



# A binary-constrained Geometric Semantic Genetic Programming for feature selection purposes



João Paulo Papa<sup>a,\*</sup>, Gustavo Henrique Rosa<sup>a</sup>, Luciene Patrici Papa<sup>b</sup>

<sup>a</sup> Department of Computing, São Paulo State University, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, Bauru, 17033-360, Brazil

<sup>b</sup> São Paulo Southwestern College, Av. Prof. Célso Ferreira da Silva, 1001, 14-01, Avaré, 18707-150, Brazil

## ARTICLE INFO

### Article history:

Received 9 April 2017

Available online 5 October 2017

### Keywords:

Feature selection

Geometric Semantic Genetic Programming

Optimum-path forest

## ABSTRACT

Feature selection concerns the task of finding the subset of features that are most relevant to some specific problem in the context of machine learning. By selecting proper features, one can reduce the computational complexity of the learned model, and to possibly enhance its effectiveness by reducing the well-known overfitting. During the last years, the problem of feature selection has been modeled as an optimization task, where the idea is to find the subset of features that maximize some fitness function, which can be a given classifier's accuracy or even some measure concerning the samples' separability in the feature space, for instance. In this paper, we introduced Geometric Semantic Genetic Programming (GSGP) in the context of feature selection, and we experimentally showed it can work properly with both conic and non-conic fitness landscapes. We observed that there is no need to restrict the feature selection modeling into GSGP constraints, which can be quite useful to adopt the semantic operators to a broader range of applications.

© 2017 Elsevier B.V. All rights reserved.

## 1. Introduction

Machine learning techniques have been the forerunner of several advances in Computer Science and application-driven areas, which range from medical image understanding to video summarization, just to name a few. Deep learning techniques are now in the spotlight, since they have obtained outstanding results in a number of applications, with performance quite near to the human level.

However, even the most accurate approaches may have their performance (i.e., effectiveness and/or efficiency) degraded due to the high dimensionality of the datasets. In this context, *feature selection* arises to mitigate that problem by selecting the subset of the most representative features, which is somehow modeled as an optimization problem. A common approach is to select the subset of features that maximize some classifier's recognition rate, the so-called *wrapper approaches*. On the other hand, one can use any kind of fitness value that measures the quality of the feature space, such as its separability or compactness.

A number of works modeled the problem of feature selection as a nature-inspired-based optimization task. Nakamura et al. [21] and Rodrigues et al. [30] proposed the Binary Bat Algorithm

for feature selection purposes, being the optimization problem guided by the accuracy of the Optimum-Path Forest (OPF) [24–26] classifier over a validating set. [11] were one of the first to introduce the term *swarm feature selection*, where the well-known Particle Swarm Optimization (PSO) was used to select features in the context of hyperspectral remote sensing image classification. Non-wrapper approaches can be referred to as well, such as the work by [22], which employed evolutionary optimization for feature construction in benchmarking datasets and symbolic learning.

A Binary Cuckoo Search approach was proposed in context of theft detection in power distribution systems [29], and the Binary Flower Pollination Algorithm was also presented for feature selection purposes and compared against PSO, Harmony Search and Firefly Algorithm [31]. Evolutionary-oriented optimization techniques have been also used to find out the most representative features. [38], for instance, used Genetic Algorithms together with Neural Networks for feature selection purposes. Genetic Programming (GP) [17] was also employed for the very same purpose, either representing classifiers instanced with different subsets of features [19,28] or using a two-stage approach [7]. Even further, Grammatical Evolution was also employed under the context of feature construction and selection [12].

Surprisingly, there are a few works that attempted at using GP for feature selection purposes only. Since the idea of using Genetic Programming to select features is plausible and quite simple, we propose here to use only logical operators at the function nodes,

\* Corresponding author.

E-mail address: [papa@fc.unesp.br](mailto:papa@fc.unesp.br) (J.P. Papa).

being the terminal nodes encoded by binary vectors that represent randomly chosen features ('1' = feature selected, and '0' the opposite situation.) This approach concerns our *baseline* for comparison purposes, being the OPF classifier used to guide the optimization process. As far as we are concerned, that is the first time such sort of approach is used for feature selection purposes.

However, the main contribution of this work is related to the Geometric Semantic Genetic Programming (GSGP) technique [20], which encodes the semantic (meaning) of individual trees when performing mutation and crossover operations. GSGP has been employed to a number of problems very recently, such as electoral redistricting problem [6] and real-life applications [35]. One strong point of geometric semantic operators concerns their ability in inducing unimodal fitness landscapes on some problems where one knows the matching between the input and the output data. However, as far as we are concerned, GSGP has never been considered in the context of feature selection up to date, which turns out to be the main contribution of this paper. Additionally, we showed GSGP can also work well in situations where the assumption of unimodal fitness landscapes is not guaranteed in the context of feature selection.

Therefore, the main contributions of this paper are twofold:

- to introduce GSGP in the context of feature selection; and
- to show feature selection can be addressed by GSGP in non-unimodal fitness landscapes.

This paper is an extension of the work by [32], which firstly introduced GSGP for feature selection purposes.

The remainder of the paper is organized as follows. Sections 2 and 3 present the theoretical background related to GSGP and the proposed approach for feature selection purposes, respectively. Section 4 describes the methodology, and Section 5 discusses the experimental results. Finally, Section 6 states conclusions and future works.

## 2. Geometric semantic genetic programming

Genetic Programming [17] is an evolutionary-based optimization algorithm that models each solution as an individual, which is usually represented as a tree composed of *function* and *terminal* nodes. The function nodes encode the arithmetic operators used over the terminal nodes in order to evaluate the trees, and the terminal nodes represent constant values. At each iteration, specific operations over the current population are performed to design the next generation of individuals, being the most used ones: (i) mutation, (ii) crossover and (iii) reproduction. Mutation and crossover aim at allowing a greater variability to the population of individuals, while reproduction tries to maintain the best ones to the next generation. In short, mutation operations change each individual without considering others, i.e., given a mutation point, we can simply generate a new random subtree at that point, while crossover switch branches between two distinct trees.

