Genetic Algorithms as a Tool for Restructuring Feature Space Representations

Haleh Vafaie and Kenneth De Jong
Computer Science Department
George Mason University
Fairfax, VA 22030

Abstract

This paper describes an approach being explored to improve the usefulness of machine learning techniques to classify complex, real world data. The approach involves the use of genetic algorithms as a "front end" to a traditional tree induction system (ID3) in order to find the best feature set to be used by the induction system. This approach has been implemented and tested on difficult texture classification problems. The results are encouraging and indicate significant advantages of the presented approach.

1.0 Introduction

In recent years there has been a significant increase in research on automatic image recognition in more realistic contexts. The corresponding increase in difficulty in designing effective classification procedures for the important components of these more complex recognition problems has resulted in the application of different methods to reduce the number of features used to represent the problem spaces. Many of these recognition problems are not properly represented using the reduced feature set, which makes them very sensitive to the features used to define their space. Hence, there is a strong motivation to find an appropriate representation space. This is a difficult task especially when performed manually, since the space of potential representations is very large. This has led to an interest in machine learning techniques as a possible strategy for automating the process of changing representation spaces. This paper describes part of a larger effort to apply machine learning techniques to such problems in an attempt to improve classification process required for various recognition tasks. The immediate problem attacked is that of texture recognition in the context of noise and changing lighting conditions. In this context standard induction systems such as ID3 produce classification trees that may be sub-optimal in two respects. First, there is a need to minimize the number of features actually used for classification, since each feature used adds to the design and manufacturing costs as well as the running time of a recognition system. At the same time there is a need to achieve high recognition rates in the presence of noise and changing environmental conditions.

This paper describes a methodology being explored to improve the usefulness of machine learning techniques for such problems. The approach described here involves the use of genetic algorithms as a "front end" to a decision tree induction system in order to find an adequate feature space through selection and/or construction of a useful set of features to be used by the tree induction system. The results presented suggest that genetic algorithms are a useful tool for solving difficult recognition problems in which the performance of the underlying system is an important design consideration.

2.0 Background

The problem of finding good representation spaces is certainly not a new one. Both feature selection and feature construction present difficult search problems, in that they require exploration of very large search spaces. A review of previous work in these area reveals that the strategies applied are based on greedy or directed search algorithms in order to search the space more efficiently. However, these methods are very brittle and may get trapped on local peaks and produce inadequate results [8].

As a consequence, there is a significant opportunity for improving the usefulness of machine learning techniques for automatically generating useful classification procedures for mis-represented problems if effective means were available for finding feature sets that produce optimal performance. Since genetic algorithms are best known for their ability to efficiently search large spaces about which little is known and have proved to be robust, they seem to be an excellent choice as a search strategy for selecting and constructing new features to be used in designing a recognition system [2].

3.0 The proposed architecture

The proposed GA-based method for transforming initial feature sets into more useful ones is shown in Figure 1. An important property of this architecture is that feature selection and feature construction are independently selectable modules. The advantage of this is considerable

