Improving the Performance of a Rule Induction System Using Genetic Algorithms

Haleh Vafaie and Kenneth De Jong
Center for Artificial Intelligence
George Mason University
Fairfax, VA 22030

Abstract

Concept acquisition is a form of inductive learning that induces general descriptions of concepts from specific instances of a given concept. AQ15 is a conceptual inductive learning program that uses feature-based examples of concepts to generate rules which describe the underlying concept. This program has been successfully applied to a wide variety of domains. We have been exploring the use of AQ15 on a difficult class of image processing problems, namely, to infer descriptions of textures from noisy examples. In this context, we find that AQ15 produces rules that are sub-optimal from two view points: 1) we need to minimize the number of features actually used for classification; and 2) we need to achieve high recognition rates with noisy data. Our multistrategy approach is to apply genetic algorithms to select the best feature subset to use in conjunction with AQ15 to achieve our goals. The proposed approach has been implemented and applied to an initial set of randomly selected texture data. Our results are encouraging and indicate significant advantages to a multistrategy approach in this domain.

1. Introduction

In recent years there has been a significant increase in research on automatic image recognition, classification and texture analysis. Many researchers are now interested in texture classification based on discriminant properties (features) of a given image. We have been exploring the use of machine learning techniques on a difficult class of image processing problems, namely, the use of rule induction methodologies (in particular, AQ15) to infer descriptions of textures from noisy examples. In this context, we find that AQ15 produces rules that are sub-optimal from two view points: 1) we need to minimize the number of features actually used for classification; and 2) we need to achieve high recognition rates with noisy data. In order to achieve our goals we have adopted a multistrategy approach involving the use of genetic algorithms to select the best feature subset to use in conjunction with AQ15.

The appropriate selection of these properties plays a crucial role in several aspects of the design of robust and feasible recognition systems. Since feature extraction is generally a costly combination of special purpose hardware and cpu-intensive computations, reducing the number of features required for classification will permit the design of recognition systems that were otherwise infeasible. In addition, by reducing the number of features (by eliminating redundant or irrelevant properties), the performance of the rule induction system itself (AQ15) can be improved as well as the classification performance of the rules produced.

2. Feature Selection

A review of previous work in this area indicates that there are two main approaches that the image processing community has taken to feature selection. One approach selects features independent of their effect on classification performance. The other approach selects features based on the overall effectiveness of the performance of the classification system.

The first approach involves transforming the original features according to procedures such as those presented by Karhunen-Loeve or Fisher to form a new set of features. Then, it selects a subset of these transformed features by choosing the first "n" transformed features where the selected subset has lower dimensionality than the original one (Dom 89). The smaller set of features is assumed to preserve most of the information provided by the original data and be more reliable because it removes redundant and noisy features (Dom 89).

The second approach directly selects a subset "d" of the available "m" features based on some effectiveness criteria, without significantly degrading the performance of the classifier system (Ichino 84a). Many researchers have adopted this method and have created their own variations on this approach. For example, Blanz, Tou and Heydorn, and Watanabe after ordering the features use different methods (such as, take the first "d", throw away the last 'm-d', Branch and Bound, etc.) to select a subset of these features (Dom 89). The main issue for this approach is how to account for dependencies between features when ordering them initially and selecting an effective subset in a later step (Dom 89).

A related strategy involves simply selecting feature subsets and evaluating their effectiveness (Dom 89). This process then requires a "criterion function" and a "search procedure" (Foroutan 85). The evaluation of feature set effectiveness has been studied by many researchers. Their solutions vary considerably and include using the Bayes probability of error, the Whitney and Stern estimate of probability of error based on the k-nearest neighbor bound (Ichino 84b), or a measure based on statistical separability (Dom 89).

All of the above mentioned techniques involve the assumption is that some a priori information (such as a probability density function) about the data set is available. However, quite frequently very little is known about the distribution function, or this property is not necessarily related to the classifier's performance and the performance must be estimated using the available data (Foroutan 85).

