A Grammatical Evolution based Hyper-Heuristic for the Automatic Design of Split Criteria

Márcio P. Basgalupp
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
S. J. dos Campos, SP, Brazil
basgalupp@unifesp.br

Rodrigo C. Barros
Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, RS, Brazil
rodrigo.barros@pucrs.br

Tiago Barabasz
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
S. J. dos Campos, SP, Brazil
tbarabasz@unifesp.br

ABSTRACT

Top-down induction of decision trees (TDIDT) is a powerful method for data classification. A major issue in TDIDT is the decision on which attribute should be selected for dividing the nodes in subsets, creating the tree. For performing such a task, decision trees make use of a split criterion, which is usually an information-theory based measure. Apparently, there is no free-lunch regarding decision-tree split criteria, as is the case of most things in machine learning. Each application may benefit from a distinct split criterion, and the problem we pose here is how to identify the suitable split criterion for each possible application that may emerge. We propose in this paper a grammatical evolution algorithm for automatically generating split criteria through a context-free grammar. We name our new approach ESC-GE (Evolutionary Split Criteria with Grammatical Evolution). It is empirically evaluated on public gene expression datasets, and we compare its performance with state-of-the-art split criteria, namely the information gain and gain ratio. Results show that ESC-GE outperforms the baseline criteria in the domain of gene expression data, indicating its effectiveness for automatically designing tailor-made split criteria.

Categories and Subject Descriptors

I.2.6 [**Induction and Knowledge Acquisition**]: Learning

General Terms

Algorithms

Keywords

Grammatical Evolution, Hyper-Heuristics, Split Criterion.

1. INTRODUCTION

Top-down induction of decision trees is a powerful method for data classification. Given a training dataset, decision trees are created by recursively dividing the input space such that the training data examples in each partition can be classified with increasingly smaller uncertainty. The process continues until a given stopping condition is satisfied — usually, the tree growth stops when all training data examples in a given node belong to the same class [24].

A major issue in top-down induction of decision trees is the decision on which attribute should be selected for dividing the node in subsets at each step of the recursive algorithm. For the case of *axis-parallel* decision trees (also known as *univariate*), the problem is to choose the attribute that better discriminates the input data. A decision rule based on such an attribute is thus generated, and the input data is filtered according to the outcomes of this rule. For performing such a task, decision trees make use of a *split criterion*, which is usually an information-theory based measure commonly regarded as an *impurity function*. Such a function examines heuristically the possible tests over the training dataset attributes, locally optimizing the best node splits by analyzing the estimates of the class distributions.

The rationale behind well-known split criteria such as the *information gain* [22] or the *gain ratio* [23] is that one should seek to minimize the class entropy in a given node in order to maximize the acquired gain in information. The value of entropy decreases as the probability distribution of the classes in a node become more heterogeneous. Therefore, the selected attribute is the one that generates a partition in which the examples are distributed less randomly over the classes [9].

However, each split criterion in the literature has disadvantages. For instance, the information gain tends to favor attributes with more values [15]. The χ^2 criterion [18] generates very large trees, since it favors binary attributes that lead to very narrow trees with great depth. The gain ratio criterion [23], in turn, may be undefined for some cases, and it favors attributes with highly skewed value-frequency [33]. The Gini index [5] also shares the disadvantage of being bias towards multi-valued and highly-skewed attributes.

Apparently, there is *no free-lunch* regarding decision-tree split criteria, as is the case of most things in machine learning. In the recent years, researchers have investigated hyper-heuristic approaches for developing customized solutions that can be reused in several problems. Within the machine learning literature, we can cite the work of Pappa and Freitas [21], which proposed a genetic programming based hyper-heuristic to automatically design full rule induction

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2576768.2598327

algorithms. Similarly, the work of Barros et al. [2, 3] proposed a hyper-heuristic to automatically design full decision-tree induction algorithms. Both studies aimed at avoiding the *no free-lunch theorem* by customizing solutions to particular application domains.

Bearing in mind the strategies above-mentioned, and also the fact that each application domain may benefit from a distinct decision-tree split criterion, the problem we pose here is how to identify the suitable criterion for each possible application that may emerge. Since the manual design of tailored split criteria is unfeasible, given the huge amount of different existing applications, we propose in this paper a *grammatical evolution* (GE) based hyper-heuristic for automatically generating split criteria tailored to particular application domains.

This study goes beyond the work of Barros et al. [2, 3] considering that it proposes the automatic construction of a building block of decision-tree induction algorithms (the split criterion) instead of selecting and combining existing building blocks, enhancing the granularity level of the hyper-heuristic. Since GE is a grammar-based evolutionary algorithm, our novel method incorporates knowledge regarding the problem of finding the best split criterion in a context-free grammar. We name our approach ESC-GE (Evolutionary Split Criteria through Grammatical Evolution).

This paper is organized as follows. Section 2 describes related work in split criteria for TDIDT. Section 3 presents ESC-GE, our new approach for the automatic generation of split criteria. Section 4 details the methodology employed during the experiments, which are in turn presented in Section 5. We end this paper with our conclusions and future work suggestions in Section 6.

