To Appear in the Special Issue of Evoluationary Computation
- Evolution, Learning, and Instinct: 100 Years of the Baldwin Effect

Using Learning to Facilitate the Evolution of Features for Recognizing Visual Concepts

J. Bala

Datamat Systems Research, Inc
8260 Greensboro Drive
Suite 255
McLean, VA 22102
jbala@dsri.com

K. De Jong, J. Huang, H. Vafaie, and H. Wechsler
Department of Computer Science
School of Information Technology and Engineering
George Mason University
Fairfax, VA 22030
{kdejong, rhuang, hvafaie, wechsler}@cs.gmu.edu

Abstract: This paper describes a hybrid methodology that integrates genetic algorithms and decision tree learning in order to evolve useful subsets of discriminatory features for recognizing complex visual concepts. A genetic algorithm (GA) is used to search the space of all possible subsets of a large set of candidate discrimination features. Candidate feature subsets are evaluated by using C4.5, a decision-tree learning algorithm, to produce a decision tree based on the given features using a limited amount of training data. The classification performance of the resulting decision tree on unseen testing data is used as the fitness of the underlying feature subset. Experimental results are presented to show how increasing the amount of learning significantly improves feature set evolution for difficult visual recognition problems involving satellite and facial image data. In addition, we also report on the extent to which other more subtle aspects of the Baldwin effect are exhibited by the system.

Key Words: Evolutionary computation, Baldwin effect, decision trees, feature selection, hybrid systems, induction, pattern recognition.

1. Introduction

Pattern recognition or classification, a difficult but fundamental task for intelligent systems, depends heavily on the particular choice of features used by the classifier. One usually starts with a given set of features and then attempts to derive an optimal subset of features leading to high classification performance. There are two general approaches to identifying optimal subsets of features: 1) strategies which evaluate subsets using abstract measures felt to be relevant to important properties of good feature sets (such as orthogonality, information content, low variance, etc.), and 2) strategies which involve actually building a classifier from the feature subset and evaluating its performance on actual classification tasks. The first approach is generally a much less expensive process, but may result in suboptimal performance if the abstract measures do not correlate well with actual performance. The second approach should produce better classification performance in theory, but is frequently suboptimal in practice because the cost of building and testing classifiers prohibits any kind of systematic evaluation of feature subsets.

These issues are very clearly in evidence when one attempts to develop classifiers for complex, real world image classification problems such as determining land usage patterns from Landsat images or locating important landmark features such as eyes and nose in facial images. In such cases there are generally a large number of candidate features which could be extracted from the low level (noisy) image data for use in classification. However, finding good subsets of features which can be used to build effective classifiers is a difficult, resource intensive process.

In this paper we describe a novel approach that we have developed to address some of these difficulties. We use a genetic algorithm (GA) to search the space of all possible subsets of a user-provided set of candidate features. An important component of the fitness of a feature set is determined by using C4.5, a tree induction algorithm (Quinlan, 1986), to construct a classifier in the form of decision tree from the given features and a set of sample data, and then testing the performance of the classifier on previously unseen data.

The resulting hybrid system in which the GA searches the genotype space of feature subsets and C4.5 explores the phenotype space of decision trees via inductive learning is proving to be surprisingly effective in identifying good feature sets as well as effective classifiers based on those features. In addition this system provides new evidence that learning can facilitate evolution in subtle and indirect ways which biologists refer to as the Baldwin effect.

After summarizing related work in section 2, we describe the system architecture in more detail in section 3. In section 4 we discuss the experimental methodology used to evaluate its effectiveness, and summarize the data obtained on two difficult visual classification problems in section 5. In section 6 we discuss various aspects of the Baldwin effect in evidence. We briefly note some additional evidence of the robustness of this approach in section 7, and provide summary conclusions in section 8.

2. Background

Any object or pattern that has to be recognized and/or classified must possess a number of discriminatory properties or features. The first step in any recognition process, performed either by a machine or by a human being, is to choose candidate discriminatory features and evaluate them for their usefulness. Feature selection in pattern recognition involves the derivation of salient features from the raw input data in order to reduce the amount of data used for classification and simultaneously provide enhanced discriminatory power. The number of features needed to successfully perform a given classification task depends on the discriminatory qualities of the selected features.

The selection of an appropriate set of features is one of the most difficult tasks in the design of pattern classification system. At the lowest level, the raw feature data is derived from noisy sensor data (such as spectral properties) the characteristics of which are complex and difficult to characterize. In addition, there is considerable interaction among low level features which must be identified and exploited. The typical number of possible features, however, is so large as to prohibit any systematic exploration of all but a few possible interaction types (e.g., pairwise interactions). In addition, any sort of performance oriented evaluation of feature subsets involves building and testing the associated classifier, resulting in additional overhead costs.

As a consequence, a fairly standard approach is to attempt to pre-select a subset of features using some abstract measures felt to be relevant to important properties of good feature sets (such as orthogonality, information content, low variance, etc.). This approach has been recently dubbed the "filter" approach (Kohavi and John, 1995) and is generally much less resource intensive than building and testing the associated classifiers, but may result in suboptimal performance if the abstract measures do not correlate well with actual performance.