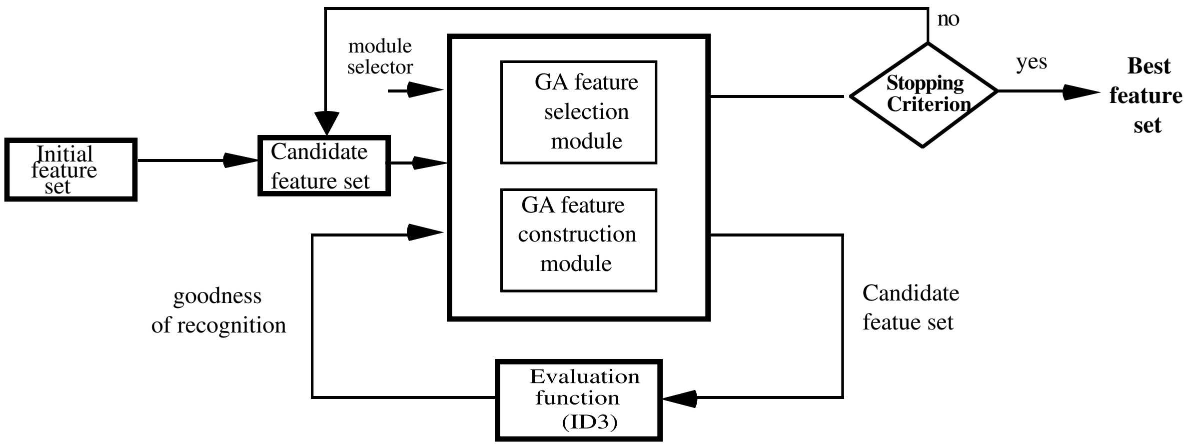

Figure 1. Block diagram of the proposed architecture

flexibility in controlling the way in which the two processes interact. In particular, there is sufficient generality here to allow the modules to be selected dynamically depending on the state of the problem.

In this approach it is assumed that an initial set of features will be provided as input as well as a training set in the form of feature vectors extracted from actual data and representing positive and negative examples of the various classes for which classification is to be performed. Depending on the state of the problem either the feature selection or construction method is applied to search the problem space in order to improve the recognition rate of a given classification system. Both the feature selection and construction methods are based on genetic algorithms that use an evaluation function as a feedback to guide the search.

The performance of each of the selected feature subsets is measured by invoking an evaluation function with the correspondingly modified feature space and training set, and measuring the specified classification result. The best feature subset found is then output as the recommended set of features to be used in the actual design of the recognition system.

3.1 Genetic Algorithms

The main issues in applying GAs to any problem are selecting an appropriate representation and an adequate evaluation function. The representation for feature selection is trivial in that a simple binary representation has proved to be very effective [6], [7]. However, this is a difficult issue for feature construction and a more complex representation is required in order to effectively apply a genetic algorithm.

Although they use different representations, both the feature selection module and feature construction module are designed to evolve better subsets. Hence, a single evaluation procedure can be used for both modules. The following sections describe these methods in detail.

3.2 GA Representation for Feature Construction

The most natural way to construct useful new features is by forming combinations of existing features via a well-chosen set of operators. For example, in image processing domains initial features are frequently real-valued and useful new features derived by combining existing features using simple arithmetic operations (such as +, -, *, /). Hence, new features can be expressed as expressions such as (F1 - (F2 + F3)) or (F4 * F4) and represented naturally as tree structures.

However, we are interested in evolving sets of features which work well together, so an individual in this case is a variable-length structure representing a set of features, some of which may be from the original feature set and some which are expressions. For example,

((F1 - (F2 + F4)), (F4 * F4), F9)

would represent a set of 3 features, two of which have been constructed from initial features. Such feature sets (individuals) are naturally represented as tree structures as illustrated in Figure 2.Closely related to the choice of representation is the selection of useful forms of the genetic operators of recombination and mutation. In the following experiments the crossover operation will follow Koza's proposed crossover operator for his genetic programming paradigm [4]. This crossover operator has proved to be well suited for variable length hierarchical structures. Figure 2 illustrates how such an operator might swap selected features from two features sets. In order to maintain the necessary variation in the population, mutation involves randomly selecting and replacing an operator (or feature) with a member of the initial set of operators (or features).

3.3 GA Representation for Feature Selection

For feature selection, the main interest is in representing the space of all possible subsets of the given feature set. Hence, the simplest form of representation is to have one binary gene for each feature, and the value of a gene