2. RELATED WORK

In this section, we make use of the following notation. $\mathbf{X}$ is the set of N training instances, a_i is the i^{th} predictive attribute of $\mathbf{X}$, and y is the class vector with k rows (classes).

The most well-known split criteria in the literature are based on information-theory, following the concept of Shannon's entropy [26]. Entropy is known to be a unique function which satisfies the four axioms of uncertainty. It represents the average amount of information when coding each class into a codeword with ideal length according to its probability. Some interesting facts regarding entropy are:

- For a fixed number of classes, entropy increases as the probability distribution of classes becomes more uniform;
- If the probability distribution of classes is uniform, entropy increases logarithmically as the number of classes in a sample increases;
- If a partition induced on a set $\mathbf{X}$ by an attribute a_j is a refinement of a partition induced by a_i, then the entropy of the partition induced by a_j is never higher than the entropy of the partition induced by a_i (and it is only equal if the class distribution is kept identical after partitioning). This means that progressively refining a set in sub-partitions will continuously decrease the entropy value, regardless of the class distribution achieved after partitioning a set.

The first split criterion that arose based on entropy is the *global mutual information* (GMI) [11, 25, 28]. Ching et al. [8] propose the use of GMI as a tool for supervised discretization. They name it *class-attribute mutual information*, though the criterion is exactly the same. GMI is bounded by zero (when a_i and y are completely independent) and its maximum value is $max(\log_2 |a_i|, \log_2 k)$ (when there is a maximum correlation between a_i and y). The authors reckon this measure is biased towards attributes with many distinct values.

Information gain [7,12,22] is another example of measure based on Shannon's entropy, being employed in the well-known decision-tree induction algorithm ID3 [22]. The goal of information gain is to maximize the reduction in entropy due to splitting each individual node. Wilks [31] has proved that as $N \rightarrow \infty$, $2 \times N \times GMI$ (or similarly replacing GMI by information gain) approximates the χ^2 distribution. This measure is often regarded as the *G statistics* [17,18]. Instead of using the value of this measure as calculated, we can compute the probability of such a value occurring from the χ^2 distribution on the assumption that there is no association between the attribute and the classes. The higher the calculated value, the less likely it is to have occurred given the assumption. The advantage of using such a measure is making use of the levels of significance it provides for deciding whether to include an attribute at all.

Quinlan [22] acknowledges the fact that the information gain is biased towards attributes with many values. This is a consequence of the previously mentioned particularity regarding entropy, in which further refinement leads to a decrease in its value. Quinlan proposes a solution for this matter called *gain ratio* [23]. It basically consists of normalizing the information gain by the entropy of the attribute being tested. The gain ratio compensates the decrease in entropy in multiple partitions by dividing the information gain by the attribute self-entropy. Nevertheless, the gain ratio has two deficiencies: (i) it may be undefined (*i.e.*, the value of self-entropy may be zero); and (ii) it may choose attributes with very low self-entropy but not with high gain. For solving these issues, Quinlan suggests first calculating the information gain for all attributes, and then calculating the gain ratio only for those cases in which the information gain value is above the average value of all attributes.

Several variations of the gain ratio have been proposed, such as the *normalized gain* [14] and the *average gain* [30], though the gain ratio is still considered the state-of-the-art in the split criteria literature, being employed in the well-known decision-tree induction algorithm C4.5 [23].

With respect to similar hyper-heuristic approaches, we can cite the work of Barros et al. [1–4] called HEAD-DT, which is a hyper-heuristic that automatically designs full decision-tree induction algorithms. HEAD-DT, however, does not generate novel split criteria. It simply selects from a fixed list of 15 available criteria in the literature. In a similar note, the work of Vella et al. [29] proposes a hyper-heuristic that evolves rules to allow the choice of existing split criteria. An example of rule is: "if x% of the attributes have an entropy value below a given threshold, then use the existing split criterion Y to partition the nodes in subsets". Therefore, the work of Vella et al. [29] also does not generate novel split criteria through genetic programming. To the best of our knowledge, this work is the first to propose the automatic generation of split criteria through a grammatical evolution based hyper-heuristic.

3. ESC-GE

Evolutionary Split Criteria through Grammatical Evolution (ESC-GE) is a hyper-heuristic that automatically designs split criteria for top-down decision-tree induction algorithms. Hyper-heuristics (HHs) operate on a different level of generality from metaheuristics. Instead of guiding the search towards near-optimal solutions for a given problem, a HH approach operates on the heuristic level, guiding the search towards the near-optimal heuristic that can be further applied to different application domains. HHs are therefore assumed to be problem-independent and can be easily utilized by experts and non-experts as well [20]. It can be seen as a high-level methodology which, when faced with a particular problem instance or class of instances, and a number of low-level heuristics, automatically designs a suitable combination of the provided components to effectively solve the respective problem(s) [6].