Filter approaches vary considerably in how they search for good feature subsets. Since there are in general 2^N possible candidate subsets of N features, problems involving large numbers of candidate features cannot be handled by any form of systematic search. A standard technique for avoiding this combinatorial explosion simply ranks the features according to some criteria involving orthogonality, information content, low variance, etc., and then deletes lower ranked features. For difficult image classification tasks simple rank selection of features generally results in suboptimal classification performance since non-linear interactions among features are ignored and criteria like orthogonality, etc. are related to but not sufficient to guarantee good classification performance.

The problem of identifying and exploiting non-linear interactions among features can be addressed via the use of more sophisticated search techniques such as genetic algorithms (GAs) which provide efficient heuristic methods for searching large spaces (e.g., Vafaie and De Jong, 1994). It is still difficult, however, to develop abstract feature space measures of optimality which guarantee optimality in classification performance. In practice this is best achieved by including some form of performance evaluation of the feature subsets while searching for good subsets. This approach, recently dubbed the "wrapper" approach (Kohavi and John, 1995), typically involves building a classifier based on the feature subset being evaluated, and using the performance of the classifier as a component of the overall evaluation. This approach should produce better classification performance than the filter approaches, but adds considerable overhead to an already expensive search process. This in turn usually restricts the number of alternative feature subsets one can afford to evaluate, and thus may also produce suboptimal results.

Focusing purely on classification performance ignores another important issue, namely, the need to minimize the number of features used for classification. In image processing this shows up in terms of the need to minimize the overhead of extracting features from raw imagery data during the preprocessing stage. Practical solutions require balancing the need for accuracy with the need for cost effective implementations.

To address these issues we have developed a hybrid "filter + wrapper" approach (Figure 1) which uses a GA to search the space of possible feature subsets, and which evaluates feature subsets on the basis of performance, information content, and cost. Performance evaluation is obtained by using C4.5, an efficient decision tree induction algorithm, to produce a decision tree classifier based on the given feature subset, and then measure its classification accuracy on unseen images. Information content is estimated via an information theoretic calculation (see section 3 for details), and cost is estimated in terms of the number of features in a feature subset. Experimental results indicate that this approach significantly improves feature set evolution for difficult visual recognition problems involving satellite and facial image data.

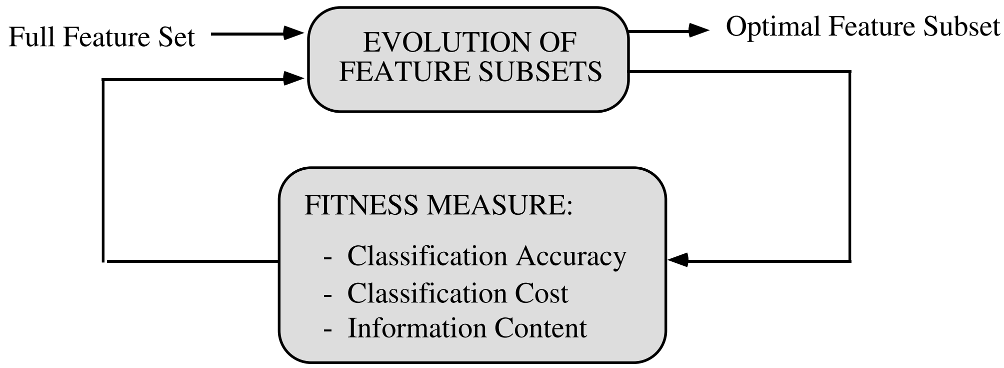


Figure 1. A Hybrid Approach for Feature Selection

Systems that employ several strategies have been shown to offer significant advantages over single-strategy systems in other contexts. Examples of such systems include ensembles of neural networks and decision trees (Gutta and Wechsler, 1995), genetic algorithms and neural networks (Gruau and Whitley, 1993), and genetic algorithms and rule-based systems (Bala et al, 1994; Bala and Wechsler, 1995; Vafaie and De Jong, 1994).

The integration of genetic algorithms and decision tree learning described in this paper is designed to allow evolution and learning to work synergistically in a variety of ways. At one level evolution (genotype learning) can be viewed as a coarsely grained global search strategy that only has to get close to the goal. Learning at the phenotype level can then be viewed as a more finely grained, local search strategy for tuning behavior (Mühlenbein and Kinderman, 1989). Although Darwinian theory does not allow for the inheritance of acquired characteristics (Lamarckian evolution), learning (acquired behaviors) can still influence the course of evolution in subtle and interesting ways which biologists refer to as the Baldwin effect. Understanding the computational implications of such effects is an active area of research (Whitley *et al*, 1994). The (computational) Baldwin effect assumes that local search is employed to estimate the fitness of (chromosome) strings, but that the acquired improvements do not change the genetic encoding of the individual. Rather, genetic changes are affected indirectly via selection due to the enhanced fitness feedback provided by learning. Computationally, the genotype can be viewed as establishing a context or bias in which learning takes place (Turney, 1996). Evolution of the genotype corresponds to a bias shift due to previous learning, resulting in new contexts which facilitate future learning.