Also, the search procedure used in these techniques to select a subset of the given features plays an important role in the success of the approach. Exhaustively trying all the subsets is computationally prohibitive when there are a large number of features. Non-exhaustive search procedures such as sequential backward elimination and sequential forward selection pose many problems (Kittler 78). These search techniques do not allow for back tracking; therefore, after a selection has been made it is impossible to make any revisions to the search. In order to avoid a combinatorial explosion in search time, these search procedures generally do not take into consideration any inter-dependencies that may exist between the given features when choosing a subset.

Since genetic algorithms are best known for their ability to efficiently search large spaces about which little is known, and spaces involving noise, they seem to be an excellent choice for the combinatorially explosive problem of searching the space of all possible subsets of a given feature set for a minimally effective set of features for use with AQ15 to infer texture descriptions. We describe this approach in more detail to the following sections.

3. A Multistrategy Approach

The overall architecture of our system is given in Fig. 1. We assume that the low level feature extraction will be done for us, and that we will start with a feature set from which an optimal subset is to be selected. As indicated, genetic algorithms are used to explore the space of all subsets of a given feature set. Each of the selected feature subsets is evaluated (its fitness measured) by invoking AQ15 and measuring the recognition rate of the rules produced. Each of these components is described in more detail in the following sections.

3.1 Genetic Algorithms as the Search Procedure

Genetic algorithms (GAs), a form of inductive learning strategy, are adaptive search techniques initially introduced by Holland (Holland 75). Genetic algorithms derive their name from the fact that their operations are similar to the mechanics of genetic models of natural systems.

Genetic algorithms maintain a constant-sized population of individuals. The initial population is commonly a randomly generated collection of individuals representing samples of the space to be searched. Each individual is evaluated, selected and recombined with other individuals on the basis of its overall fitness with respect to the given application domain. Therefore, high performing individuals may be chosen for replication several times. This eventually leads to a population that has improved fitness with respect to the given goal.

New individuals (offspring) for the next generation are formed by using two main genetic operators, crossover and mutation. Crossover operates by randomly selecting a point in the two selected parents gene structures and exchanging the remaining segments of the parents to create new

