Feature Selection and Classification Using Age Layered Population Structure Genetic Programming

Anthony Awuley
Department of Computer Science
Brock University
St. Catharines, ON, Canada L2S 3A1
Email: anthony.awuley@gmail.com

Brian J. Ross
Department of Computer Science
Brock University
St. Catharines, ON, Canada L2S 3A1
Email: bross@brocku.com

Abstract—This paper presents a new algorithm called Feature Selection Age Layered Population Structure (FSALPS) for feature subset selection and classification of varied supervised learning tasks. FSALPS is a modification of Hornby’s ALPS algorithm – an evolutionary algorithm reknown for avoiding pre-mature convergence on difficult problems. FSALPS uses a novel frequency count system to rank features in the GP population based on evolved feature frequencies. The ranked features are translated into probabilities, which are used to control evolutionary processes such as terminal-symbol selection for the construction of GP trees/sub-trees. The FSALPS meta-heuristic continuously refines the feature subset selection process whiles simultaneously evolving efficient classifiers through a non-converging evolutionary process that favors selection of features with high discrimination of class labels. We compared the performance of canonical GP, ALPS and FSALPS on some high-dimensional benchmark classification datasets, including a hyperspectral vision problem. Although all algorithms had similar classification accuracy, ALPS and FSALPS usually dominated canonical GP in terms of smaller and efficient trees. Furthermore, FSALPS significantly outperformed canonical GP, ALPS, and other feature selection strategies in the literature in its ability to perform dimensionality reduction.

I. INTRODUCTION

Classification is one of the major tasks in machine learning, in which of features or attributes are used to predict class labels [1]. A feature is “an independent measurable property of a process been observed” [2]. Noisy attributes in data increase the complexity of the learning space and reduce performance of learning algorithms with corresponding high computational requirement in data analysis and machine learning. Feature selection is the process of refining input data by removing irrelevant and/or redundant features [3]. Therefore, to improve the performance of classification algorithms, feature selection is often performed. For a data set \( D \) having \( m \) features, a feature selection algorithm selects a subset of \( n \leq m \) features by eliminating irrelevant and redundant features such that it improves or retains the prediction accuracy of the classifier algorithm. According to [4], relevant feature subset selection for learning systems may improve understanding of the data, leads to the design of better classifier models, eliminate irrelevant data with benefits such as enhanced data visualization, reduced computational cost for constructing learning models and improve generalization of constructed models.

Genetic programming (GP) has been widely used for classification [5]. GP evolves an expression that uses a subset of features from the problem feature vectors to predict class label(s). In fact, feature reduction is said to automatically arise when using GP, since most evolved expressions will normally use reduced semi-optimized sets of features selected from the overall feature set. Feature extraction with GP is also possible [5], although we do not consider it in this paper.

Some examples of using GP for feature reduction are as follows. Smith et al. [6] used GP and GA for feature extraction and feature selection respectively as a preprocessor for a C4.5 decision tree–learning algorithm. The approach proved more successful than direct application of the standard C4.5 decision tree–learning algorithm on the same dataset. Research in [7] and [8] used C4.5 decision tree learning algorithm as a classifier for a GP feature construction problem. Oechsle et al. [9] separate the feature selection and feature extraction stages, and used GP to evolve them independently on a vision problem. The GP classification methodology involved evolution of a set of partial solutions for a single class. This enables the use of GP for classification of high multi-class data, and is faster than conventional GP. Lin et al. [10] used a multi–population layered GP to perform feature selection. Badran et al. [11] used a multi-objective GP to perform feature extraction as a preprocessing stage to a multi-class problem. This involved the reduction of the original dimensionality space to a new reduced and optimized multi-dimensional decision space. Both Bedran et al. [11] and Lemczyk et al. [12] used GP to handle a multi-class dataset by addressing the classification problem through task decomposition.

This paper presents a new evolutionary feature selection and classification strategy called Feature Selection Age Layered Population Structure (FSALPS) [13]. FSALPS extends Hornby’s ALPS algorithm [14][15] to evolve a classifier and perform feature subset selection with high classification accuracy. The ALPS algorithm overcomes the problem of premature convergence by the regular introduction of new individuals into the population. This results in evolutionary search that is constantly exploring different parts of the search space. We extend ALPS by introducing the ability to evolve feature subsets with high discrimination of class labels. FSALPS uses a novel frequency-based feature-ranking system.
that will affect the probability of feature assignment to new randomly created individuals. The intention is that the features used will naturally be refined during evolution, which should improve the overall performance of results over time.

