A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces

Pablo Nascimento da Silva* Alexandre Plastino* Alex A. Freitas†

Abstract

Feature selection methods have been widely adopted to prepare high-dimensional feature spaces for the classification task of data mining. However, in many real-world datasets, the feature space is formed by binary features related via generalization-specialization relationships, also known as hierarchical feature spaces. Although there are many methods for the traditional feature selection problem, methods which properly consider hierarchical features are still very underexplored. In this work, we propose a novel genetic algorithm (GA) for hierarchical feature selection. The proposed GA has two novel hierarchical mutation operators tailored to deal with redundant features in hierarchical feature spaces. The computational experiments show that our proposed approach exhibited better predictive performance than two state-of-the-art hierarchical feature selection methods (SHSEL and HIP) and also than two traditional feature selection methods (ReliefF and CFS).

Keywords: Hierarchical Feature Spaces, Feature Selection, Genetic Algorithms, Classification, Bioinformatics.

1 Introduction

Classification is one of the most important tasks in the data mining field [23]. In this task a previously trained classification model automatically assigns a class label to a new instance, based on the values of its features. In many interesting real-world problems, each instance is formed by a set of hierarchically organized binary features. In other words, in each instance, a feature value is deemed positive (negative) when the property associated with the feature has been (has not been) observed for that instance. Besides, such features are related via generalization-specialization relationships, characterizing hierarchical feature spaces [7, 12, 16, 18, 19, 20, 21]. In a generalization-specialization hierarchy, for any given instance t, if a feature x has positive value in t, denoted $(x = 1)$, then all ancestors of x in the feature hierarchy also have positive value in t. In contrast, if a feature x has negative value in t, denoted $(x = 0)$, then all descendants of x in the feature hierarchy also have negative value in t.

One example of data often characterized by hierarchical feature spaces is biological data [18, 19, 20]. For instance, in this work we address the problem of hierarchical feature selection on datasets of ageing-related genes [2]. Ageing is a complex biological process that affects nearly all animal species, even though it is still poorly understood [14]. However, the increasing amount of available ageing-related data allows the use of data mining methods to discover novel patterns that could improve the understanding of the biological ageing process. In the datasets explored in this work, each instance represents a gene, and each gene is associated with terms derived from the Gene Ontology [17], as described later. In these datasets, a general feature (e.g., reproduction) would be the ancestor of more specific features (e.g., asexual reproduction).

Real-world datasets often have a large number of features, many of which can be redundant (highly correlated with other features) or irrelevant for classification (having no significant correlation with the class variable). The problem of redundant features, in particular, is a recurrent issue in hierarchical feature spaces, since hierarchically related features (i.e., features on the same path within the hierarchy) tend to be highly correlated (redundant) with each other, as will be seen later. Hence, by removing hierarchically redundant features in a data preprocessing phase one can improve the classifier's predictive accuracy, speed up the learning process and improve the interpretability of the classifier.

Existing hierarchical feature selection methods [7, 12, 16, 18, 21] usually find a suitable feature subset by keeping features with good relevance values and removing redundancy among hierarchically related features. However, none of the existing methods employs an effective search method; they just use a simple criterion for removing hierarchically redundant features.

In this work, we introduce a new genetic algorithm (GA) with two mutation operators specifically designed to cope with hierarchically redundant features, in order to increase predictive accuracy. These mutation operators are based on two principles: (i) exploiting generalization-specialization relationships among

*Universidade Federal Fluminense, {psilva, plastino}@ic.uff.br
†University of Kent, a.a.freitas@kent.ac.uk

the features; and (ii) removing hierarchical redundancy among the features. In essence, the proposed mutation operators attempt to reduce the number of hierarchically redundant features by assigning to each feature in a candidate feature subset a different probability of mutation. This probability is determined by the degree of correlation among hierarchically redundant features.

Experiments showed that the proposed GA achieved better predictive accuracy than two traditional and two hierarchical feature selection methods.

The remainder of this work is organized as follows. Section 2 reviews essential concepts of feature selection for classification, hierarchical feature spaces and genetic algorithms. Section 3 describes the related work. Our novel genetic algorithm is introduced in Section 4. Section 5 introduces the two novel mutation operators based on the hierarchy of features. In Section 6, we report the computational results. Finally, Section 7 presents the conclusions.

2 Background

2.1 Feature Selection for Classification The predictive accuracy of classifiers is significantly influenced by the quality of the input features [10, 11]. The main goal of feature selection methods is to increase predictive accuracy by selecting a subset of features that are relevant for classification and non-redundant (with little or no correlation among the features).

Feature selection methods can be divided into embedded, filter and wrapper methods [10, 11]. Embedded methods select features during the training of the classifier; whilst wrapper and filter methods are used in a data preprocessing phase. Filter methods evaluate the quality of a feature subset using specific measures, without using the target classification algorithm. By contrast, wrapper methods evaluate the quality of a feature subset by measuring the predictive accuracy of a classifier built using that subset. Hence, wrapper methods select a feature subset tailored specifically for the target classification algorithm, which increases the chances of maximizing predictive accuracy for that algorithm. We follow the wrapper approach in this work.