Hinton and Nowlan (1987) were among the first to consider computational ways in which evolution of strings and the learning of individuals might interact. Many additional examples can be found in Belew and Mitchell (1996) and Balakrishnan and Honavar (1995). Recently, Bala et al (1995, 1996), Terano and Ishino (1995), and Turney (1995) have addressed the same issue in the context of data analysis for pattern classification, marketing decisions on noisy questionnaire data, and medical data, respectively. As an indication of the wide applicability of such an approach, the system developed by Bala et al (1996) expands on the research done by Johnson et al (1994) within the framework of behavior-based AI in which pattern classifiers are viewed as autonomous visual agents navigating on facial landscapes.

3. Hybrid System Architecture

The integration of genetic algorithms and inductive decision tree learning for optimal feature set selection is achieved via the hybrid architecture illustrated in Figure 2. The overall goal of the system is to find a small number of features which can be used to achieve high classification accuracy on difficult visual concepts. The underlying hypothesis on which this architecture is based is that evolution can be significantly facilitated by learning.

The GA module explores the space of all possible feature subsets defined over some (large) user-provided feature set with the goal of finding small feature subsets that can be used to achieve the best classification performance on a given task.

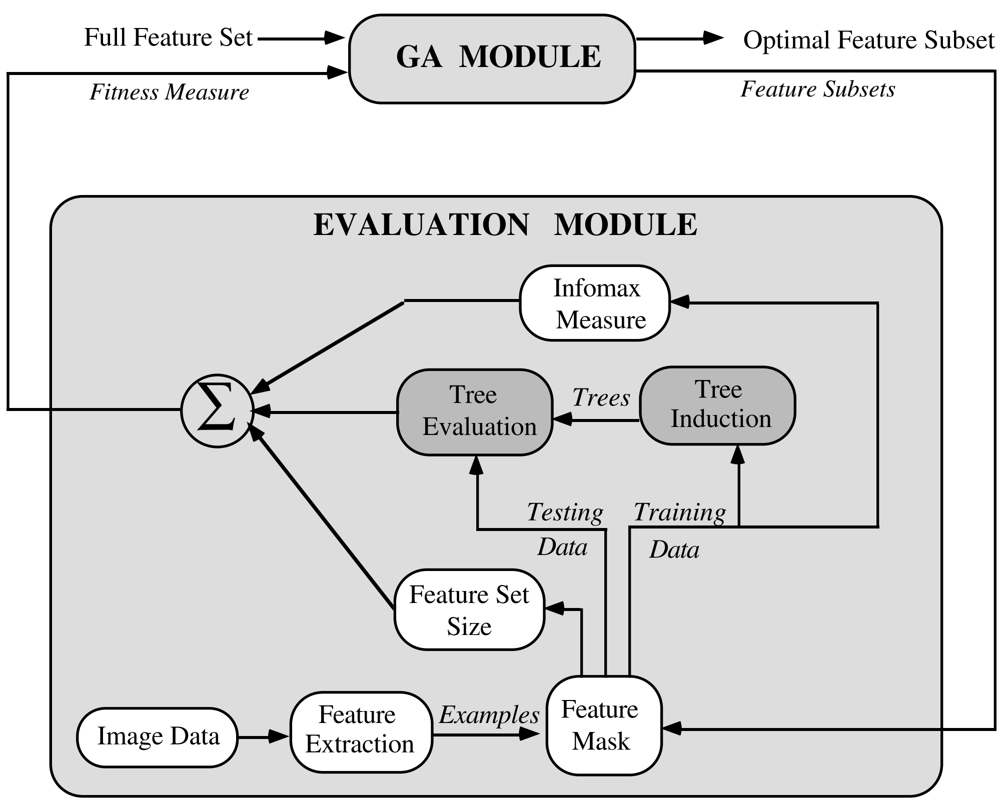


Figure 2. Hybrid Architecture for Evolution Enhanced by Learning

It is the job of the evaluation module to provide fitness feedback for each candidate feature subset *F* produced by the GA module. This fitness feedback is expressed as a function to be maximized and consists of the sum of three independent measures:

$$Fitness(F) = Inf(F) - Cost(F) + Acc(F) \tag{1}$$

The first term *Inf(F)* is an information theoretic measure based on a standard technique in the literature in which the discriminatory power of each feature is estimated independently using an entropy measure *E*. In our case the extracted feature values were all discrete and so the entropy is given by:

$$E(f) = \sum_{k=1}^{n} \sum_{i=1}^{m_j} \left[-x_{i,k}^{+} \log_2\left(\frac{x_{i,k}^{+}}{x_{i,k}^{+} + x_{i,k}^{-}} \right) - x_{i,k}^{-} \log_2\left(\frac{x_{i,k}^{-}}{x_{i,k}^{+} + x_{i,k}^{-}} \right) \right] \tag{2}$$

where n is the number of classes and m_j is the number of distinct values that feature *f* can take on. $x_{i,k}^{+}$ is the number of positive examples in class k for which feature *f* takes on its i^{th} value. Similarly $x_{i,k}^{-}$ is the number of negative examples in class k for which feature *f* takes on its i^{th} value.