FSALPS is tested on high dimensional benchmark datasets and compared to canonical GP, ALPS GP, and others. Combining ALPS search power with GPs dynamic tree encoding to perform feature subset selection should lead to improvements in aspects of classification problems involving high dimensional datasets. We show that FSALPS GP significantly reduces the number of features while maintaining or improving classification accuracy. This reduces the computational effort needed to design a classifier system as compared to other feature selection strategies that rely on external classifiers. We also did this without consideration of the effect of bloat (intron) code within the trees, whose existence makes the frequency estimations less accurate than possible.

Section II reviews the ALPS algorithm. Section III introduces the FSALPS algorithm. In Section IV, the performance of canonical GP, ALPS GP and FSALPS are compared on a number of classification problems. A very-high dimension hyperspectral computer vision problem is explored in Section V. In Section VI, FSALPS is compared to related works. Concluding remarks and future research are in Section VII.

II. REVIEW: ALPS

The Age Layered Population Structure (ALPS) algorithm introduced by Hornby[14][15] seeks to reduce the problem of premature convergence in stochastic algorithms. In ALPS, age is used as a property of individuals to restrict competition and breeding in the population. ALPS segregates the population of individuals into age-layers. Age is measured by how long an individual’s genotypic material has been evolving in the population. ALPS outperformed canonical EAs in various test problems [14][15].

An aging scheme is used to separate individuals into age layers (see Table I). These values are multiplied by an age gap parameter to determine the maximum age per layer. Given an exponential aging scheme with an age gap of 10 and 6 layers, the maximum ages for the layers will be 10, 20, 40, 80, 160, 320. Individuals within a layer are not allowed to outgrow the maximum allowed age for that layer. Rather, an attempt is made to move such individuals to the next higher layer. Unsuccessful migrants are destroyed. The maximum age per layer allows evolution to proceed long enough in the preceding layer before individuals are old enough to migrate to the next available layer. It also allows individuals to improve in fitness before being pushed to the next higher layer. The last layer has no age limit, hence allows for the accumulation of the best individuals. According to [14], an individual in the last layer is only guaranteed to remain there provided it is the global optimum or else it will be replaced by other highly fit individuals from the lower layers.

The layered approach of evolution does not only restrict competition between individuals in the entire population but also serves as a way of transferring genotypic materials from different fitness basins to higher layers [14]. In ALPS, the bottom layer is regularly replaced with randomly generated individuals. The periodic introduction of such individuals in the bottom layer results in an EA that is never completely converged [14]. By using age to restrict breeding, the possibility of highly fit old individuals dominating the population is reduced, which most often leads to early convergence in canonical EA.

III. THE FSALPS ALGORITHM

<table>
<thead>
<tr>
<th>Ageing-scheme</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Fibonacci ($n$)</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>13</td>
<td>21</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Polynomial ($n^2$)</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>16</td>
<td>25</td>
<td>49</td>
<td>64</td>
</tr>
<tr>
<td>Exponential ($2^n$)</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
</tr>
</tbody>
</table>

Table I: Examples of age schemes for ALPS [14]

FSALPS enhances the ALPS algorithm in Section II, by introducing feature selection mechanisms into the general ALPS strategy. FSALPS performs feature selection by providing a loop back mechanism (see Figure 1) where evolved feature-counts are translated into feature-ranking probabilities. These probabilities are used for selecting terminal symbols in tree construction during initialization or for sub-tree construction during mutation. The FSALPS heuristic performs regular refinement of feature probabilities by reanalyzing the terminal distributions throughout the GP population during a run.

The FSALPS algorithm is shown in Algorithm 1. FSALPS calls which differ from the standard ALPS algorithm are shown in boldface. Lines 2 – 5 initialize the FSALPS algorithm. Initially, the system uses a uniform probability for

![Fig. 1: FSALPS feature selection refinement loop. The frequencies of features in the population are used to determine probabilistic tree generation for layer 0.](image)
A frequency count of features in a classifier expression is by a direct count of features in the GP tree. Terminal probabilities are directly computed from the frequency counts of $n$ terminals seen throughout the entire population. If feature $i$ has a frequency $f_i$, then its probability is:

$$ P_i = \frac{f_i}{\sum_{j=1}^{n} f_j} $$

The frequency count values are then converted into probabilities on line 8 of Algorithm 1 by invoking Algorithm 2. These newly computed probabilities for the features are used in the random creation of new individuals for the first layer when the FSALPS system enters re-initialization mode (line 7). They are also used during tree mutation (line 12) to select terminals during the construction of random sub-trees.