In the particular case of ESC-GE, it is considered a HH approach since it automatically designs a split criterion (mathematical function) that is problem-independent, *i.e.*, such a function can be used by a top-down decision-tree induction algorithm to split data from any classification dataset. Nevertheless, the underlying assumption of ESC-GE is that tailor-made split criteria are capable of being more effective than a single general-use split criterion. For instance, instead of using the same split criterion for all classification datasets, ESC-GE is trained with datasets that share a particular application domain, under the hypothesis that the automatically-designed split criterion that was tailored to such a domain will be more effective than traditional criteria such as the information gain and gain ratio.

ESC-GE is guided by a grammatical evolution search (GE) [19], which is considered the state-of-the-art among the available grammar-based genetic programming techniques [16]. GE tries to mimic the process of generating a protein from the genetic material of an organism. Individuals are generated by binary strings (codons) equivalent to the double helix of DNA; the integer string decoded from the binary string is the equivalent of the transcription of DNA to RNA; finally, the mapping of the integer string to the grammar production rules and the subsequent generation of a computer program (or function) is the equivalent of the translation of RNA to the sequence of amino acids that are contained within the protein molecule. Figure 1 depicts this rationale.

The remainder of this section presents the main features of ESC-GE: its grammar (Section 3.1), its genetic operators (Section 3.2), and its fitness function (Section 3.3).

3.1 Grammar

Figure 2 presents the context-free grammar employed by ESC-GE in order to generate split criteria for TDIDT. The non-terminals and terminals are detailed as follows.

Non-terminals:

- $+$: the sum of two scalars.
- $-$: the subtraction of two scalars.
- $*$: the multiplication of two scalars.
- $/$: the protected division of two scalars. The denominator cannot be 0 (zero).
- *log*: the protected log_{10} function over a non-zero scalar.
- *sin*: the sin function over a scalar.
- *cos*: the cos function over a scalar.

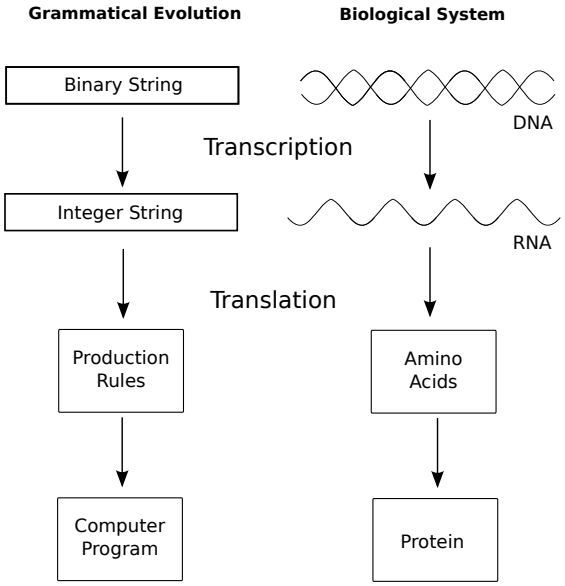

Figure 1: Comparison between the GE system and a biological genetic system. Adapted from [19].

```
1) <start>::= <esc>
2) <esc>::= (+ <esc> <esc>) | (- <esc> <esc>) | (/ <esc>
<esc>) | (* <esc> <esc> ) | (log <esc>) | (sum <vector>)
| (numClass) | (numEx) | (sin <esc>) | (cos <esc>) |
(perBagPerClass)
3) <vector> ::= (perClass) | (perBag) | (** <esc> <vec-
tor>) | (// <vector> <esc>)
```

Figure 2: Context-free grammar for generating split criteria.

- *sum*: the summation over a vector. The result is a scalar resulting from the sum of all the elements of the vector.
- $**$: the multiplication of a scalar and a vector. Each element of the vector is multiplied by the scalar, resulting in a new vector.
- //: similar to the (**) operator, but now resulting in a new vector in which each element is divided by the scalar. Scalars are not allowed to be 0 (zero).

Terminals:

- numEx: scalar representing the number of instances in the current node.
- perBag: vector representing the number of instances in each partition (bag) of the current node.
- perClass: vector representing the number of instances from each class of the current node.
- perBagPerClass: matrix representing the number of instances in each partition grouped by their corresponding class.

Given the grammar in Figure 2, ESC-GE evolves a function S within the following equation:

$$criterion = (1 - unknownRate) \times \sum_{p=1}^{np} \sum_{c=1}^{k} S \qquad (1)$$

where *unknownRate* is the percentage of missing values, np is the number of partitions (bags), and k is the number

of classes. Note that ESC-GE only evolves function S in Eq. 1, whereas the other terms are fixed and *criterion* should be maximized. The first term $(1 - unknownRate)$ simply weights the criterion proportionally to its amount of missing values. ESC-GE avoids the further complexity of evolving the $\sum$ operators by fixing both sums over partitions and classes. The evolution of a more detailed split criterion with non-fixed $\sum$ operators is left for future work.

3.2 Genetic Operators

In the beginning of the evolutionary process, the individuals of the initial population in ESC-GE are randomly initialized. They have variable length, with a minimum size of five codons each, and a chance of 85% that new codons will be incrementally added to the individual. Each codon is comprised of 1 byte (8 bits), which is randomly generated.