Features with higher *E* values provide better discrimination among the classes. The information content (infomax) of a particular feature subset *F* is simply the sum of the *E* values of the individual features in the subset:

$$Inf(F) = \sum_{f \in F} E(f) \tag{3}$$

However, simply maximizing the information content of feature subsets would lead to the inclusion of most (if not all) of the available features. In practice this must be balanced against the cost of extracting features from the raw imagery data. Although in general different features involve different extraction costs, we use a simple measure of cost which is directly proportional to the number of features in the feature subset:

$$Cost(F) = cardinality(F) \tag{4}$$

With just cost and infomax feedback the system would represent a GA-based filter approach. However, the final term in the fitness function, *Acc(F)*, is a measure of the classification accuracy of the feature subset F, and is obtained as indicated by the shaded boxes in Figure 2. C4.5 is used to induce a decision tree based on these features using a randomly selected part of the set of training data. The classification performance of the induced decision tree is measured by applying it to the internal testing data, and the observed classification accuracy is used as the measure of performance.

Each feature subset selected by the GA module for evaluation is used by the feature masking module to preprocess the training data so that it is described only in terms of those features. The feature masking module also randomly divides the learning examples into the training and testing data, respectively. The training data is used by the tree induction module to induce trees and is also used to calculate the infomax measure, while the testing data is used by the tree evaluation module to measure the classification accuracy of the induced trees.

We selected C4.5 for the decision tree learning algorithm for several reasons. In addition to being fast and widely used, C4.5 itself uses entropy as an information measure during tree derivation. This fits well conceptually with our use of entropy in the infomax measure. Self-organization in perceptual networks and the development of receptive fields has been shown to be driven by such a principle. As an example, Linsker (1988) has reported that a perceptual system can develop the ability to recognize relevant features of its environment using the infomax principle.

The C4.5 algorithm, in an iterative fashion, determines the feature which is most discriminatory and then it dichotomizes (splits) the data into classes categorized by this feature. The next significant feature of each of the subsets is then used to further partition them and the process is repeated recursively until each of the subsets contain only one kind of labeled data. The resulting structure is called a decision tree, where nodes stand for feature discrimination tests, while their exit branches stand for those subclasses of labeled examples satisfying the test. An unknown example is classified by starting at the root of the tree, performing the sequential tests and following the corresponding branches until a leaf (terminal node) is reached indicating that some class has been decided on. Decision trees are disjunctive, since each branch leaving a decision node corresponds to a separate disjunctive case. After decision trees are constructed a tree pruning mechanism is invoked which helps to improve the generality of the trees and reduces the effects of noise in the training data. It discards some of the unimportant sub-trees and retains those covering the largest number of examples. The remaining tree thus provides a more general description of the learned concept.

4. Experimental Methodology

In order to test our ideas the hybrid architecture described in the previous section has been implemented using standard, readily available software packages with default parameter settings. For the GA module we used GENESIS 4.5 (Grefenstette, 1991) with a constant population size of 50, a crossover rate 0.6 and a mutation rate 0.001. The search space of all possible subsets of a given set of N features was represented as a fixed length bit string of N bits with "1" representing the presence of a feature in the subset and "0" representing the absence of that feature. For the tree induction module we used C4.5 (Quinlan, 1993) with the pruning confidence level set to default 25%. These choices remained constant across all experiments; no attempt was made to tune the software to particular problems.

Our testing methodology operated as follows. Since GAs are stochastic algorithms all of the experiments involving a GA component were run a minimum of 10 times in order to provide reliable estimates of average behavior. For each image classification problem we divided the sample images provided into two sets. One third of the images were set aside to be used as unseen data for final evaluation of the feature sets produced. The remaining two thirds were made available for use by our hybrid architecture to identify good feature sets. This data was subdivided into two subsets in which 40% is available for training and 60% for testing by the learning module. Since the performance of C4.5 can be affected somewhat by the particular choice of data for the training set, C4.5 was run 10 times for each feature set with randomly generated 40-60 splits. The results from the best of these 10 runs were used as the estimate of classification accuracy. Because the underlying image data was noisy, returning the best rather than the average accuracy over 10 runs proved to be more effective.

With machine learning techniques in general there is always a danger of "overfitting" in the sense that a classifier is induced which gives highly accurate results on the training data, but performs at a much lower level when presented with previously unseen data. To avoid this problem, we subjected each of the feature subsets produced to a final independent evaluation. For each final feature subset output by our system as "the best subset found", we used C4.5 to produce the associated classifier and then tested it on the remaining one third of the images (described above) that had been set aside to be used as unseen data for final evaluation. All of the classification accuracy results reported in the experimental section are based on the results from the unseen testing data rather than the training data.

There were several objectives to our experimental study. The first was to assess the ability of our hybrid architecture to identify feature sets which result in good classification performance. A second goal was to understand more clearly the role that learning played in this process. A final goal was to understand the extent to which various aspects of the Baldwin effect were present.

To achieve these objectives we designed a series of experiments involving two difficult image classification problems, one involving satellite images and one involving facial images. For each of these problems we first performed the following two baseline experiments to serve as a point of comparison for our hybrid system:

1. Unfiltered C4.5

This experiment represents the standard situation of learning without evolution. In this case C4.5 is given the full set of features and run on the training data. The decision tree produced is then evaluated on the unseen testing data. The accuracy of the decision tree produced as well as the number of features actually used by C4.5 are recorded.

2. C4.5 with an Evolutionary Filter and No Wrapper

In this experiment the GA module is used as a filter to search for good feature subsets using only infomax and cost as the fitness measure:

$$Fitness(F) = Inf(F) - Cost(F)$$

The final feature subset produced by this evolutionary filter is given to C4.5 along with the training data. The resulting decision tree is then evaluated on the unseen testing data. The accuracy of the decision tree produced as well as the number of features actually used by C4.5 are recorded. This configuration gives us a somewhat better baseline to study the behavior of our system.