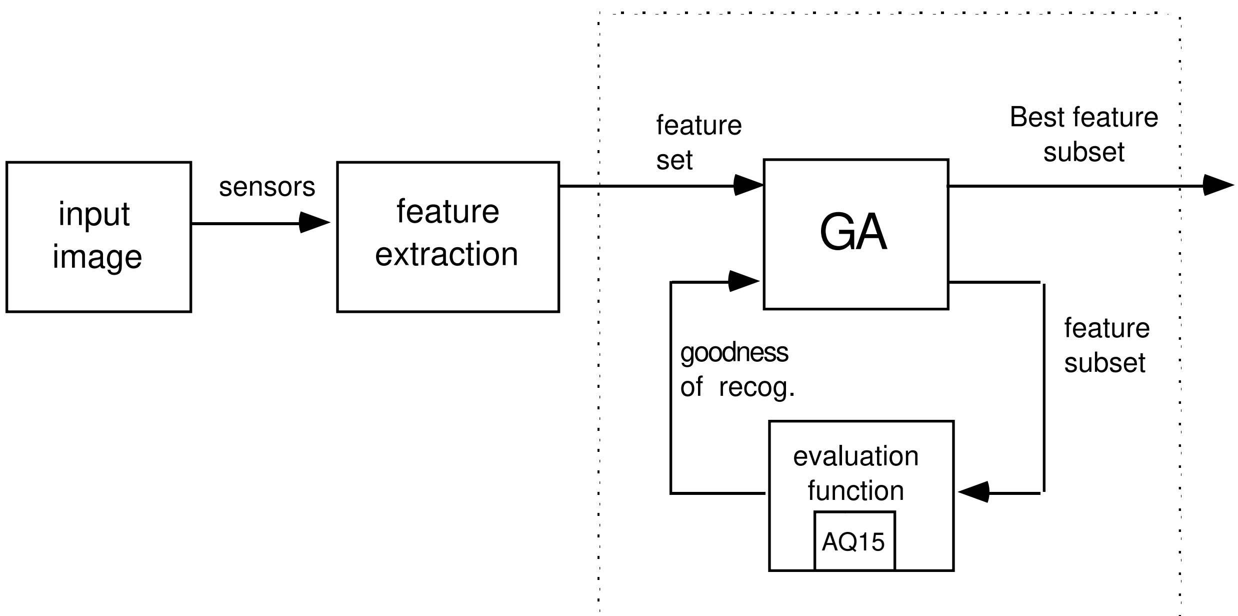

Figure 1: Block diagram of the adaptive feature selection process

offspring. Therefore, crossover combines the features of two individuals to create two similar offspring. Mutation operates by randomly changing one or more components of a selected individual. It acts as a population perturbation operator and is a means for inserting new information into the population. This operator prevents any stagnation that might occur during the search process.

Genetic algorithms have demonstrated substantial improvement over a variety of random and local search methods (De Jong 75). This is accomplished by their ability to exploit accumulating information about an initially unknown search space in order to bias subsequent search into promising subspaces (De Jong 88). Since GAs are basically a domain independent search technique, they are ideal for applications where domain knowledge and theory is difficult or impossible to provide (De Jong 88).

The main issues in applying GAs to this problem are selecting an appropriate representation and an adequate evaluation function. We discuss both of these issues in more detail in the following sections.

3.2 Representation Issues

The first step in applying GAs to the problem of feature selection is to map the search space into a representation suitable for genetic search. Since we are only interested in representing the space of all possible subsets of the given feature set, the simplest form of representation is to consider each feature in the candidate feature set as a binary gene (Holland 75). Then, each individual consists of fixed-length binary string representing some subset of the given feature set. An individual of length ‘d’ corresponds to a d-dimensional binary feature vector ‘X’, where each bit represents the elimination or inclusion of the associated feature. For example, $X_i=0$ represents elimination and $X_i=1$ indicates inclusion of the ith feature. Hence, an individual of the form <1101> represents the subset containing all but the third 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.

3.3 Evaluation function

Choosing an appropriate evaluation function is an essential step for successful application of GAs to any problem domain. Evaluation functions provide GAs with the feed-back about the fitness of each individual in the population. GAs then use this feed-back to bias the search process so as to provide an improvement in the population’s average fitness.

It is essential for the success of our system to properly and accurately measure the effectiveness of the image classification. A feature subset is evaluated based on its ability to recognize the appropriate images or proper partitioning of classes. Performance of a feature subset is measured by applying the evaluation function presented in Figure 2. The evaluation function as shown 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 reduced. This is done by removing the values for features that are not in the selected subset offeature vectors. The second step is to apply a rule induction process (AQ15) to the new reduced training data to generate the decision rules for each of the given classes in the

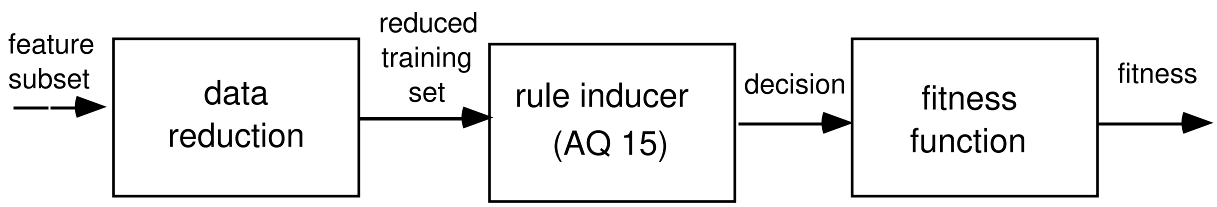

Figure 2: Selected evaluation function

training data. The last step is to evaluate the rules produced by the AQ algorithm in order to perform the classification and hence, recognition with respect to the test data. Each of these steps will be discussed in more detail.

