generalisation in that, rather than checking the value of a single attribute at each node, it tests a linear combination of attributes. Feature space is consequently partitioned by oblique hyperplanes. The third method is an ensemble of neural networks that are trained on different features patterns, including one network that incorporates all patterns. Finally, we have also evolved fuzzy rules using evolutionary programming.

2 A Brief Overview of Connectionist, Decision Tree and Fuzzy Classification

Neural networks are connectionist systems that are widely used for regression and function approximation. They comprise input nodes, which are used to provide training data to the network and output nodes used to represent the dependent variables. Input-Output mapping is achieved by adjusting connection weights between the input and output nodes, but more usually, through an intermediate layer of nodes. This characteristic can be modified for classification problems by specifying desirable network outputs to be binary values. Neural networks are reliably used in classification of complex data.

Fuzzy classification is a rule-based approach in which IF-THEN rules are used to categorise data. The rules relate generalised or imprecise groupings of input data, and the decision of a given rule represents a degree of belonging to a given output class. This type of classifier is particularly useful when it is necessary to provide interpretability to the classification system in the form of linguistic rules. However, the process of creating classification rules is often difficult and time-consuming. Several studies have attempted to cope with this problem using learning algorithms, and in particular the use of neural networks.

Decision trees are a well established technique based on structuring the data, according to information theory, into mutually exclusive crisp regions. Classification is generally performed by starting at the tree's root node and checking the value of a single attribute, and depending on its values the appropriate link is followed. This process is repeated at succeeding nodes until a leaf is reached and a classification is assigned. Generating the optimal decision tree is generally NP-hard, and therefore a sub-optimal tree is induced instead, using greedy or randomised hill climbing algorithms for instance. The resulting tree is often pruned; subtrees are replaced by leaf nodes if this reduces the expected error rates. This results in potentially smaller and more accurate trees.

3 Evolutionary Fuzzy Classification

Evolutionary techniques have not been previously applied to fuzzy classification problems. The closest related work studies use genetic algorithms to optimise fuzzy classification rules and their membership functions. One disadvantage with this approach is that it is invariably necessary to pre-specify the structure of the rules, which often results in sub-optimal classification. In this paper, we have proposed a new technique in which fuzzy classification rules of arbitrary size and structure can be generated using genetic programming. This is desirable for complex problems, with large numbers of input features, for which it is not feasible to formulate the structure of rules manually. Furthermore, with such large numbers of features it is usually the case that certain features are not significant in classification of different output classes. Hence, in this case, we can say that genetic programming is used for unconstrained rule discovery and optimisation.

Genetic programming is an evolutionary technique in principle to Holland's genetic algorithms. The main differences are, (1) the structure of a genetic program is a tree, (2) the nodes of the trees are functions (or terminals), which enables the trees to interpreted as programs and (3) the size of each tree in a population is variable, unlike most genetic algorithms where all individuals are the same size. Otherwise, standard operators applicable to genetic algorithms are used in genetic programming.

In this study, the normalised input space was partioned into three fuzzy membership functions, *negative, zero* and *positive* [5]. Let N, Z, P be the fuzzy membership functions. We assumed simple rule constructs comprised of two inputs and one output. The non-terminal nodes of the GP trees represent these simple rules, which are combined to form the complex classification rule. Each simple rule is evaluated by matching its inputs against the fuzzy antecedents and the output is obtained using an AND operator, namely MIN. Thus, a node expressed as ZP(x,y) is interpreted as:

$$\text{IF } (x = Z \text{ AND } y = P) \text{ THEN MIN}(\mu_Z\,|_x, \mu_P\,|_y)$$

where $\mu_Z\mid_x$ is the degree of belonging of x to the fuzzy membership function Z. This type of rule construct is preferable to direct combination of the input parameters because it assists in interpretability of the classification system. There are nine different fuzzy rules which can be formed from the combination of the three membership functions.