To evolve the current generation of individuals, the following mutually-exclusive genetic operations can be performed: crossover, mutation, and duplication. Crossover has a chance of 90% of being performed, and both duplication and mutation have a chance of 5% of being applied. These operations are executed until all individuals of the new population are generated.

For crossover to be performed, two individuals are chosen via tournament selection. After the individuals are selected, they take part in a standard one-point crossover operation, generating two children. In the duplication process, one individual is also selected using tournament selection, and then two codons of the individual are randomly selected, copying all codons located between these two selected codons to the end of the individual. Finally, mutation requires one individual to be selected via tournament selection. This individual is traversed codon by codon, where each codon has a 10% probability of having its value replaced by a randomly-generated 8-bit value. The three operators are illustrated in Figure 3.

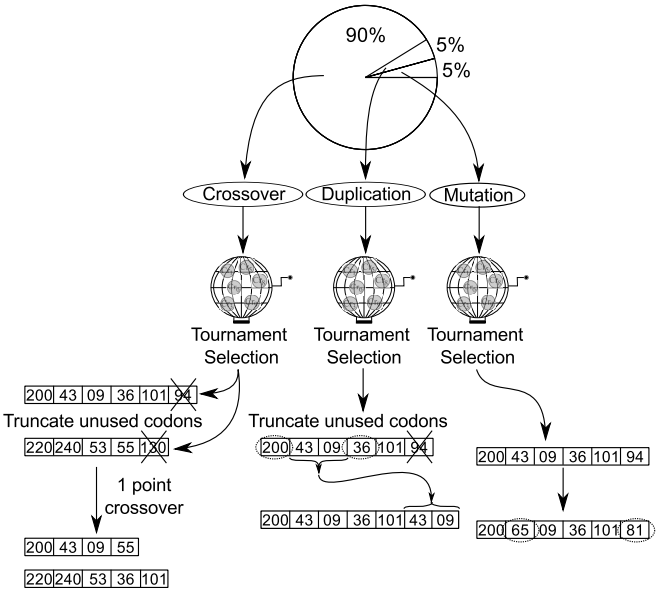

Figure 3: ESC-GE genetic operators.

3.3 Fitness Function

In ESC-GE, each individual represents a possible split criterion to be used in TDIDT. After each individual is decoded into a function, we fit such a function into the body of J48 (the Java version of C4.5) [32]. In order to computer the fitness of each individual, we evaluate the corresponding modified version of J48 in a *meta-training set*. In contrast, a *meta-test set* is used to assess the quality of the evolved split measure function, which is the best individual produced by ESC-GE. Note that there is no overlapping of instances between the meta-training and meta-test sets, which allows us to measure the generalization ability of the evolved split criterion.

The fitness evaluation process is based on [1–3], where we have multiple datasets comprising the meta-training set, and multiple (but different) datasets comprising the meta-test set. In this approach, each dataset is described by a different set of predictive attributes, so each dataset corresponds to a different classification problem.

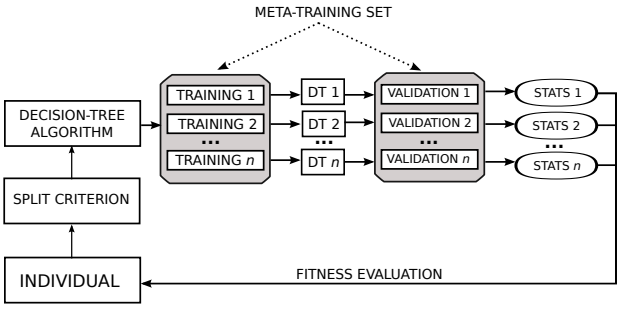

Figure 4: ESC-GE fitness evaluation.

In Figure 4, we can observe how the fitness evaluation of a split criterion occurs. First, a given individual is mapped into its corresponding split criterion, and then it is incorporated into a decision-tree induction algorithm. Next, each dataset from the meta-training set is partitioned into a training set and a validation set — typical values are 70% for training and 30% for validation [32]. The term "validation set" is used in here instead of "test set" to avoid confusion with the meta-test set, and also due to the fact that we are using the performance measure of a candidate split criterion on those validation sets to guide the evolutionary search for a better function. The same cannot be done with test sets, which are exclusively used for assessing the predictive performance of a decision-tree algorithm using the evolved function as split criterion.

After dividing each dataset from the meta-training set into "training" and "validation", we induce a decision tree for each training set available. For evaluating the predictive performance of these decision trees, we use the corresponding validation sets. Statistics regarding the predictive performance and the size of each decision tree are recorded (*e.g.*, accuracy, F-Measure, precision, recall, total number of nodes/leaves, etc.), and can be used individually or combined as the fitness function of ESC-GE. In this work, we use as fitness function the average F-Measure of the decision trees generated by a given individual for each dataset in the meta-training set. The well-known F-Measure (also known as F-score or F1 score) is the harmonic mean of precision and recall, as shown in the equations below:

$$precision = \frac{tp}{tp + fp} \quad (2)$$

$$recall = \frac{tp}{tp + fn} \quad (3)$$

$$fmeasure = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

$$Fitness = \frac{1}{n} \sum_{i=1}^{n} fmeasure_i \quad (5)$$

where tp (tn) is the numbers of true positives (negatives) in the validation set, fp (fn) is the numbers of false positives (negatives) in the validation set, $fmeasure_i$ is the F-Measure obtained in dataset i and n is the total number of datasets in the meta-training set.