3.2.1 The AQ algorithm

The AQ algorithm is a conceptual inductive learning methodology used to produce a complete and consistent rule-based description of classes of examples (Michalski 86, 83). The AQ methodology works by selecting an uncovered single positive event or example, called the seed, in the class for which it is producing the decision rules. This seed is then maximally generalized through the space of the given problem without satisfying any negative examples (i.e. examples describing other classes). This process continues until decision rules are found that satisfy all the positive examples and none of the negative ones.

A class description is formed by a collection of disjuncts of conjuncts describing all the training examples given for that particular class. A decision rule is then a collection of conjuncts of allowable selector (feature) values.

GEM, an AQ15 based program, provided the required rules used for classification process. This system (for more detail see (Reinke 84)) basically uses a set of parameters, feature descriptions, and a training set as input. It then uses the given parameters to direct the AQ algorithm in its process of searching for a complete and consistent class description. The following presents a simplified version of decision rules formed as output of GEM for a given class containing total of nine features which can take on integer values:

1. [Feature1=3 to 7] and [Feature4= 4 to 9] and [Feature8=1 to 10] (total:9, unique 6)

2. [Feature1=6] and [Feature4= 4 or 5 or 9] (total:5, unique 3)

3. [Feature1=4] and [Feature3= 3 to 9] and [Feature7=2 to 6] and [Feature9=1 to 8] (total:2, unique 1)

This class is described by three decision rules, where each covers a total of 9, 5, and 2 training events (examples) respectively. The term “unique” refers to the number of positive training examples that were uniquely covered by each rule.

3.2.2 Fitness function

In order to use genetic algorithms as the search procedure, it is necessary to define a fitness function which properly assesses the decision rules generated by the AQ algorithm. The fitness function must be able to discriminate between correct and incorrect classification of examples, given the AQ created rules. Finding an appropriate function is not a trivial task, due to the noisy nature of most image data. The following procedure was followed to best achieve our goal.

The fitness function takes as an input a set of feature or attribute definitions, a set of decision rules created by the AQ algorithm, and a collection of testing examples defining the feature values for each example. The fitness function then views the AQ generated rules as a form of class description that, when applied to a vector of feature values, will evaluate to a number.

The fitness function is evaluated for the given feature subset by applying the following steps. For every testing example a match score (which will be described in more detail) is evaluated for all the classification rules generated by the AQ algorithm, in order to find the rule(s) with the highest or best match. At the end of this process, if there is more than one rule having the highest match score, one rule will be selected based on the chosen conflict resolution process. This rule then represents the classification for the given testing example. If this is the appropriate classification, then the testing example has been recognized correctly. After all the testing examples have been classified, the overall fitness function will be evaluated by adding the weighted sum of the match score of all of the correct recognitions and subtracting the weighted sum of the match score of all of the incorrect recognitions, i.e.

$$F = \sum_{i=0}^{n} S_i * W_i - \sum_{j=n}^{m} S_j * W_j$$

where :

S_i is the best match score evaluated for the test example 'i'

n is the number of testing examples that were classified correctly

m is the total number of testing examples (i.e. m-n is the total number of incorrectly classified testing examples).

w_i is the weight allocated to the class associated with test example 'i'.

In order to normalize and scale the fitness function 'F' to a value acceptable for GAs, the following operations were performed:

$$\text{Fitness} = 100 - [(F / \text{total testing examples}) * 100]$$

As indiciated in the above equation, after the value of 'F' was normalized to the range [-100,100], the subtraction was performed in order to ensured that the final evaluation is always positive, as required by GAs, (i.e., fitness falls in the range [0,200]). Then, the final fitness value decreases as the classification is improved. For example, a fitness value of zero indicates that all the testing examples have been classified correctly.

There are two main parameters that define how the fitness function evaluates the performance of the decision rules on the testing examples. These are the weight associated with each classification, and the way that the degree of matching is evaluated for a given selector.