ALPS requires regular inter-layer migration for individuals that are older than the allowed age limit for a layer (line 19). An individual’s age is compared to the maximum allowed age of its layer, and if it is older, an attempt is made to move the individual to a higher layer. All layers are scanned for the layer ($L_h$) with an age range that can accommodate a new individual from a lower layer ($L_l$). When moving over-aged individuals ($L_l$) to a higher layer, a reverse tournament selection strategy is used to select the weakest individual in a random tournament of size $k$, which is then replaced by the migrated individual.

### IV. Classification Experiments

#### A. Experimental Design

<table>
<thead>
<tr>
<th>Name</th>
<th>Attributes</th>
<th>Examples</th>
<th>Pos</th>
<th>Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pima Indians Diabetes</td>
<td>8</td>
<td>336</td>
<td>111</td>
<td>225</td>
</tr>
<tr>
<td>Wisconsin Breast Cancer</td>
<td>30</td>
<td>569</td>
<td>212</td>
<td>357</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>34</td>
<td>351</td>
<td>225</td>
<td>126</td>
</tr>
<tr>
<td>Sonar</td>
<td>60</td>
<td>208</td>
<td>111</td>
<td>97</td>
</tr>
</tbody>
</table>

1) **Datasets**: Four datasets with varying number of features and training examples are examined (Table II). The datasets are from the UCI ML database [16]. All use continuous (floating-point) feature values, and all are binary classification problems. The labels Pos and Neg refers to diabetes diagnoses (Pima), tumour malignancy (Wisconsin), good or bad radar prediction (Ionosphere), and metal cylinder or rock detection (Sonar). Incomplete data is removed from all the datasets.

2) **Fitness Function**: For each feature subset, a classifier is used to test the prediction accuracy of class labels. In a binary classification problem, the fitness of individuals on a test data with $P$ positive instances and $N$ negative instances is translated into metrics such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Class labels that were correctly discriminated by the evolved GP classifier make up TP and TN. The sum of TP and FN equals the sum of positive class labels in the test data. We use classification accuracy for fitness measurements:

$$ fitness = \frac{(TP + TN)}{(P + N)} $$

Positive and negative examples are not distinguished during sampling, and so their natural distribution in the dataset determines their frequency as fitness cases.
Training and testing is done using 20-fold cross validation. Each dataset is shuffled and divided into 20 subsets. We then iteratively use 19 subsets for training, and one subset for testing. We designate the individual with the highest training fitness over a run as being the evolved solution for that run.

3) GP Language: The functions listed in Table III are used. Functions meet closure requirements to avoid illegal operations during tree evaluation. ERCs reside in the range $-1.0 \leq \text{ERC} \leq 1.0$.

4) Parameters: The parameters used for canonical GP are in Table IV. See [17] for discussion of basic GP parameters. Each evolutionary run will complete a total of 250,000 evaluations.

Parameter settings used in the FSALPS and ALPS systems are listed in Table V. Five layers are used in each ALPS system using the polynomial (1, 2, 4, 9 and 16) ageing scheme. When selecting parent(s) for breeding, the selection pressure parameter determines how often a parent is selected from either the current or immediate lower layer. A selection pressure of 0.8 implies 80% of parents are selected from current layer and 20% from the immediate lower layer.

5) GP system and Hardware Configuration: All the algorithms tested (Canonical GP, ALPS, FSALPS) are implemented in ECJ [18].

We used a cluster with 9 nodes with 6 cores each, making a total of 54 cores with 2 different hardware configurations:

1) Intel Core i7 920 2.66GHz with 12GB ram
2) AMD Phenom X6 1055 3.6GHz with 8GB ram

The runs were deployed in parallel such that each run was handled by one compute node.

B. Results

1) Classification Accuracy: The prediction accuracy of a classifier is a measure of how the predictor performs on all training examples. The best solution tree in the training phase is tested on a test dataset. Using k-fold cross validation, all instances of the datasets are used for testing. Table VI contains the average testing performance (classification accuracy measured in percentage) for each k-subset of the best-evolved training solution tree. The results recorded for all three strategies were not significantly different when compared at a 95% confidence interval using Tukey’s HSD ANOVA test.

