Neurocomputing 273 (2018) 554–569

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# A Genetic Programming approach for feature selection in highly dimensional skewed data

Felipe Viegas[b], Leonardo Rocha[a,*], Marcos Gonçalves[b], Fernando Mourão[a], Giovanni Sá[a], Thiago Salles[b], Guilherme Andrade[b], Isac Sandin[a]

[a] Department of Computer Science, Universidade Federal de São João del Rei, São João del Rei, MG, Brazil
[b] Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil

ARTICLE INFO

*Article history:*
Received 24 March 2016
Revised 23 August 2017
Accepted 30 August 2017
Available online 7 September 2017

Communicated by H. Zhang

*Keywords:*
Feature selection
Classification
Genetic Programming

ABSTRACT

High dimensionality, also known as the curse of dimensionality, is still a major challenge for automatic classification solutions. Accordingly, several feature selection (FS) strategies have been proposed for dimensionality reduction over the years. However, they potentially perform poorly in face of unbalanced data. In this work, we propose a novel feature selection strategy based on Genetic Programming, which is resilient to data skewness issues, in other words, it works well with both, balanced and unbalanced data. The proposed strategy aims at combining the most discriminative feature sets selected by distinct feature selection metrics in order to obtain a more effective and impartial set of the most discriminative features, departing from the hypothesis that distinct feature selection metrics produce different (and potentially complementary) feature space projections. We evaluated our proposal in biological and textual datasets. Our experimental results show that our proposed solution not only increases the efficiency of the learning process, reducing up to 83% the size of the data space, but also significantly increases its effectiveness in some scenarios.



## 1. Introduction

The organization and extraction of useful knowledge from the huge amount of data available in distinct applications is one of the biggest challenges in Computer Science nowadays. Machine learning (ML) techniques, such as Automatic Document Classification (ADC), have demonstrated to be a viable path towards facing such challenges. Particularly, ADC techniques aim at building effective models to associate documents with well-defined semantic categories in an automated way. ADC techniques are the core component of many important applications such as spam filtering [1], organization of topic directories [2], identification of writing styles or authorship [3], delayed chaotic systems [4], digital analysis [5,6], among many others.

ADC methods usually exploit a supervised learning paradigm [7], which learns a model for classifying new unseen examples, given a set of previously labeled examples. More formally, given a training set $\mathbb{D}_{train} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, in which $x_i$ is a feature vector that represents an example and $y_i$ its class, the objective is to learn a classification model that predicts each label of $\mathbb{D}_{test} = \{(x'_1, ?), \ldots, (x'_k, ?)\}$. There exists a wide range of proposed algorithms for ADC, although several challenges continue to receive significant attention from the research community [8,9].

In this paper, we deal with two major challenges for ADC: (i) high dimensional classification with low density of examples (i.e., sparse input spaces), also known as the curse of dimensionality; and (ii) skewed data classification (i.e., class imbalance problem). We deal with both problems *at the same time*. Learning with high dimensionality (also known as the $p \gg N$ problem—where $p$ denotes the input space dimension (i.e., the dimensionality of the feature vectors) and $N$ is the number of training examples) in sparse input spaces is undoubtedly one of the major challenges in machine learning research. As the dimensionality $p$ of the input space increases, the number of labeled instances required to produce proper models also increases, but in an exponential fashion [10]. Such requirement becomes a critical factor that limits the applicability of learning techniques in real world problems. Indeed, when learning with highly dimensional data, the identification of patterns in the input space may become a complex task, motivating the prior use of dimension reduction techniques [11,12].

* Corresponding author.
*E-mail addresses:* frviegas@dcc.ufmg.br (F. Viegas), lcrocha@ufsj.edu.br (L. Rocha), mgoncalv@dcc.ufmg.br (M. Gonçalves), fhmourao@ufsj.edu.br (F. Mourão), giovannisa@ufsj.edu.br (G. Sá), tsalles@dcc.ufmg.br (T. Salles), gnandrade@dcc.ufmg.br (G. Andrade), isac@ufsj.edu.br (I. Sandin).

http://dx.doi.org/10.1016/j.neucom.2017.08.050
0925-2312/

Several dimensionality reduction strategies have already been proposed, such as the feature selection methods. A number of feature selection metrics have been explored to assess the features' discriminative power (and ultimately guide the selection process), such as Information Gain [13,14], $\chi^2$ [9], Odds-Ratio [15], among others. Such metrics estimate a score for each feature (i.e., dimensions of the input space) in order to measure its importance for pattern discrimination [9]. Hence, an effective selection of features contributes not only to the learning efficiency, reducing memory and processing demands, but also to the learning effectiveness, as less informative or noise features are filtered out.

In addition to the curse of dimensionality problem, real world data are often skewed. Skewness happens when the number of examples belonging to one class largely outnumbers the other classes, causing some learning algorithms to be biased towards the largest class. Besides hampering learning effectiveness, skewed data also makes feature selection harder: typical feature selection strategies become sub-optimal (or even prejudicial to classification effectiveness), by exacerbating the class imbalance effect on classification, due to a larger bias towards the majority class. This problem has been receiving an increasing attention by the ML community due to its wide spread in real world problems [8]. For instance, it has been shown [16] that metrics commonly used for feature selection are biased and may not work properly with skewed data, making learning tasks even more challenging. To the best of our knowledge, there is no agreement towards a proper use of feature selection solutions in such scenario, motivating us to combine widely used metrics in order to come up with a better selection strategy in face of skewed data. As argued in [9], "feature selection should then be relatively more important in difficult, high-skew situations".

The main hypothesis of our work is that existing feature selection metrics have some biases, thus producing distinct projections over the feature space, based on different criteria, which can be influenced by several factors, such as data skewness. In this sense, instead of combining the metrics themselves in a exhaustive way, in order to reach a richer set of discriminative features, we here propose a Genetic Programming based strategy to combine these projections on sparse, highly dimensional skewed data. Hence, instead of combining the metrics themselves, we combine these projections using GP. Our solution proposes to seek throughout the space of possible combinations of a set of features selected by basic metrics (for instance, Information Gain, $\chi^2$, Odds Ratio) to determine an unbiased estimator of the discriminative power of the features.

By means of the proposed GP-based feature selection approach, we judiciously combine several feature space projections, optimizing for classification accuracy, and ensuring that we properly capture the strongest (and with highest class discrimination capabilities) feature-class relationships that otherwise could be hindered by data skewness. This approach is stronger than traditional single metric approaches since it takes the advantages of all considered metrics while, at the same time, minimizes the influence of their drawbacks, in a pure automatic way. This strategy not only avoids the problem of weighting and combining numeric values ranging on distinct scales but also avoids poor selection of features due to data skewness.

Finally, we stress here that, due to the general nature of our GP-based feature selection approach, it is resilient to some characteristics of the data under analysis, specially in terms of data skewness: it should handle both balanced and imbalanced data in the exact same way. This aspect directly contributes to a wider applicability of the proposal.

In our experiments we adopt the Naive Bayes algorithm, one of the most widely used classification techniques due to its simplicity, lack of parameters and high effectiveness in several tasks. As we are exploring a search for feature combinations by means of a GP-based approach, it is important to have an efficient classifier to evaluate each subset combination. We compare the feature subsets selected by our proposed approach with those selected by each individual feature selection metric and the original feature space.

We evaluate the effectiveness of our approach considering an entire spectrum of data skewness scenarios, ranging from no skewness at all to highly imbalanced settings. More specifically, we assess the effectiveness of our proposed GP-based feature selection approach by considering four textual datasets, namely, Four Universities (4UNI), Reuters (REUT90), 20 Newsgroups (20NG), ACL-BIN and one biological dataset containing examples of p53 proteins classified as cancer suppressor or not. Experimental results on the biological dataset showed that the proposed method can significantly reduce the feature space (up to 83%) while increasing the classification effectiveness when compared to traditional feature selection metrics (with gains of 29% in macro-averaged $F_1$ and 16% in micro-averaged $F_1$ when compared to the best feature selection baseline). Substantial results in terms of dimensionality reduction without compromising effectiveness were also obtained when applying our GP-based feature selection method in the textual datasets.

In sum, our main contribution in this paper is the proposal of an original strategy that drastically reduces the feature space without penalizing, or even improving, the classification effectiveness in scenarios with highly dimensional (including skewed) data. This paper expands and advances our previous work [17] in several ways, including:

- We show that our solution *generalizes* for other domains and datasets. In the original work, we experimented with only one dataset—a biological one. We now introduce new experiments with four new *textual* datasets with different characteristics in terms of size (number of documents), dimension (number of features) and skewness level.
- We introduce a new set of experiments in which we compare the results of several feature selection methods. Such experiments show the complementarity of those methods, better justifying our proposal to combine them.
- We introduce new thorough analyses to better understand the gains achieved by our strategy. Our results demonstrate that our method is the best, when compared to the baselines, for balancing between positive and negative features over all classes, independently of the level of skewness, either in terms of documents or features.

The remainder of this paper is organized as follows. We discuss some related work and briefly describes the explored feature selection metrics that form the basis of our solution in next section. Following, we describe our proposed GP-based strategy to determine an unbiased feature selection metric better suited to handle skewed data sets. Next, we report and discuss our experimental evaluation. Finally, we conclude and discuss some future work in the last section.

## 2. Related work and background

This section starts by introducing formally the feature selection problem. Then, we present the notation adopted in all following discussions. We conclude the section with a review of the main works related to the feature selection problem.

### 2.1. Problem definition

The problem of Dimensionality Reduction [18] can be decomposed into two steps: feature extraction and feature selection. Feature extraction is a preprocessing transformation (for instance,

**Table 1**
Dependency tuples used by feature selection metrics.

| | | Feature | |
|---|---|---|---|
| | | Presence ($t_j$) | Absence ($\overline{t_j}$) |
| **Class** | Inside ($c_i$) | ($c_i$, $t_j$) | ($c_i$, $\overline{t_j}$) |
| | Outside ($\overline{c_i}$) | ($\overline{c_i}$, $t_j$) | ($\overline{c_i}$, $\overline{t_j}$) |

linear or non-linear space embedding) over the feature space [19]. In turn, feature selection aims to select relevant and informative features, considering distinct criteria, such as enhancement of performance or classification effectiveness [20]. The premise is that, usually, datasets contain features that are irrelevant, redundant or noisy and, hence, may be removed without loss of useful information. Indeed, many studies have showed that a proper feature selection may improve efficiency or even effectiveness of learning methods, since they simplify the resulting models and reduce chances of overfitting [21]. Thus, the goal is to filter out the maximum number of 'unnecessary' features from the input space. This filtering process determines feature space projections that represent subsets of features able to better describe the data, by defining a score for each feature in order to assess its discriminative power in the learning task.

Feature selection techniques can be divided into three groups: (1) filter methods, corresponding to strategies that select features without using a learning predictor; (2) wrapper methods, which use learning algorithms as "black boxes" to score a subset of features according to the classification effectiveness; and (3) embedding methods that adopt learning predictors to perform feature selection, but in this case, the selection process is injected into the training of a learning classifier. In [22], a comprehensive study is presented comparing different types of feature selection approaches. The authors show that filters are usually faster than wrappers, although the latter using a simple classification algorithm may be faster than the former. Conversely, wrappers often achieve better classification performance than filters. Also, feature subsets obtained from wrappers allows enhancing the performance of several classification algorithms. Accordingly, we propose a new wrapper method using Genetic Programming to guide the search over all possible feature combinations generated by traditional feature selection metrics.