We then ran two experiments involving our hybrid system in order to understand better the effects of having evolution and learning interact in a tightly coupled manner:

1. C4.5 with an Evolutionary Wrapper and Filter

This experiment provides the data for evaluation of the standard configuration of our hybrid system in which evolution and learning interacted in a tightly coupled manner. For each feature subset F produced by the GA module, C4.5 is invoked in order to compute *Acc(F)* as described in section 3 and used along with infomax and cost as the fitness feedback:

$$Fitness(F) = Inf(F) - Cost(F) + Acc(F)$$

The final decision tree produced by this evolutionary wrapper is evaluated on the unseen testing data. The accuracy of the final decision tree as well as the number of features actually used are recorded.

2. Reduced C4.5 with an Evolutionary Wrapper and Filter

We were also interested to know to what extent the *amount* of learning affects the ability of our hybrid system to evolve good feature sets. To assess this we ran an additional experiment which differed from the previous experiment in that only 50% of the available training data is provided to C4.5 when inducing the decision tree associated with a given feature set. More precisely:

$$Fitness(F) = Inf(F) - Cost(F) + Reduced\ Acc(F)$$

These experiments were performed on two quite different image classification problems, one involving satellite images and one involving facial images. The results are presented in the next section and illustrate nicely the symbiotic relationship between evolution and learning. In addition trace information about the evolutionary and learning components was collected in order to assess the presence of various aspects of the Baldwin effect. These results are presented in section 6.

5. Experimental Results

5.1 Satellite Data

The satellite (LANDSAT) image database is summarized in Table 1 and consists of the multi-spectral values of pixels in 3x3 neighborhoods, along with a classification associated with the central pixel in each neighborhood which corresponds to actual land use categories of the scenes represented. The goal of this set of experiments is to find good feature sets capable of being used to correctly classify land use in such images.

Characteristic	Description
Number of Examples	Learning set consists of 4435 examples and test set consists of 2000 examples
Number of Features	4 spectral bands * 9 pixels in neighborhood yield 36 features
Feature Values	Numerical, range 0 to 255
Number of Classes	6 decision classes: 1 to 6

Table 1. Satellite Data Characteristics

One frame of LANDSAT imagery consists of four digital images of the same scene in different spectral bands (see Figure 3). Two of these images are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and the other two images are in the (near) infra-red. Each pixel is a 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80mx80m. Each image contains 2340x3380 such pixels.

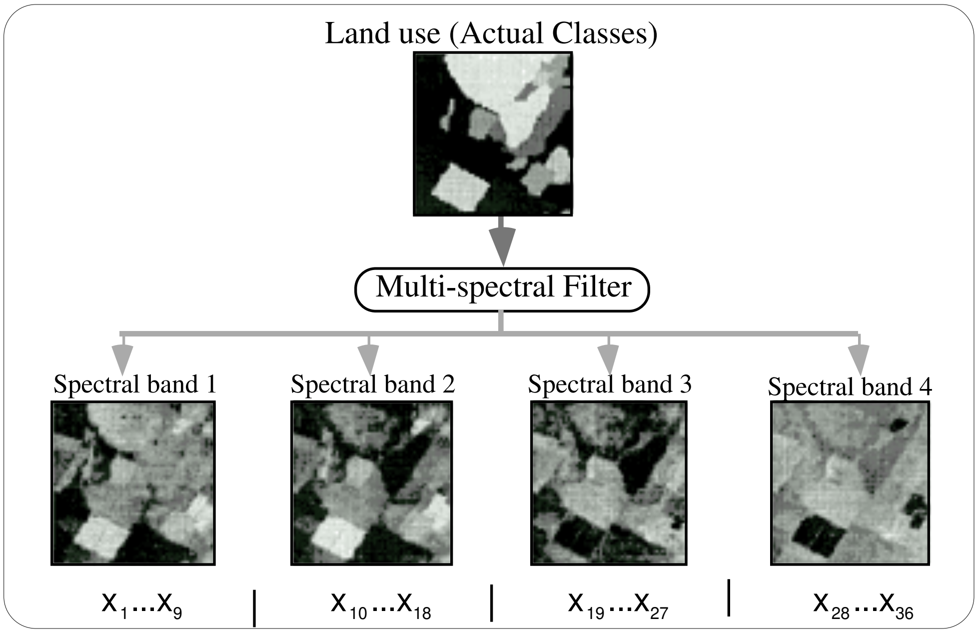


Figure 3. Four Spectral Image Bands of Satellite Data

The results of the experiments involving the satellite data are presented in Table 2. The data from experiments involving GAs represents average values with standard deviations in parentheses. The error rate data represents the classification accuracy of the corresponding C4.5 produced tree as measured on the unseen images held in reserve for final testing. The remaining two columns show the number of features in the feature subsets given to C4.5 when constructing the decision tree and the actual number of features present in the decision tree C4.5 produced.

The results in Table 2 show quite clearly the advantages of a system in which learning and evolution interact symbiotically. The wrapper approach results in a significant decrease in both classification error and in cost as measured by the number of features used. The final experiment involving a reduced amount of training data illustrates an interesting tradeoff between cost and accuracy that the system discovered.