2.2 Feature Selection and Hierarchical Spaces We use the following notation. The i-th instance of a dataset D consists of a d-dimensional vector of binary features $(x_{i1}, x_{i2}, \ldots, x_{id})$, $x_{ij} \in \{0, 1\}$ for all $1 \leq j \leq d$. In this work the feature set X of D is a hierarchical feature space, more precisely a Direct Acyclic Graph (DAG), where each vertex (node) represents a feature and each edge represents a generalization-specialization relationship between two features. An edge $(X_a \rightarrow X_b)$ shows that feature X_a is a parent of feature X_b, and conversely X_b is a child of X_a. More generally, a feature X_a is an ancestor (descendant) of a different feature X_b if and only if there is a sequence of edges leading from X_a to X_b (from X_b to X_a) in the feature DAG. In generalization-specialization hierarchies ("IS-A hierarchies"), for each instance t, if a feature x has positive value in t ($x = 1$), then all ancestors of x in the hierarchy also have positive values in t. In contrast, if a feature x has negative value in t ($x = 0$), then all descendants of x in the hierarchy also have negative values in t. Note that IS-A hierarchies lead to hierarchical redundancy among features, since a specific feature value logically implies the values of all its ancestors or descendants, as explained above.

Hierarchical feature selection methods are a special case of feature selection methods that exploit characteristics of the feature DAG to improve the predictive accuracy of classifiers. This is typically done by removing hierarchically redundant features [16, 19].

2.3 Genetic Algorithms (GAs) A GA is a stochastic search method inspired by Charles Darwin's natural evolution theory [15]. A GA works with a population of individuals (candidate feature subsets in this work) that iteratively undergo selection and modification, evolving towards a good solution for a given problem. In essence, a GA works as follows. First, an initial population of individuals is randomly created. Then, the quality of each individual is evaluated by a fitness function. At each generation (iteration), the best individuals (those with the highest fitness values) are selected more often for reproduction. The selected individuals undergo genetic operations, like crossover (which combines parts of two individuals to create a new individual) and mutation (where a small part of an individual is replaced according to a randomly generated value). The reproduction process produces offspring which will replace the parents, creating a new generation of individuals which are expected to be better than the previous generation's individuals. This process is repeated until a stopping criterion (e.g., a fixed number of generations) is satisfied.

3 Related Work

Traditional (non-hierarchical) feature selection methods – e.g., Correlation-based Feature Selection (CFS) [4] and ReliefF [8] – can be employed in hierarchical feature spaces by completely ignoring the structure of the feature hierarchy, i.e., treating the features as a flat set of features. However, this is a naive approach to cope with hierarchically redundant features. When the feature space is hierarchical, intuitively the use of hierarchical feature selection methods is more likely to effectively

cope with hierarchically redundant features, by exploiting the generalization-specialization relationships in the feature hierarchy. Next, we briefly review existing hierarchical feature selection methods.

SHSEL [16] is based on the principle that, if there is an edge between two features in the hierarchy (i.e., one is a parent of the other), in general they are highly correlated and tend to be redundant for classification. Hence, for each pair of features connected by an edge in the hierarchy, SHSEL removes the most specific (child) feature if the correlation between those two features is above a user-defined threshold. Next, using only the remaining features, for each path in the feature hierarchy, SHSEL keeps the features with relevance higher than the average relevance of features in that path. A related method, Greedy Top-Down search strategy (GTD) [12], selects the features with the highest relevance value in each path from each leaf to the root node in the hierarchy. Moreover, Tree-Based Feature Selection (TSEL) [7] has been used in the special case of tree-structured (rather than DAG-structured) features. A recent work showed that SHSEL achieved better results than TSEL and GTD [16]. Thus, TSEL and GTD are no longer considered in this work.

Some hierarchical feature selection methods follow the lazy learning paradigm, selecting a different feature subset for each new test instance to be classified. Such lazy methods are the Select Hierarchical Information-Preserving Features (HIP) method [19], the Select Most Relevant Features (MR) method [19], and the hybrid HIP-MR method [18, 19]. Since the HIP method obtained better results than MR and HIP-MR in [21], hereafter we only consider the HIP method.

For each new instance to be classified, HIP selects the subset of the most specific positive-valued features (which imply their ancestors) and the most general negative-valued features (which imply their descendants). As a result, the values of the features selected by HIP for an instance imply the values of all other features for that instance, so that this method completely removes the hierarchical redundancy in the original feature set. Actually, HIP selects only features whose values are non-hierarchically redundant, i.e., features whose values cannot be inferred from the values of other features. However, HIP has the limitation of not explicitly taking into account the relevance (the degree of correlation with the class variable) of the features.

Note that all above methods follow the filter approach. Our proposed GA seems to be the first hierarchical feature selection method based on the wrapper approach.

In this work, we compare our proposed methods against the state-of-the-art hierarchical feature selection methods HIP and SHSEL, as well as against the traditional feature selection methods CFS and ReliefF.

4 A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces

The problem of redundant features is a recurrent issue in the classification task. In datasets with hierarchical features, the structure of the feature space can be used to mitigate this matter through the elimination of hierarchically connected features. In fact, this removal is performed by all hierarchical feature selection methods proposed so far. However, none of these methods employ an effective heuristic search to deal with this problem. So, in this work, we propose a new genetic algorithm (GA), named Genetic Algorithm for Hierarchical Feature Selection (GA-HFS). We focus on GAs because they perform a global search, less likely to get trapped in local optima than local search methods [3, 15]; and they have been successfully employed in non-hierarchical feature selection [3, 22].