<u>weight (W):</u>

This parameter is useful when encoding user or expert's knowledge into the evaluation function. However, for our preliminary experiments all the classes were assumed to have equal weight or importance (i.e. weight=1).

<u>Match score (S):</u>

The degree of match between a rule and a testing example may be treated as strict or partial.

In strict match, the degree of consonance between a selector and a feature value of a testing example is treated as a boolean value. That is, it evaluates to zero if the example's feature value is not within the selector's specified range, and is one otherwise.

A rule's overall match score is calculated as the minimum of the degree of match for all the selectors in that rule. Therefore, for each selector in a given rule, if all the selectors match the example value, the match score is '1', otherwise it is '0'.

$$S_s = \min (S_i)$$

In the partial form of matching we allow for real valued numbers which are evaluated in the following manner. For each selector in a given rule, if the variable or feature in that selector has linear domain , then find the closest distance as its match score. This measure is evaluated using the following formula:

$$S = (1- |a_j - a_k| /n)$$

where:

a_j is the selectors closest feature value to the testing example's value a_k,

a_k is the testing example's feature value , and

n is the total number of values given for the variable or feature.

Otherwise, the domain is non-linear and the match score for a given selector is '1' when a match occurs, and is zero in other situations. The total partial match score for a given rule is evaluated by averaging all the selectors match score for the given rule.

$$S_p = \text{Average } (S_i)$$

Aside from the weight associated to each class and match score of a given testing example, conflict resolution process plays an important role in the classification step.

Conflict resolution:

For any testing example, after the appropriate match score is evaluated for all the rules generated by the AQ algorithm, the rule with best match score must be selected. However, there may be situations where more than one rule satisfies this condition. This is when the conflict resolution process plays an important role in selecting a single rule. This process is performed by selecting a rule with highest typicality, that is, a rule that covers the most training examples as specified by the total number of examples it covers (see example of section 2.2.1). However, there may be situations that all the selected rules have the same typicality. In this situation a rule is randomly selected as the one having the best match score, and hence, is assumed to have the proper classification.

4. Experimental results

In performing the experiments for this research we used GEM, an AQ15 based inductive learning system, and GENESIS, a general purpose genetic algorithm program. In order to search for an optimum subset of features, two important steps were followed. The first step was to find the optimum set of parameters for the GEM system, so that it can provide the best possible performance in terms of rule induction. After careful consideration, is was concluded that there are only a few (four) parameters with a small number of legal values that affect the AQ's classification performance. Therefore, using a genetic approach for parameter tuning with such a small space was ruled out, and instead we decided to perform a systematic search for optimal settings. This test was performed on two different type of domains in order to find a more robust parameter settings. It was concluded that both domains result in best classification performance with the same type of parameter settings. These AQ parameter settings were then used through out all our experiments.

The second step was to select appropriate parameter settings for GENESIS. For our preliminary experiments, the standard parameter settings recommended by De Jong were used: a population size=50, a mutation rate= 0.001, and a crossover rate=0.6.

The proposed approach was tested on four texture images randomly selected from

Brodatz (Brodatz 66) album of textures. These images are depicted in Figure 3. Two hundred feature vectors, each containing 18 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 then divided equally for training generation of decision rules, and testing examples.

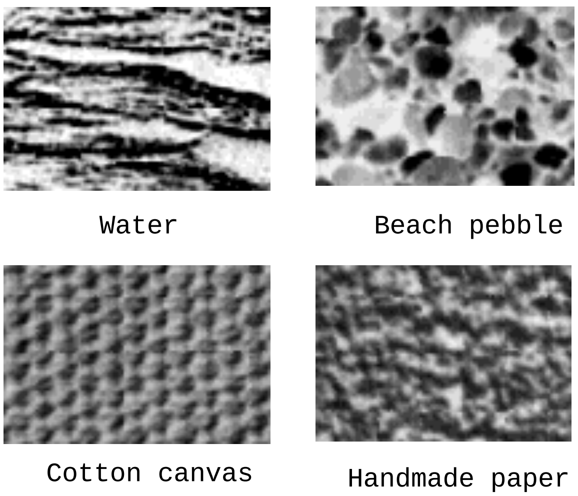

Figure 3: The texture images used in our experiments.