Experimental setup	**Error Rate**	**Number of Features Available to C4.5**	**Number of Features Used by C4.5**
Unfiltered C4.5	19.6%	36	34
C4.5 with a GA Filter	21.5% (3.58)	29.7 (4.6)	29.0 (3.5)
C4.5 with GA Wrapper	16.7% (0.29)	19.7 (3.1)	19.3 (2.5)
Reduced C4.5 with GA Wrapper	18.3% (0.46)	14.0 (2.6)	14.0 (2.7)

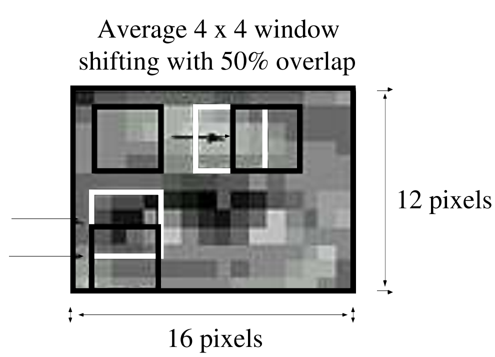
Table 2. Experimental Results for Satellite Data

One of the interesting things we observed in comparing the GA Filter and GA Wrapper experiments was that, in addition to producing better final results, the GA Wrapper system was also converging much more quickly in the sense that far fewer trials were required to locate feature subsets of equivalent accuracy, thus supporting our intuition that integrated learning accelerates evolutionary convergence.

5.2 Face Data

The ability to detect salient facial landmarks is an important component of any face recognition system. Among the many facial landmarks available it appears that the eyes play the most important role in both face recognition and social interaction. Detecting the eyes serves first of all an important role in face normalization and thus facilitates further localization of the other facial landmarks. It is eye detection that allows one to focus attention on salient facial configurations, to filter out structural noise, and to achieve eventual face recognition. The difficulty of identifying good feature sets for eye detection make this an excellent and extremely challenging problem for our hybrid system.

The face data used in these experiments comes from the FERET data base [Gutta et al, 1995]. 102 eye and 102 other facial region examples are made available during learning while 52 eyes and 52 other facial region examples are used as test data in our experiments. The eye images or the images of other facial regions, of resolution 16 by 12 (column by row) pixels, are cut from face images of size 64x72 pixels, and 105 features are produced for each example. The configuration of those features is shown in Figure 4.



x_1 to x_{35}: Average of a 4 x 4 window with 50% overlap
x_{36} to x_{70}: Standard deviation of a 4 x 4 window with 50% overlap
x_{71} to x_{105}: Entropy of a 4 x 4 window with 50% overlap

Figure 4. Extraction of Face Features

The results of the experiments involving the face data are presented in Table 3 and are much the same as those obtained on the satellite data.

Experimental Setup	Error Rate	Number of Features Available to C4.5	Number of Features Used by C4.5
Unfiltered C4.5	38.5%	105	18
C4.5 with GA Filter	40.6% (2.21)	84.7 (4.2)	9.3 (0.6)
C4.5 with GA Wrapper	30.4% (4.79)	45.3 (3.8)	8.7 (0.6)
Reduced C4.5 with GA Wrapper	36.8% (1.31)	47.3 (3.2)	10.0 (1.0)

Table 3. Experimental Results for Face Data

6. Observable Aspects of the Baldwin Effect

It is clear from both the biological and the evolutionary computation literature that various aspects of the Baldwin effect are difficult to define precisely and even more difficult to measure. On the biological side these effects are frequently described as a way of explaining how learned traits can show up as inherited traits in future generations without having to adopt a Lamarckian viewpoint (in which acquired traits lead to immediate genome changes which are then passed on to the next generation).

Computationally, this issue arises in the design of systems in which the genome serves as the starting point and context for a local search algorithm (typically hill climbing) designed to find points nearby with even better fitness. When such individuals are found, one can pass back both the new fitness and the new genome in a Lamarckian manner or just the new fitness. In the latter case the effect of discovering and returning a higher fitness results in more offspring potential for the starting point genome. Since offspring reside near their parents with high probability, over several generations the (Baldwin) effect would accelerate convergence toward the more fit genomes discovered by local search without "back coding" them in the genome.

In our system the Baldwin effect could show up in several different forms. Our genomes are simply bit strings representing a feature subset which defines the set of features that could be used for classification. C4.5 uses that feature set together with some training data to build an effective classifier. As we have seen the classifiers produced using both evolution and learning are consistently more fit than those without learning. In general the actual number of features in the classifier is significantly less than that specified by the genome. This information is *not* back-coded into the genome; only the improved fitness is returned.

As we have seen in the previous sections, one effect of learning is to significantly accelerate convergence to feature sets displaying higher fitness. A more subtle effect would be that the features actually used by high performance classifiers (a subset of those specified by the genome) are those which dominate the genomes in later generations even though there was no back-coding of this information. Another subtle effect would be that the number of unused features in the genome declines over time.