### 2.2. Adopted notation

In the following discussions, we denote $t$ and $\bar{t}$ as the presence or absence of a feature, respectively. Probabilities $P(c)$ and $P(t)$ indicate the prior occurrence probability of a class and a feature, respectively.[1] Table 1 summarizes the four dependency tuples that may compose the feature selection metrics. These dependency tuples may be represented by conditional or joint probabilities. We denote as "active" the class $c$ wherein we observe the presence of $t$, whereas all other classes are denoted "inactive" ones. Features with higher probability of being observed in an active class are *positive features*, whereas features with higher probability of being observed in any inactive class are *negative features*. Finally, $N$ denotes the number of input examples.

### 2.3. Literature review

We start by reviewing an extensive study regarding twelve feature selection metrics [9]. This work contrasted the effectiveness of distinct metrics using a benchmark of 229 text classification problem instances. Furthermore, the effects of data skewness were studied and it has been shown that standard metrics, such as *Information Gain*, were adversely impacted by it. The authors also proposed an alternative metric, called Bi-normal Separation, which performed consistently better than traditional metrics, even in face of skewed data. However, the original metric was designed to work with discrete feature values and its extension to nominal or real-valued features is not trivial.

In [16], the authors analyzed certain biases associated with the metrics *Information Gain,* $\chi^2$, Odds-Ratio and Correlation Coefficient. According to the authors, feature selection metrics can be grouped into one-sided and two-sided metrics. The first group corresponds to metrics that select the most indicative features of class membership (i.e., positive features), while the second one corresponds to metrics that select features either indicative of class membership or non-membership (i.e., negative features). It has become well accepted that negative features are indeed important for learning accurate models, since they may contribute to a higher recall [9]. Regarding two-sided metrics, when dealing with balanced data sets, both positive and negative features are selected with a proportion similar to the actual distribution observed in the dataset. However, when dealing with skewed data, this behavior does not hold and two-sided metrics tend to become biased towards positive features. Indeed, in [16] the authors showed that the two-sided metrics *Information Gain* and $\chi^2$ are biased towards positive features. Hence, most of the selected features belong to the largest classes (since these classes, typically, have more features than smaller ones). In order to address such issue, the authors proposed a wrapper model to combine explicitly positive and negative features, selected by a base metric, and find an optimal ratio between them. Empirical assessments demonstrated that such wrapper model was able to produce a more effective feature selection. However, no pattern regarding the optimal ratio was found, neither recommended, since it depends on the data set, the learning algorithm and the feature selection metric used as a base metric. Thus, finding such optimal balance between positive and negative features is not a trivial task.

In [23], the authors proposed a feature selection metric to deal with skewed data, by evaluating features considering the Area Under the ROC Curve (AUC). More specifically, the proposed technique learns a linear classifier considering each single feature, evaluating it at distinct boundary regions (that is, varying the classification threshold) and computing the AUC. Each feature is then ranked according to such metric. The authors came up with an unbiased estimator to assess the discriminative power of features that is better suited to handle skewed data. Following the direction of providing unbiased estimators for feature scoring, in [24] the authors proposed some modifications over the previously defined Gini-Index Text (GIT) metric for scoring features in text classification. The authors studied a series of drawbacks of this metric when exposed to skewed data and outlined some modifications in order to come up with an unbiased estimator, which they called Complete GIT feature selection. Such proposal, according to the reported experimental results, leads to a more effective selection of features, being better handled for skewed data.

In [25], the authors proposed an ensemble-based wrapper approach for feature selection from highly imbalanced datasets. The proposed algorithm keeps the advantages of wrapper-based feature selection while maximizing data usage and minimizing selection bias, simultaneously. It does that by training multiple base classifiers with balanced subsets (samples). Following this direction, the authors of [26] studied the challenges of feature selection in imbalanced data with Bayesian learning. They introduced two feature selection approaches to deal with high-dimensional imbalanced data: the Hellinger distance-based method and an approach based on class distributions. In [27], the authors explored both high

[1] When learning models in real domains, usually, all these probabilities are estimated from a training set, through Maximum Likelihood Estimates (MLE).

dimensional and class imbalance issues by introducing a family of methods based on a backward elimination for feature ranking and embedded classification using Support Vector Machines.

In [20], the authors presented some feature selection strategies using heuristic search algorithms, such as, Genetic Algorithm (GA). The heuristic search algorithms evaluate different subsets to optimize the objective function (i.e., classification tasks). Different subsets are generated either by searching around in a search-space or by generating solutions to the optimization problem. Another work [28] proposed feature construction and selection approach for classification tasks using Genetic Programming. Genetic Programming (GP) using a tree-based representation can be used for both feature construction and implicit feature selection. The authors tested different combinations of the constructed and/or selected features and compared them on seven high-dimensional gene expression problems. The results showed that selected sets of features may significantly reduce the dimensionality and maintain or even increase the classification accuracy in most of the cases. Similar work is presented in [29,30]. While in the first one the authors evaluate GP approaches to perform feature selection and classification of mass spectrometry data for detecting diseases and discovering drugs in Bioinformatics, the second work uses a GP based method for diabetes classification.

In [31], the authors proposed a Feature Engineering Wrapper (FEW) that uses Genetic Programming to represent and evolve individual features tailored to the machine learning method with which it is paired. In order to maintain feature diversity, the authors introduced a $\varepsilon$-lexicase survival, a method based on $\varepsilon$-lexicase selection. This survival method preserves semantically unique individuals in the population based on their ability to solve difficult subsets of training cases, thereby yielding a population of uncorrelated features. Experimental results showed that FEW was able to improve model predictions for the evaluated problems.

Recently, other evolutionary computation techniques, such as Particle Swarm Optimization (PSO) and Gray-wolf optimizer, have been adapted to perform feature selection in high dimensional scenarios. In [32], the authors propose three new initialization strategies and three new personal best and global updating mechanisms in PSO to develop novel feature selection approaches with the goals of maximizing the classification performance. In [33], a classification accuracy-based fitness function is proposed based on a gray-wolf optimizer to find an optimal feature subset.

In [34], the authors applied support vector machine with ant colony algorithm for synchronous feature selection and parameter optimization in fault diagnosis of rotating machinery. The failure of rotating machinery can result in fatal damage and economic loss since rotating machinery plays an important role in the modern manufacturing industry. The proposed method was able to find relevant features when compared to other methods in the literature.

A comprehensive survey of the state-of-the-art works on evolutionary computation for feature selection is present in [35]. The authors identify the contributions of different algorithms, presenting some challenges that must be addressed in future work, such as scalability. Most works deal with datasets on which the number of features and the number of instances are significantly increasing. Although the article cites more than two hundred papers, we observe that just one is focused on text mining [36], a challenging scenario and focus of the present work.

Motivated by the challenges of determining an unbiased estimator for feature scoring, which may work well even in highly skewed settings, we combine the projections defined by distinct feature selection metrics, taking into account the characteristics of a variety of widely used basic feature selection metrics. Our proposal is to employ Genetic Programming in order to find a way to combine such projections efficiently and effectively, as discussed next. As presented in [35], only our previous work (which this article greatly extends) exploited such avenue [17].

### 2.4. Traditional feature selection metrics

The relevance and applicability of feature selection metrics in practice have boosted the number of studies on this topic recently [21]. Thus, many distinct metrics have been proposed. Since our work aims at exploiting existing metrics, in this section, we briefly introduce four distinct metrics (one and two-sided) widely used for this task: Information Gain [13,14], $\chi^2$ [9], Odds-Ratio [15] and Correlation Coefficient [13,37], described below.

Information Gain (IG): quantifies how much information we obtain about a class when we know that a certain feature exists or not in a sample. It is a two-sided metric defined as follows:

$$\begin{aligned} IG(t) = &-\sum_{i=1}^{|\mathbb{C}|} P(c_i)\log P(c_i) \\ &+ P(t)\sum_{i=1}^{|\mathbb{C}|} P(c_i|t)\log P(c_i|t) \\ &+ P(\bar{t})\sum_{i=1}^{|\mathbb{C}|} P(c_i|\bar{t})\log P(c_i|\bar{t}), \text{ where} \\ &|\mathbb{C}| \text{ is the number of classes.} \end{aligned} \quad (1)$$

Chi-square $\chi^2$: used in statistical analysis to test whether two events are independent. In the context of feature selection, it is used to measure the association between features and classes. It is two-sided metric defined as:

$$X^2(t, c_i) = \frac{N[P(t, c_i)P(\bar{t}, \bar{c}_i) - P(t, \bar{c}_i)P(\bar{t}, c_i)]^2}{P(t)P(\bar{t})P(c_i)P(\bar{c}_i)} \quad (2)$$

Correlation Coefficient (CC): used to estimate the correlation between classes and the interrelation among features. It is a variation of the Chi-square metric, where $CC^2 = \chi^2$. This metric can be seen as a one-sided Chi-square metric, defined as:

$$CC(t, c_i) = \frac{\sqrt{N}[P(t, c_i)P(\bar{t}, \bar{c}_i) - P(t, \bar{c}_i)P(\bar{t}, c_i)]}{\sqrt{P(t)P(\bar{t})P(c)P(\bar{c}_i)}} \quad (3)$$

Odds Ration (OR): measures the chances of an feature occur in the positive class normalized by that of the negative class. The basic idea is that the features distribution of a relevant document is different from the distribution of documents not relevant. It is a one-sided metric defined as:

$$OR(t, c_i) = \log\frac{P(t|c_i)[1 - P(t|\bar{c}_i)]}{[1 - P(t|c_i)]P(t|\bar{c}_i)} \quad (4)$$

## 3. Combining feature space projections

In this section, we outline our proposal to model the problem of feature selection using Genetic Programming (GP). First, we present the motivation for using such evolutionary approach and then, we describe, in general terms, the operation of a GP algorithm. Finally, we detail our modeling strategy to tackle the feature selection problem.

Each individual feature selection metric may select different sets of features, since they exploit different criteria in their selection process. As outlined in the previous section, these different sets may contain good discriminative features as well as not so relevant ones. Furthermore, good features selected by one method will not necessarily be selected by a different one. Thus, our hope is that, by combining the basic metrics, we might find an unbiased

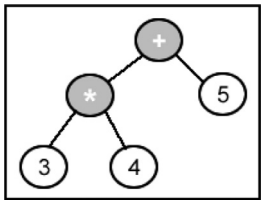



**Fig. 1.** Example individual: $(+\,(*\,3\,4)\,5) = (3*4)+5$.

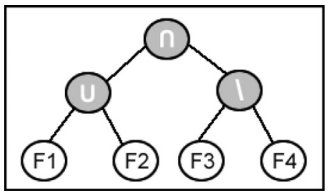



**Fig. 2.** Hypothetical individual under our modeling strategy: $(\cap\,(\cup\, f_1\, f_2)\,(\setminus\, f_3\, f_4)) = (f_1 \cup f_2) \cap (f_3 \setminus f_4)$.

estimator which achieves a better (near-optimal) proportion of selected positive and negative features, retaining the most informative ones. Clearly, the search space over all possible combinations of basic feature selection metrics is very large and a brute force search is not a feasible choice. So, our proposal is to employ GP to guide the search process more efficiently.

GP is an evolutionary algorithm extensively (and successfully) used to solve complex optimization and learning problems, where the search space for an optimal solution is prohibitively large. It works by mimicking the evolution process of a population of individuals, following Darwin's principle of *survival of the fittest*. In such framework, each individual is usually represented by a tree, consisting of terminals (leaf nodes) and non-terminals (functions, for instance, arithmetic or logic operators). In Fig. 1 we illustrate an arithmetic operation represented by a tree.