Geometric Semantic Genetic Programming introduces the concept of semantic operators [20], which can encode the meaning of the *programs* (individual trees/solutions) during the convergence process. On the other hand, standard GP ignore the knowledge about a problem and manipulate the solutions only considering their syntax. In order to cope with this problem, [20] proposed four geometric semantic operators, being two of them related to binary-constrained optimization problems, which is the case of feature selection. Roughly speaking, each possible solution is encoded by a binary array that basically turns on (i.e., the decision variable takes the value '1') or off (i.e., the decision variable takes the value '0') a given bit that corresponds to the presence or absence of some specific feature.

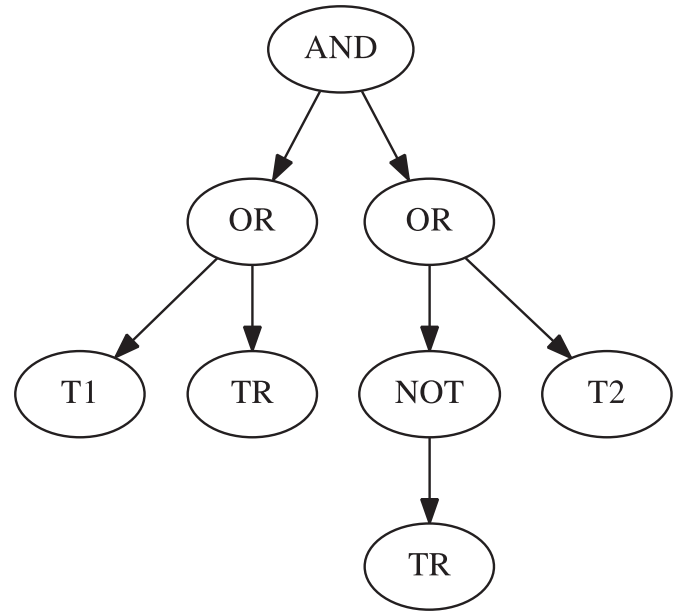


Fig. 1. Offspring generated by means of the semantic crossover defined in Eq. (1).

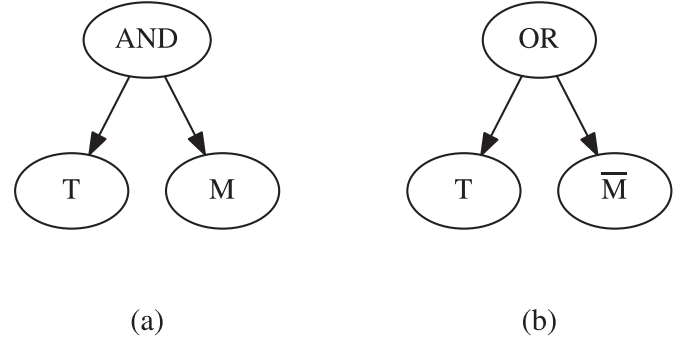


Fig. 2. Tree-like representation concerning the following expressions: (a)  $T \text{ AND } M$ , and (b)  $T \text{ OR } \bar{M}$ .

Let  $T_1$  and  $T_2$  be two logic functions<sup>1</sup>, such that  $T_1, T_2: \{0, 1\}^n \rightarrow \{0, 1\}$ . A geometric semantic crossover operator over  $T_1$  and  $T_2$  outputs the following offspring boolean function:

$$T_3 = (T_1 \text{ OR } T_R) \text{ AND } (\bar{T}_R \text{ OR } T_2), \quad (1)$$

where  $T_R$  is a randomly generated boolean function. Fig. 1 depicts a graphical representation of the offspring function  $T_3$ . The boolean function  $T_R$  can be any tree generated at random that contains only logic function nodes.

Notice that Eq. (1) is a geometric semantic operator when the fitness function used to guide the optimization problem is based on the Hamming distance [20]. A similar definition is also applied to the geometric semantic mutation operator, which states that a given parent function  $T: \{0, 1\}^n \rightarrow \{0, 1\}$  is a semantic mutation operator when the fitness function is based on the Hamming distance [20].

The geometric semantic mutation operator outputs the following boolean offspring  $T_M$ :

$$T_M = \begin{cases} T \text{ AND } M & \text{with probability 0.5} \\ T \text{ OR } \bar{M} & \text{otherwise,} \end{cases} \quad (2)$$

where  $M$  stands for a random minterm of all input variables. Fig. 2 depicts the above formulation in a tree-like structure.

<sup>1</sup> By logic function we mean an "OR" or "AND" operator, for instance.

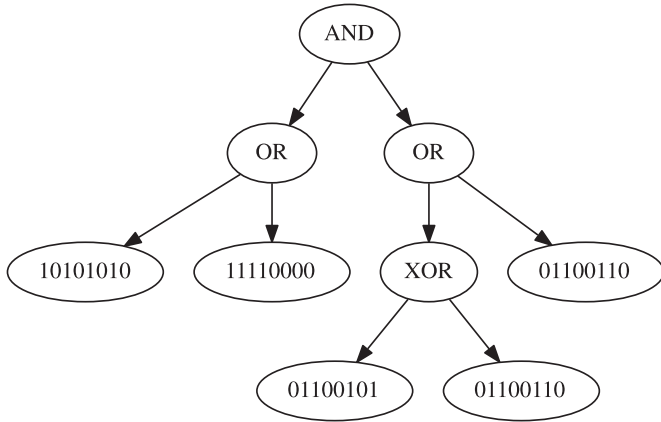


Fig. 3. How to use Genetic Programming in the context of feature selection problem.

### 3. Problem modeling

Feature selection aims at finding the subset of features that maximizes some fitness function, such as the classifier's effectiveness or some quality measure of the feature space. Each possible solution is represented as a binary string, where the symbol '1' means that a given feature is turned on, and the symbol '0' stands for the opposite situation. Concerning Genetic Programming-based feature selection, each terminal node encodes that binary string, while the function nodes encode logical functions, such as logical AND, OR, and XOR (exclusive OR), for instance. Fig. 3 depicts an example of an individual concerning the problem of feature selection.

