

Genetic Algorithm Technique in Hybrid Intelligent Systems for Pattern Recognition

Mansoor Farooq

PhD Scholar, Department of Computer Science, Shri Venkateshwara University, U.P. India

ABSTRACT: In this research paper we use combination of stochastic algorithms and hybrid intelligent systems for pattern recognition or classification. Stochastic algorithm has been used for model selection of components or classification, especially when the dimension of the patterns is high. Genetic algorithms are one of the commonly used techniques for stochastic classifiers. Hybrid Intelligent Systems (HIS) combine intelligent techniques in synergistic architectures in order to provide solutions for complex problems. These systems utilize at least two of the three techniques: fuzzy logic, genetic algorithms and neural networks. The aim of their combination is to amplify their strengths and complement their weaknesses. A good architecture for a hybrid system should match each of its tasks to the appropriate intelligent technique and provide an efficient means for their integration. However, it is not always obvious or easy to build HIS architectures that achieve the higher intelligence goal. A good architecture for a hybrid system should match each of its tasks to the appropriate intelligent technique and provide an efficient means for their integration.

KEYWORDS: Genetic Algorithm, Hybrid Intelligent Systems and Pattern Recognition

I. INTRODUCTION

Pattern recognition is an established field of research with a wide variety of applications in fields such as speech recognition, image processing, and signal analysis. Many of the developed techniques of pattern recognition [2] have been applied successfully to a variety of real problems. However, there are always new applications that have different types of patterns that render the existing algorithms unable to handle them. The primary goal of pattern recognition is pattern classification [2]. In general, pattern classification techniques can be divided into two main categories [2]: supervised and unsupervised techniques. In supervised classification the classes and their boundaries are determined or at least defined by the designer of the classification system before training the classifier. On the other hand, in unsupervised learning the system learns the boundaries of the classes based on the similarity between groups of patterns.

Pattern classification techniques can also be categorized based on the type of algorithm used. An important group of techniques is statistical pattern classification. In these techniques a given pattern is assigned to a certain category based on a vector of features. Each class is assumed to define a probability density function for these features. The conditional probability of a pattern belonging to a certain class is computed and decision rules like Bayesian decision rule or maximum likelihood are applied [2] to determine the class of a given pattern. More recently, neural network techniques and methods imported from statistical learning theory have been receiving increasing attention. Stochastic methods and unsupervised learning techniques are also among the actively studied techniques.

Despite the number of different classification techniques, there are many applications that still require more powerful classification systems. A promising way for developing these systems is to combine more than one technique in unified classifier architectures. This type of combination is also called hybridization. The field of hybrid systems is still a new field of research and there is a lot to be accomplished in it. The design of a recognition system for a specific application requires careful attention to the following issues: definition of pattern classes, sensing environment, pattern representation, feature extraction and selection, cluster analysis, classifier design and learning procedure [2]. In this paper, we will focus on hybrid intelligent systems for pattern recognition.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

II. RELATED WORK

Most reported systems in the literature deal with the voluminous data collection using some feature extraction method. Note that the original pattern is basically a 2-D function that relates time to the measured attribute reading. Each reading point is considered to be a feature. The main two techniques used for feature extraction select 60 or 72 points per 480. After that, only six of the extracted features are put together to form a vector that represents the pattern to the classifier. The main concept is to reduce the 2-D large sequence of points describing the pattern to a 1-D vector representation with a minimal number of elements that enable reasonable classification performance. Thus each pattern is converted to a point in a higher dimensional space, e.g. 6 instead of being described by a large sequence of points in 2-D space. Obviously, this method involves loss of information since the location in the time domain where the original important feature happened is lost. The main driver for this approach might be the dependence classification techniques like Principal Component Analysis and Linear Discriminant Analysis that can handle vectors with limited numbers of elements (dimensions).

In the real domain the number of values representing a pattern is huge. For Example 1024 values to represent each pattern. This large number of values is not suitable for commonly used pattern classification algorithms such as clustering and neural network techniques. This problem is usually called the problem of high dimensionality [11]. A commonly used technique for overcoming this problem is feature extraction. The basic idea is to extract the important features of the pattern and use these in the pattern classification instead of the full pattern description in the real domain. Therefore to achieve our goal of being able to handle large numbers of classes we developed the concept of the abstract domain. In this domain the pattern is described by a set of features made out of simple shapes rather than a selected vector. Simple shapes can be triangles, rectangles, or any other polygonal shape. The idea here is take a sequence of values from the large data vector and represent them with the shape. Note that we also keep the location of the shape.