The evolution of the population is driven by the generation and the combination of its individuals, according to their fitnesses. The following main steps summarize the evolution process of a population in a GP framework:

1. Random generation of individuals (initial population), using the available functions and terminals. A widely used strategy is the so called ramped half-and-half generation [38]. In this strategy half of the individuals are created using full method and the other half are generated using the grow method. Full method and grow method generates individuals, such as the example in Fig. 1, but the trees generated by the full method tend to be richer and more computationally complex than the trees generated by the grow method. The maximum depth of the trees generated is ramped, such that individuals are created in a range of sizes.
2. Iterative generation of a new population by means of the following sub-steps, until a stopping criterion is met:
   (a) Assessment of each individual's fitness, according to the problem at hand (i.e., the quality of the solution represented by it).
   (b) Probabilistic selection (based on fitness) of one or two individuals from the population to participate in the genetic operations detailed in (c). A commonly used strategy is the tournament selection,[2] which will be employed here.
   (c) Creation of a new individual using any of the following genetic operators:
      (i) *Reproduction:* the selected individuals are copied to the new population.
      (ii) *Mutation:* a new individual is created and added to the new population after a random change in some node of the tree.
      (iii) *Crossover:* a new individual is created and added to the new population after recombining parts of two randomly selected individuals (trees).
3. In each population, the best created individual is saved as the result of the iteration. Once the stopping criterion is satisfied, the best program produced is designated as the (perhaps approximate) solution to the problem.

Recall that the problem we aim to solve is to determine subsets of features which better capture the characteristics of the classes, ultimately yielding a more compact representation of the input examples (i.e., a reduced dimensionality) without decreasing (potentially increasing) the effectiveness of learning algorithms when discriminating between classes. To adapt the GP framework to such problem, we model each individual as a possible combination of a set of "basic" feature selection metrics, and let the GP search for individuals which yield the most effective combination of metrics.

More specifically, consider the individual tree representation adopted by the GP framework. Each terminal node (leaf) consists of a ("basic") feature selection metric, which returns a set of features considered highly discriminative by such metric (i.e., a function $f : \mathbb{D} \mapsto \mathbb{S}$, where the input $\mathbb{D}$ is the training set and the output $\mathbb{S}$ is the set of most discriminative features, according to $f$). Two feature selection metrics $f_i$ and $f_j$ (sibling nodes in the tree) are combined according to a set operation specified by their parent node (non-terminal node). The set operations may be union ($\cup$), intersection ($\cap$), set difference ($\setminus$), and so on. Fig. 2 illustrates an hypothetical individual considering our modeling strategy. As one can observe, the subsets of features, selected by feature selection metrics $f_1$ and $f_2$ are combined by operator $\cup$. Moreover, subsets of features, selected by feature selection metrics $f_3$ and $f_4$ are combined by operator $\setminus$. At the end, these two new subsets are combined by $\cap$, generating a final subset of features, which we hope will better contribute not only to the learning efficiency, but also to its effectiveness.

During the GP evolutionary process, the quality of every individual is evaluated according to a pre-defined (problem dependent) fitness function. Recall that our main goal is to come up with a subset of features that, besides aggressively reducing the input dimensionality, yields a better learning effectiveness. Hence, we define the fitness function according to the classification quality, evaluated after filtering the input examples according to the subset of features selected by the individual, for instance, obtained after applying all operators of the individual. The GP framework, thus, tries to maximize the fitness function, generating individuals whose associated feature subsets lead to an improved classification effectiveness. In the following, we present the experimental setup (that is, the explored data set, the set of "basic" feature selection metrics and the adopted classifier) and report the evaluation of our approach.

## 4. Experimental setup

In this section, we describe the experimental setup used to perform our experiments, including the exploited feature selection metrics, the hypothesis demonstration, the adopted classification algorithm, the evaluation metrics, among other practical issues.

[2] Tournament selection involves picking a number of individuals chosen at random from the population and and staging a tournament to determine which one gets selected. The winner of the tournament is the fittest one.

**Table 2**
General information about the datasets.

| Dataset | Size | # Features | Density | Class distribution | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | # Classes | Minor class | 1°quartile | Median | Mean | 3°quartile | Major class |
| 4UNI | 8277 | 40,195 | 139.275 | 7 | 137 | 343 | 930 | 1182 | 1382 | 3759 |
| REUT | 8184 | 24,985 | 42.230 | 8 | 113 | 254.75 | 442 | 1023 | 946.5 | 3930 |
| 20NG | 18,805 | 61,050 | 129.511 | 20 | 628 | 955 | 979 | 94,025 | 990 | 999 |
| ACL-BIN | 27,677 | 1,110,351 | 181.509 | 2 | 13,795 | 13816.75 | 13838.5 | 13838.5 | 13,860 | 13,882 |

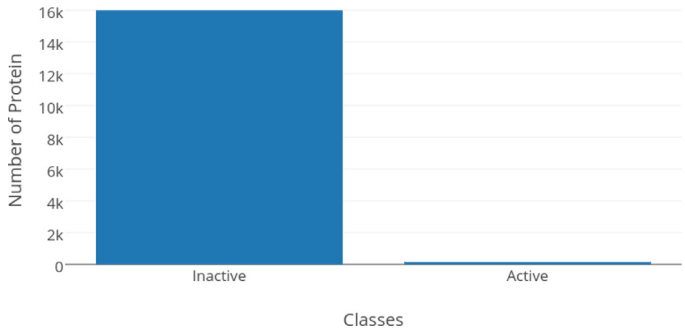



**Fig. 3.** Distribution of classes of the biological dataset.

### 4.1. Datasets

*Biological dataset:* We adopt in our experiment the k8 cancer-rescue mutants data set [39], a highly skewed data set with examples of *p*53 proteins, characterized by 5408 features composed by: 2D electrostatic and surface based features (4826 features) and 3D distance based features (582 features). This dataset is composed by 16,715 samples, classified as active (143 samples) and inactive (16,572 samples), where "active" and "inactive" denotes the functional state of the *p*53 mutants. Note that, when analyzing this data set, the ultimate goal (in a classification setting) is to discriminate the minority (i.e., "active") samples. Fig. 3 illustrates the distribution of classes of the biological dataset.

*Textual datasets:* Considering the textual domain, we used four real-world datasets, namely, 20 Newsgroups, Four Universities, Reuters and ACL-BIN datasets. For all datasets, we performed a traditional preprocessing task: we removed stopwords, using the standard SMART list. Next, we give a brief description of each dataset.

- 4 Universities (4UNI), a.k.a, WebKB: this dataset contains Web pages collected from Computer Science departments of four universities by the Carnegie Mellon University (CMU) text learning group. There is a total of 8277 web pages, classified in 7 categories (such as student, faculty, course and project web pages).
- Reuters (REUT): this is a classical text dataset, composed by news articles collected and annotated by Carnegie Group, Inc. and Reuters, Ltd. We consider here a set of 8184 articles, classified into 8 categories.
- 20 Newsgroups (20NG): this dataset is a dataset containing 18,805 newsgroup documents, partitioned almost evenly across 20 different newsgroups categories. 20NG has become a popular dataset for experiments in text applications of machine learning techniques, such as text classification and text clustering.
- ACL-BIN: this dataset contains documents regarding product reviews from Amazon [40], which are classified as positive or negative. This dataset has approximately 1.110.351 terms, related to four different domains: Books, DVDs, Electronics and kitchenware.

Table 2 shows some of the characteristics of the datasets used in our experiments. The first column indicates the name of the dataset, the second column is the number of documents in the dataset, the third shows the number of features (words) represented in the dataset, the fourth corresponds to the average number of words (density) per document, and the last columns show the class distribution of the dateset.

Fig. 4 illustrates the distribution of classes of the four textual datasets. Notice that the 4UNI and REUT datasets are very skewed and that 20NG has few classes with much less documents that the average of the other classes. Although the ACL-BIN classes are balanced, there is another type of imbalance in it; an *attribute imbalance*. In other words, there are more positive attributes for one class than for the other. We included this dataset to check whether our proposed method can also deal with this type of imbalance.

### 4.2. Exploited feature selection metrics

We used in our analyses all feature selection metrics previously discussed: Information Gain [13,14], $\chi^2$ [9], Odds-Ratio [15] and Correlation Coefficient [13,37]. In our GP modeling, we consider the terminal nodes as features sets selected by such metrics.

These metrics were chosen because they are widely used in the literature and show distinct characteristics. The fact that they are one (CC and OR) and two-sided (IG and $\chi^2$) metrics was also determinant for this choice: this enforces the search for the best set of features to be performed by the GP to consider both the positive and negative features.

### 4.3. Classification algorithm

As we are dealing with continuous features, the automatic classification algorithm used in our experiments was the Gaussian Naive Bayes [10]. This classifier is one of the most widely used techniques, due to its simplicity and efficiency in several scenarios, especially when applied to scenarios in which the attributes are independent, making their "naive" assumption more reliable. Since we are exploiting only univariate feature selection metrics, the subset of attributes selected by these metrics are somehow benefiting the independence assumption of Naive Bayes. During the GP iterations, many classification tasks are executed and evaluated, thus efficiency is paramount. We note here that the proposed method is general enough to be instantiated with another classifier, such as the SVM or kNN (with the drawback that parameter optimization may be required at each iteration, usually a very costly procedure).

### 4.4. Fitness function

As previously mentioned, the fitness function is problem-dependent and must reflect the ultimate goal of the optimization/learning problem. Here, our main goal is to provide a high quality classification. For this purpose, our fitness function must be a metric to assess the classification quality, enabling the GP framework to search for individuals that maximize it. As one of our goals in this paper is to deal with highly skewed data, we

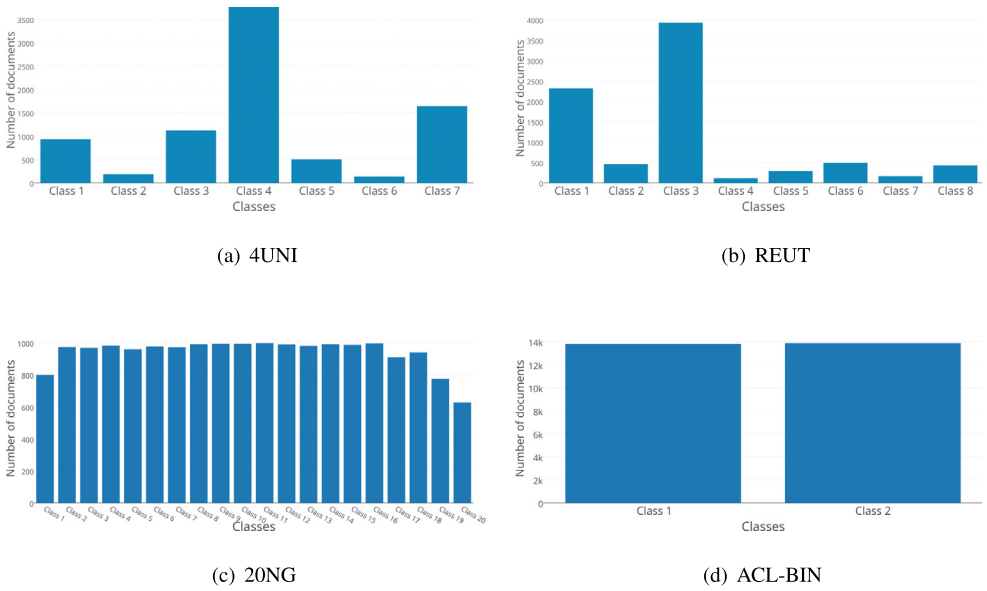



**Fig. 4.** Distribution of classes of the textual datasets.

used the Macro$F_1$ metric to compose the fitness function. In fact, the Macro$F_1$ metric better captures the classification effectiveness for each classes individually (unlike the Micro$F_1$ metric, which assess a global classification effectiveness). The description about the Macro$F_1$ metric can be found in Appendix A. This is important when dealing with skewed data, since the effectiveness in discriminating the minority classes, usually the most important ones for most applications, is taken into account. Since the fitness is a minimization function, we defined as:

$$fitness = 1.0 - \left(\frac{2p_{macro}r_{macro}}{p_{macro} + r_{macro}}\right) \quad (5)$$

On the other hand, on balanced datasets, Micro$F_1$ and Macro$F_1$ become almost equivalent. Therefore Macro$F_1$ covers both cases: skewed and balanced datasets.