Our preliminary results suggested several advantages to the adopted approach as indicated in graph of figure 4 and Table 1.

The results obtained from our experiments with 80 generations is shown in Figure 4. This figure shows the changes in the evaluation function and improvement in the classification process over the generations of population.

Table 1 summarizes the results of our experiments comparing the best performance of the AQ generated classification rules to that of our multistrategy approach.

In all our experiments classification was improved with respect to that of running the AQ algorithm independent of GAs as shown in figure 4. The best performance found (Table 1), indicated an 8.0% improvement in the recognition rate of the AQ generated rules. This improvement can play an important role when dealing with crucial decision making (such as military).

The result of the feature selection process was to reduce the initial feature set consisting of 18 elements to a subset of having only 9 elements for the best performing individual. This represented a 50% reduction in the number of features.

Another important advantage of using this approach is that choosing the appropriate subset of features reduces the time required to perform the rule induction on textures or images. This is a direct result of feature selection process. In our experiments the time required for the AQ15 to generate decision rules used for classification was reduced by 36.54%. This reduction in learning time may then permit using the AQ algorithm as a means of classification for many problems that were otherwise computationally prohibitive.

It should also be noted that feature selection also reduces the complexity of AQ generated class descriptions. In this case the complexity of the generated rules (Table 1) was defined to be the number of complexes (conditions) per class.

5.0 Summary and Conclusions

The experimental results obtained indicate the power of applying a multistrategy approach to the problem of learning rules for classification of texture images. Our initial

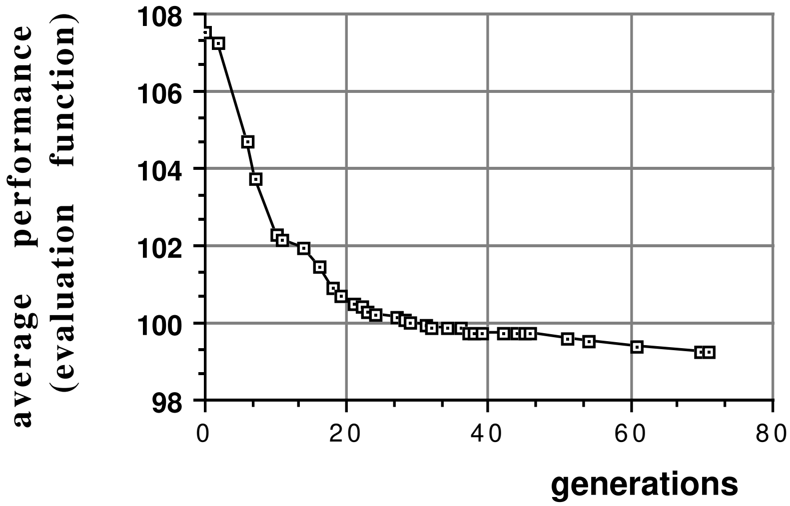

Figure 4: The evaluation function for the depicted textures

	Best performance [0,200]	Learning time (sec)	complexity (#complexes/class)
AQ15	111.5	5.2	301.5
GA-AQ	95.5	3.3	211.5

Table 1: Results of our experiments

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. This is a step towards development of real-time classification systems.

More testing is needed in order to substantiate our results. This includes examining any changes to the classification or recognition when increasing one or all of the numbers of texture classes, features, and testing and training examples. This method could also be extended to more complex problems that exist in general field of computer vision.

The classification can be further improved by using genetic algorithms to refine the AQ generated rules. This was demonstrated by Bala et al.(Bala 91).

One future direction would be to use GAs with some background knowledge. For example, including user/expert provided estimates of the relative importance of various features for texture classification could further improve speed and performance.