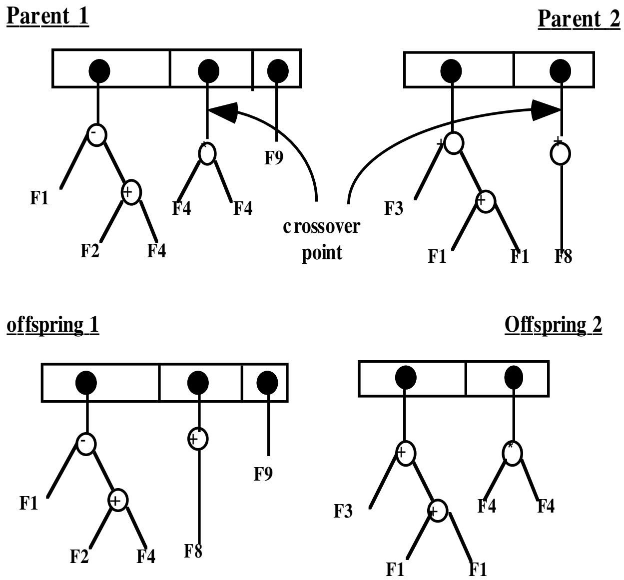

Figure 2. An example of the crossover operation

represent the presence (or absence) of that feature in the feature set. Then, each individual consists of fixed-length binary string representing some subset of the given feature set. An individual of length l corresponds to an l-dimensional binary feature vector X, where each bit represents the elimination or inclusion of the associated feature.

The advantage to this representation is that the classical GA's operators as described before (binary mutation and crossover) can easily be applied to this representation without any modification. This eliminates the need for designing new genetic operators, or making any other changes to the standard form of genetic algorithms.

3.4 GA Evaluation Procedures

In order to evolve better feature subsets, both the feature selection module and the feature construction module require a procedure for estimating the fitness of a given future subset.

The evaluation procedure is divided into three main steps. After a feature subset is selected, the initial training data, consisting of the entire set of feature vectors and class assignments corresponding to examples from each of the given classes, is modified. This is done by adding/removing the values for features that are not present in the initial set of feature vectors. The second step is to apply ID3 as the classification procedure to the new modified training data to generate the decision tree for the given classes in the training data. The last step is to evaluate the produced decision tree with respect to their classification performance on the test data.

4.0 Experiments

In order to evaluate and test our ideas we have implemented a prototype version of the system. For each of the GA modules, we selected an existing program that best suited the required representation and recombination operators. The necessary changes were made to these programs in order to satisfy the design requirements. For the feature construction component, we modified the simple genetic algorithm GAL that was developed by Bill Spears to accommodate our goals, and GENESIS [3] was used for the feature selection module. We also used without modification C4.5, a standard implementation of ID3 [5], to build up the decision trees for the evaluation procedure. For all components, standard default parameter settings from the literature were used. For GA modules, this resulted in a constant population size of 50, a crossover rate 0.6 and a mutation rate of 0.001. For C4.5, the pruning confidence level was set to default 25%.

An initial set of experiments in texture domain has been performed to assess the performance of the presented system. In this experiment feature selection module was applied first, followed by the feature construction module. The generated subsets of optimal features for recognizing visual concepts in texture data have been compared with the previously presented GA-based method which used only the feature selection module. The error rates on unseen texture data have been used as the basis for comparison.

In these experiments four texture images were randomly selected from Brodatz [1] album of textures and are shown in Figure 3. Two hundred feature vectors, each containing 8 features were then randomly extracted from an arbitrary selected area of 30 by 30 pixels from each of the chosen textures. These feature vectors were divided equally between training examples used for the generation of the decision tree, and testing examples used to measure the performance of the produced decision tree.

The training data was then divided into two data sets to be used for generating an optimal feature set. The data set consisting of 67% of the training examples was used for inducing decision trees and the remaining 33% were used for evaluation of the feature subset.

In our experiment we selected to apply the feature selection module and feature construction module only once. Since, in general we are faced with large number of features and search spaces, the feature selection module was applied first in order to reduce the number of features and the search space.

The feature construction module used the feature subset that was produced during the best run as input in order to find a more suitable representation by searching the space of all possible combinations of the given features (using simple arithmetic operators). Figure 4 shows average performance of the selected set over 10 runs using the feature construction module. The feature set that was produced during the best run (lowest error rate) was output as the final feature set to be used for the recognition system.

Table 1 shows the results of our experiments together with the corresponding performance for the set including all the initial features and the set obtained using only the feature selection strategy. The feature selection method reduced the feature set size by 50%. The feature construction module further reduced the feature set cardinality by proposing only three features. This final feature set included two of the four input features along with a feature that was constructed using a combination of all of the input features. Note that in addition to reducing the dimensionality of the final feature set, recognition performance improved as well.