## 5. Experimental evaluation

In this section, we describe the proposed experimental design, followed by the analysis and discussion about the experimental results.

### 5.1. Hypothesis demonstration

As mentioned, each feature selection metric exploits different strategies to select the most discriminative feature subspace. To demonstrate our hypothesis, we compare the feature subsets produced by each single feature selection metric in order to show their differences and complementarity. Our goal is to demonstrate the feasibility of their combination.

In order to evaluate the similarity among these subspaces, we measure the Jaccard distance for each pair of feature subsets. The Jaccard distance is measured by the Eq. (6), which is the complement of the Jaccard Index, also known as the Jaccard similarity coefficient. The Jaccard distance measures the distance between two sets, i.e., the greater the Jaccard distance, the more dissimilar these sets are. Fig. 5 illustrates the dissimilarity between the most discriminative feature subspaces selected by each traditional feature selection metric.

$$Jaccard\_Distance(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

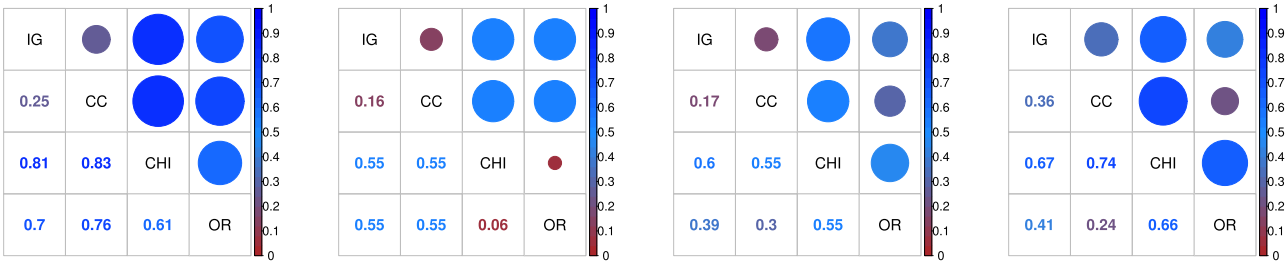
These figures contain circles on the upper diagonal and the actual Jaccard distance on the lower diagonal. The distances located on the lower diagonal are the Jaccard distances between the features selected by each FS metric. These values can range from 0 to 1.0, where values next to 1.0 indicate maximum dissimilarity. For example, observing the Fig. 2(a), the distance 0.83 indicates that features selected by CC and CHI are very dissimilar. The same observation can be made regarding the upper diagonal. The size and color of the circles indicate the level of dissimilarity – big and purple circles indicate high dissimilarity, i.e., indicates distances next to 1.0, while small orange/red circles indicates similarity. Taking the same example of Fig. 2(a), the distance of the features selected by CC and CHI generates a big purple circle.

Going to the results, as mentioned, for the 4UNI dataset, the Jaccard distance between the feature space selected by $\chi^2$ and *CC* is 0.83, meaning that the most discriminative features selected by these two feature selection metrics are quite dissimilar. On the other hand, the features selected by the *GI* metric are somewhat similar to the features selected by the *CC* metric, since the Jaccard distance is 0.25. Note that we are not interested in the relevance order of the selected features, but in the feature subset selected by the traditional metrics. As we can observe, no feature selection metric selected the same feature subset, which shows that there is a divergence among the most discriminative features selected by each traditional metrics. This divergence motivates the use of the GP-based approach, allowing us to explore new feature combinations. Thus, the feature space built by the proposed approach may be more discriminative and effective than the feature space selected by each of the traditional feature selection metrics.

### 5.2. GP experimental evaluation

#### 5.2.1. Experimental design

Fig. 6 illustrates the proposed experimental design whose goal is to contrast the proposed strategy against traditional feature selection metrics. First, we partition the dataset into two equal partitions: *test* and *validation* keeping the original distribution of the classes in each half. The first partition (test) is the set of unseen examples used in the final evaluation of the techniques, whereas the second one (validation) corresponds to examples used by the tuning process (i.e. selection of the best subset and parameters




(a) 4UNI (b) REUT (c) 20NG (d) ACL_BIN

**Fig. 5.** Jaccard distance of the most discriminative feature subset selected by each traditional FS metric. Distances next to 1.0 indicate dissimilarity between the features selected by each FS metric. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

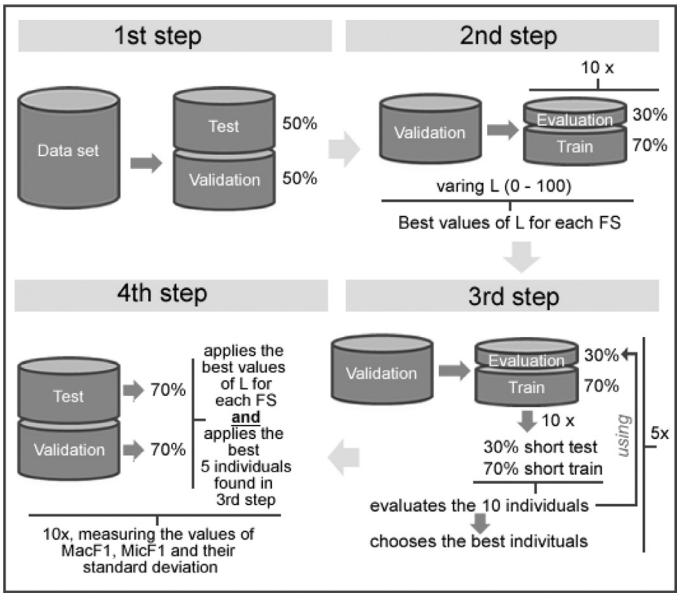



**Fig. 6.** Experimental architecture in detail.

**Algorithm 1** Step 3.

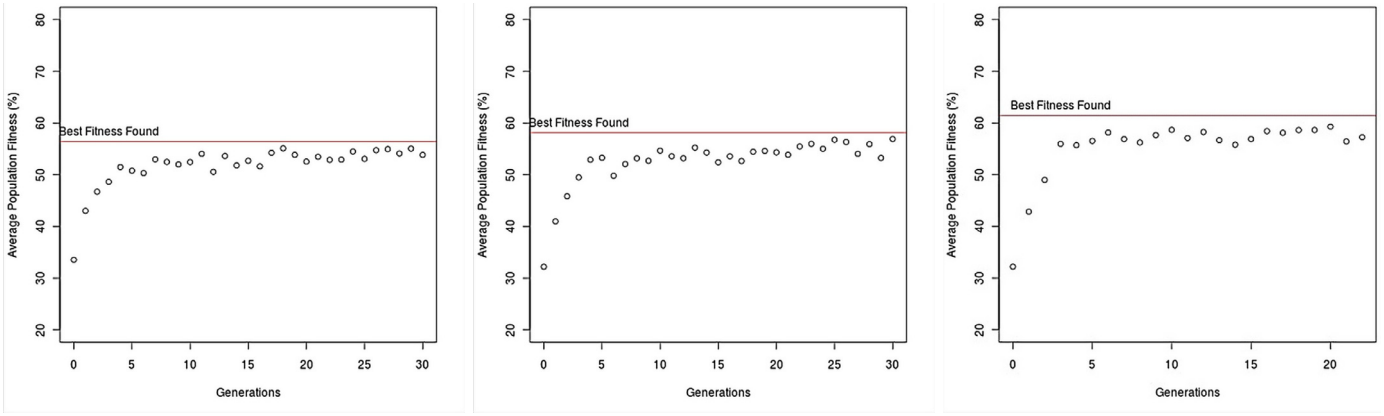
```
1: function STEP3(ValidationSet)
2:     metrics ← [ig, cc, χ², or]
3:     for all fs ∈ metrics do
4:         bestL[fs] ← STEP2(ValidationSet, fs)
5:     for r ← 1, 5 do
6:         bestIndividuals[r] ← GP_CONFIGURATION(ValidationSet, bestL)
7:         r ← r + 1
8:     return bestIndividuals
9: function STEP2(ValidationSet, fs)
10:    for i ← 1, 10 do
11:        Train, Evaluation ← RANDOM_SUBSAMPLE(ValidationSet)
12:        rankedFeatures ← FS_METRIC(fs, Train)
13:        for L ← 0.05, 1.0 do
14:            filteredTrain ← FILTER_TOP_FEATURES(L, Train, rankFeature)
15:            filteredEvaluation ← FILTER_TOP_FEATURES(L, Evaluation, rankedFeatures)
16:            results[i] ← NAIVE_BAYES_CLASSIFIER(filteredTrain, filteredEvaluation)
17:            L ← L + 0.05
18:        i ← i + 1
19:    bestL ← COMPUTE_BEST_L(results)
20:    return bestL
21: function GP_CONFIGURATION(ValidationSet, metrics, bestL)
22:    for i ← 1, 10 do
23:        Train, Evaluation ← RANDOM_SUBSAMPLE(ValidationSet)
24:        for all fs ∈ metrics do
25:            rankedFeatures ← FS_METRIC(fs, Train)
26:            subSets[fs] ← SELECT_TOP_FEATURES(Train, fs, bestL[fs])
27:        terminalNodes ← [subSets[ig], subSets[cc], subSets[χ²], subSets[or]]
28:        functionNodes ← [∪, ∩, /]
29:        individuals[i] ← EXECUTE_GP(Train, Evaluation, terminalNodes, functionNodes)
30:        i ← i + 1
31:    bestIndividual ← SELECT_BEST_INDIVIDUAL(individuals)
32:    return bestIndividual
```

settings) of the feature selection metrics, as well as by our GP-based approach.

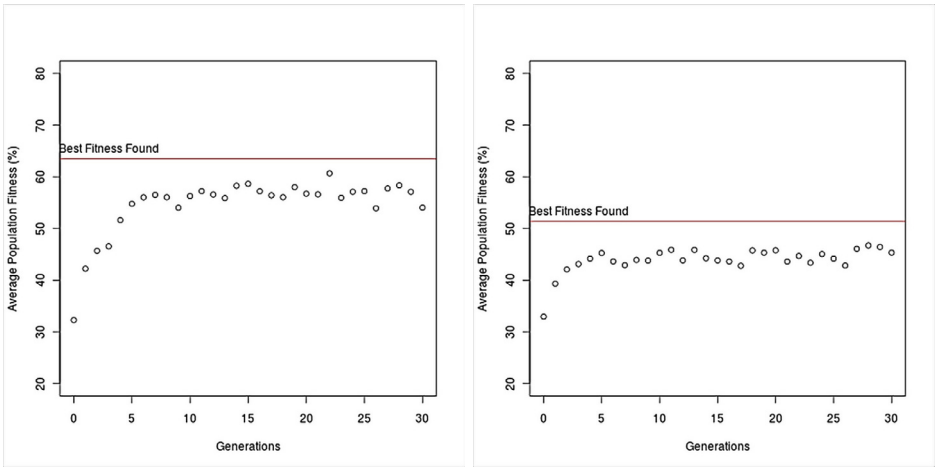
In the second step, for each individual metric, we find the feature subset with the highest discriminative power, by means of random subsampling over the *validation* set, as indicated in the Algorithm 1. Each evaluated metric scores all features according to its estimated discriminative power. The goal is to find the top $L\%$ most discriminative features (i.e., with the highest scores), which would lead to better classification effectiveness. More specifically, considering the *validation* set, we apply a hold-out process 70/30 [41]. In other words, 70% of the *validation* set is used for training the models, while the remaining 30% corresponds to the *evaluation* set. For each feature selection metric, the algorithm selects as the best top $L\%$, the subset of features with highest mean Macro$F_1$, considering 10 repetitions, where in each repetition the subset of features is varied by 5% from 5% up to 100%.