GA-HFS uses new mutation operators to guide the search towards feature subsets with few hierarchically redundant features. These mutation operators use the hierarchical structure of the feature space to determine the value of a biased mutation probability for each feature – i.e., the probability of changing a feature's status from selected to non-selected.

4.1 Individual Representation Individuals are represented by d-dimensional binary vectors (using the value 1 for selected features and 0 for non-selected features), where d is the number of features in the dataset. Figure 1 shows an example of an individual representation. Each letter from A to N represents a feature in the hierarchy, and an edge represents a generalization-specialization relationship – e.g., the edge from A to B shows that A is a parent of B. Nodes in black (white) represent selected (non-selected) features.

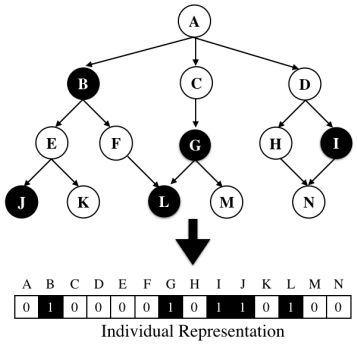

Figure 1: Example of an individual representation used by the proposed GA-HFS.

4.2 Fitness Function This function evaluates the quality of a feature subset. We employ a lexicographic multi-objective fitness function. In this approach, two or more objectives are taken into account to measure the fitness of each individual, where each objective has a distinct predefined priority. GA-HFS' lexicographic fitness function has two objectives: to maximize predictive accuracy (higher priority), measured by the Geometric Mean (GM) of Sensitivity and Specificity [6] of the classifier; and to minimize the number of selected features (lower priority). This latter objective is used only as a tie-breaking criterion in tournament selection – described below. The GM is computed by an internal 5-fold cross-validation procedure on the training set, since we follow the wrapper approach [3].

4.3 Elitism This procedure preserves the ϵ (a parameter) best individuals from the previous generation.

4.4 Selection Procedure At each generation (iteration), tournament selection [15] is used to select individuals to act as parents for the crossover and mutation operators. Tournament selection randomly samples k individuals from the population, where k (the tournament size) is a user-defined parameter. These individuals play a tournament based on the lexicographic multi-objective approach. That is, if an individual has a GM value higher than the others in the tournament, the former is selected as the tournament winner. Otherwise (i.e., the individuals in the tournament have the same GM value), to break the tie, the tournament winner is the individual with the smallest number of selected features. Tournament selection is performed as many times as needed to produce new individuals (for the next generation), until reaching the fixed population size.

4.5 Crossover Operation Each pair of individuals (parents) selected by a tournament can undergo crossover to create two child individuals. GA-HFS uses uniform crossover. For each position in the feature vector of the two parents, this operator randomly decides if the binary values in that position remain the same in each parent or are swapped between the two parents. The crossover operator is performed with a given user-defined probability.

4.6 Mutation Operation GA-HFS has two new biased mutation operators (introduced in the following section), which exploit the hierarchy of features to generate new individuals.

4.7 Stopping criteria GA-HFS runs until a fixed number of iterations has been performed or until the algorithm converges (i.e., there is no difference between the population's highest and lowest fitness values).

5 One Standard Mutation and Two Novel Hierarchical Mutation Operators

Mutation operators randomly replace the value of a gene (indicating whether or not a feature is selected) in an individual. Mutation contributes to more diversity in the population, because it can introduce new gene values that do not occur in any individual of the population. The mutation probability is a user-defined parameter (in general a relatively small value) defining the probability of mutating each gene in an individual. We propose three versions of GA-HFS, using three mutation operators (one operator per GA-HFS version): a standard mutation and two novel ones, as described next.

5.1 Bitwise Mutation This is the most common mutation operator for binary representation. This operator simply flips the bit value of a gene in an individual with a user-defined mutation probability. I.e., it flips a gene value from a selected feature (1) to a non-selected feature (0) or vice-versa.

5.2 Simple Hierarchical Elimination (SHE) Mutation The new Simple Hierarchical Elimination (SHE) mutation operator is a modified version of bitwise mutation with biased mutation probabilities, as explained below. It relies on the assumption that a feature subset with a large amount of hierarchical redundancy often decreases predictive accuracy [12, 16]. The SHE mutation aggressively removes hierarchically redundant features from a feature set. Hence, after applying this operator, the reduced feature subset is expected to have fewer hierarchically redundant features and consequently a higher fitness value.

The SHE mutation is described in Algorithm 1. It works by assigning to each feature f in the input individual a mutation probability value that depends on the selection status of that feature and its ancestors/descendants in the individual. If f is marked as selected and any of f's ancestors/descendants is also selected (involving hierarchical redundancy), then f will mutate with a biased probability (bp) – lines 2 and 3 of the algorithm. If the mutation is applied, it will remove feature f, changing its status from selected to non-selected. If the condition in line 2 is not satisfied, then f will mutate with a standard probability (sp) – lines 4 and 5. Both sp and bp are user-defined parameters, where bp should be higher than sp. Hence, the probability of removing a currently selected feature with at least one currently selected ancestor/descendant is

greater than the probability of changing the status of other features in an individual. During the GA run, the SHE operator is applied many times, reducing the number of redundant features in the individuals at each generation. Therefore, in the long run, it is expected that individuals with relatively few hierarchically redundant features and higher values of fitness will be present in the last generation's population.