In the above example, the terminal nodes encode a binary string concerning a problem with 8 features with some standard logical operators. Notice that any other operator with output constrained in  $\{0, 1\}$  can also be used. Since logical operators are usually quite fast to be computed, the process of evaluating a given tree is pretty much standard, thus not requiring a high computational burden. The only restriction concerns the total number of individuals' evaluation: since we have  $2^n$  possible solutions for a problem with  $n$  features, to employ GP for feature selection makes sense when  $n \rightarrow \infty$ , thus becoming unfeasible to compute all the  $2^n$  possible configurations of features.

The evaluation of a given individual (tree) outputs a final binary string, which is then usually employed to map the original dataset to a new one composed of the selected features only. Later, a classifier is trained over that modified dataset for the further classification of the remaining samples.

### 4. Methodology

We propose to model the problem of selecting suitable features by means of a meta-heuristic optimization task. As aforementioned in Section 1, feature selection stands for selecting the most representative features of a given problem, thus reducing its complexity and dimensionality. Roughly speaking, the proposed approach aims at selecting the set of features that minimizes the classification error of some supervised classifier over the validation set (i.e. the so-called wrapper approach). This procedure is hereinafter called "Experiment A".

Although one can use any supervised pattern recognition technique, we opted to use the Optimum-Path Forest (OPF) classifier [24,25], since it is parameterless and fast for training. The OPF is a graph-based technique that models the problem of pattern recognition as a graph partition task, where the nodes encode the feature vectors extracted from the samples, and a complete

graph connects them all. The arcs are weighted by the distance among samples, and a reward-competition process takes place by choosing some key samples from each class called *prototypes*. Such special nodes try to conquer the remaining samples by offering them optimum-paths according to some path-cost function, and the whole process ends up partitioning the graph into optimum-path trees, each one rooted at a prototype node.

However, "Experiment A" does not guarantee a unimodal fitness landscape, since the fitness function is not based on the Hamming distance. In order to fulfill such requirement, we designed the "Experiment B", where four datasets with a reasonable amount of features were chosen to validate GSGP under unimodal fitness landscapes in the context of feature selection. Roughly speaking, the main idea is to find the best subset of features as the one that maximizes the OPF accuracy over a validation set. The best subset is considered among all possible subsets, say  $2^n$ , where  $n$  stands for the number of features. Finally, the best subset is taken as our gold standard, and the fitness function now aims at minimizing the Hamming distance between the current solution and that gold standard. Further, we computed the OPF accuracy over a test set using the subset of features selected by the optimization procedure to evaluate the robustness of the feature selection process.

#### 4.1. Datasets

Table 1 presents the datasets employed in this work. We considered 24 datasets with different number of samples, classes and features to validate the proposed approach under distinct scenarios. The datasets were downloaded from the LibSVM project<sup>2</sup>, being already processed for missing numbers and nominal features are quantized.

Since the optimization process is guided by the results over the validating set, we partitioned the training sets of all datasets in 50% to compose the validating set, and the remaining 50% to be part of the new training set.

#### 4.2. Experimental setup

In this work, we compared Geometric Semantic Genetic Programming against five approaches for feature selection purposes, say that:

- Bat Algorithm (BA);
- Firefly Algorithm (FA);
- Genetic Programming (GP);
- Particle Swarm Optimization (PSO); and
- Baseline OPF classification (i.e., no feature selection has been applied).

In order to provide a statistical analysis by means of Wilcoxon signed-rank test [37], we conducted a 2-fold cross-validation with 15 runnings for both experiments (Experiment "A" and Experiment "B"). We employed 15 agents over 25 iterations for convergence considering all techniques and experiments. Table 2 presents the parameter configuration for each optimization technique<sup>3</sup>. In regard to the source-code, we used the optimization library LibOPT [27]<sup>4</sup>, and the development library LibDEV<sup>5</sup>. Concerning the OPF classifier, we used the LibOPF library<sup>6</sup>.

With respect to BA, we have the minimum and maximum frequency ranges,  $f_{\min}$  and  $f_{\max}$ , respectively, as well as the loudness

<sup>2</sup> <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>.

<sup>3</sup> Notice these values have been empirically chosen.

<sup>4</sup> <https://github.com/jppbsi/LibOPT>.

<sup>5</sup> <https://github.com/jppbsi/LibDEV>.

<sup>6</sup> <https://github.com/jppbsi/LibOPF>.

**Table 1**  
Datasets considered in the work.

	# Training Set	# Testing Set	# Features	# Classes
Adult [34]	1605	30,956	122	2
BASEHOCK [18]	997	996	4862	2
COIL20 [4]	770	770	1024	20
DNA [15]	2000	1186	180	3
Gisette [10]	6000	1000	5000	2
Isolet [10]	3899	3898	617	26
Letter [15]	15,000	5000	16	26
Lung [2]	102	101	3312	5
Madelon [14]	2000	600	500	2
MPEG7_BAS [25]	700	700	180	70
MPEG7_Fourier [25]	700	700	126	70
Mushrooms [33]	4062	4062	112	2
ORL [5]	200	200	1024	40
PCMAC [18]	972	971	3289	2
Pendigits [1]	7494	3498	16	10
Protein [36]	17,766	6621	357	3
Scene [3]	1211	1196	294	6
Segment [15]	1155	1155	19	7
SenseIT [8]	78,823	19,705	100	3
Sonar [13]	104	104	60	2
Splice [23]	1000	2175	60	2
USPS [16]	7291	2007	256	10
Vehicle [15]	423	423	18	4
Yeast [9]	1500	917	103	14