Having determined the $L\%$ value for each feature selection metric, the third step generates the best individuals of the GP approach.[3] Again, we split the *validation* set into two subsets: *train* and *evaluation*, according to a hold-out 70/30. As before, we perform 10 repetitions. For each repetition, as we can see in Algorithm 1, the function *gp_configuration* uses the selected $L\%$ to generate the subset of features corresponding to each feature selection metric, which will then be used in the GP as the terminal

[3] For more information about the GP library, as well as, the configuration setup, see Appendix B.




(a) 1ª execution. (b) 2ª execution. (c) 3ª execution.




(d) 4ª execution. (e) 5ª execution.

**Fig. 7.** Evaluating of the average population fitness of *K8 Collection*. We evaluate the evolution of the average fitness metric of each population during the GP evolutionary process. The average fitness metric in the population increases, showing that the GP is able to search for good solutions.

nodes, as well as the set operations will be used as function nodes. Finally, the algorithm executes the GP evolutionary process (line 29) and save the best individual of the iteration. Then, we evaluate the best individuals found in all repetitions considering the *evaluation* set, in order to select the best one (i.e., the one with better generalization capability). This step is repeated five times as shown in the lines 5–7 in Algorithm 1, deriving 5 best individuals. To be clear, at this step we are only interested in finding the subset of features selected by the best individuals.

Therefore, the steps 1–3 were performed in order to fairly define the best values of *L*% for each feature selection metric, as well to derive the best 5 individuals provided by our GP approach. The last step consists of evaluating our approach. Accordingly, first, random samples were selected from the validation set (70%) in order to create classifications models, filtering the *L*% most discriminative features selected by each *FS* metric, as well as the 5 best individuals found in the third step. After that, the classification effectiveness of the generated models was evaluated on the Test set, using a random sample (70%). We repeated this process 10 times, evaluating the effectiveness in terms of Macro$F_1$ and Micro$F_1$. To compare the average results on our random subsamples experiments, we assess the statistical significance of our results by means of a paired *t*-test with 95% confidence.

*5.2.2. Results*

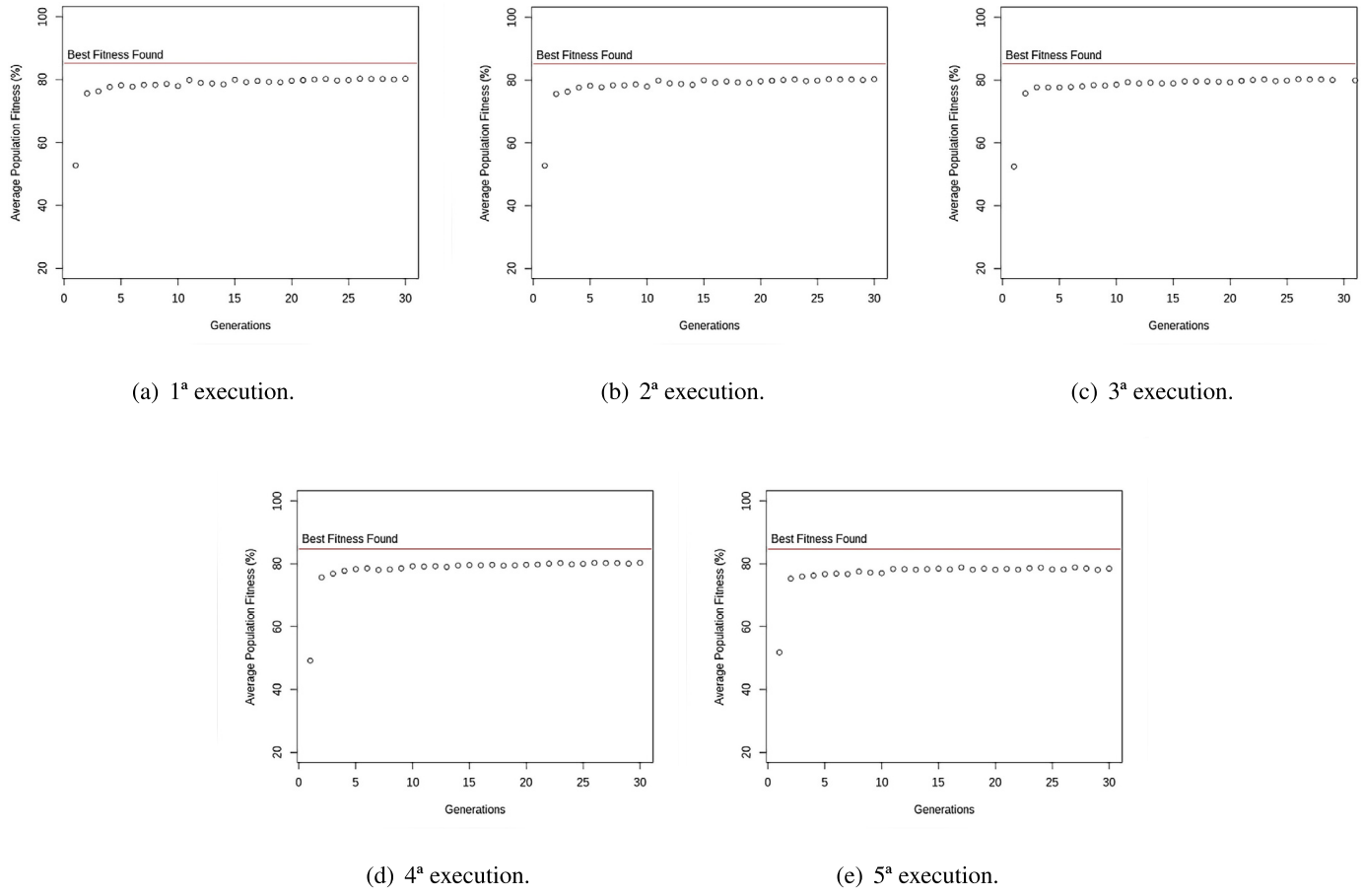
In this section we discuss the results found by applying the experimental design described in Fig. 6 in each dataset. All experiments were run on a Quad-Core Intel Xeon E5620, running at 2.4 GHz, with 16GB RAM.

As mentioned, after partitioning the dataset into *test* and *validation*, we found the most discriminative feature subspace for each traditional FS metric. The results were, considering, respectively, the feature selection metrics, *OR, GI,* $\chi^2$ e *CC*, :

- for the biological dataset, the most discriminative feature subspaces were achieved for *L* adjusted for 10%, 15%, 10% and 10%,
- for 4UNI 50%, 65%, 25% and 65%;
- for Reuters 95%, 50%, 95% and 50%;
- for 20news 60%, 45%, 85% and 50%;
- and for ACL-BIN 10%, 15%, 15% and 10%.

As we can see, results vary greatly across datasets and feature selection metrics.

In Figs. 7 and 8, we present the evolution of the average fitness metric of each population during the GP evolutionary process (third step in Fig. 6) in two datasets. We highlight that the evolutionary process is similar for all individuals. Moreover, we run an analysis in which we calculated the average pairwise similarity




(a) 1ª execution. (b) 2ª execution. (c) 3ª execution.

(d) 4ª execution. (e) 5ª execution.

**Fig. 8.** Evaluating of the average population fitness of *REUT Collection*. We evaluate the evolution of the average fitness metric of each population during the GP evolutionary process. The average fitness metric in the population increases, showing that the GP is able to search for good solutions.

of the subsets selected by the top 50 individuals in the datasets and the average Jaccard was above 0.95. This means that the feature subsets selected by them are very similar and consequently the classification results would also be. Thus, for sake of efficiency, we restricted our analysis to the 5 best individuals to measure the effectiveness of our strategy. As we can observe, as the process goes forward, the population quality (considering the average fitness metric in the population) increases, showing that the GP is, in fact, able to search for good solutions. Similar results are achieved for the individuals in the remaining textual collections, but due to space reasons, we do not include them in the article. As we shall see, even in the cases in which the achieved solution is not optimal (the GP process does not guarantee an optimal solution), the achieved results in terms of classification effectiveness are better or tied to the ones in which we adopt the traditional metrics alone and, usually, with a much smaller feature subspace. The expressions that represent the best five individuals for all datasets can be found in Appendix C.

The classification effectiveness of the feature subspaces selected by each traditional FS metric (second step) and the best individuals selected by the GP approach can be seen in Tables 3 and 8, respectively. We summarize the best results achieved by each metric and the best individuals of our approach in Table 5. In that table, we included as baseline the Feature Engineering Wrapper (FEW) [31], which uses Genetic Programming for feature learning that interfaces with other machine learning methods to compose effective data representations. FEW is able to effectively search the feature space to generate models with a smaller set of optimized features.

**Table 3**
NB results combined with each feature selection metric.

| Collection | Strategy | $L$(%) | # Features | Mac.$F_1$(%) | Mic.$F_1$(%) |
|---|---|---|---|---|---|
| K8 | NB+CC | 10 | **540** | **49.00 ± 0.63** | 82.52 ± 0.82 |
| | NB+$\chi^2$ | 10 | **540** | 39.79 ± 1.12 | 61.32 ± 2.25 |
| | NB+GI | 15 | 811 | 44.99 ± 0.87 | 73.58 ± 1.68 |
| | NB+OR | 10 | **540** | **49.66 ± 0.78** | **83.92 ± 1.16** |
| | NB | 100 | 5408 | 35.66 ± 0.58 | 52.20 ± 1.56 |
| 4UNI | NB+CC | 65 | 24,120 | 54.90 ± 0.64 | 61.32 ± 1.08 |
| | NB+$\chi^2$ | 25 | **9277** | **55.66 ± 0.76** | **62.87 ± 0.8** |
| | NB+GI | 65 | 24,120 | **55.27 ± 0.73** | 61.92 ± 0.98 |
| | NB+OR | 50 | 18,554 | **55.43 ± 0.72** | 61.84 ± 0.86 |
| | NB | 100 | 37,108 | 52.67 ± 1.58 | 61.90 ± 0.87 |
| REUT | NB+CC | 50 | **10,590** | 82.35 ± 0.54 | 91.95 ± 0.29 |
| | NB+$\chi^2$ | 95 | 20,120 | **80.28 ± 1.14** | 91.90 ± 0.50 |
| | NB+GI | 50 | **10,590** | **82.33 ± 0.78** | 92.01 ± 0.37 |
| | NB+OR | 95 | 20,120 | **80.66 ± 1.08** | 92.18 ± 0.42 |
| | NB | 100 | 21,180 | 80.45 ± 1.36 | **92.62 ± 0.45** |
| 20NG | NB+CC | 50 | 28,896 | **83.02 ± 0.38** | **83.74 ± 0.34** |
| | NB+$\chi^2$ | 85 | 49,125 | 82.76 ± 0.43 | **83.58 ± 0.45** |
| | NB+GI | 45 | **26,007** | **83.03 ± 0.34** | **83.70 ± 0.34** |
| | NB+OR | 60 | 34,676 | **83.12 ± 0.40** | **83.88 ± 0.40** |
| | NB | 100 | 57,794 | **83.12 ± 0.51** | 83.02 ± 0.45 |
| ACL-BIN | NB+CC | 10 | **85,942** | 86.10 ± 0.77 | 86.13 ± 0.72 |
| | NB+$\chi^2$ | 15 | 128,914 | 81.55 ± 2.89 | 81.97 ± 2.61 |
| | NB+GI | 15 | 128,914 | 85.67 ± 0.48 | 85.73 ± 0.46 |
| | NB+OR | 10 | **85,942** | 86.00 ± 0.42 | 86.03 ± 0.40 |
| | NB | 100 | 859,431 | **86.72 ± 0.30** | **86.73 ± 0.30** |