These equations are directly applicable in the case of binary classification problems, *i.e.*, the case where a dataset has only two classes: positive and negative. Nevertheless, they can be trivially extended to multi-class problems. For instance, we can compute the value of a measure for each class — assuming each class to be the "positive" class in turn, and considering all the other classes as the "negative class" — and then compute a (weighted) average of the per-class measure.

Although accuracy is still a very popular measure of predictive performance, it is important to notice that it tends to be a misleading measure in datasets with a very unbalanced class distribution. For instance, suppose we are classifying a dataset whose class distribution is very skewed: 10% of the instances belong to the positive class and 90% to the negative class. An algorithm that always classifies instances as belonging to the negative class would achieve 90% of accuracy, even though it never predicts the positive class. In this case, assuming that the positive class is equally important to (or even more so than) the negative class, we would prefer an algorithm with a somewhat lower accuracy value, but which correctly predicts some instances as belonging to the rare positive class.

Having in mind that most datasets used in our experiments have very unbalanced class distributions, the average F-Measure is a more suitable fitness function than, say, the average accuracy, since it is well-known that the F-measure copes much better with unbalanced class-distribution problems than the accuracy measure.

4. METHODOLOGY

In this section, we present the methodology employed during the empirical analysis. We present the baseline split criteria and ESC-GE parameters in Section 4.1, whereas the public gene expression datasets are described in Section 4.2 and the statistical analysis process in Section 4.3.

4.1 Baseline Criteria and Parameters

We compare the split criterion generated by ESC-GE to well-known and widely-used entropy-based decision-tree split criteria: information gain [22], and gain ratio [23]. All the criteria employed were plugged into the J48 algorithm [32], in order to allow a fair comparison among them.

Table 1 shows the user-defined parameter values used in ESC-GE. No attempt to tune these parameter values was made. Parameter optimization is a topic left for future research.

Table 1: Configurable ESC-GE parameters.

Parameter	Value
Initialization Probability	85%
Number of Individuals	100
Minimum Individual Size	5
Number of Generations	20
Crossover Probability	90%
Duplication Probability	5%
Mutation Probability	5%
Mutation per Codon Probability	10%
Tournament Size	3
Elite	2

4.2 Datasets

The empirical analysis presented in this paper is based on 20 public datasets from the gene expression application domain [27]: alizadeh-v1, alizadeh-v2, alizadeh-v3, armstrong-v1, armstrong-v2, bittner, chen, chowdary, golub-v1, gordon, khan, lapointe-v1, lapointe-v2, liang, nutt-v3, pomeroy-v1, pomeroy-v2, ramaswamy, shipp-v1, and tomlins. In brief, microarray technology enables expression level measurement for thousands of genes in parallel, given a biological tissue of interest. Once combined, results from a set of microarray experiments produce a gene expression dataset. The datasets employed here are related to different types or subtypes of cancer, *e.g.*, patients with prostate, lung, skin, and other types of cancer. The classification task refers to labeling different examples (instances) according to their gene (attribute) expression levels.

Table 2 provides details about these datasets. They were randomly divided into two groups: A and B. Datasets belonging to A were used as the meta-training set, whereas datasets belonging to B were used as the meta-test set.

Table 2: Summary of the 20 Gene Expression datasets. For each dataset, we present the total number of instances, total number of attributes, imbalanced ratio (rate between over- and under-represented class), and total number of classes.

	Data Set	# Instances	# Attributes	IR	# Classes
A	alizadeh-v3	62	2093	0.43	4
	armstrong-v1	72	1081	0.5	2
	lapointe-v2	110	2496	0.27	4
	nutt-v3	22	1152	0.47	2
	tomlins	104	2315	0.38	5
B	alizadeh-v1	42	1095	1	2
	alizadeh-v2	62	2093	0.21	3
	armstrong-v2	72	2194	0.71	3
	bittner	38	2201	1	2
	chen	179	85	0.72	2
	chowdary	104	182	0.68	2
	golub-v1	72	1868	0.53	2
	gordon	181	1626	0.21	2
	khan	83	1069	0.38	4
	lapointe-v1	69	1625	0.28	3
	liang	37	1411	0.11	3
	pomeroy-v1	34	857	0.36	2
	pomeroy-v2	42	1379	0.4	5
	ramaswamy	190	1363	0.33	14
	shipp-v1	77	798	0.33	2

4.3 Statistical Analysis

To evaluate the statistical significance of the experimental results, we present the results of statistical tests by following the approach proposed by Demšar [10]. In brief, this

approach seeks to compare multiple algorithms on multiple datasets, and it is based on the use of the Friedman test with a corresponding post-hoc test. The Friedman test is a non-parametric counterpart of ANOVA, as follows. Let R_i^j be the rank of the j^{th} of k algorithms on the i^{th} of N datasets. The Friedman test compares the average ranks of algorithms, $R_j = \frac{1}{N}\sum_i R_i^j$. The Friedman statistic, given by:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \quad (6)$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are large enough.