**Table 2**  
Parameter configuration.

Technique	Parameters
BA	$f_{\min} = 0$ , $f_{\max} = 2$ , $A = 0.5$ , $r = 0.5$
FA	$\gamma = 1.0$ , $\beta_0 = 1.0$ , $\alpha = 0.2$
PSO	$c_1 = 1.7$ , $c_2 = 1.7$ , $w = 0.7$

parameter  $A$ , and pulse rate  $r$ . Considering FA, we have  $\alpha$  for computing the randomized parameter, as well as attractiveness  $\beta_0$  and the light absorption coefficient  $\gamma$ . Finally, PSO defines  $w$  as the inertia weight, and  $c_1$  and  $c_2$  as the control parameters. In regard to GSGP and GP parameters, we employed the following configuration:

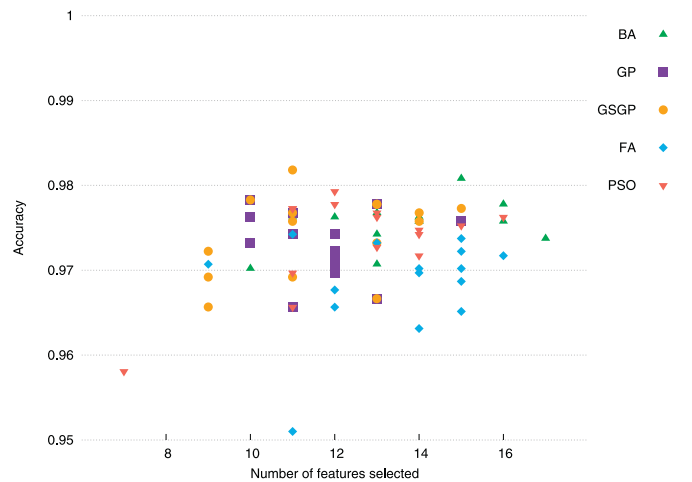
- tree generation: GROW [17] with minimum-depth equal to 2 and max-depth equal to 5;
- reproduction rate: 0.3; mutation rate: 0.3; crossover rate: 0.4;
- function nodes: AND (logical and), OR (logical or), XOR (logical xor) and NOT (logical not);
- terminal nodes: we used 1,000 random generated numbers within the interval of each decision variable to compose the terminal nodes.

## 5. Experiments

In this section, we present the results concerning the aforementioned methodologies. Section 5.1 presents the results concerning “Experiment A”, i.e. feature selection guided by OPF accuracy, while Section 5.2 presents the results regarding “Experiment B”, i.e. the idea is to minimize the Hamming distance between the best current solution and the gold standard.

### 5.1. Experiment A

Table 3 presents the mean accuracy results over the test set considering the OPF classifier. The best results according to Wilcoxon statistical test are in bold. Once can observe GSGP obtained very competitive results in all datasets, being the top technique in 11 out of 24 datasets. From this viewpoint, one can observe



**Fig. 4.** Trade-off between the number of selected features and accuracy over Segment dataset.

GSGP is quite suitable to be employed for feature selection problems even when one can not hold the unimodal fitness landscape assumption. Although baseline OPF achieved the best accuracies in almost all datasets, it obtained the worst results in others, but still being close to the top accuracies. As a matter of fact, it is also known that some features may degrade the classifier’s performance, which justifies that learners designed in some datasets with less features may produce more accurate results.

Table 4 presents the average number of selected features by each technique considered in this work, where the one in bold stands for the technique that selected the smaller number of features. In this context, a feature selection technique usually aims at achieving the best trade-off between the number of features and effectiveness, i.e., a small number of features that allows a reasonable accuracy is then expected. One can observe GSGP selected the smaller number of features in 14 out of 24 datasets, which means 58.33% of the datasets considered in this work.

Fig. 4 presents a pictorial example to help us analyze the trade-off between the number of features and the accuracy over Segment

**Table 3**

Average accuracy over the test set considering all datasets.

	BA	FA	GP	GSGP	PSO	Baseline
Adult	<b>64.56%</b>	<b>65.45%</b>	<b>63.11%</b>	<b>63.98%</b>	<b>64.89%</b>	<b>65.29%</b>
BASEHOCK	<b>79.66%</b>	<b>80.32%</b>	<b>78.93%</b>	<b>79.33%</b>	<b>77.88%</b>	<b>80.43%</b>
COIL20	98.86%	<b>99.21%</b>	98.88%	98.88%	98.85%	<b>99.19%</b>
DNA	75.51%	75.76%	75.24%	75.34%	74.72%	<b>77.85%</b>
Gisette	<b>91.22%</b>	91.15%	91.00%	90.93%	90.90%	<b>91.90%</b>
Isolet	<b>90.98%</b>	90.71%	<b>90.62%</b>	<b>90.76%</b>	<b>90.86%</b>	<b>91.18%</b>
Letter	94.83%	96.17%	93.57%	93.37%	94.75%	<b>97.43%</b>
Lung	87.59%	90.89%	87.04%	88.36%	87.24%	<b>92.41%</b>
Madelon	<b>62.73%</b>	61.73%	<b>62.30%</b>	<b>61.66%</b>	59.36%	<b>64.37%</b>
MPEG7_BAS	<b>89.20%</b>	88.37%	<b>89.19%</b>	<b>89.22%</b>	<b>89.19%</b>	<b>89.17%</b>
MPEG7_Fourier	71.25%	<b>72.14%</b>	70.29%	71.13%	71.13%	<b>72.33%</b>
Mushrooms	<b>96.78%</b>	95.11%	<b>97.63%</b>	<b>95.04%</b>	<b>96.61%</b>	<b>95.43%</b>
ORL	91.95%	92.82%	91.57%	91.69%	91.79%	<b>93.50%</b>
PCMAC	<b>73.45%</b>	<b>73.13%</b>	<b>72.64%</b>	71.59%	<b>72.30%</b>	<b>72.51%</b>
Pendigits	97.98%	97.96%	97.25%	97.10%	97.79%	<b>98.74%</b>
Protein	<b>58.82%</b>	<b>58.69%</b>	<b>58.56%</b>	<b>58.68%</b>	<b>58.49%</b>	<b>59.09%</b>
Scene	79.10%	80.33%	79.25%	78.89%	79.14%	<b>81.09%</b>
Segment	<b>97.53%</b>	96.85%	<b>97.30%</b>	<b>97.43%</b>	<b>97.35%</b>	97.22%
SensIT	72.78%	71.87%	72.66%	72.96%	72.92%	<b>73.42%</b>
Sonar	79.96%	83.38%	<b>81.98%</b>	<b>82.40%</b>	<b>82.87%</b>	<b>84.64%</b>
Splice	<b>72.74%</b>	72.62%	<b>73.25%</b>	<b>73.25%</b>	<b>73.43%</b>	<b>73.35%</b>
USPS	92.53%	92.69%	92.05%	92.14%	92.05%	<b>94.00%</b>
Vehicle	<b>77.09%</b>	76.73%	<b>77.03%</b>	76.58%	<b>77.91%</b>	<b>77.68%</b>
Yeast	<b>57.12%</b>	<b>57.33%</b>	<b>56.93%</b>	<b>56.84%</b>	<b>56.89%</b>	56.67%