**Table 4**
The five best individual found and correspond results.

| Collection | Ind. | # Features | Mac.$F_1$(%) | Mic.$F_1$(%) |
|---|---|---|---|---|
| K8 | 1 | 9 | **66.65 ± 1.65** | **98.59 ± 0.12** |
| | 2 | 6 | 60.43 ± 0.90 | 97.59 ± 0.09 |
| | 3 | 6 | 56.24 ± 0.70 | 97.83 ± 0.31 |
| | 4 | 5 | 58.37 ± 0.93 | 98.01 ± 0.14 |
| | 5 | 9 | 55.49 ± 0.75 | 97.82 ± 0.12 |
| 4UNI | 1 | 2824 | **55.46 ± 0.68** | **62.85 ± 0.94** |
| | 2 | 4234 | 54.62 ± 1.26 | 61.16 ± 0.71 |
| | 3 | 4187 | 54.39 ± 1.12 | 60.73 ± 0.89 |
| | 4 | 4108 | 53.72 ± 1.39 | 60.52 ± 0.62 |
| | 5 | 4173 | 52.37 ± 1.13 | 60.28 ± 0.62 |
| REUT | 1 | 4466 | **85.25 ± 0.89** | **93.10 ± 0.35** |
| | 2 | 4804 | **85.10 ± 0.98** | **93.31 ± 0.34** |
| | 3 | 4595 | **85.13 ± 0.98** | **93.31 ± 0.34** |
| | 4 | 5085 | 84.76 ± 1.07 | **93.25 ± 0.31** |
| | 5 | 4550 | 84.63 ± 0.84 | 92.97 ± 0.32 |
| 20NG | 1 | 16,280 | **82.39 ± 0.41** | 83.06 ± 0.40 |
| | 2 | 16,364 | 82.37 ± 0.45 | 83.04 ± 0.42 |
| | 3 | 16,427 | **82.39 ± 0.42** | 83.07 ± 0.39 |
| | 4 | 17,546 | **82.46 ± 0.42** | 83.14 ± 0.40 |
| | 5 | 17,953 | **82.40 ± 0.45** | 83.10 ± 0.42 |
| ACL-BIN | 1 | 58,882 | **86.86 ± 0.20** | **86.86 ± 0.20** |
| | 2 | 46,123 | 86.16 ± 0.23 | 86.17 ± 0.23 |
| | 3 | 49,534 | **86.80 ± 0.21** | **86.80 ± 0.21** |
| | 4 | 42,096 | 86.22 ± 0.21 | 86.22 ± 0.21 |
| | 5 | 49,005 | 86.14 ± 0.22 | 86.14 ± 0.22 |

**Table 5**
Comparison of the best NB results of each step.

| Collection | Strategy | Mac.$F_1$(%) | Mic.$F_1$(%) | # Features |
|---|---|---|---|---|
| K8 | NB+OR | 49.66 ± 0.78 | 83.92 ± 1.16 | 540 |
| | $FEW_{NB}$ | 50.43 ± 0.02 | **99.05 ± 0.01** | 160 |
| | GP | **66.65 ± 1.65** | 98.59 ± 0.12 | **9** |
| | Diff. | +32.16 ▲ | −0.464 ● | |
| 4UNI | NB+$\chi^2$ | **55.66 ± 0.76** | **62.87 ± 0.8** | |
| | $FEW_{NB}$ | 29.56 ± 0.15 | 44.83 ± 0.02 | *Orig. feat.* |
| | GP | **55.46 ± 0.68** | **62.85 ± 0.94** | **2824** |
| | Diff. | −0.36 ● | −0.032 ● | |
| REUT | NB+CC | 82.35 ± 0.54 | 91.95 ± 0.29 | 10,590 |
| | $FEW_{NB}$ | 58.95 ± 0.05 | 80.22 ± 0.02 | *Orig. feat.* |
| | GP | **85.25 ± 0.89** | **93.10 ± 0.35** | **4466** |
| | Diff. | +3.461 ▲ | +1.243 ▲ | |
| 20NG | NB+OR | **83.12 ± 0.40** | **83.88 ± 0.40** | 34,676 |
| | $FEW_{NB}$ | 65.74 ± 0.01 | 66.53 ± 0.01 | *Orig. feat.* |
| | GP | 82.46 ± 0.42 | 83.14 ± 0.40 | **17,546** |
| | Diff. | −0.797 ● | −0.91 ● | |
| ACL-BIN | NB | **86.72 ± 0.30** | **86.73 ± 0.30** | 859,431 |
| | $FEW_{NB}$ | – | – | – |
| | GP | **86.86 ± 0.20** | **86.86 ± 0.20** | **58,882** |
| | Diff. | +0.161 ● | +0.15 ● | |

In order to allow a fair comparison, we paired FEW with the Gaussian Naive Bayes ($FEW_{NB}$) and applied the appropriate adjustments for the GP parameters. We marked in bold the best results that are statistically superior or tie with others. When the difference is not significant, the symbol is a ●. If it is significant, a ▲ represents a positive variation. The statistical difference among results is defined as $\frac{(PS-BL)}{BL}$ [42], where *PS* is the result of the proposed solution based on GP and *BL* is the result of the best baseline.

Observing the results in Table 5 (the summarization table), we can see, for instance, that our GP-based feature selection strategy provided substantial improvements over the traditional feature selection metrics in the biological dataset K8. Comparing the best individual feature selection metric with our GP-based approach, we reduced, in this dataset, the feature space from 504 ($L = 10\%$) to only 9 features. Moreover, even selecting less than 2% of the original feature space, we improved Macro$F_1$ from 50.43% to 66.65%, a very significant improvement (gains of 32.16% in Macro$F_1$). Regarding the comparison with $FEW_{NB}$, although we tied with it in terms of Micro$F_1$, we surpassed it in terms of Macro$F_1$, with a much larger reduction in the number of features in this dataset.

Somewhat similar results can be observed for the REUT dataset, in which our strategy was able to (statistically) improve the classification effectiveness, besides significantly reducing the size of the feature subspace. Compared to the best individual feature selection (Correlation Coefficient) with our GP-based approach, we reduced the number of features from 10,590 ($L = 50\%$) to 4466 features, a 42% reduction. Finally, observing the results related to the other textual datasets, although our proposal was not able to produce effectiveness gains over the "best" traditional feature selection metrics, our GP strategy presents a much more aggressive reduction on the number of features in each dataset, without compromising the classification effectiveness (i.e. 68% for ACL-BIN, 30, 4% for 4UNI and 50, 5% for 20NG). The FEW strategy was not able to reduce the original feature space in the textual datsets and did not outperform our strategy in any of them. Moreover, in case of ACL-BIN, with 1,110,351 features, FEW was not able to execute due memory overflow. We believe that is due to the fact that FEW was not designed to work on highly dimensional and sparse feature spaces. Our method, on the contrary, can properly deal with these cases by exploiting the results of the combined FS methods.

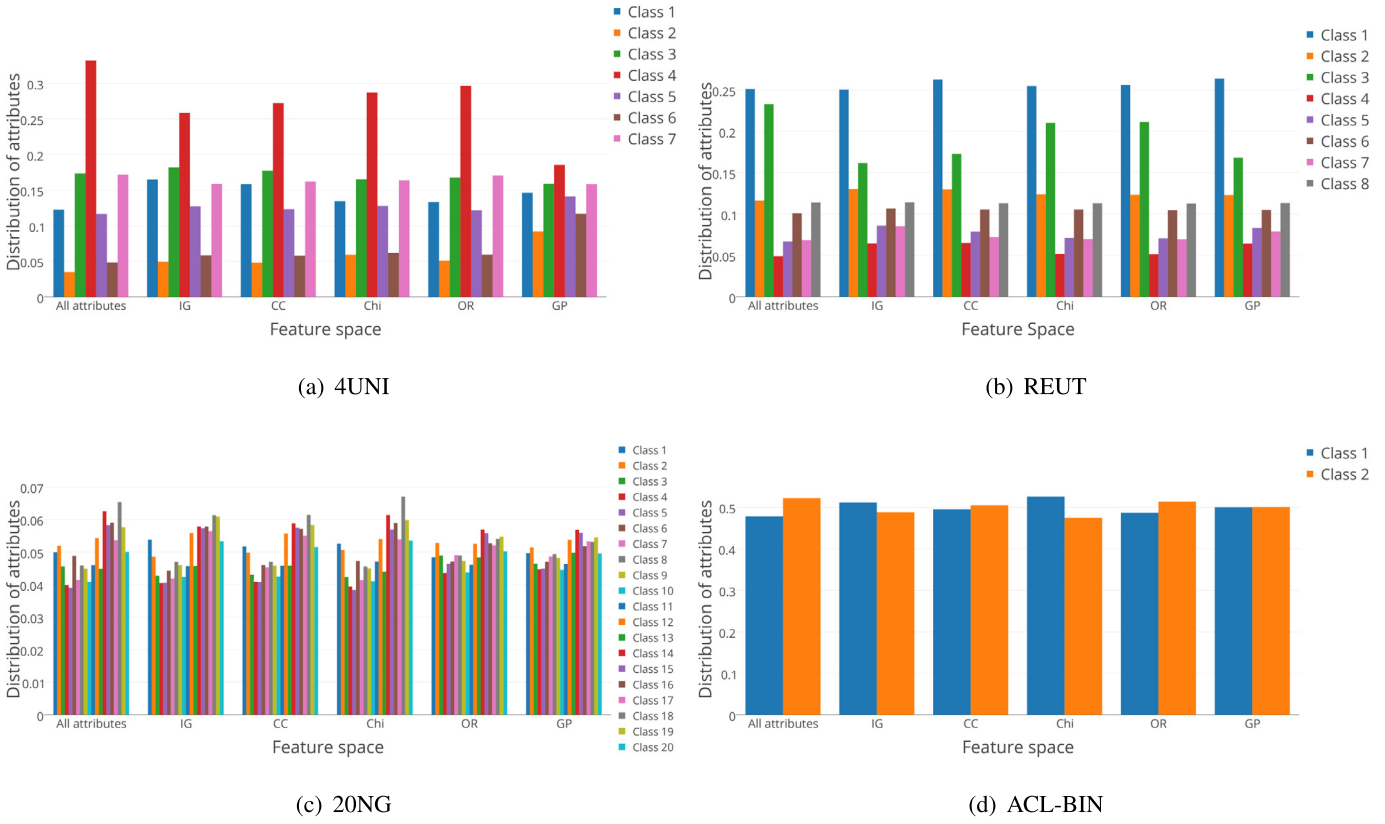
In order to better understand the gains achieved by our GP approach, we assess the feature distribution across the classes in the original datasets, considering all features, and after applying feature selection, considering the most discriminative features selected by each traditional FS metric and our GP approach and compare them. For example, in 4UNI there are 37,108 features.[4] We calculate how these features are distributed across the classes. Considering the OR metric and 4UNI, according to Table 3, we have that the best results are achieved using 18,554 features. Therefore, we also calculate the distribution of these features across the classes. We repeat this process for all FS metrics, as well for the GP approach. The results achieved is presented in Fig. 9.