As we have chosen a binary decision tree as the overall architecture for decision tree classifiers, the tree building process starts with a training pattern set S_{train} , then a classification rule is chosen and a node is added to the tree to capture this rule. The training set is subdivided into subsets based on the chosen rule, which will be discussed later. The same process for choosing rules and adding nodes is repeated recursively until the pattern subset contains patterns from one class. At this point a leaf is added to the tree and the class information is attached to it. The classification tree is fully built when all the classes of the training set have been represented by leaves in the tree. The classification rules we use in HIS is expressed as:

IF ($\text{Sim}(x, m) > t$) THEN visit the Pass-Child node ELSE visit Fail-Child node, where $\text{Sim}(x, m)$ is a similarity measure between a signal pattern x and a set of important features, called the mask, m and t is a similarity cutoff value [8]. The roles of the three components of the decision tree classification rules. The feature mask component represents a set of reference characteristics that describe clusters of classes while the similarity measure is the comparison tool for comparing patterns in the abstract domain. The rule cutoff value defines class boundaries in the abstract domain. This is different from the basic pattern classification where the boundaries are defined in the input data domain.

III. GENETIC ALGORITHM IN HYBRID INTELLIGENT SYSTEM FOR PATTERN RECOGNITION

Genetic algorithms are great tool for global search and optimization but their performance is not guaranteed [1]. The goal of HIS is to complement the weaknesses of these techniques and integrate their strengths.

IF ($\text{Sim}(x, m) > t$) THEN visit the Pass-Child node ELSE visit Fail-Child node. The main component of the rule antecedent is computation of the similarity of the pattern x to a set of important features m , called a mask. In general, a feature mask is a set of features that might be composed of a partial pattern, a subset of features from more than one pattern or even a set of features that are not found in any of the training pattern set S_{train} . Obviously, the problem of selecting high quality feature masks can be formulated as a search problem over the space of all features related to a training set S_{train} . The goal of the search algorithm is to find the feature masks with the highest ability to discriminate among the members of the training set S_{train} . It is clear that the search space is infinite since we not only allow feature masks to be composed of any subsets of all features in S_{train} but also from features that do not exist in S_{train} . Our HIS implementation uses Matlab [5] due to its flexibility in system modelling and the availability of many toolboxes that

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

implement commonly used algorithms. We used a publicly available Matlab toolbox GAOT [53] due to its capability of handling real number encoding of chromosomes.

Finding Pattern Masks Using GA

Genetic algorithms are excellent at searching large spaces to find solutions with the highest performance [10]. Their high efficiency is attributed to the way they generate and evaluate many individuals making up a generation. Crossover processes help preserving the best traits through the generations. Mutation introduces a random minor perturbation to the selected individuals to allow exploration of different regions in the search space instead of allowing the search to become trapped near local extreme points. Our goal is to develop a hybrid intelligent classifier that employs the different intelligent techniques to solve pieces of the problem. The first step to apply a genetic algorithm to the problem of feature mask selection is encoding the problem in some chromosome format and defining the fitness function that guides the genetic algorithm. The second step is to determine the genetic operators and then tune them to the best performance.

GA Parameters

A genetic algorithm [10] is utilized to find the best pattern mask for each node in the classification tree. Each pattern mask is made of n features that can vary among pattern masks. We discuss below the basic constructs used for the genetic algorithm in hybrid intelligent system pattern classifiers.

- Encoding in hybrid pattern classifier is done by using strings of real-valued numbers. Each feature in the mask is represented by its attributes, i.e., start and end locations, height and area. The mask string is composed of the sequence of its encoded features.
- Initial population in hybrid pattern classifier is generated by picking m signal patterns from the pattern set associated with the node. For each pattern in m we pick the biggest n features in area and encode them to make an individual mask string.
- Fitness is very crucial to the performance of hybrid pattern classifier. There are many possibilities to measure the fitness of a pattern mask. One fitness measure based on the infomax concept used in decision trees [6]. The more evenly the rule divides the pattern set, the higher its fitness score. This will lead to a shallower classification tree and lower classification cost. Therefore we devised a fitness measure that takes into account the characteristics of the clusters generated by the pattern mask.
- Selection methods include roulette wheel selection and ranking selection. Any of these methods is applicable to pattern classifiers. Our current implementation uses roulette wheel selection.
- Crossover methods that apply to real numbers can be used in pattern classifier since the pattern masks are encoded as real number strings.
- Mutation in hybrid pattern classifier is done by using non-uniform mutation; however any of the mutation methods applicable to real number strings can be used as well.
- Termination condition in hybrid pattern classifier can be set to a certain level of mask fitness or a limit on the number of generations. Note that the termination condition can be different for different nodes in the tree.