**Table 4**

Average number of best features found over the validation set considering all datasets.

	BA	FA	GP	GSGP	PSO	Baseline
Adult	80.13	78.93	76.47	<b>75.27</b>	82.47	122
BASEHOCK	3,143.53	3,178.13	3,080.53	<b>3,063.13</b>	3,154.53	4862
COIL20	665.07	662	655.60	<b>654</b>	660.13	1024
DNA	115.07	118.47	112.80	<b>111.67</b>	115.73	180
Gisette	3,245.53	3,233.87	<b>3,097.23</b>	3,127.53	3,233.63	5000
Isolet	406.80	402.47	397.33	<b>389.40</b>	396.67	617
Letter	11.80	12.53	<b>10.67</b>	10.73	11.33	16
Lung	2147	2,150.04	2,097.53	<b>2,078.33</b>	2,151.53	3312
Madelon	320.53	322.53	<b>313.33</b>	319	320.47	500
MPEG7_BAS	119.40	119.40	118.20	<b>116.40</b>	118.33	180
MPEG7_Fourier	82	84.67	<b>77.93</b>	79.73	82.13	126
Mushrooms	74	74.80	<b>71.27</b>	72.60	72.13	112
ORL	668.40	664.27	655.07	<b>651.20</b>	663.87	1024
PCMAC	2,133.13	2,143.27	2,064.53	<b>2,058.73</b>	2115	3289
Pendigits	12.07	12.53	10.80	<b>10.20</b>	11.73	16
Protein	235.33	231.27	226.07	<b>225.07</b>	231.73	357
Scene	188.53	192.20	185.87	187	<b>185.13</b>	294
Segment	13.47	13.40	<b>11.73</b>	<b>11.73</b>	12.47	19
SensIT	66.20	65.20	<b>60.93</b>	62.67	64.80	100
Sonar	38.67	38.87	37.8	<b>36</b>	39.87	60
Splice	41.60	39.80	<b>37.73</b>	38	38.53	60
USPS	168.20	164.33	<b>163.80</b>	165	165.47	256
Vehicle	12.87	12.87	12	<b>11.93</b>	12.8	18
Yeast	68.87	67.80	<b>66.33</b>	68.67	68.80	103

dataset. Since we considered a 2-fold cross-validation with 15 runnings for all datasets, one can observe 15 points for each technique, being the idea to have those points at the upper-left corner of the chart, i.e. one aims at obtaining the best recognition rates with the smallest number of features. One can observe GSGP consistently obtained very much suitable trade-offs, followed by naïve GP.

Fig. 5 shows a possible bottleneck of GSGP, which is related to its computational burden. On average, GSGP required much less computational load when compared to BA, for instance, but some peaks can be observed at runnings #6, #10 and #15, for instance. Since we are using trees with maximum depth equals to 5 concerning both GP and GSGP, the evaluation of each individual (tree) does not take too much time, which makes GP slight faster than BA and PSO, for instance. However, with respect to GSGP, when we perform the semantic crossover (Eq. (1)) and mutation (Eq. (2)) operators, one has the additional complexity to evaluate that new

tree defined by the operators. Fig. 6 displays the convergence graph among all techniques considered in this work. The fitness value (y label) used for this graph is the minimization of the classification error over the validation set. Note that on this particular dataset, almost everyone reached the same minimum point. Nevertheless, one can observe the FA technique was the fastest one for convergence, followed by PSO, GP, BA, and finally GSGP.

Fig. 7 depicts the number of selected features versus the OPF recognition accuracy over Yeast dataset. In this situation, one can observe more competitive results, where BA appears as a good choice, closely followed by GSGP ( $\approx 68$  features and  $\approx 58\%$  of recognition accuracy). Fig. 8 displays the computational load concerning the feature selection process over Yeast dataset. Once again, one can observe some peaks of higher computational load concerning GSGP, but with reasonable efficiency on average. Finally, Fig. 9 displays the convergence graph among all imple-



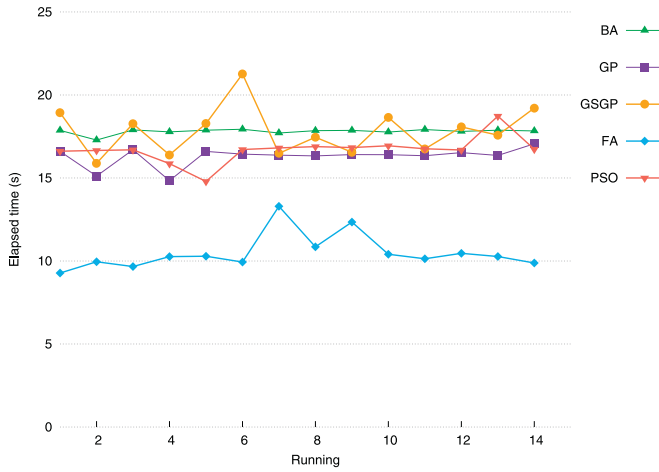


Fig. 5. Computational load concerning the feature selection process over Segment dataset.