Algorithm 1 Simple Hierarchical Elimination (SHE) Mutation

Input : an *individual*, *bp* (biased probability value) and *sp* (standard probability value)
Output: a mutated individual

```
1: for each gene (feature) f ∈ individual do
2:    if f is selected and f has selected ancestors/descendants then
3:       Mutate f with biased probability bp
4:    else
5:       Mutate f with standard probability sp
6:    end if
7: end for
```

5.3 Correlation-based Hierarchical Elimination (CbHE) Mutation The second new mutation operator follows the same basic principle of SHE, i.e., it sets biased mutation probabilities in order to reduce the number of hierarchically redundant selected features and to try to achieve a higher predictive accuracy. Note, however, that although SHE favors feature elimination from a candidate feature subset, it does not consider the actual degree of correlation between features. Hence, the second new operator, named Correlation-based Hierarchical Elimination (CbHE) mutation, attempts to guide the search towards a good candidate solution by setting biased mutation probabilities for the individual's selected features based on the correlation between features in the hierarchy.

CbHE assigns mutation probabilities as follows. First, it assigns to each non-selected feature in the individual a standard probability value. Second, it assigns to each gene marked as selected in the individual a biased mutation probability based on the correlation level between the feature and its ancestors/descendants also marked as selected in the individual.

The basic idea is that if a feature marked as selected in an individual is strongly (weakly) correlated with its ancestors/descendants also marked as selected in that individual, then the probability of mutating the feature's status to non-selected should be high (low). Note that, like in the SHE mutation, the decision to use a biased or standard mutation probability in the CbHE mutation varies not only across features, but also across individuals. The goal is evolving the population towards candidate solutions with few hierarchically redundant features. However, in SHE the biased mutation probability value is fixed throughout the GA run, whereas CbHE dynamically computes the biased mutation probability values in a data-driven way.

CbHE is described in Algorithm 2. For each feature f marked as selected in the individual, CbHE sets a mutation probability value based on the correlation between f and its selected ancestors/descendants. If f is selected, then CbHE calculates the average correlation between f and its selected ancestor/descendant features in the individual (lines 3 to 7). As a measure of correlation, CbHE uses the symmetrical uncertainty coefficient [10], which takes normalized values in the [0,1] range. Then, f undergoes mutation with a biased mutation probability given by the average correlation between f and its selected ancestors/descendants. In other words, a strong (weak) correlation means that there is a high (low) probability that the status of f in the individual will mutate from selected to non-selected – line 8. Note that, if f is selected and none of its ancestors/descendants is selected, then the status of f remains the same (no mutation). In contrast, if f is not selected in the individual, then CbHE assigns to f a standard mutation probability value (line 10).

Algorithm 2 Correlation-based Hierarchical Elimination (CbHE) Mutation

Input : an *individual* and *sp* (standard probability value)
Output: a mutated individual

```
1: for each gene (feature) f ∈ individual do
2:    if f is selected then
3:       corr ← 0
4:       AD ← selected ancestors/descendants
5:       for each feature v ∈ AD do
6:          corr ← corr + Correlation(f,v) / |AD|
7:       end for
8:       Mutate f with biased probability corr
9:    else
10:      Mutate f with standard probability sp
11:   end if
12: end for
```

6 Computational Experiments

6.1 Datasets Following the methodology used in [19, 20], we built 28 datasets of ageing-related genes, involving the effect of genes on an organism's longevity. These datasets were built by integrating data from the Human Ageing Genomic Resources (HAGR) GenAge database (version: Build 17) [13] and the Gene Ontology (GO) database (version: 2015-10-10) [17]. HAGR is a database with information about ageing- and

longevity-related genes in four model organisms: *C. elegans* (worm), *D. melanogaster* (fly), *M. musculus* (mouse) and *S. cerevisiae* (yeast). The GO database provides three ontology types: biological process (BP), molecular function (MF) and cellular component (CC). Each ontology has a separate set of GO terms (features), i.e., a distinct feature hierarchy (a DAG). So, for each of the 4 model organisms, we built 7 datasets, with 7 combinations of feature types (feature hierarchies), denoted: BP, CC, MF, BP.CC, BP.MF, CC.MF, BP.CC.MF.

Hence, each dataset contains instances (genes) from a single model organism. Each instance is formed by a set of binary features indicating whether or not the gene is annotated with each GO term in the GO hierarchy and a binary class variable indicating if the instance is either positive ("pro-longevity" gene) or negative ("anti-longevity" gene) according to the HAGR database. In order to avoid overfitting, GO terms annotated for less than three genes were discarded.

Information about the datasets is shown in Table 1. For each of the 4 model organisms, each of the 7 rows describes a specific dataset. The first and second columns show the organism name and the feature hierarchies used in each dataset. The other columns show the number of features (#F), the number of edges in the feature DAGs (#E), the number of instances (#I), the percentage of positive-class instances (% Pos) and the percentage of negative-class instances (% Neg).

Table 1: Detailed information about the datasets used in the experiments.