Iman and Davenport [13] showed that Friedman's χ_F^2 is undesirably conservative and derived an adjusted statistic:

$$F_f = \frac{(N-1)\times\chi_F^2}{N\times(k-1)-\chi_F^2} \quad (7)$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

If the null hypothesis of similar performances is rejected, we proceed with the Nemenyi post-hoc test for pairwise comparisons. The performance of two classifiers is significantly different if their corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}} \quad (8)$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$.

5. RESULTS AND DISCUSSION

Table 3 shows the predictive accuracy values for ESC-GE, information gain, and gain ratio. It presents the average accuracy over a 5-fold cross-validation procedure. Observe that the decision-tree induction algorithm that employs the split criterion automatically designed by ESC-GE generates trees with the best accuracy values in 10 out of 15 datasets.

Table 3: Average accuracy of J48 with its corresponding split criterion.

	ESC-GE	Info Gain	Gain Ratio
alizadeh-v1	**0.76**	0.69	0.69
alizadeh-v2	**0.95**	0.90	0.90
armstrong-v2	0.75	**0.78**	0.74
bittner	**0.76**	0.53	0.55
chen	**0.91**	0.83	0.86
chowdary	**0.91**	0.89	0.89
golub-v1	**0.93**	0.76	0.83
gordon	**0.96**	0.93	0.94
khan	0.83	0.82	**0.88**
lapointe-v1	0.70	**0.77**	0.67
liang	**0.78**	0.70	0.70
pomeroy-v1	0.79	**0.88**	**0.88**
pomeroy-v2	**0.60**	0.52	0.57
ramaswamy	0.54	0.56	**0.62**
shipp-v1	**0.79**	0.74	0.77
Number of victories	10	2	3
Average Rank	1.47	2.43	2.1

To evaluate the statistical significance of the accuracy results, we calculated the average Friedman rank for ESC-GE, information gain, and gain ratio: 1.47, 2.43, and 2.1, respectively. The average rank suggests that ESC-GE outperforms the baseline split criteria regarding predictive accuracy. The calculation of Iman's F statistic resulted in $F_f = 4.45$. Critical value of $F(k-1,(k-1)(n-1)) = F(2,28)$ for $\alpha = 0.05$ is 3.34. Since $F_f > F_{0.05}(2,28)$ $(4.45 > 3.34)$, the null-hypothesis is rejected. We proceed with a post-hoc Nemenyi test to find which method provides better results in a pairwise fashion. The critical difference $CD = 0.86$. The differences between the average rank of ESC-GE and the rank of the baseline split measures are 0.96 and 0.63, respectively. Given that the difference between ESC-GE and information gain is greater than the CD value, we can confidently argue that the performance of the split criterion evolved by ESC-GE is significantly better than the performance of information gain. Even though there is no significant difference between ESC-GE and the remaining criterion (gain ratio), we can observe that only the split criterion evolved by ESC-GE outperforms another measure with statistical significance regarding accuracy.

Table 4 shows the F-measures values for ESC-GE, information gain, and gain ratio. ESC-GE generates the best trees regarding F-Measure once again in 10 out of 15 datasets, whereas both information gain and gain ratio do it in 3 datasets. In the pomeroy-v1 dataset, the best F-Measure values are provided by both information gain and gain ratio. In terms of statistical analysis, the computed value of $F_f = 4.45$. Since $F_f > F_{0.05}(2,28)$ $(4.45 > 3.34)$, the null-hypothesis is rejected, and thus we can argue that there is significant differences among the criteria regarding F-measure. If we proceed with a post-hoc Nemenyi test, the critical difference is once again $CD = 0.86$, and we can observe that the difference between ESC-GE and information gain (0.96) is greater than CD $(0.96 > 0.86)$. It is important to notice that ESC-GE once again achieves the lowest average rank, indicating it is the most suitable option for the domain of gene expression data.

Table 4: Average F-Measure of J48 with its corresponding split criterion.

	ESC-GE	Info Gain	Gain Ratio
alizadeh-v1	**0.76**	0.69	0.69
alizadeh-v2	**0.95**	0.90	0.90
armstrong-v2	0.75	**0.78**	0.74
bittner	**0.76**	0.52	0.55
chen	**0.91**	0.83	0.86
chowdary	**0.91**	0.89	0.89
golub-v1	**0.93**	0.76	0.83
gordon	**0.96**	0.93	0.94
khan	0.83	0.82	**0.88**
lapointe-v1	0.70	**0.77**	0.67
liang	**0.76**	0.69	0.69
pomeroy-v1	0.79	**0.88**	**0.88**
pomeroy-v2	**0.57**	0.49	0.54
ramaswamy	0.54	0.57	**0.61**
shipp-v1	**0.80**	0.75	0.77
Number of victories	10	3	3
Average Rank	1.47	2.43	2.1

Figure 5 shows the Nemenyi's critical diagram, as suggested by Demsăr [10]. In this diagram, a horizontal line represents the axis on which we plot the average rank values of the methods. The axis is turned so that the lowest (best) ranks are to the right since we perceive the methods on the right side as better. When comparing all the criteria

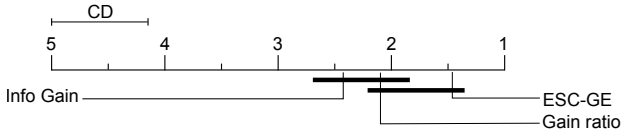

Figure 5: Critical diagram: Accuracy and F-measure.