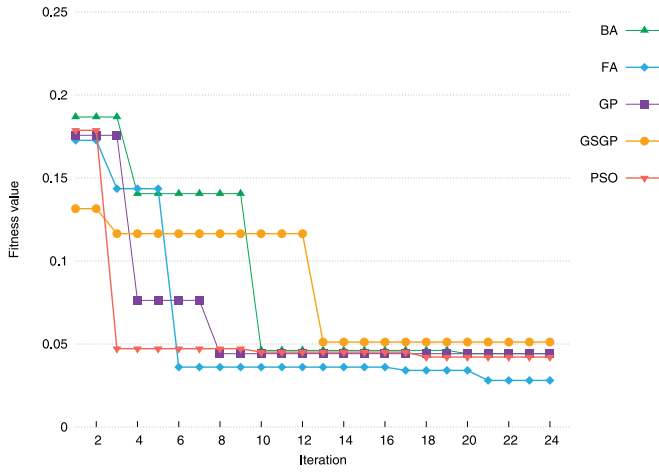


Fig. 6. Convergence graph concerning all implemented algorithms over Segment dataset.

mented techniques. Note that, once again, GSGP has obtained the best convergence among all algorithms.

## 5.2. Experiment B

In this section, we present the results concerning the experiment that holds the assumption the fitness landscape is unimodal, according to Section 2. In regard to this experiment, since we need to find out the gold standard by means of an exhaustive search over  $2^n$  possibilities, where  $n$  stands for the number of features, we opted to use four datasets from the ones presented in Table 1 that have the smaller number of features, say that:

- Letter: 16 features ( $2^{16} - 1 = 65,535$  possibilities);
- Pendigits: 16 features ( $2^{16} - 1 = 65,535$  possibilities);
- Segment: 19 features ( $2^{19} - 1 = 524,287$  possibilities); and
- Vehicle: 18 features ( $2^{18} - 1 = 262,143$  possibilities).

The gold standard subset of features is the one that maximizes the OPF accuracy over the validating set, as displayed in Table 5. In this case, '1' denotes a given feature has been selected, and '0' the opposite situation.

As aforementioned in Section 4, the idea is to find out the subset of features that minimizes the Hamming distance with respect to the gold standard. Table 6 presents the average Hamming distance concerning the aforementioned datasets and the techniques

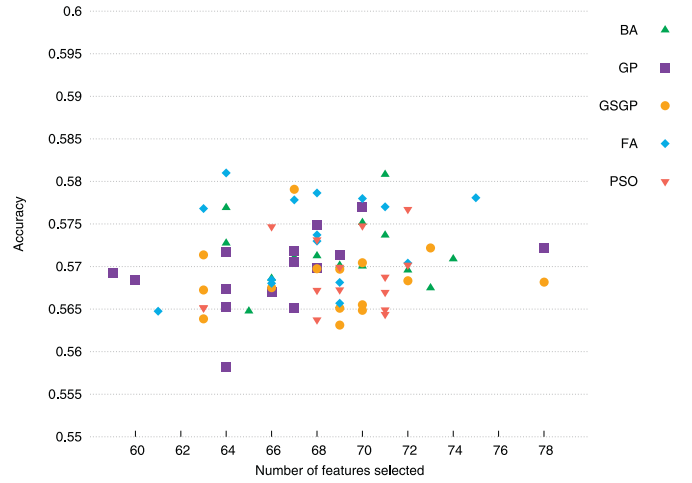


Fig. 7. Trade-off between the number of selected features and accuracy over Yeast dataset.

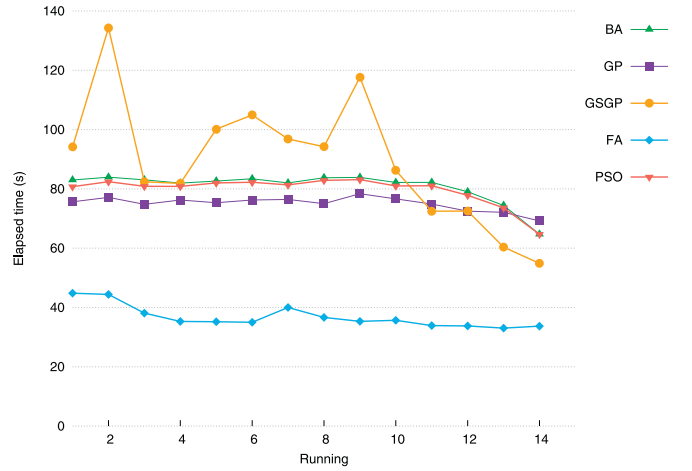


Fig. 8. Computational load concerning the feature selection process over Yeast dataset.

Table 5

Gold standard concerning the datasets used in the "Experiment B".

	Gold standard
Letter	1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
Pendigits	1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1
Segment	1 1 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 0
Vehicle	1 1 1 0 1 1 0 0 1 1 1 1 0 1 0 0 0 0

Table 6

Average Hamming distance considering the datasets used in the "Experiment B".

	BA	FA	GP	GSGP	PSO
Letter	<b>0.71</b>	1.35	2.02	1.87	<b>0.74</b>
Pendigits	<b>2.90</b>	4.07	4.22	4.20	<b>2.91</b>
Segment	<b>2.50</b>	3.76	3.34	3.47	<b>2.66</b>
Vehicle	<b>2.07</b>	2.94	2.95	2.88	<b>1.99</b>

used in the previous experiment. The best results are in bold according to Wilcoxon statistical test. In this situation, the smaller the distance, the better the technique is.