Group	Dataset	#F	#E	#I	% Pos	% Neg
C. elegans	BP	991	1707	657	34.40	65.60
	CC	178	277	484	36.36	63.64
	MF	263	331	504	37.70	62.30
	BP.CC	1169	1984	664	34.34	65.66
	BP.MF	1254	2038	663	34.24	65.76
	CC.MF	441	608	566	36.22	63.78
	BP.CC.MF	1432	2315	667	34.33	65.67
D. melanogaster	BP	800	1355	132	71.97	28.03
	CC	89	130	122	70.49	29.51
	MF	146	182	126	70.63	29.37
	BP.CC	889	1485	133	71.43	28.57
	BP.MF	945	1536	133	71.43	28.57
	CC.MF	234	311	130	70.77	29.23
	BP.CC.MF	1034	1666	133	71.43	28.57
M. musculus	BP	1333	2406	109	68.81	31.78
	CC	143	214	107	68.22	31.78
	MF	240	289	106	67.92	32.08
	BP.CC	1475	2619	109	68.81	31.19
	BP.MF	1572	2694	109	68.81	31.19
	CC.MF	382	501	109	68.81	31.19
	BP.CC.MF	1714	2906	109	68.81	31.19
S. cerevisae	BP	844	1511	331	13.29	86.71
	CC	145	230	331	13.29	86.71
	MF	221	277	331	13.29	86.71
	BP.CC	989	1741	331	13.29	86.71
	BP.MF	1065	1788	331	13.29	86.71
	CC.MF	366	507	331	13.29	86.71
	BP.CC.MF	1210	2018	331	13.29	86.71

6.2 Experimental Methodology We implemented all feature selection methods used in this work within the open-source WEKA data mining tool [5]. The Naïve Bayes (NB) from WEKA was used as the classification algorithm to evaluate the quality of the feature subsets selected by each feature selection method. NB was chosen due to its good performance on related work [16, 21] and its fast speed. The predictive accuracy was measured by 10-fold cross validation. The methods were evaluated on 24 datasets, since the 4 datasets with the BP.CC feature hierarchies were used only for tuning the parameters of all methods. Since GAs are stochastic search methods, we run GA-HFS using 10 different random seeds for each of the 10 cross-validation folds (i.e., 100 GA-HFS runs in total), and the reported results are averaged over all those 100 runs.

As shown in Table 1, the majority of the datasets have imbalanced class distributions, so we evaluated the methods' predictive accuracy by using the Geometric Mean (GM) of Sensitivity and Specificity as well as the Area Under the Precision-Recall Curve (AUCPR) measures. GM is defined as follows: $GM = \sqrt{Sensitivity * Specificity}$. Sensitivity is the proportion of positive class instances correctly predicted as positive, whereas Specificity is the proportion of negative class instances correctly predicted as negative [6]. The AUCPR plots the precision of the classifier as a function of its recall, then the area under this curve is used to evaluate the classifier's predictive accuracy (the higher the area, the better) [6].

To determine whether the differences in predictive accuracy are statistically significant, as recommended by Demsar [1], we ran the Friedman test followed by the Nemenyi post-hoc test. First, the Friedman test was executed with the null hypothesis that the accuracies of all methods are equivalent. The alternative hypothesis is that there is a difference between the accuracies of all methods as a whole. If the null hypothesis is rejected, we run the Nemenyi post-hoc test to identify pairs of methods with significantly different accuracies. Both the Friedman and Nemenyi tests were used at the 0.05 significance level.

6.3 Parameter Tuning To tune the parameter settings of all feature selection methods we used the irace tool [9]. To use irace, we selected 4 out of the 28 datasets (one from each model organism). We selected

the 4 datasets with the BP.CC feature hierarchies, since they have a medium number of features. Irace was run with default parameters and a maximum budget of 250. For each feature selection method, the best parameter setting found by irace was used in the experiments to measure predictive accuracy, using the other 24 datasets. Table 2 shows the ranges of parameter settings used by the irace tool. The last three columns show the best parameter setting found by irace for each of the three GA-HFS versions, each with a different mutation operator (Bitwise, SHE and CbHE).

Table 2: Ranges of parameter settings used by irace and the best parameter setting found, for the three GA-HFS versions (each with a distinct mutation type).

GA-HFS				
Parameter	Range	Bitwise	SHE	CbHE
# Population	[50, 150]	138	62	146
# Generations	[50, 150]	80	96	149
Elitism Size	[2,10]	6	2	4
Tourn. Size	[2,10]	4	5	6
Crossover Prob.	[0.70,1.00]	0.93	0.98	0.95
Mutation Prob.	[0.01,0.10]	0.02	0.03	0.06
SHE's Prob.	[0.01,0.30]	–	0.19	–
CbHE's Prob.	auto	–	–	auto

Irace was also used to tune the parameters of ReliefF and SHSEL. These methods have only one parameter to be tuned, making their parameter tuning easier than for GA-HFS. ReliefF's parameter and SHSEL's parameter are thresholds that calibrate the number of features to be removed, and irace considered both parameters' values in the range [0.00,1.00]. The best parameter values found by irace were 0.04 and 0.98 for ReliefF and SHSEL, respectively. HIP and CFS have no parameters to be tuned.

Table 3: GM values (%) obtained by Naïve Bayes with the 8 feature selection approaches.