2) Feature Analysis: Feature reduction is analyzed for the best solutions in each of the 20 runs. Table VII summarizes feature reduction per dataset, in terms of the average/min/max number of features within best solution trees per experiment. Feature reduction is most evident in the FSALPS strategy.
Fig. 2: Sonar: Frequency histogram of features found in solutions (average 20 runs).

Table VIII: Pearson correlation coefficients for feature frequency distributions.

<table>
<thead>
<tr>
<th></th>
<th>Canonical vs ALPS</th>
<th>Canonical vs FSALPS</th>
<th>ALPS vs FSALPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pima</td>
<td>-0.183</td>
<td>-0.254</td>
<td>0.914</td>
</tr>
<tr>
<td>Breastcancer</td>
<td>0.545</td>
<td>0.587</td>
<td>0.803</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>0.624</td>
<td>0.675</td>
<td>0.786</td>
</tr>
<tr>
<td>Sonar</td>
<td>0.333</td>
<td>0.391</td>
<td>0.589</td>
</tr>
</tbody>
</table>

Table IX: Comparing mean difference between feature selection performed for each strategy at the 0.05 significance level using Tukey’s HSD ANOVA test for Breastcancer(⊗), Ionosphere(⊕), Pima(⊖) and Sonar(⊘) dataset for all 20 runs. An arrow points to the dominant strategy while a — means there is no significant difference between the two strategies.

<table>
<thead>
<tr>
<th></th>
<th>ALPS</th>
<th>FSALPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canonical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALPS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2 show the frequencies of the individual objectives as found within solutions in the Sonar runs, averaged over 20 runs. The frequencies are affected by the overall tree sizes of solutions. Some features were almost completely eliminated for each subsequent run, while others are more commonly found in solutions. This shows that features vary in importance with respect to their value during classification.

We compared the distribution shapes of the objective frequencies by computing Pearson correlation coefficients of the frequencies between the three strategies (Table VIII). ALPS and FSALPS show a high correlation to each other in terms of objective frequencies, while there is less correlation of them with canonical GP. In Table IX, Tukey’s HSD ANOVA test shows that FSALPS is the superior algorithm for feature reduction (95% confidence interval).

Figure 3 is a visualization of the feature frequencies in the lowest and highest layers of ALPS and FSALPS. This is measured in one single run of each algorithm.

3) Tree Size Analysis: Figure 4 shows the average tree size growth rate per generation for the three strategies, in 2 of the experiments. Tree growth rate is very high for canonical GP compared to ALPS and FSALPS. In Table X, tree size of the best solution tree used for testing was compared at a 95% confidence interval using Tukey’s HSD test. The results show a significant size difference between the canonical GP and ALPS variants for all but one dataset.

Execution time was significantly lower for ALPS and FSALPS compared to canonical GP. This follows directly from the smaller tree sizes of these algorithms. Details are in [13].

C. Discussion

We were surprised that there is no significant difference in classification accuracy between the 3 strategies. Nevertheless, there is a significant difference in feature reduction. First, canonical GP outperformed ALPS in terms of feature reduction. This is because ALPS continually reintroduces all features into the population at regular intervals, which negatively impacts its feature selection capabilities. FSALPS fixes this issue by selecting a refined set of features to introduce into the population. This resulted in significant reduction of features compared to both canonical GP and ALPS.

ALPS and FSALPS also create smaller trees than canonical GP. This affects the time efficiency for initializing and
Fig. 3: Sonar experiments: Percentage contribution of each feature in layers 0 and 4 in example ALPS and FSALPS runs. Each of the 60 features is uniquely colour-coded. In (a), ALPS layer 0 shows the regular random introduction of all features through the run, while (c) shows that FSALPS introduces a reduced set of features into level 0. Layer 4 in both algorithms is more stable, although the FSALPS case has a reduced feature set (i.e. fewer colour bands).

Fig. 4: Average population tree size per generation (avg 20 runs).

V. EXPERIMENT: HYPERSPECTRAL IMAGE CLASSIFICATION

A. Experimental Setup

<table>
<thead>
<tr>
<th>Table XI: Number of samples used per training class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Corn–notill</td>
</tr>
<tr>
<td>Soybean–mintill</td>
</tr>
</tbody>
</table>
Hyperspectral imaging allows for simultaneous collection of image data in hundreds of narrow adjacent spectral bands [19] and is achieved using hyperspectral remote sensors, which are usually airborne on planes or satellites. Hyperspectral images are ideal for classifying various physical phenomena such as vegetation or mineral resources [19]. This is possible because organic and inorganic materials produce unique absorption and reflection properties at particular bandwidths. GP has been used to evolve hyperspectral classifiers for a number of applications [20] [21] [22] [23] [24].