One can observe that BA and PSO obtained the best results, followed by FA, GSGP and GP. However, an interesting point concerns a direct comparison between GSGP and naïve GP, given the first one obtained slightly better results. Table 7 presents the mean

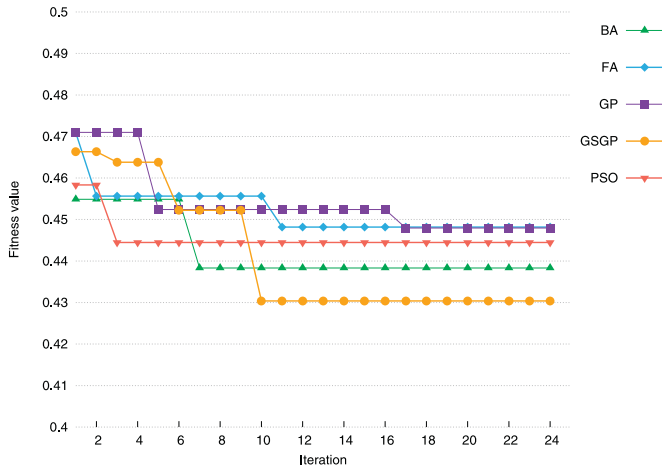


Fig. 9. Convergence graph concerning all implemented algorithms over Yeast dataset.

Table 7

Average elapsed time (s) considering the datasets used in the “Experiment B”.

	BA	FA	GP	GSGP	PSO
Letter	0.0012	0.0036	0.0035	0.5447	<b>0.0010</b>
Pendigits	0.0014	0.0042	0.0036	0.5199	<b>0.0011</b>
Segment	0.0013	0.0057	0.0035	0.5611	<b>0.0011</b>
Vehicle	0.0013	0.0039	0.0035	0.5669	<b>0.0011</b>

Table 8

GSGP comparison between “Experiment A” and “Experiment B”.

	GSGP-A	GSGP-B
Letter	93.37%	<b>97.01%</b>
Pendigits	<b>97.10%</b>	<b>97.74%</b>
Segment	<b>97.43%</b>	<b>97.53%</b>
Vehicle	76.58%	<b>78.74%</b>

computational load for feature selection purposes, being PSO the fastest approach, closely followed by BA, GP and FA. Once again, GSGP appeared as the most costly technique, although its computational burden is quite acceptable.

We performed an additional experiment to evaluate GSGP over both non-unimodal and unimodal fitness landscapes. Table 8 presents a comparison between GSGP under “Experiment A” (GSGP-A) and GSGP under “Experiment B” (GSGP-B), being the best techniques in bold according to Wilcoxon statistical test. In this experiment, we considered the best subset of features selected by both experiments to train and evaluate an OPF classifier in order to assess GSGP behavior under that different conditions, i.e., we would like to assess whether GSGP would benefit or not from unimodal fitness landscapes concerning the problem of feature selection. One can observe that both GSGP-based techniques obtained similar results in 2 out 4 datasets, being GSGP over unimodal fitness the best one in all situations, which was expected, since we assume the operators are “really semantic”. Concerning the Letter dataset, for instance, GSGP-B obtained a considerably more accurate result than GSGP-A. In short, the results showed us GSGP is suitable for feature selection, as well as it can work well in non-unimodal fitness landscapes, thus broadening its number of applications.

## 6. Conclusions

Feature selection appears to be one of the most important problems in machine learning. Despite selecting the subset of features that lead to better recognition rates and lower computational load, to discard some features might be much more important than the classification effectiveness itself. In medical-related data, some features are usually too much expensive to be obtained, or even too much invasive to the patient.

In this paper, we tackled the problem of feature selection as an evolutionary optimization problem, where the idea is to find the subset of features that maximizes/minimizes some fitness function. Specifically, we introduced the Geometric Semantic Genetic Programming for feature selection problems. In this case, we considered two situations:

- the first one aims at maximizing the classification accuracy over a validating set, but such approach does not guarantee one will have a unimodal fitness landscape, which means we can no longer hold the assumption that one is using semantic operators; and
- in the second experiment, we aim at minimizing the Hamming distance between the best solution and the gold standard, thus following the guidelines for using semantic operators [20].

We showed GSGP can obtain very much reasonable results in 24 public datasets without the guarantee one has unimodal fitness landscapes (GSGP-A). A second experiment (GSGP-B) used four datasets in order to obtain a gold standard to be used as the fitness function. Nevertheless, one can notice that finding the gold standard of a dataset is very time-consuming, being infeasible on problems with many features. Thus, the purpose of this experiment was to validate GSGP in the context of feature selection under unimodal fitness functions. In 2 out of 4 datasets, GSGP-A obtained similar results to GSGP-B, being the latter one the best in all four datasets, as expected. We believe the results presented in this paper can make even broader the applications of Geometric Semantic Genetic Programming. In regard to future works, we intend to compare GSGP with different meta-heuristic techniques.

As a take-home message, one can conclude GSGP is suitable for the feature selection problem based on wrapper approaches, as well as GSGP can work well in situations where the assumption of unimodal fitness landscapes can not be held. One possible shortcoming of GSGP is related to its computational load, which is slightly heavier (on average) than the compared techniques.

## Acknowledgment

The authors would like to thank FAPESP grants #2010/15566-1, #2013/07375-0, #2014/16250-9, #2014/12236-1, #2015/25739-4, and #2016/19403-6, Capes and CNPq grant #306166/2014-3.

## References

- [1] F. Alimoglu, E. Alpaydin, Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition, in: *Proc. of the 5th Turkish Artificial Intelligence and Artificial Neural Networks Symposium*, Citeseer, 1996.
- [2] A. Bhattacharjee, W.G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, et al., Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses, *Proc. National Acad. Sci.* 98 (24) (2001) 13790–13795.
- [3] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, *Pattern Recognit.* 37 (9) (2004) 1757–1771.
- [4] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (8) (2011) 1548–1560.
- [5] D. Cai, X. He, Y. Hu, J. Han, T. Huang, Learning a spatially smooth subspace for face recognition, in: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–7.