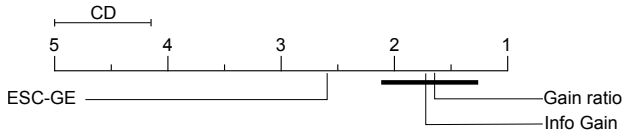

Figure 6: Critical diagram: Tree Size.

against each other, we connect the groups of criteria that are not significantly different through a bold horizontal line. We also show the critical difference given by the Nemenyi's test above the graph. Figure 5 shows both the results of accuracy and F-Measure (since the ranking values are the same). We can see that ESC-GE is connected only to gain ratio (no significant difference), though their difference is almost at the limit of the critical difference. ESC-GE is significantly better than information gain (no line connecting them), whereas gain ratio is not.

Finally, Table 5 shows the size of trees generated by J48 with the criterion evolved by ESC-GE, and also with the baseline criteria information gain and gain ratio. Results show that both information gain and gain ratio generate smaller trees in most datasets, whereas ESC-GE generates the smaller tree in only 3 datasets.

Table 5: Average tree size of J48 with its corresponding split criterion.

	ESC-GE	Info Gain	Gain Ratio
alizadeh-2000-v1	22.8	6.20	**5.80**
alizadeh-2000-v2	8.20	**5.00**	**5.00**
armstrong-2002-v2	10.2	**8.20**	9.00
bittner-2000	15.6	**6.20**	**6.20**
chen-2002	**11.2**	17.0	18.4
chowdary-2006	**9.40**	10.2	9.80
golub-1999-v1	9.80	**6.00**	6.20
gordon-2002	15.6	8.20	**7.20**
khan-2001	26.2	10.6	**9.80**
lapointe-2004-v1	14.0	10.4	**10.2**
liang-2005	5.60	**5.00**	**5.00**
pomeroy-2002-v1	12.2	**6.40**	**6.40**
pomeroy-2002-v2	70.8	**11.4**	11.6
ramaswamy-2001	**11.0**	57.2	55.4
shipp-2002-v1	17.4	**8.80**	9.40
Number of victories	3	8	8
Average Rank	2.6	1.73	1.67

Regarding the statistical analysis, the computed value of $F_f = 5.21$. Since $F_f > F_{0.05}(2, 28)$ $(5.21 > 3.34)$, the null-hypothesis is rejected. Once again, we proceed with a post-hoc Nemenyi test, and the critical difference $CD = 0.86$. The differences between the average rank of ESC-GE and the baseline split measures – information gain and gain ratio – are 0.87 and 0.93, respectively. Thus, we can assert that both gain ratio and information gain generate decision trees significantly smaller than the ones generated by ESC-GE. It is also important to note that there is no significant differences between the baseline split criteria in terms of tree size (see the critical diagram in Figure 6).

The fact that two baseline split criteria generate smaller trees than ESC-GE should not be a concern, since smaller trees are only preferable in scenarios where the predictive performance of the models is similar. The analysis previously presented clearly indicates that ESC-GE generates trees that outperform the baselines regarding predictive performance in terms of both accuracy and F-Measure. The Occam's razor assumption that among competitive hypotheses, the simpler is preferred, does not apply in this case.

The split criterion evolved by ESC-GE is presented in Eq. 9:

$$f = -N \times \sum_{p=1}^{np} \sum_{c=1}^{k} \frac{1}{(N - i_{p,c})} \times \frac{1}{i_{p,c}} \quad (9)$$

where $i_{p,c}$ is the number of instances in partition p that belong to class c. We plan on further studying this criterion to verify its boundary values and real applicability in other domains.

6. CONCLUSION AND FUTURE WORK

In this work, we presented ESC-GE, a hyper-heuristic grammatical evolution algorithm that automatically generates split criteria for a particular application of top-down induction of decision trees. Since the human manual approach for designing tailor-made split criteria for every emerging application domain of decision trees would be unfeasible, we believe that the evolutionary search of ESC-GE constitutes a robust and efficient solution for the problem.

We performed a thorough experimental analysis in which the split criterion designed by ESC-GE was compared to state-of-the-art split criteria information gain [22] and gain ratio [23] in 20 public gene expression datasets. We assessed the performance of ESC-GE's evolved criterion through 2 different performance measures, accuracy and F-Measure, and also a complexity measure, tree size. The experimental analysis suggested that ESC-GE can generate a criterion which outperforms both information gain and gain ratio in terms of predictive performance, though generating significantly larger trees. Bearing in mind that an accurate prediction system is widely preferred over a less-accurate (but simpler) system, we believe that ESC-GE arises as an effective alternative for generating tailor-made split criteria for future applications of decision trees.