	Datasets	NoFS	CFS	HIP	ReliefF	SHSEL	GA-HFS	GA-HFS-SHE	GA-HFS-CbHE
C. elegans	BP	62.0	61.3	61.4	51.8	57.9	65.4	**67.2**	64.3
	CC	65.7	63.0	**68.6**	60.3	62.9	66.2	66.5	68.0
	MF	57.6	49.6	50.9	47.9	41.6	61.0	**62.4**	62.2
	BP.MF	61.9	63.5	63.2	61.5	60.1	66.8	**68.6**	65.6
	CC.MF	64.2	61.1	61.4	55.6	56.9	66.2	**67.6**	65.6
	BP.CC.MF	62.4	61.6	63.0	58.3	61.6	66.5	**68.4**	65.7
D. melanogaster	BP	59.4	58.3	66.1	39.8	53.7	64.1	66.5	**67.6**
	CC	66.7	68.1	68.4	51.2	59.1	69.7	71.8	**73.7**
	MF	58.0	52.2	57.5	55.4	46.0	54.7	**60.9**	60.3
	BP.MF	57.3	61.4	**68.1**	43.7	59.0	59.3	63.6	62.8
	CC.MF	65.8	59.1	65.7	57.4	57.1	**68.9**	65.0	67.4
	BP.CC.MF	59.4	59.5	**75.1**	63.8	60.9	60.0	66.2	66.4
M. musculus	BP	59.1	52.8	67.3	59.7	66.1	67.7	**70.4**	69.6
	CC	64.1	50.4	58.3	61.6	55.5	67.7	67.7	**69.7**
	MF	63.5	59.6	65.8	62.1	64.0	66.9	68.4	**70.3**
	BP.MF	64.9	63.6	68.4	59.5	65.1	71.4	71.1	**71.3**
	CC.MF	61.6	55.4	68.1	56.4	61.2	66.8	67.7	**69.2**
	BP.CC.MF	70.2	63.1	70.5	57.4	69.2	**74.4**	73.8	71.8
S. cerevisae	BP	61.5	63.4	68.8	68.2	49.6	68.1	70.0	**70.6**
	CC	57.6	40.9	47.8	57.1	0.00	**61.4**	57.6	**61.4**
	MF	34.2	26.1	42.8	35.6	14.1	31.8	42.0	**44.9**
	BP.MF	62.1	60.0	68.3	66.6	49.8	67.5	**69.5**	**69.5**
	CC.MF	**59.9**	36.8	57.4	51.6	25.1	58.3	55.0	58.9
	BP.CC.MF	62.8	61.4	66.4	**69.6**	53.5	67.7	68.4	68.6
	Avg. Rank	5.0	6.5	3.7	6.3	6.9	3.3	2.3	**1.9**
	#Wins	1.0	0.0	3.0	1.0	0.0	3.0	7.5	**9.5**

{GA-HFS-CbHE,GA-HFS-SHE} $\succ$ {SHSEL,CFS, ReliefF, NoFS} and {GA-HFS,HIP} $\succ$ {SHSEL,CFS,ReliefF}

6.4 Results This section compares the predictive accuracies obtained by Naïve Bayes with 8 feature selection approaches: 3 GA-HFS versions (one with standard bitwise mutation and the two others with a new mutation operator (SHE and CbHE)); two traditional (non-hierarchical) feature selection methods (CFS and ReliefF), two state-of-the-art hierarchical methods (SHSEL and HIP); and, as a baseline, Naïve Bayes using the whole feature set (NoFS).

The results for the measures GM and AUCPR are shown in Tables 3 and 4, where the first column shows the organism name and the feature hierarchies of each dataset. The other columns show the GM or AUCPR values obtained by Naïve Bayes with the aforementioned 8 feature selection approaches. The best results for each dataset are highlighted in bold type.

The last two rows of each table show, for each method, its average rank (Avg. Rank) and number of wins (#Wins). The lower the Avg. Rank, the better the performance of the method. In the row right below Table 3 and Table 4, the symbol $\succ$ means a statistically significant difference between some methods, such that $\{a\} \succ \{b, c\}$ means that a is significantly better than b and c.

Table 3 shows that GA-HFS-CbHE achieved the best average rank and the highest number of wins in terms of GM; whilst GA-HFS-SHE achieved the second best average rank and number of wins. The Friedman test detected a significant difference among the methods, and the Nemenyi test showed that the two GA-HFS versions with novel mutation operators (GA-HFS-CbHE and GA-HFS-SHE) are significantly better than SHSEL, CFS, ReliefF and NoFS. There was no significant difference between those two best GA-HFS versions and GA-HFS using bitwise mutation and the HIP method; but GA-HFS-CbHE and GA-HFS-SHE

Table 4: AUCPR values (%) obtained by Naïve Bayes with the 8 feature selection approaches.