The study used GP source code (lilgp) developed at Michigan State University [6]. This was used to evolve a set of complex fuzzy classification rules, one for each output class of data. The function set is comprised of the simple rules described above, while the terminal set comprised random constants and common arithmetic operators. The fitness of the trees on their evaluated outputs were determined against targets of 1.0 for the correct data class and 0.0 otherwise. As an example, the following is a small portion of a complex rule:

```
(ZP (NP a b) (ZN d c) )
```

It can be interpreted as:

IF d is Z AND c is N THEN temp1 = MIN($\mu_Z\mid_d, \mu_N\mid_c$)
IF a is N AND b is P THEN temp2 = MIN($\mu_N\mid_f, \mu_P\mid_b$)
IF temp1 is Z AND temp2 is P THEN
out= MIN($\mu_Z\mid_{\text{temp1}}, \mu_P\mid_{\text{temp2}}$)

4 Voting Schemes

Because neural networks are trained on limited sample sets of representative data there are always significant errors in the generalisation of complex functions. One way to overcome this problem is to train multiple neural networks on independent data sets and then use voting schemes to determine an overall classification [7]. The two popular schemes are commonly known as ensemble and modular networks. In ensembles or committees redundant networks are trained to carry out the same task, with voting schemes being applied to determine an overall classification. On the other hand, it is pointless to train identical neural networks and consideration is thus often given to using different topologies, data sets or training algorithms. In modular networks the classification problem is decomposed into subtasks. For example, neural networks are trained to respond to one class or a group of classes.

The output of a classification network can be used to indicate the degree to which the input features are matched to the different classes. Therefore a simple approach to network combination is to sum the activation levels of corresponding output nodes. A refinement to this scheme is to scale the output levels within a network such that they sum to one. This allows the contributions across networks to be more comparable.

An alternative approach is based on the confusion matrix, a table containing entries c_{ij}, which indicate the frequency that data samples from class i were labeled as class j. Such a table is useful for analysing the performance of a classifier. Our approach is based on the classification accuracies for each class given by

$$\frac{a_{ii}}{\sum_j a_{ij}}$$

The contributions of each classifier are weighted by these expected accuracies. An example of a confusion matrix is shown in Table 1. It can be seen that the classifier is capable of consistently correctly classifying all instances of class 1, but only 75% of class2.

	1	2	3	4	5	6	7	8	9	% Accuracy
1	11	0	0	0	0	0	0	0	0	100.0
2	0	9	3	0	0	0	0	0	0	75.0
3	0	2	15	1	0	0	0	0	0	83.3
4	1	0	3	12	0	0	0	1	0	70.6
5	0	2	3	3	0	1	1	0	1	0.0
6	0	3	1	1	0	3	2	1	1	25.0
7	1	1	0	0	0	0	16	0	1	84.2
8	0	1	1	2	3	3	1	1	1	7.7
9	0	2	1	0	1	0	0	0	13	76.5

Table 1: Confusion matrix

5 Experiments

The experiments compared the performance of four classifiers on the shape classification problem. The training and testing data both consist of 130 samples each containing seventeen continuous attributes of shape. Following standard practice the data was normalised prior to presentation to the neural and fuzzy classifiers, whereas this was not necessary for the decision trees.

The first set of experiments concentrated on evaluating each individual classifier. Examples of their outputs are shown in Figures 2 to 4. Their performances are shown in Table 2. The bottom three entries correspond to alternative means to deriving a classification based on neural output node activation levels. These are standard winner-take-all (WTA), and using the two decision trees (OC1 and C4.5). It is of interest to note that the simple winner-take-all performs better than the more complex decision trees in combining the outputs of the neural networks.

The second set of experiments compared various voting schemes applied to neural network ensembles. The neural networks were trained on the five subgroups of shape properties, and an additional neural network was trained on the combined seventeen properties. The voting schemes either select only the maximum activation level within a network (winner-take-all), or alternatively all the activations levels are used. The levels may first be scaled, and weighted by the confusion matrix, and are then summed over the networks. The result in Table 3 shows that better performance is achieved when the network ensemble includes the network trained with all properties. Furthermore, the use of the confusion matrix generally improves performance. However, the modular approaches provide little improvement over the basic single network clasification result.

The final set of experiments investigated hybridisation of three classifiers, namely, neural network, fuzzy-GP and C4.5, using the confusion matrix method. This does not rely on potentially incommensurate likelihood values that may be produced by different classifiers. We see from Table 4 that further gains in performance have been achieved, indicating that the neural network and decision tree provide useful complimentary information.

6 Conclusions

This paper has presented a comparison of different classification techniques on a difficult shape analysis problem. Similar to other reports in the literature [8, 9] it has been shown there is no significant differences between the individual techniques on our classification problem. However, we have shown that improvements can be achieved through different