As future work, we intend to enhance ESC-GE's grammar so it can generate more complex criteria. Also, we intend to develop a multi-objective fitness function, allowing the trade-off between predictive performance and parsimony. Optimizing the evolutionary parameters of ESC-GE is also a topic left for future research.

Acknowledgment

The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) for funding this research.

7. REFERENCES

[1] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and A. A. Freitas. A hyper-heuristic evolutionary algorithm for automatically designing decision-tree algorithms. In *14th Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 1237–1244, 2012.

[2] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and A. A. Freitas. Automatic Design of Decision-Tree Algorithms with Evolutionary Algorithms. *Evolutionary Computation*, 21(4), 2013.

[3] R. C. Barros, M. P. Basgalupp, A. A. Freitas, and A. C. P. L. F. de Carvalho. Evolutionary Design of Decision-Tree Algorithms Tailored to Microarray Gene Expression Data Sets. *IEEE Transactions on Evolutionary Computation*, in press, 2014.

[4] R. C. Barros, A. T. Winck, K. S. Machado, M. P. Basgalupp, A. C. P. L. F. de Carvalho, D. D. Ruiz, and O. S. de Souza. Automatic design of decision-tree induction algorithms tailored to flexible-receptor docking data. *BMC Bioinformatics*, 13, 2012.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[6] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu. A survey of hyper-heuristics. Technical Report Computer Science Technical Report No. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science and Information Technology, University of Nottingham, 2009.

[7] R. Casey and G. Nagy. Decision tree design using a probabilistic model. *IEEE Transactions on Information Theory*, 30(1):93– 99, 1984.

[8] J. Ching, A. Wong, and K. Chan. Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):641–651, 1995.

[9] R. L. De Mántaras. A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6(1):81–92, 1991.

[10] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[11] M. Gleser and M. Collen. Towards automated medical decisions. *Computers and Biomedical Research*, 5(2):180–189, 1972.

[12] C. Hartmann, P. Varshney, K. Mehrotra, and C. Gerberich. Application of information theory to the construction of efficient decision trees. *IEEE Transactions on Information Theory*, 28(4):565– 577, 1982.

[13] R. Iman and J. Davenport. Approximations of the critical region of the friedman statistic. *Communications in Statistics*, pages 571–595, 1980.

[14] B. Jun, C. Kim, Y.-Y. Song, and J. Kim. A New Criterion in Selection and Discretization of Attributes for the Generation of Decision Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):1371–1375, 1997.

[15] I. Kononenko, I. Bratko, and E. Roskar. Experiments in automatic learning of medical diagnostic rules. Technical report, Jozef Stefan Institute, Ljubljana, Yugoslavia, 1984.

[16] R. Mckay, N. Hoai, P. Whigham, Y. Shan, and M. O Neill. Grammar-based Genetic Programming: a survey. *Genetic Programming and Evolvable Machines*, 11(3):365–396, 2010.

[17] J. Mingers. Expert systems - rule induction with statistical data. *Journal of the Operational Research Society*, 38:39–47, 1987.

[18] J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3(4):319–342, 1989.

[19] M. O'Neill and C. Ryan. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358, 2001.

[20] E. Özcan, B. Bilgin, and E. E. Korkmaz. A comprehensive analysis of hyper-heuristics. *Intelligent Data Analysis*, 12(1):3–23, 2008.

[21] G. L. Pappa and A. A. Freitas. *Automating the Design of Data Mining Algorithms: An Evolutionary Computation Approach*. Springer Publishing Company, Incorporated, 2009.

[22] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[23] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, San Francisco, CA, USA, 1993.

[24] L. Rokach and O. Maimon. Top-down induction of decision trees classifiers - a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(4):476 – 487, 2005.

[25] I. K. Sethi and G. P. R. Sarvarayudu. Hierarchical Classifier Design Using Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(4):441–445, 1982.

[26] C. E. Shannon. A mathematical theory of communication. *BELL System Technical Journal*, 27(1):379–423, 625–56, 1948.

[27] M. Souto, I. Costa, D. de Araujo, T. Ludermir, and A. Schliep. Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, 9(1):497, 2008.

[28] J. Talmon. A multiclass nonparametric partitioning algorithm. *Pattern Recognition Letters*, 4(1):31–38, 1986.

[29] A. Vella, D. Corne, and C. Murphy. Hyper-heuristic decision tree induction. *World Congress on Nature & Biologically Inspired Computing*, pages 409–414, 2010.

[30] D. Wang and L. Jiang. An improved attribute selection measure for decision tree induction. In *4th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 654–658, 2007.

[31] S. S. Wilks. *Mathematical Statistics*. John Wiley & Sons Inc., 1962.

[32] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, October 1999.

[33] X. Zhou and T. Dillon. A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):834–841, 1991.