	Datasets	NoFS	CFS	HIP	ReliefF	SHSEL	GA-HFS	GA-HFS-SHE	GA-HFS-CbHE
C. elegans	BP	55.1	55.9	58.4	50.5	56.4	57.9	59.7	**60.1**
	CC	56.3	54.0	57.2	56.5	49.8	56.8	56.6	**59.6**
	MF	50.2	48.0	50.7	50.3	45.6	53.0	**54.7**	54.4
	BP.MF	53.6	55.4	57.0	50.8	54.8	56.9	**59.1**	58.2
	CC.MF	54.8	51.2	54.2	55.7	53.1	56.1	**57.9**	56.4
	BP.CC.MF	54.0	58.1	58.1	54.7	54.7	59.2	**61.3**	60.3
D. melanogaster	BP	83.1	83.3	**87.6**	82.8	82.8	84.7	85.8	85.4
	CC	87.6	87.9	**88.6**	84.6	86.6	86.5	88.0	88.5
	MF	81.9	78.1	82.9	81.9	79.3	81.4	83.3	**84.5**
	BP.MF	84.7	82.8	84.5	79.7	84.2	86.3	**86.5**	85.9
	CC.MF	88.1	87.3	**88.6**	85.2	86.5	87.5	88.2	87.6
	BP.CC.MF	85.4	86.0	87.2	82.2	85.7	86.4	**88.4**	87.5
M. musculus	BP	82.5	82.6	86.0	81.4	85.1	87.1	**89.3**	89.1
	CC	84.5	79.5	80.0	81.2	82.3	**86.7**	**86.7**	86.0
	MF	87.1	86.0	85.5	83.8	86.0	86.7	87.5	**87.9**
	BP.MF	81.7	86.7	87.2	83.7	87.4	88.3	89.7	**90.0**
	CC.MF	85.7	82.5	84.9	82.3	84.9	87.6	88.3	**88.6**
	BP.CC.MF	83.0	86.1	88.0	83.4	87.6	88.6	89.9	**90.0**
S. cerevisae	BP	45.6	48.4	46.3	45.6	45.1	51.2	53.6	**55.1**
	CC	34.0	27.0	30.4	28.6	26.5	**38.8**	38.3	38.1
	MF	26.8	27.8	27.0	31.7	28.2	25.8	34.6	**36.9**
	BP.MF	41.8	47.3	46.0	45.0	45.3	49.3	54.5	**54.9**
	CC.MF	33.9	26.6	32.0	33.1	35.0	38.4	37.9	**39.4**
	BP.CC.MF	44.4	46.1	46.0	49.9	44.3	50.5	**53.9**	53.3
	Avg. Rank	5.8	6.0	4.3	6.6	6.1	3.5	**1.8**	**1.8**
	#Wins	0.0	0.0	3.0	0.0	0.0	1.5	8.5	**11.0**

{GA-HFS-CbHE,GA-HFS-SHE} $\succ$ {HIP,SHSEL,CFS, ReliefF, NoFS}; {GA-HFS} $\succ$ {SHSEL,CFS, ReliefF, NoFS} and {HIP} $\succ$ {ReliefF}

had a much better average rank and number of wins. Moreover, GA-HFS with bitwise mutation and HIP were significantly better than SHSEL, CFS and ReliefF.

Table 4 reports the AUCPR results. GA-HFS-CbHE and GA-HFS-SHE achieved the joint best average rank, but GA-HFS-CbHE had the highest number of wins followed by GA-HFS-SHE. The Friedman test detected a significant difference among the methods. The Nemenyi test showed that both GA-HFS-CbHE and GA-HFS-SHE were significantly better than HIP, SHSEL, CFS, ReliefF and NoFS. Besides, GA-HFS was significantly better than SHSEL, CFS, ReliefF and NoFS. Also, HIP was significantly better than ReliefF. There was no significant difference between GA-HFS with bitwise mutation and the two GA-HFS versions using the hierarchical mutation operators. However, the latter two methods obtained a much better number of wins and average rank than the former.

Figure 2 presents the average percentage of features selected by each method – averaged over the 24 datasets and the 10 cross-validation folds. As expected, comparing the three GA-HFS versions, the percentage of selected features is reduced when a hierarchical mutation operator (SHE or CbHE) is applied. Since GA-HFS-SHE and GA-HFS-CbHE were the two best methods regarding GM and AUCPR, these results broadly support the hypothesis that reducing the number of hierarchically redundant features increases predictive accuracy. Note that, among the three GA mutation types, SHE tends to remove more hierarchically redundant features. Indeed, GA-HFS-SHE selects on average less than half of the features selected by GA-HFS (with traditional bitwise mutation) and about two thirds of the features selected by GA-HFS-CbHE.

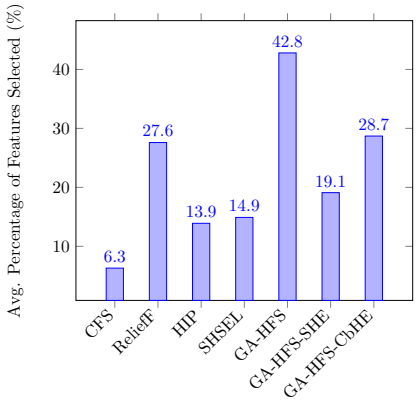

Figure 2: Average percentage (%) of features selected by the 7 feature selection methods.

Finally, we computed the relative frequency (%) of selection of each feature, out of all runs of GA-HFS-CbHE (the best method overall) on all datasets which originally included that feature. We computed these results per organism and for all organisms as a whole. To identify the most relevant features (Gene Ontology (GO) terms) for the biology of ageing, we focus on very frequently selected GO terms, but ignoring very generic, non-informative terms. Briefly, the most frequently selected GO terms include: (a) GO:0016209, "Antioxidant activity", with selection frequency over 90% for organisms yeast and worm; (b) GO:0000003, "Reproduction", with selection frequency over 92% for yeast, worm and fly; (c) GO:0045202, "Synapse" (a structure connecting neurons), with selection frequency over 86% for worm, fly and mouse. These GO terms were ranked 2nd, 4th and 6th, respectively, in terms of relative selection frequencies across all organisms.