We have instrumented our system to produce trace data as a means of detecting such effects. Figure 5 and Table 4 provide some preliminary evidence of such effects. The data is trace information from the face recognition experiment presented in section 5.2. In this case the genome was 105 bits long. The shaded areas represent the subset of features to be evaluated from the initial 105 user-supplied features. Figure 5 illustrates the time evolution of the genome of the best individual in the population during a single run. Each row represents the makeup of the best individual's genome after *t* trials (births). To keep the size of Figure 5 manageable only the rows corresponding to values of *t* which are multiples of 100 are displayed. Figure 5 also shows the corresponding time evolution of the actual subset of features chosen by C4.5 to be used in the classifier. Although not conclusive, there is a fairly clear pattern of C4.5-selected features dominating the genome in future generations.

Similarly, Table 4 shows the continuous decrease in the size of the feature subsets being evaluated by the GA module, and a corresponding decrease in the number of features not used by C4.5. The data is clearly preliminary but quite suggestive of the presence of such subtle effects.

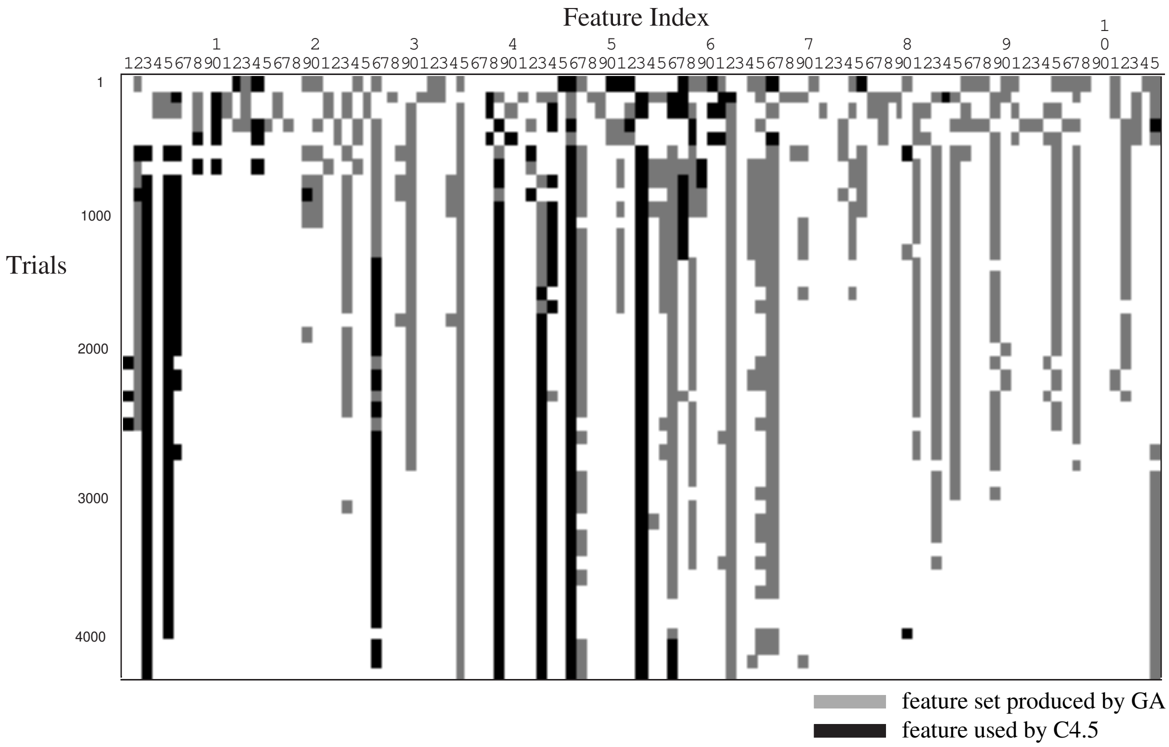


Figure 5. The Time Evolution of the Genome for Face Data

Trials	Average feature subset size produced by GA	Number of features used by C4.5
1-500	39.32	8.62
501-1000	32.74	8.25
1001-1500	29.61	7.80
1501-2000	24.79	7.42
2001-2500	18.90	7.25
2501-3000	16.35	7.00
3001-3500	13.83	7.00
3501-4000	11.43	6.67

Table 4. The Time Evolution of Feature Subsets on the Face Data

7. Feature Set Robustness

Although our experience has been to the contrary, one possible concern regarding the tightly coupled role that C4.5 plays in our system is that the resulting feature sets are brittle in the sense that they work well with C4.5 but not with other classification techniques. To get a more quantitative sense of this we tested the robustness of several of the better feature sets by using them with a standard K nearest neighbor algorithm (KNN) to construct a quite different classifier from those produced by C4.5, and measured the KNN classification accuracy on the unseen data as well.

Tables 5 and 6 provide initial evidence that in fact the feature sets are quite robust in the sense that they show similar improvements in classification accuracy when used with KNN.

KNN Experiments	**Ave. Error Rate**	**Standard Deviation**
GA Filter	15.65%	2.07%
GA Wrapper	13.14%	0.38%

Table 5. Experimental Results for Satellite Data

KNN Experiments	**Ave. Error Rate**	**Standard Deviation**
GA Filter	46.96%	1.92%
GA Wrapper	43.19%	0.96%

Table 6. Experimental Results for Face Data

8. Conclusions

We have introduced in this paper a hybrid methodology that integrates genetic algorithms and decision tree learning in order to evolve useful subsets of discriminatory features for classifying complex visual concepts. The experimental results reported, using both satellite and facial image data, indicate that learning does indeed help evolution in several important ways. The error rates on the underlying classification tasks were observed to decrease significantly when learning and evolution are allowed to dynamically interact. In addition, learning was observed to speed up the rate of evolutionary convergence.