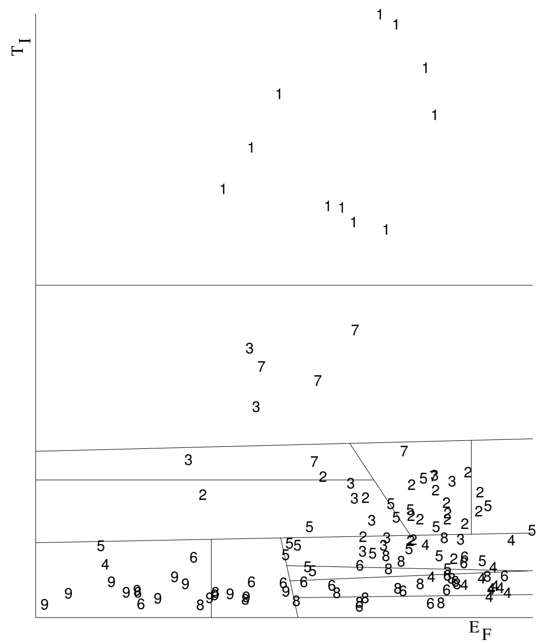

Figure 2: An example of the partitioning of feature space for the bean classification task by the OC1 decision tree using only two properties. The horizontal axis property is elliptiocity, and the vertical axis property is triangularity.

```
Rule 4:
     property 16 <= 0.225411
->  class 6

Rule 7:
     property 1 > 63.7441
     property 1 <= 65.4694
     property 4 > 0.161205
     property 4 <= 0.179735
     property 11 <= 0.967232
     property 13 > 0.710997
     property 15 <= 0.061688
     property 16 > 0.225411
     property 16 <= 0.329981
->  class 8

Rule 8:
     property 1 > 65.4694
     property 4 <= 0.179735
     property 15 <= 0.061688
->  class 6
```

Figure 3: A typical set of rules generated by C4.5

```
(ZP (NP b
        (PN (- (PN a 0.83860)
               (NP (ZP (- (- (ZN d c)
                             (NZ -0.02504 d))
                          (PZ a f)) h)
                   (PZ e h))) g))
    (ZP a
        (ZP (- (ZN d c)
               (NP d
                   (ZN d c)))
            (ZP (NP f b)
                (ZN d c)))))
```

Figure 4: A portion of a complex fuzzy-GP rule in prefix notation. Upper case letters denote fuzzy rules, and lower case letters denote ionput shape property values.

METHOD	ACCURACY %
C4.5	53
OC1	54
Fuzzy-GP	44
NN + WTA	57
NN + C4.5	52
NN + OC1	51

Table 2: Accuracy for single classifiers

METHOD	ACCURACY %	
	Subgroups	Subgroups + Everything
WTA, Sum of activations	39	46
WTA, Sum of activations with scaling	40	46
WTA, Sum of CM	55	58
WTA, Sum of activations, weighted by CM	44	55
WTA, Sum of activations with scaling, weighted by CM	49	56
Sum of activations	42	49
Sum of activations, with scaling	43	51
Sum of activations, weighted by CM	50	56
Sum of activations, with scaling, weighted by CM	51	58

Table 3: Accuracy for NN voting schemes

METHODS	ACCURACY %
NN + C4.5	63
NN + Fuzzy-GP	58
Fuzzy-GP + C4.5	57
NN + C4.5 + FGP	61

Table 4: Hybrid Classifiers

combinations of these techniques [10].

A new approach to generate fuzzy clasification rules using genetic programming, on other hand, has not demonstrated satisfactory result. This may be due to the limited partitioning of the fuzzy input space into only three membership functions implemented in the study and further work will investigate the potential of this method.

References

[1] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision.* Chapman and Hall, 1993.

[2] P.L. Rosin. Measuring shape: Ellipticity, rectangularity, and triangularity. *submitted for publication.*

[3] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[4] S.K. Murthy, S. Kasif, and S. Salzberg. System for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–33, 1994.

[5] D. A. Linkens and H. O. Nyongesa. Genetic algorithms for fuzzy control - part i. *IEE Proc - Control Theory Appl.*, 142:161–176, 1995.

[6] D. Zongker and B. Punch. *http://isl.cps.msu.edu/GA/software/lil-gp.* 1996.

[7] A.J.C. Sharkey, editor. *Combining Artificial Neural Networks.* Springer-Verlag, 1999.

[8] H. Chen, P. Buntin, L. She, S. Sutjahjo, C. Sommer, and D. Neely. Expert prediction, symbolic learning, and neural networks: An experiment on greyhound racing. *IEEE Expert*, 9:21–27, 1994.

[9] M. Mulholland, D.B. Hibbert, P.R. Haddad, and P. Parslov. A comparison of classification in artificial intelligence, induction versus neural networks. *Chemometrics and Intelligent Laboratory Systems*, 30:117–128, 1995.

[10] L.Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Trans. Systems Man and Cybernetics*, 22:418–435, 1992.