We will use GP to evolve classifiers for detecting specific class labels (crops) on the Indian Pines hyperspectral data, which was obtained from [25]. The images consist of sets of 145 by 145 pixel data obtained over the Indian Pines test site in northwestern Indiana using the AVIRIS sensor. The site predominantly consists of agriculture and other vegetation. We use 200 image bands with a wavelength range of 0.4 – 2.5μm meters. Since each band is considered a feature, this is a very high dimensional problem to test the capabilities of FSALPS.

Fig 5a shows the reflectance image of the Indian Pines data at the 2.09μm bandwidth. The ground truth image contains a total of 16 classes with known samples (see Fig 5b). Here, we will evolve classifiers for two cases: Corn–notill and Soybean–mintill. Positive and negative examples are obtained from image regions according to the ground truth image in Figure 5b. For each image band, we normalize the RGB color channel between 0.0–1.0. The training examples were manually selected (Table XI). We selected pixels such that a wide range of points of the ground truth image was sampled. At the same time, we did not examine the hyperspectral data during sampling. Hence a similar quality of training examples would likely have arisen if random sampling were performed. 95% of training examples are from regions with 100% intensity of observable reflectance of class labels. The remaining 5% are boundary regions with reduced intensity of resident spectra. The Corn–notill class contains 5.26% of the total samples available on each hyperspectral band while Soybean–mintill covers 8.9% of each band.

As before, we will compare canonical GP, ALPS and FSALPS classifiers on this data. We use the same experimental setup used earlier (Tables III, IV V). The 200 bands are represented as 200 features (terminals), and each returns the band value at that pixel being classified. An image pixel is processed during one tree evaluation, and a positive or negative classification decision for that pixel is determined from the classifier expression.

### B. Results

1) Classification Accuracy: We found that the 3 strategies did not show signs of premature convergence (Figure 6). The best–evolved classifier obtained from training is applied to the test data. Classification performance of the best classifiers are shown in Figure 7. The raw percentage scores of classification accuracy are recorded in Table XII. All three strategies performed well on both cases. An ANOVA test using (95% confidence interval) did not reveal any statistically significant differences.

2) Feature reduction: Table XIII highlights the feature reduction seen. FSALPS reduced the features the most, and ALPS the least. We used an ANOVA test at 95% significance interval to compare feature reduction between all 20 runs.

### Table XII: Average and best classification testing accuracy (%) for Indian Pines data.

<table>
<thead>
<tr>
<th>Class</th>
<th>Canonical</th>
<th>ALPS</th>
<th>FSALPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corn–notill</td>
<td>Avg 92.52</td>
<td>93.69</td>
<td>92.16</td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td>92.52</td>
<td>93.69</td>
</tr>
<tr>
<td>Soybean–mintill</td>
<td>Avg 91.30</td>
<td>92.43</td>
<td>91.40</td>
</tr>
</tbody>
</table>

### Table XIII: Minimum (Min), maximum(Max), average (Avg) and Standard Deviation (StD) number of unique objectives found in the solution set for each dataset. Total objectives considered for each dataset including ERC is 201.

<table>
<thead>
<tr>
<th>Class</th>
<th>Canonical</th>
<th>ALPS</th>
<th>FSALPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corn–notill</td>
<td>Min 18</td>
<td>28</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Max 47</td>
<td>81</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Avg 30.15</td>
<td>51.85</td>
<td>18.25</td>
</tr>
<tr>
<td></td>
<td>StD 7.62</td>
<td>11.95</td>
<td>6.15</td>
</tr>
<tr>
<td>Soybean–mintill</td>
<td>Min 17</td>
<td>29</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Max 49</td>
<td>78</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Avg 30.6</td>
<td>49.55</td>
<td>20.55</td>
</tr>
<tr>
<td></td>
<td>StD 8.26</td>
<td>13.31</td>
<td>8.46</td>
</tr>
</tbody>
</table>
for all algorithms (see Table XIV). FSALPS significantly outperformed canonical and ALPS for both crop types on feature reduction. Figure 8 compares the frequency of features in layer 0 for ALPS and FSALPS for two hyperspectral runs.

3) Tree Size: Canonical GP has the highest rate of tree growth, while ALPS and FSALPS grew at the same rate (see Figure 9). An ANOVA test (95% significance) confirmed that the tree sizes of ALPS and FSALPS are significantly smaller than canonical GP.