Each component of the algorithm has different parameters that need to be set to optimize hybrid pattern classifier's performance. Selection of these parameters is based on running many tests, varying one parameter at a time to determine the value that gives the best performance. Our system design evolution process is an iterative process where each iteration includes a tuning step to optimize the added component parameters. Selection of the GA parameters fits in the tuning step. In this first iteration our focus was to utilize and tune the GA to find the best feature masks. However we also needed to define similarity cutoff values and the similarity measure.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

Cutoff values were encoded at the end of the feature mask chromosomes. The chromosomes' fitness was computed as the ratio of the smaller subset size, pass or fail subset, to the total pattern set size associated with the node for which the GA is working on finding the feature mask. In each case, the classification tree was built to have leaves each containing one class only. Table 1 shows the task assignment of hybrid pattern classifier while Table 2 shows the parameters of the test. Table 3. shows the default parameters of the GA for testing.

Hybrid Pattern Classifier	Intelligent Component
Select Feature Masks	GA
Set Similarity Cutoff	GA
Similarity Measure	Area Overlap

Table 1 Hybrid Pattern Classifier task assignment for the first iteration.

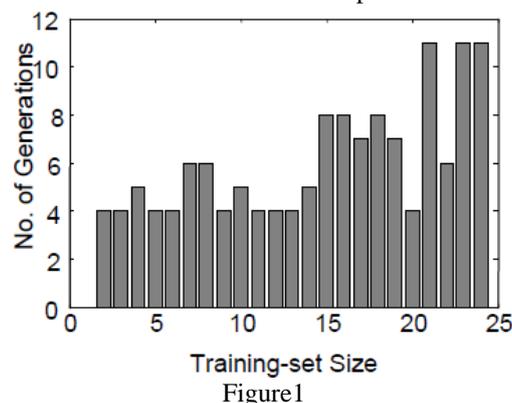
Parameter	Value
Number of patterns	2-24
Number of points per pattern	100
Maximum number of features per pattern	17

Table 2 Test set parameters for the first iteration.

GA Parameter	Default Value
Number of features per mask	5
Population Size	8
Number of Generations	8
Selection Function	Normal Geometric Selection Probability of selecting best = 0.08
Crossover Function	Arithmetic Crossover
Mutation Function	Non Uniform Mutation Shape Parameter = 3
Fitness Function	Equal size of Fail and Pass subsets
Termination Function	Maximum Number of Generations

Table 3 Default GA Parameters for testing Hybrid Pattern Classifier.

In this initial set of tests the performance of hybrid pattern classifier was measured by its ability to correctly classify each pattern in the given training set. Figures 1, 2 and 3 show the parameter values that enable the correct classification of each pattern to itself in most cases. Hashed bars denote that some patterns were not correctly classified.



International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

Show the parameter values that enable the correct classification of each pattern to itself in most cases. Hashed bars denote that some patterns were not correctly classified. Number of generations used to satisfy the performance measure for each training-set size when population size was set to 2 [9].

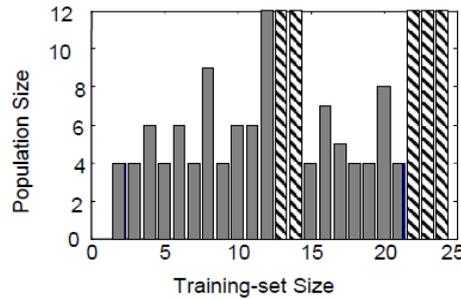


Figure 2

Show the parameter values that enable the correct classification of each pattern to itself in most cases. Hashed bars denote that some patterns were not correctly classified. Population size used to satisfy the performance measure for each training-set size when number of generations is fixed to 4 and number of features per rule to 5 [9].

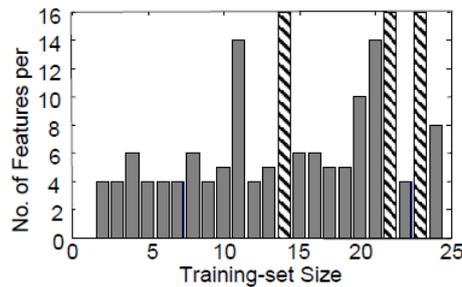


Figure 3

Show the parameter values that enable the correct classification of each pattern to itself in most cases. Hashed bars denote that some patterns were not correctly classified. Number of features per rule used to satisfy the performance measure for each training-set size when number of generations is fixed to 4 and population size to 4 [9].