Notice that the percentage of occurrence of features is larger for the majority classes and smaller for minority classes, considering the original distribution, for all datasets. After applying a FS metric and the GP approach, we can see that the percentage of occurrence of features decreases for the majority classes and increases for smallest ones, making the feature distribution more balanced across classes. Particularly in the case of GP, we find that the feature distribution is more balanced than the ones achieved by other FS metrics. This is particularly the case in 4UNI, in which there was a dramatic result in the original imbalance, mainly regarding class 4, the majority one. But this is also true in the case of the ACL-BIN, in which we did not have a document imbalance, but a feature imbalance across classes. In that case, GP found a perfect feature balance.

### 5.3. Discussion

Based on the results in Section 5.2.2, we can conclude that each feature selection metric has advantages, but are also biased towards features belonging to the majority classes, leading to potentially sub-optimal feature subsets. Our GP approach, on the other hand, seems to be capable of selecting a better set of discriminative features (with a better ratio between positive and negative features, according to the observed data), avoiding the potential bias when facing skewed data.

[4] Note that, as the data are skewed, some features may not occur in the *train* set.




(a) 4UNI (b) REUT

(c) 20NG (d) ACL-BIN

**Fig. 9.** Feature distribution in classes. We compare the occurrence of features in the classes before and after applying feature selection. These results show that the GP-based approach is able to better distribute the features in a more balanced way.

**Table 6**
Jaccard distance between the best individuals found by the GP approach with the same most discriminative feature selected by each traditional FS metric. Highest distances indicate dissimilarity between the features selected by best individuals of the GP approach compared with same number of features selected by each FS metric.

| | | GI | CC | $\chi^2$ | OR |
|---|---|---|---|---|---|
| 4UNI | GP1 | 0.80 | 0.81 | 0.88 | 0.83 |
| | GP2 | 0.82 | 0.83 | 0.87 | 0.83 |
| | GP3 | 0.82 | 0.83 | 0.87 | 0.84 |
| | GP4 | 0.82 | 0.83 | 0.87 | 0.84 |
| | GP5 | 0.82 | 0.83 | 0.87 | 0.84 |
| REUT | GP1 | 0.40 | 0.37 | 0.75 | 0.72 |
| | GP2 | 0.34 | 0.32 | 0.74 | 0.71 |
| | GP3 | 0.39 | 0.38 | 0.75 | 0.73 |
| | GP4 | 0.37 | 0.34 | 0.75 | 0.73 |
| | GP5 | 0.40 | 0.38 | 0.76 | 0.73 |
| 20NG | GP1 | 0.38 | 0.34 | 0.62 | 0.33 |
| | GP2 | 0.38 | 0.34 | 0.62 | 0.33 |
| | GP3 | 0.38 | 0.35 | 0.62 | 0.33 |
| | GP4 | 0.42 | 0.38 | 0.62 | 0.33 |
| | GP5 | 0.43 | 0.38 | 0.62 | 0.33 |
| ACL-BIN | GP1 | 0.43 | 0.42 | 0.75 | 0.44 |
| | GP2 | 0.44 | 0.43 | 0.74 | 0.47 |
| | GP3 | 0.37 | 0.36 | 0.71 | 0.40 |
| | GP4 | 0.40 | 0.39 | 0.75 | 0.43 |
| | GP5 | 0.47 | 0.46 | 0.76 | 0.50 |

To reinforce such conclusions, we run a final set of experiments comparing the feature subsets selected by the traditional feature selection metrics and by our approach, considering the *same* number of most discriminative features. First, we measure the Jaccard distance among the feature subspaces selected by each strategy. Table 6 illustrates the dissimilarity between these feature subspaces, showing that no traditional feature selection metric was capable of selecting the same feature subspace exploited by the GP approach with the *same* number of most discriminative features. We removed the K8 dataset from Table 6 because the subset of features selected by the GP approach was so small (up to 9 features) that all the Jaccard distances were very close to zero.

**Table 7**
A comparison between the feature selection metrics considering the Top-9 features of collection K8.

| | Mac.$F_1$(%) | Std. dev. | Mic.$F_1$(%) | Std. dev. |
|---|---|---|---|---|
| GP | **66.65** | **1.65** | 98.59 | 0.12 |
| GI | 49.83 | 0.40 | 98.37 | 0.55 |
| OR | 50.87 | 1.64 | **99.13** | **0.06** |
| $\chi^2$ | 50.06 | 0.52 | 98.82 | 0.22 |
| CC | 50.15 | 0.72 | **99.14** | 0.05 |

More specifically, observing the results for the ACL_BIN collection, the Jaccard distance between the feature subspace selected by *CC* and *GP2* is 0.43, meaning only a moderate similarity. On the other hand, the features selected by the $\chi^2$ metric are very dissimilar when compared to the features selected by *GP2* resulting in a distance equivalent to 0.74. We also evaluate the classification effectiveness of the subsets selected by the traditional feature selection metrics in this scenario, as reported in Table 7. For instance, considering the 9 top ranked features selected by the OR metric in the K8 dataset, the learned model was able to achieve Macro$F_1$ and Micro$F_1$ of $50.87 \pm 1.64$ and $99.13 \pm 0.06$, respectively. Contrasting these values with the ones reported in Table 5 we can observe an improvement in Micro$F_1$, with a statistically tie in Macro$F_1$. This means that, after applying the OR filters, the learned model became more biased towards the largest class. On the other hand, with the same number of features, the GP-based approach

**Table 8**
A comparison between the feature selection metrics considering the Top-4466 features of collection REUT.

| | Mac.$F_1$(%) | Std. dev. | Mic.$F_1$(%) | Std. dev. |
|---|---|---|---|---|
| GP | **85.25** | **0.89** | **93.10** | **0.35** |
| GI | 76.18 | 0.68 | 88.20 | 0.34 |
| OR | 71.12 | 2.50 | 83.37 | 2.57 |
| $\chi^2$ | 67.58 | 1.17 | 74.58 | 1.29 |
| CC | 76.26 | 0.98 | 88.27 | 0.47 |

**Table 9**
A comparison between the feature selection metrics considering the Top-42,096 features of collection ACL-BIN.

| | Mac.$F_1$(%) | Std. dev. | Mic.$F_1$(%) | Std. dev. |
|---|---|---|---|---|
| GP | **86.22** | **0.21** | **86.22** | **0.21** |
| GI | 85.23 | 0.49 | 85.32 | 0.47 |
| OR | 85.29 | 0.89 | 85.37 | 0.84 |
| $\chi^2$ | 78.56 | 6.87 | 79.51 | 5.95 |
| CC | 85.65 | 0.64 | 85.71 | 0.61 |

**Table 10**
A comparison between the feature selection metrics considering the Top-16,280 features of collection 20NG.

| | Mac.$F_1$(%) | Std. dev. | Mic.$F_1$(%) | Std. dev. |
|---|---|---|---|---|
| GP | **82.39** | **0.41** | **83.06** | **0.40** |
| GI | 80.19 | 0.40 | 80.84 | 0.40 |
| OR | 80.24 | 0.47 | 80.98 | 0.47 |
| $\chi^2$ | 79.92 | 0.38 | 80.70 | 0.34 |
| CC | 80.28 | 0.44 | 80.94 | 0.43 |

**Table 11**
A comparison between the feature selection metrics considering the Top-2824 features of collection 4UNI.

| | Mac.$F_1$(%) | Std. dev. | Mic.$F_1$(%) | Std. dev. |
|---|---|---|---|---|
| GP | **55.46** | **0.68** | **62.85** | **0.94** |
| GI | 48.27 | 1.75 | 55.00 | 1.36 |
| OR | 48.35 | 1.28 | 55.64 | 1.69 |
| $\chi^2$ | 50.95 | 0.86 | 59.20 | 1.13 |
| CC | 47.68 | 1.44 | 54.40 | 1.53 |

was able to achieve a significantly higher Macro$F_1$ (with a similar Micro$F_1$), indicating that the learned classification model, after filtering the features, was more effective to classify test examples belonging to the minority class, the most important one.

The results regarding the comparison between traditional feature selection metrics and our approach, considering the *same* number of most discriminative features, regarding the textual datasets are presented in Tables 8–11. We can observe that our GP strategy achieves gains in all datasets, mainly in terms of Macro$F_1$, indicating a more effective discrimination of the minority class examples. If we observe the Jaccard distance of these feature subspaces in Table 6, we can see that in cases in which the Jaccard distance is low (Jaccard distance between 0.3 and 0.5), the small differences in the selection still produced considerable differences in classification effectiveness. This corroborates our argument that our GP based solution does a better job when selecting the best features, by better balancing positive and negative features across classes.

### 5.4. Complexity of the GP solution

The time complexity of the training phase, based on our modeling, is $O(N_g \times N_i) \times T_e$, where $N_g$ is the number of evolution generations, $N_i$ is the number of individuals in the population pool, and $T_e$ is the fitness evaluation complexity of an individual.

**Table 12**
Execution time (seconds) of the step 3 of our GP solution.

| Collection | GP execution |
|---|---|
| K8 | 185188.44 |
| 4UNI | 86053.20 |
| REUT | 30236.00 |
| 20NG | 370376.87 |
| ACL-BIN | 120806.71 |

In our problem, the fitness evaluation complexity of an individual is the complexity of a single feature selection process given by $O(N_t)$, where $N_t$ is the number of training and testing samples. A single feature selection process requires preprocessing of the features of the validation, training and testing sets, the classification process and evaluation of the model. Thus, the complexity of the training is given by $O(N_g \times N_i \times N_t)$. This is the worst case scenario for the training phase. Table 12 shows the average time (in seconds) of the 3rd step of Fig. 6.

It is important to remind that the training time is not as important as the time to perform the actual feature selection with the suggested feature space, since the training phase is likely to be performed only once, and if eventually required, it can be done offline. The application of the suggested feature space is usually fast as it only requires the computation of a smaller set of features.

## 6. Conclusions and future work

In this paper we propose a Genetic Programming solution for a more effective feature selection strategy, which in addition to providing a highly effective selection of the most important features, is also robust to skewed data. Our solution learns a “compound” feature selection metric, able to take advantage of feature subsets selected by several feature selection metrics.

Our experimental results in a biological dataset, a highly skewed one, show that our GP-based approach produced large gains in effectiveness with a massive reduction in the number of features, up to 98%. Aggressive reductions were also obtained in the textual datasets up to 65%, without losses in effectiveness. These results are not matched by the the traditional feature selection metrics, which based on our experiments, are biased towards features belonging to the majority class. On the other hand, our GP approach is capable of selecting a better set of discriminative features, avoiding the potential bias when facing skewed data.

As future work, we intend to construct a computational model in which, instead of combining feature subsets, we will combine the mathematical formulas that define the feature selection metrics with the goal to achieve a more impartial formula. Moreover, based on the insights present in [35], we intend to adapt our proposal to combine feature selection with feature construction, since it can potentially improve the classification performance, and combine feature selection with instance selection, since it can potentially improve the effectiveness. We also intend to experiment with new datasets.