7 Conclusions

This work has introduced three versions of a genetic algorithm (GA) for feature selection, including two novel mutation operators tailored for feature selection in hierarchical feature spaces. These two operators are based on the principle that reducing the number of hierarchically redundant features often leads to higher predictive accuracy. The first operator, Simple Hierarchical Elimination (SHE) mutation, sets a fixed biased mutation probability to each feature with hierarchical redundancy, where the probability of removing such features is greater than the probability of changing the selection status of other features. The second mutation operator, Correlation-based Hierarchical Elimination (CbHE), sets the probability of removing a hierarchically redundant feature in a data-driven way, based on the correlation among hierarchically related features.

The experiments compared the predictive accuracy of Naïve Bayes with features selected by 8 different approaches. In summary, the two proposed GAs using the two novel hierarchical mutation operators achieved better predictive accuracies than traditional and state-of-the-art hierarchical feature selection methods. Actually, those two best GAs obtained significantly higher predictive accuracy than 4 or 5 other approaches, depending on the accuracy measure. Also, those two best GAs, using new hierarchical mutation operators, selected overall substantially fewer features than the GA using a non-hierarchical mutation operator.

Acknowledgements This work was supported by CAPES research grant PDSE 88881.132229/2016-01 (P.N. da Silva) and CNPq (A. Plastino). We also acknowledge the support of IC-UFF for access to the "OSCAR" computer cluster.

References

[1] J. Demsar, *Statistical comparisons of classifiers over multiple data sets*, J Mach. Learn. Res. 7, 1–30, 2006.

[2] F. Fabris, J.P. Magalhães and A.A. Freitas, *A review of supervised machine learning applied to ageing research*, Biogerontology 17 (2), 1–8, 2017.

[3] A.A. Freitas, *Data mining and knowledge discovery with evolutionary algorithms*, Springer, 2002.

[4] M.A. Hall, *Correlation-based feature selection for discrete and numeric class machine learning*, In: Intl. Conf. on Mach. Learn. (ICML), pp. 359–366, 2000.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten, *The WEKA data mining software: an update*, ACM SIGKDD Exploration Newsletter 11(1), 10–18, 2009.

[6] N. Japkowicz and M. Shah, *Evaluating learning algorithms: A classification perspective*, Cambridge University Press, 2011.

[7] Y. Jeong and S-H. Myaeng, *Feature selection using a semantic hierarchy for event recognition and type classification*, In: Intl. Joint Conf. on Natural Language Processing (IJCNLP), pp. 136–144, 2013.

[8] I. Kononenko, *Estimating attributes: analysis and extensions of RELIEF*, In: ECML, pp. 171–182., 1994.

[9] M. Lopes-Ibanez, J. Dubois-Lacoste, L.P. Caceres, M. Birattari and T. Stutzle, *The irace package: Iterated racing for automatic algorithm configuration*, Operations Research Perspectives 3, 43–58, 2016.

[10] H. Liu and H. Motoda, *Computational methods of feature selection*, CRC Press, 2008.

[11] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*, Springer, 2012.

[12] S. Lu, Y. Ye, R. Tsui, H. Su, R. Rexit, S. Wesaratchakit, X. Liu and R. Hwa, *Domain ontology-based feature reduction for high dimensional drug data and its application to 30-day heart failure readmission prediction*, In: Intl. Conf. on Collaborative Computing (Collaboratecom), pp. 478–484, 2013.

[13] J.P. Magalhães, A. Budovsky, G. Lehmann, J. Costa, Y. Li, V. Fraifeld and G.M.Church, *The human ageing genomic resources: online databases and tools for biogerontologistis*, Aging Cell 8(1), 65–72, 2009.

[14] J.P. Magalhães, *The biology of ageing: a primer. An introduction to gerontology*, Cambridge University Press, pp. 22–47, 2011.

[15] C.R. Reeves, *Genetic Algorithms*, In: Handbook of Metaheuristics, Springer, 2010.

[16] P. Ristoski and H. Paulheim, *Feature selection in hierarchical feature spaces*, In: DS, pp. 288–300, 2014.

[17] The GO Consortium, *Gene ontology: tool for the unification of biology*, Nat. Gen. 25(1), 25–29, 2000.

[18] C. Wan and A.A. Freitas, *Prediction of the pro-longevity or anti-longevity effect of caenorhabditis elegans genes based on bayesian classification methods*, In: IEEE Intl. Conf. on Bioinformatics and Biomedicine (BIBM), pp. 373–380, 2013.

[19] C. Wan, A.A. Freitas and J.P. Magalhães, *Predicting the pro-longevity or anti-longevity effect of model organism genes with new hierarchical feature selection methods*, IEEE/ACM Trans. Comput. Biol. Bioinform. 12(2), 262–275, 2015.

[20] C. Wan and A.A. Freitas, *Two methods for constructing a gene ontology-based feature network for a bayesian network classifier and applications to datasets of ageing-related genes*, In: ACM BCB, pp. 27–36, 2015.

[21] C. Wan and A.A. Freitas, *An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features*, Artificial Intelligence Review, 2017.

[22] B. Xue, M. Zhang, W.N. Browne and X. Yao, *A survey on evolutionary computation approaches to feature selection*, IEEE Trans. Evol. Comp. 20(4), 606–626, 2016.

[23] M.J. Zaki and W. Meira Jr., *Data mining and analysis: fundamental concepts and algorithms*, Cambridge University Press, 2014.