The above results showed the potential of the GA for finding the feature masks. However, performance started degrading when the set size was larger than 24. Therefore we developed another method for finding the cutoff values. In this iteration we selected the similarity cutoff value as the middle point of the biggest gap of similarity distribution of the node pattern set. Table 4 shows the task assignment of hybrid pattern classifier with this change.

HISPC Task	Intelligent Component
Select Feature Masks	GA
Set Similarity Cutoff	Mid point of Biggest Gap in similarity distribution
Similarity Measure	Area overlap

Table 4 Hybrid Pattern Classifier task assignment with improved cutoff placeholder.

The next round of tests was performed on a pattern set containing 64 patterns. The goal of these tests was to tune the GA component and determine which parameters have the highest effect on the overall system performance. Table 5 shows the details of the parameters of the training and test sets.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

Parameter	Value
Number of patterns	64
Number of points per pattern	100
Maximum number of features per pattern	17

Table 5 Test set parameters for the first iteration with improved cutoff placeholder.

IV. RESULTS

To guarantee that the result has statistical significance, we ran every test case 5 times and computed the average of the 5 classifiers' performance. The purpose of the iteration tests is to quickly get a feeling for what parameters have the greatest bearing on the classification performance. Therefore, we chose the trade off to measure the classification performance as the average of 5 runs only, which is a small sample from the statistical perspective rather than paying the cost of a larger number of runs.

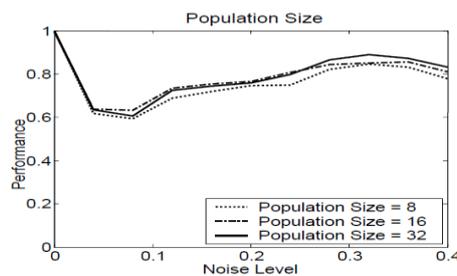


Figure 4

Shows the results of performance test when the population size parameters are 8, 16 and 32 as the average run for each parameter was 5 times which is a small sample from the perspective of statistical significance. Rather than paying cost of a larger number of runs

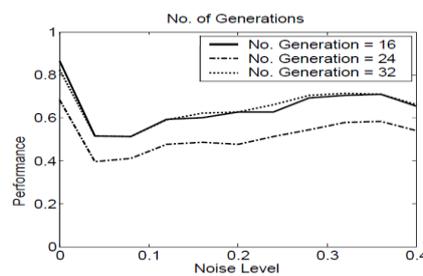


Figure 5

Result of this figure determines the number of generations the genetic algorithm would produce for each node in order to find its feature mask.

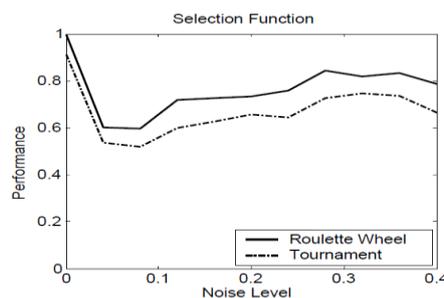


Figure 6

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

One of the widely used selection techniques is roulette wheel selection [36]. Mutation allows the search to get out of local optima and explore new areas in the search space even if they have lower fitness. Therefore, it is usually common that GAs keep the best individuals found till termination. Results of test of the roulette wheel and tournament selection functions is shown Roulette Wheel takes no parameters. For Tournament selection the number of tournaments was set to 3

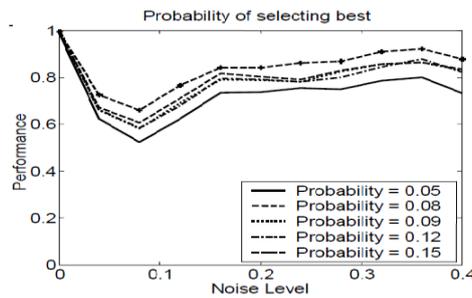


Figure 7

This selection method is ranking selection function based on the normalized geometric distribution [53]. Q defines the normalization factor. Results of testing the normalized geometric selection function with different probabilities are shown with normalized geometric selection Q = 0.05, 0.08, 0.09, 0.12, 0.15.