These experiments also provided some preliminary evidence of the presence of more subtle aspects of the interaction of learning and evolution. Those features actually used by C4.5 early in the evolutionary process were seen to dominate the genome in future generators even though there was no Lamarckian "back coding" of this information.

The resulting feature sets also appear to be robust in the sense that they can be used as the basis for building effective classifiers using other approaches than simple C4.5 decision trees.

As a consequence we are quite optimistic about the synergistic role evolution and learning can play in general, and the usefulness of such systems for evolving effective feature sets and their corresponding classifiers. We are currently extending these techniques to even more challenging visual classification problems.

Acknowledgments

The research reported here was supported in part by US Army Research Lab under contract DAAL01-93-K-0099. We also wish to thank the anonymous referees for their thoughtful and constructive comments.

References

Bala, J., P. Pachowicz, and K. De Jong. (1994). *Multistrategy Learning from Engineering Data by Integrating Inductive Generalization and Genetic Algorithms*, in R.S. Michalski and G. Tecuci (Eds.), *Machine Learning: A Multistrategy Approach, Vol. IV,* Morgan Kaufmann, San Mateo, CA., pp. 121-138, 1994.

Bala, J., and H. Wechsler. (1996). Shape Analysis Using Multistrategy Learning, *Pattern Recognition* , 29 (8), pp. 1323-1334.

Bala, J., K. DeJong, J. Huang, H. Vafaie, and H. Wechsler. (1995). Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification, *14th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Montreal, Canada.

Bala, J., K. DeJong, J. Huang, H. Vafaie, and H. Wechsler. (1996). Visual Routine for Eye Detection Using a Hybrid Genetic Architecture, *13th Int. Conf. on Pattern Recognition (ICPR)*, Vienna, Austria.

Balakrishnan, K. and V. Honavar (1995). Evolutionary design of neural architectures: A preliminary taxonomy and guide to the literature. Artificial Intelligence Research Group, Department of Computer Science, Iowa State University, Technical Report CS TR #95-01.

Belew, R.K. and M. Mitchell (1996). Editors, "Adaptive Individuals in Evolving Populations: Models and Algorithms", Massachusetts: Addison-Wesley.

Gruau F., and D. Whitley. (1993). Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect*, Evolutionary Computation*, Vol.1, No.3, pp. 213-234.

Grefenstette, J., L. David & D. Cerys. (1991). *Genesis and OOGA: Two Genetic Algorithms System*, TSP: Melorse, MA.

Gutta, S., and H. Wechsler. (1996). Face Recognition Using Hybrid Classifiers, *Pattern Recognition* (to appear).

Gutta, S., J. Huang, I. Shah, D. Singh, B. Takacs, and H. Wechsler. (1995). Benchmark Studies on Face Recognition *Int. Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland.

Hinton, G. E., and S. J. Nolan. (1987) How Learning Can Guide Evolution, *Complex Systems*, Vol.1, pp. 495-502.

Johnson, M. P., P. Maes, and T. Darell. (1994). Evolving Visual Routines, in R.A. Brooks and P. Maes (Eds.), *Artificial Life IV*, MIT Press.

Kohavi, R., and G. John. (1995), Wrappers for Feature Subset selection. Technical Report, Computer Science Department, Stanford University.

Linsker, R. (1988). Self-Organization in a Perceptual Network, *Computer*, Vol. 21, No. 3, pp. 105-117.

Mühlenbein, H., and J. Kinderman. (1989). The dynamics of Evolution and Learning. Toward Genetic Neural Networks, in R. Pfeifer, Z. Schreter, F. Fogelman-Soulie, and L. Steels (Eds.), *Connectionism in Perspective*, Elsevier Science, pp. 173-197.

Quinlan, J. R. (1986). The Effect of Noise on Concept Learning, in R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), *Machine Learning: an Artificial Intelligence Approach,* Morgan Kaufmann, San Mateo, CA, pp. 149-166.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Terano, T., and T. Ishino, Y. (1995). Data Analysis Using Simulated Breeding and Inductive Learning Methods, IJCAI'95 Workshop on Data Engineering and Inductive Learning, pp. 60-69, Aug., 20, Montreal, Canada.

Turney, P. (1996). How to Shift Bias: Lessons from the Baldwin Effect, *Evolutionary Computation* (to appear).

Turney, P. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm, *Journal of Artificial Intelligence Research,* 2, pp. 369-409.

Vafaie, H., and K. De Jong. (1994). Improving a Rule Induction System Using Genetic Algorithms, in R.S. Michalski and G. Tecuci (Eds.), *Machine Learning: A Multistrategy Approach, Vol. IV*, Morgan Kaufmann, San Mateo, CA., pp. 453-469.

Whitley, D., V. S. Gordon, and K. Mathias. (1994). Lamarckian Evolution, the Baldwin Effect, and Function Optimization, in Y. Davidor, H.P. Schwefel, and R. Manner (Eds.), *PPSN III*, Springer Verlag, pp. 6-15.