C. Discussion
The results for the hyperspectral experiments follow the earlier findings in Section IV. Classification accuracy was equivalent between canonical GP, ALPS, and FSALPS. FSALPS was the best in feature reduction, and FSALPS and ALPS both generated smaller trees than canonical GP.

VI. COMPARISONS TO RELATED WORK
Harper [27] used a variation of ALPS called spatial co-evolution ALPS (SCALPS). SCALP was found to produce
TABLE XVI: Comparing feature selection of some published strategies and FSALPS.

<table>
<thead>
<tr>
<th>Research</th>
<th>Pima</th>
<th>WBC New</th>
<th>Ionosphere</th>
<th>Sonar</th>
</tr>
</thead>
<tbody>
<tr>
<td>[26]</td>
<td>Min –</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Avg 6.4</td>
<td>15.7</td>
<td>19.7</td>
<td>38.0</td>
<td></td>
</tr>
<tr>
<td>FSALPS</td>
<td>Min 3</td>
<td>5</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Avg 6.4</td>
<td>9.3</td>
<td>12.75</td>
<td>16.85</td>
<td></td>
</tr>
<tr>
<td>Tot: 8</td>
<td>30</td>
<td>34</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

needed to confirm these results, and especially using similar experimental methodologies.

Our primary goal in the hyperspectral experiments is to study a very high-dimensional classification problem, rather than focus on the quality of the hyperspectral classifiers themselves. Preliminary investigation shows that our classifiers are competitive with those in the literature [13]. However, more work is required for accurate comparisons to be made.

VII. CONCLUSIONS AND FUTURE WORK

FSALPS is a new heuristic for GP that performs dynamic feature reduction during evolution. FSALPS emerged dominant over canonical GP and ALPS in terms of feature reduction capabilities. Surprisingly, on the problems investigated here, FSALPS solutions did not score significantly higher classification accuracy than canonical and ALPS GP contrary to the usual expectations that simpler search space means more accurate classifiers. Canonical GP and ALPS are able to compensate and sacrifice feature selection for classification accuracy. FSALPS was also shown to be competitive with other approaches in the literature, in terms of classification accuracy, while often superior in feature reduction capabilities. Therefore, FSALPS is worthy of consideration in classification problems, given that it creates accurate classifiers that are efficient to execute.
There are many directions for future work. First, the feature analysis done by FSALPS could be improved. Because FSALPS uses the entire tree for frequency calculations, blot (intron) sub-expressions will contribute to inaccurate feature counts. Although blot will affect our frequencies in our experiments, the results show that feature reduction still arose. Since FSALPS trees are smaller, blot is probably not as prevalent as with standard GP. More effective feature probabilities could be computed if blot expressions could be reduced, perhaps via size-fair crossover [28], expression simplification, or linear GP. Since classification accuracy was not affected in our results, another approach might be to take a classifier evolved from standard GP and perform source-code simplification on it. On the other hand, FSALPS performs significant blot reduction automatically, with minimal overhead.

We used classification accuracy to measure fitness of GP individuals. Additional criteria could be used, such as sensitivity, specificity, information gain, maximum relevance, feature correlation and mutual information [2]. The use of more sophisticated fitness evaluation may result in improved classifier performance and feature reduction.

Other than Figure 2, we did not examine in detail the specific value of particular features in different problems. Future work should investigate this further, especially with respect to information gain measurements.

Feature extraction [2][5] could be incorporated into the FSALPS algorithm. There are situations where the original feature vectors are not representative of the given problem and will require the discovery of interesting hidden relationships between the features. Thus it is possible to exploit GPs representational power to construct a system that performs feature selection and feature extraction as done in [10]. We believe that an FSALPS-like system will facilitate the discovery of new relevant features leading to improved classification accuracy and a more refined feature subset.

In our work on the Indians Pines hyperspectral data, we used a very basic image classification language. Improved results would arise when spatial operators and other spectral operators [24] are added to the GP language. These additions could easily facilitate detection of more difficult regions (e.g. boundaries and edges) on the hyperspectral image.

The problems investigated here involve binary classifications. A new direction would be to investigate multi-classification problems using FSALPS. For example, we could evolve a classifier that simultaneously discriminates multiple classes on hyperspectral images.

Acknowledgements: Thanks to Cale Fairchild for his help with system issues. Research supported by NSERC DG 138467.

REFERENCES