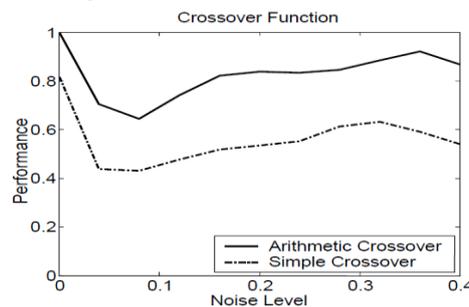


Figure 8

Arithmetic crossover takes two real strings and generates two strings by linear interpolation of the elements' values using random a parameter. Simple crossover picks a point in the two parent strings; cuts the strings then cross concatenate them to generate the children. Results of testing the arithmetic and simple crossover functions is shown

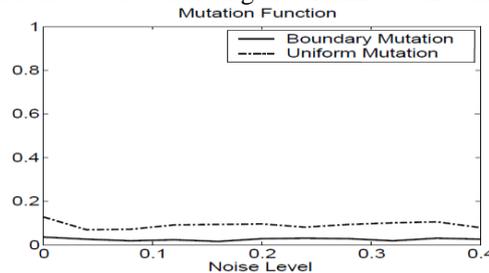


Figure 9

Boundary mutation uses uniform distribution probability to select one element of the input string then changes it to either the lower or the upper boundary value for that element. Results of testing the boundary and uniform mutation functions are shown.

The above results showed the most important parameters of the GA and their best values from among those tested. Based on these results we fixed the GA parameters so that for the following iterations test results are only affected by the added intelligent component. Table 6 shows the default GA parameters that will be used for the following Hybrid Pattern Classifier design iterations.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

GA Parameter	Default Value
Number of features per mask	5
Population Size	8
Number of Generations	8
Selection Function	Normal Geometric Selection Probability of selecting best = 0.08
Crossover Function	Arithmetic Crossover
Mutation Function	Non Uniform Mutation Shape Parameter = 3
Fitness Function	Similarity cluster quality using NN
Termination Function	Maximum Number of Generations

Table 6 GA parameters for the first iteration with improved fitness function.

V. CONCLUSION

In the course of this paper we have covered many aspects of pattern classification using hybrid intelligent systems. The paper has tried to answer questions in pattern classification, hybrid intelligent systems and genetic algorithm approach in general. The goals set of this paper has been met as shown by the various testing results. The hybrid pattern classification developed through this paper provided an excellent means for studying ways of combining different intelligent techniques. It also enabled investigation of the effects of the various parameters of each technique on the overall performance. We have also introduced the concept of feature masks that allow the classifier to define class boundaries in terms in subsets of features instead of one feature at a time. For example in regular decision trees every node usually has classification rules based on one parameter of the pattern. However, we used a pattern mask that combined a subset of features along with their location to define the class boundaries. Each technique has its effectiveness in solving part of the problem. Genetic algorithms are powerful in searching the large space of similarity rules and are very effective in setting the class boundaries cutoff values and evaluating feature mask fitness.

REFERENCES

- [1] A. Abraham and B. Nath, "Hybrid Intelligent Systems Design: A Review of a Decade of Research", IEEE Transactions on Systems, Man & Cybernetics (Part-C) August 2000 (Submitted).
- [2] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, John Wiley & Sons Inc., 2001.
- [3] J. H. Holland, Adpatations in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
- [4] J. H. Holland, K. J. Holyoak and R. E. Nisbett and P. R. Thagard, Induction: Processes of Inference, Learning and Discovery, MIT Press, Cambridge, 1986.
- [5] Matlab version 12, www.mathworks.com.
- [6] J. Quinlan, "Induction of Decision Trees", Machine Learning, vol. 1, pp. 81-106, 1986.
- [7] M. D. Vose, "The Simple Genetic Algorithm: Foundations and Theory (Complex Adaptive Systems)", MIT Press, August, 1999.
- [8] R. S. Youssif and C. N. Purdy, "Combining Genetic Algorithms and Neural Networks to Build a Signal Pattern Classifier", Proc. of Second International Conference on Hybrid Intelligent Systems, Chile, 2002.
- [9] R. S. Youssif and C. N. Purdy, "A Multistrategy Signal Pattern Classifier", 45th IEEE Midwest Symposium on Circuits and Systems, MWSCAS conference, Tulsa, Oklahoma, August 2002.
- [10] Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
- [11] R. Guiterrez-Osuna, "Pattern Analysis for Machine Olfaction: A Review", IEEE Sensors Journal, vol. 2, No. 3, pp. 189-202, 2002.
- [12] T. Back, D. B. Fogel and T. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators, Institute of Physics Publishing, 2000.