Table 1: Experimental Results for Texture Data

Full feature set *8 Features*	**Reduced feature set feature selection module** *4 Features*	**Final Reduced feature set** *3 features*
Error Rate	Error Rate	Error Rate
34.2%	30.7%	29.6%

5.0 Summary and conclusions

The initial experimental results obtained indicate the potential advantages of using GA-based feature selection and construction techniques to improve classification performance. Our initial experiments and results indicate a small but significant improvement in the classification and recognition of real world images. In addition, the reduction of the number of features improved the execution time required for rule induction substantially. The reported experiments and results indicate that an adaptive feature space restructuring strategy using genetic algorithms can yield a significant reduction in the number of features required for texture classification and simultaneously produce improvements in recognition rates of the decision tree produced by C4.5. This is a significant initial step towards the application of machine learning techniques for automating the of constructing classification systems for difficult image processing problems.

Clearly, more testing is needed in order to substantiate our results both in this domain as well as other more complex domains involving larger feature and data sets. We are currently involved in applying this approach to a difficult face recognition problem involving more than one hundred features.

There are also many architectural issues worth exploring. For example, in these experiments each module was only used once. More complex interactions between feature selection and feature construction are likely to be useful as these ideas are applied to more difficult problems.

Acknowledgments

The authors wish to thank B. Spears for providing his basic GA component, GAL, and J. Huang for assisting in the application of C4.5 program. We also like to thank J. Bala for furnishing us with the texture data.

This research was conducted at George Mason University. The research was supported in part by the National Science Foundation under grant No. IRI-9020266 and CDA-9309725, in part by the Defense Advanced Research Projects Agency under the grant No. N00014-91-J-1854, administered by the Office of Naval Research, and the grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, and in part by the Office of Naval Research under grant No. N00014-91-J-1351.

References

[1] Brodatz, P., *A Photographic Album for Arts and Design*, Toronto, Canada: Dover Publishing Co., 1966.
[2] De Jong, K., "Learning with Genetic Algorithms : An overview," *Machine Learning,* 1988.
[3] Grefenstette, J.J., L. David and D. Cerys , *Genesis and OOGA: Two Genetic Algorithms System*, TSP: Melorse, MA, 1991.
[4] Koza, J.R., *Genetic Programming*, Cambridge, MA: MIT Press, 1992.
[5] Quinlan, J.R. *The Effect of Noise on Concept Learning,* in Machine Learning: an Artificial Intelligence Approach, R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), Morgan Kaufmann publishers, San Mateo, CA, 1986.
[6] Vafaie, H., and De Jong, K., "Genetic Algorithms as a Tool for Feature Selection in Machine Learning," *International Conference on Tools with AI*, pp. 200-204, Arlington, VA, 1992.
[7] Vafaie, H., and De Jong, K., "Improving a Rule Learning System Using Genetic Algorithms," *Machine Learning: A Multistrategy Approach*, R.S. Michalski and G. Tecuci (Eds.), San Mateo, CA: Morgan Kaufmann, 1993.
[8] Vafaie, H., and De Jong, K., "Robust Feature Selection Algorithms," *International Conference on Tools with AI*, pp. 356-364, Boston, Massachusetts, 1993.

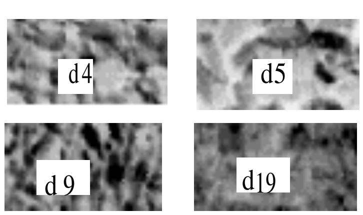

Figure 3. The texture images used for the experiments

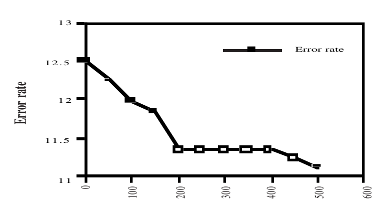

Figure 4. Average Error rate for 10 runs using the reduced feature set