Another extension involves applying our multistrategy method for constructive induction. That is, a more complex set of features could be constructed by including combinations of the given features in the initial set. The feature selection process then could be applied to this set in order to find a more effective subset.

Acknowledgments

This research was done in the Artificial Intelligence Center of George Mason University. The activities of the Center are supported in part by the Defense Advanced Research Projects Agency under grant, administrated by the office of Naval Research, No. N00014-87-K-0874, and N00014-91-J-1854 and in part by the Office of Naval Research under grant No. N00014-88-K-0226, No. N00014-88-K-0397, and No. N00014-91-J-1351 . The authors also wish to thank Dr. Pachowicz and Jerzy Bala for their informative comments.

References

Achariyapaopan, T., and Childers D.G., "Optimum and near optimum Feature selection for multivariate data," *Signal Processing*, Vol 8, No. 1, North Holland Pub. Co.

Bala, J.W., De Jong, K., and Pachowicz, Peter, "Learning Noise Tolerant Classification Procedures by Integration of Inductive Learning and Genetic Algorithms," submitted to *International Workshops on Multistrategy Learning,* Harpers Ferry, W. Va., 1991.

Bala, J.W., and De Jong, K., "Generation of feature detectors for Texture Discrimination by Genetic Search," *Proceedings of 2nd Conference on Tools for Artificial Intelligence*, Herndon, Va., 1990.

Bobrowski, L., "Feature selection based on some homogeneity coefficient," *Proceedings of the Ninth International Conference on Pattern Recognition*, New York, U.S.A., 1988.

Brodatz, P. , "*A Photographic Album for Arts and Design*," Dover Publishing Co., Toronto, Canada, 1966.

Davis, L., "Adaptive Operator Probabilities in Genetic Algorithms," *Proceedings of 3rd Conference on Genetic Algorithms*, Fairfax, Va., 1989.

De Jong, K., "*Analysis of the behavior of a class of genetic adaptive systems*," Ph.D. Thesis, Department of Computer and Communications Sciences, University of Michigan, Ann Arbor, MI., 1975.

De Jong, K. , "Learning with Genetic Algorithms : An overview," *Machine Learning* Vol. 3, Kluwer Academic publishers, 1988.

Dom, B., Niblack, W., and Sheinvald, J. , "Feature selection with stochastic complexity," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Rosemont, IL., 1989.

Dontas, K., and De Jong, K.,"Discovery of Maximal Distance Codes Using Genetic Algorithms," *Proceedings of 2nd Conference* *on Tools for Artificial Intelligence*, Herndon, Va., 1990.

Foroutan, I., and Sklansky, J. , "Feature selection for piece wise linear classifiers," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1985.

Holland, J. H. , "*Adaptation in Natural and Artificial Systems*," University of Michigan Press, Ann Arbor, MI., 1975.

Ichino, M., and Sklansky, J. , "Optimum Feature selection by zero-one Integer Programming," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 14, No. 5, 1984a.

Ichino, M., and Sklansky, J. , "Feature selection for linear classifiers," *Seventh International Conference on Pattern Recognition,* Montreal, Canada, Vol. 1, 1984b.

Kittler, J. ," Feature set search algorithms," in *Pattern Recognition and Signal Processing*, C.H. Chen, Ed., Sijthoff and Noordhoff, The Netherlands, 1978.

Michalski, R.S., Mozetic, I., Hong, J.R., and Lavrac, N., "The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *AAAI*, 1986.

Michalski, R.S., and Stepp R. E., "Learning from Observation: Conceptual clustering," *Machine Learning: An Artificial Intelligence Approach,* R. S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), Tioga, Palo Alto, CA, 1983.

Montana, D. J. , "Empirical Learning Using Rule Threshold Optimization for Detection of Events in Synthetic Images," *Machine Learning* Vol. 5, No. 4, Kluwer Academic Publishers, 1990.

Reinke, R.E. , "*Knowledge Acquisition and Refinement Tools for the Advise Meta-Expert System*," Masters thesis, University of Illinois at Urbana-Champaign, 1984.