- [6] M. Castelli, R. Henriques, L. Vanneschi, A geometric semantic genetic programming system for the electoral redistricting problem, *Neurocomputing* 154 (2015) 200–207.
- [7] R.A. Davis, A.J. Charlton, S. Oehlschlager, J.C. Wilson, Novel feature selection method for genetic programming using metabolomic 1h {NMR} data, *Chemom. Intell. Lab. Syst.* 81 (1) (2006) 50–59.
- [8] M.F. Duarte, Y.H. Hu, Vehicle classification in distributed sensor networks, *J. Parallel Distrib. Comput.* 64 (7) (2004) 826–838.
- [9] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: *Advances in neural information processing systems*, 2001, pp. 681–687.
- [10] M.A. Fanty, R. Cole, Spoken letter recognition, in: *NIPS*, 1990, p. 220.
- [11] H.A. Firpi, E. Goodman, Swarmed feature selection, in: *33rd Applied Imagery Pattern Recognition Workshop*, IEEE Computer Society, Washington, DC, USA, 2004, pp. 112–118.
- [12] D. Gavrilis, I.G. Tsoulos, E. Dermatas, Selecting and constructing features using grammatical evolution, *Pattern Recognit. Lett.* 29 (9) (2008) 1358–1365.
- [13] R.P. Gorman, T.J. Sejnowski, Analysis of hidden units in a layered network trained to classify sonar targets, *Neural netw.* 1 (1) (1988) 75–89.
- [14] I. Guyon, S. Gunn, A. Ben-Hur, G. Dror, Result analysis of the nips 2003 feature selection challenge, in: *Advances in neural information processing systems*, 2004, pp. 545–552.
- [15] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, *IEEE Trans. Neural Netw.* 13 (2) (2002) 415–425.
- [16] J.J. Hull, A database for handwritten text recognition research, *IEEE Trans Pattern Anal Mach Intell* 16 (5) (1994) 550–554.
- [17] J. Koza, *Genetic programming: on the programming of computers by means of natural selection*, The MIT Press, Cambridge, USA, 1992.
- [18] K. Lang, Newsweeder: Learning to filter netnews, in: *Proceedings of the 12th international conference on machine learning*, 1995, pp. 331–339.
- [19] J.-Y. Lin, H.-R. Ke, B.-C. Chien, W.-P. Yang, Classifier design with feature selection and feature extraction using layered genetic programming, *Expert Syst. Appl.* 34 (2) (2008) 1384–1393.
- [20] A. Moraglio, K. Krawiec, C.G. Johnson, *Geometric Semantic Genetic Programming*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 21–31.
- [21] R.Y.M. Nakamura, L.A.M. Pereira, K.A. Costa, D. Rodrigues, J.P. Papa, X.S. Yang, BBA: A Binary Bat Algorithm for Feature Selection, in: *25th SIBGRAPI Conference on Graphics, Patterns and Images*, 2012, pp. 291–297.
- [22] K. Neshatian, M. Zhang, P. Andrae, A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming, *IEEE Trans. Evol. Comput.* 16 (5) (2012) 645–661.
- [23] M.O. Noordewier, G.G. Towell, J.W. Shavlik, Training knowledge-based neural networks to recognize genes in dna sequences, *Adv Neural Inf. Process. Syst.* 3 (1991) 530–536.
- [24] J.P. Papa, A.X. Falcão, V.H.C. Albuquerque, J.M.R.S. Tavares, Efficient supervised optimum-path forest classification for large datasets, *Pattern Recognit.* 45 (1) (2012) 512–520.
- [25] J.P. Papa, A.X. Falcão, C.T.N. Suzuki, Supervised pattern classification based on optimum-path forest, *Int. J. Imaging Syst. Technol.* 19 (2) (2009) 120–131.
- [26] J.P. Papa, S.E.N. Fernandes, A.X. Falcão, Optimum-path forest based on k-connectivity: theory and applications, *Pattern Recognit. Lett.* 87 (2017) 117–126.
- [27] J.P. Papa, G.H. Rosa, D. Rodrigues, X.-S. Yang, Libopt: an open-source platform for fast prototyping soft optimization techniques, (2017). <http://adsabs.harvard.edu/abs/2017arXiv170405174P>.
- [28] R. Ramirez, M. Puiggros, *A Genetic Programming Approach to Feature Selection and Classification of Instantaneous Cognitive States*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 311–319.
- [29] D. Rodrigues, L.A.M. Pereira, T.N.S. Almeida, J.P. Papa, A.N. Souza, C.O. Ramos, X.-S. Yang, BCS: a binary cuckoo search algorithm for feature selection, in: *IEEE International Symposium on Circuits and Systems*, 2013, pp. 465–468.
- [30] D. Rodrigues, L.A.M. Pereira, R.Y.M. Nakamura, K.A.P. Costa, X.-S. Yang, A.N. Souza, J.P. Papa, A wrapper approach for feature selection based on bat algorithm and optimum-path forest, *Expert Syst. Appl.* 41 (5) (2014) 2250–2258.
- [31] D. Rodrigues, X.-S. Yang, A.N. Souza, J.P. Papa, *Recent Advances in Swarm Intelligence and Evolutionary Computation*, Springer International Publishing, Cham, pp. 85–100.
- [32] G.H. Rosa, J.P. Papa, L.P. Papa, Feature selection using geometric semantic genetic programming, in: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, in: *GECCO '17*, ACM, New York, USA, 2017, pp. 253–254.
- [33] J.C. Schlimmer, *Concept Acquisition Through Representational Adjustment*, 1987 Ph.D. thesis. AAI8724747
- [34] B. Schölkopf, C.J. Burges, *Advances in Kernel Methods: Support Vector Learning*, MIT press, 1999.
- [35] L. Vanneschi, S. Silva, M. Castelli, L. Manzoni, *Geometric Semantic Genetic Programming for Real Life Applications*, Springer New York, New York, NY, pp. 191–209.
- [36] J.-Y. Wang, *Application of support vector machines in bioinformatics*, Ph.D. thesis, National Taiwan University, 2002.
- [37] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics Bulletin* 1 (6) (1945) 80–83.
- [38] J. Yang, V.G. Honavar, Feature subset selection using a genetic algorithm, *IEEE Intell. Syst.* 13 (2) (1998) 44–49.