## Acknowledgments

This work was partially supported by Fapemig, CNPq, CAPES, and by projects InWeb (MCT/CNPq573871/2008-6), MASWeb (FAPEMIG-PRONEX APQ-01400-14).

## Appendix A

We use standard measures adopted by the information retrieval and data mining community, namely, Micro$F_1$ and Macro$F_1$.

**Table 13**
Contingency table for classification effectiveness evaluation.

| Positive class = $c_i$ | | Ground truth $c_i$ | Ground truth $\bar{c}_i$ |
|---|---|---|---|
| **Prediction** | $c_i$ | TP | FP |
| | $\bar{c}_i$ | FN | TN |

Micro$F_1$ measures the global effectiveness in terms of all decisions made by the classifier (that is, the inverse of error rate). The Macro$F_1$, on the other hand, measures the classification effectiveness regarding each class independently, by computing the $F_1$ measure (i.e., harmonic mean between precision and recall) obtained for each class and averaging them [43]. In order to describe each of these measures in a binary classification, let us consider the contingency table represented in Table 13 (also known as confusion matrix), where TP, TN, FP and FN denote, respectively, the number of true positives, true negatives, false positives and false negatives, defined as:

True Positive (TP): positive test document correctly classified into the positive class.
True Negative (TN): negative test document correctly classified into the negative class.
False Positive (FP): negative test document incorrectly classified into the positive class.
False Negative (FN): positive test document incorrectly classified into the negative class.

The $F_1$ measure is defined as the harmonic mean of the precision and the recall. The precision $p$ of a performed classification denotes the fraction of all documents assigned to the positive class $c_i$ by the classifier that really belong to $c_i$, while the recall $r$ of a performed classification denotes the fraction of all documents that belong to the positive class $c_i$ that were correctly assigned to $c_i$ by the classifier. Finally, the $F_1$ measure can be expressed as

$$F_1 = \frac{2pr}{p+r} \tag{A.1}$$

There are two conventional methods to evaluate classification algorithms when applied to problems with more than two classes, namely by micro-averaging and macro-averaging the $F_1$ measure. The micro-averaged $F_1$ (Micro$F_1$) is calculated from a global contingency table (similarly to Table 13), with the precision and recall being calculated as a sum of each entry of the table:

$$p_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i} \tag{A.2}$$

$$r_{micro} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i} \tag{A.3}$$

In contrast, the macro-averaged $F_1$ (Macro$F_1$) is calculated by first calculating the precision and recall values for each class and computing their average value:

$$p_{macro} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FP_i} \tag{A.4}$$

$$r_{macro} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FN_i} \tag{A.5}$$

Notice that the main difference between both strategies is that the Micro$F_1$ is a document pivoted measure that gives equal weights to the documents while the Macro$F_1$ measure is a class pivoted measure that gives equal weights to the classes.

**Table 14**
GP parameters.

| Parameter | Value |
|---|---|
| Population size | 100 |
| Number of generations | 30 |
| Creation type | Ramped half-and-half |
| Crossover probability | 98 |
| Maximum depth for creation | 5 |
| Maximum depth for crossover | 15 |
| Selection type | Tournament |
| Tournament size | 7 |
| Swap mutation probability | 50 |
| Shrink mutation probability | 50 |
| Elitism | True |

## Appendix B

We used the *gpc++ v0.5.2* library [44], an efficient GP library, to implement our approach. As a generic library, only the implementation of structures closely related to the problem, such as terminal nodes (in our case, the feature selection metrics), function nodes (in our case, the set operators to be applied to the set of features selected by the two children nodes) and the fitness function (in our case, a single metric to assess classification effectiveness, namely, the Macro$F_1$ metric), is required.

In order to find the parameters used in our experiments, we conducted a pilot study, inspired by results reported by the author of the library [44]. We note that such study does not fit in all cases (since it is problem and data dependent) but, as argued by the author, it is a good start towards an ideal setting.

First, we sampled 10% of the dataset in a random fashion, keeping the original distribution of the classes. We applied each feature selection metric considering such sampled data in order to find the discriminative power of each feature. Using the top 5% of the most discriminative features, for each metric in isolation, we applied the GP procedure, varying the parameters in order to find the values that maximize the fitness of the individuals (that is, those that provide highest quality classification). We vary each GP parameter according to the following strategy: The population size was set from 50 up to 100, with steps of 10. The number of generations ranged from 20 to 50, with steps of 10. The crossover probability was varied from 90% to 100%, with steps of 1%, while both the swap mutation and shrink mutation probabilities were varied from 20% to 100%, with steps of 10%. Finally, the tournament size used for individual selection was chosen between 5 and 10, with steps of 1. Such parameter tuning was based on a simple experimental design where, while one of them is varied, the remaining are kept fixed. The configuration that yielded the best results is shown Table 14, and these values were used in our experiments. Despite the good results obtained, presented in the following section, we believe that a fine tuning of these parameters for our problem may lead to even better results.

## Appendix C

### K8

Individual 1: $\chi^2 \backslash CC$
Individual 2: $OR \backslash GI$
Individual 3: $OR \backslash GI$
Individual 4: $CC \backslash GI$
Individual 5: $CC \backslash \chi^2$

### 4UNI

Individual 1: $(((((((\chi^2 \backslash OR) \cup IG) \cap (CC \cup IG)) \cap ((CC \cap (\chi^2 \backslash IG)) \cup (\chi^2 \cup IG))) \cap \chi^2) \cap (((((CC \cap (\chi^2 \backslash IG)) \cup ((\chi^2 \cup IG) \cup IG)) \cap (OR \cap$

$CC))\cap(CC\cup IG))\cap\chi^2))\cap((((((\chi^2\backslash CC)\cup(CC\cap\chi^2))\backslash((OR\backslash CC)\cup(\chi^2\cup OR)))\cup(((\chi^2\cup\chi^2)\cap(IG\cap OR))\backslash((OR\backslash\chi^2)\cap(IG\cup OR))))\cap(CC\cup IG))\cap((\chi^2\backslash OR)\cup IG)))\cap(((((((OR\cap IG)\backslash(IG\cap CC))\backslash((\chi^2\cap IG)\cup(IG\cap\chi^2)))\cup((IG\cup(IG\cap CC))\cap(CC\cup IG))\cap((CC\cup\chi^2)\backslash(OR\backslash\chi^2)))\cap\chi^2))$

Individual 2: $(((((\chi^2\backslash OR)\cup IG)\cap(CC\cup IG))\cap((CC\cap(\chi^2\backslash IG))\cup(\chi^2\cup IG)))\cap\chi^2)$

Individual 3: $(((((CC\cap(\chi^2\backslash IG))\cup(\chi^2\cup IG))\cap(OR\cap CC))\cap(CC\cup IG))\cap\chi^2)$

Individual 4: $((((((\chi^2\backslash CC)\cup(CC\cap\chi^2))\backslash((OR\backslash CC)\cup(\chi^2\cup OR)))\cup\chi^2\backslash(IG\cap OR))\cap((OR\backslash\chi^2)\cap(IG\cup OR)))\cap(CC\cup IG))\cap((\chi^2\backslash OR)\cup(IG\cup IG))$

Individual 5: $(((((((OR\cap IG)\backslash(IG\cap CC))\backslash(\chi^2\cap IG)\cup IG\cup(IG\cap CC)))\cap(CC\cup IG))\cap((CC\cup\chi^2)\backslash(OR\backslash\chi^2)))\cap\chi^2)$

**REUT90**

Individual 1: $(CC\cup GI)\cap((\chi^2\backslash OR)\cup GI)$

Individual 2: $(CC\cup GI)\cap(GI\cup OR)\cap((CC\cap(\chi^2\backslash GI))\cup(\chi^2\cup GI))$

Individual 3: $(CC\cup GI)\cap(OR\cap CC)\cap((\chi^2\backslash OR)\cup GI)$

Individual 4: $(CC\cup GI)$

Individual 5: $(CC\cup GI)\cap((\chi^2\backslash OR)\cup GI)$

**20NG**

Individual 1: $((CC\cup IG)\cap(((\chi^2\cap IG)\backslash(OR\cup IG))\cup\chi^2))$

Individual 2: $(((\chi^2\backslash OR)\cup IG)\cap(((\chi^2\cap IG)\backslash(OR\cup IG))\cup\chi^2))$

Individual 3: $((IG\cup(\chi^2\cup CC))\cap(((\chi^2\cap IG)\backslash(OR\cup IG))\cup\chi^2))$

Individual 4: $(((CC\cap(\chi^2\backslash IG))\cup(\chi^2\cup IG)\cap(((\chi^2\cap IG)\backslash(OR\cup IG))\cup\chi^2))$

Individual 5: $(((OR\cap IG)\backslash(\chi^2\backslash IG))\cap(((\chi^2\cap IG)\backslash(OR\cup IG))\cup\chi^2))$

**ACL-BIN**

Individual 1: $((((OR\backslash GI)\backslash(\chi^2\backslash GI))\cup(CC\backslash\chi^2))\backslash(((OR\backslash CC)\backslash(GI\cup\chi^2))\cap((\chi^2\cap GI)\cap(\chi^2\cap CC))))\cap(GI\cup OR)\cap((OR\cup(\chi^2\cap GI))\backslash(\chi^2\backslash GI))\cap(CC\cup GI)\cap(OR\cap GI)\cap(GI\cup(\chi^2\cup CC))$

Individual 2: $(((\chi^2\cap GI)\backslash(OR\cup GI))\cup\chi^2)\cap(GI\cup OR)\cap(CC\cup GI)\cap((OR\cup GI)\backslash((GI\backslash CC)\cap OR)\backslash((OR\backslash CC)\backslash(CC\cup GI)\backslash(\chi^2\backslash OR)))\cap(GI\backslash(CC\cup\chi^2)\cup OR)\cap(GI\cup(\chi^2\cup CC))$

Individual 3: $((((OR\backslash GI)\backslash(\chi^2\backslash GI))\cup(CC\backslash\chi^2))\backslash(((OR\backslash CC)\backslash(GI\cup\chi^2))\cap((\chi^2\cap GI)\cap(\chi^2\cap CC))))\cap(GI\cup OR)\cap((OR\cup(\chi^2\cap GI))\backslash(\chi^2\backslash GI))\cap(CC\cup GI)\cap(OR\cap GI)\cap(GI\cup(\chi^2\cup CC))\cap((\chi^2\cap GI)\backslash(OR\cup GI)\cup\chi^2)\cap((OR\cap GI)\backslash(\chi^2\backslash GI))$

Individual 4: $(((\chi^2\cap GI)\backslash(OR\cup GI))\cup\chi^2)\cap(GI\cup OR)\cap(CC\cup GI)\cap((OR\cup GI)\backslash((GI\backslash CC)\cap OR)\backslash((OR\backslash CC)\backslash(CC\cup GI)\backslash(\chi^2\backslash OR)))\cap(GI\backslash(CC\cup\chi^2)\cup OR)\cap(GI\cup(\chi^2\cup CC))\cap((((OR\backslash GI)\backslash(\chi^2\backslash GI))\cup(CC\backslash\chi^2))\backslash(((OR\backslash CC)\backslash(GI\cup\chi^2))\cap((\chi^2\cap GI)\cap(\chi^2\cap CC))))$

Individual 5: $(((\chi^2\cap GI)\backslash(OR\cup GI))\cup\chi^2)\cap(GI\cup OR)\cap(CC\cup GI)\cap((OR\cup GI)\backslash((GI\backslash CC)\cap OR)\backslash((OR\backslash CC)\backslash(CC\cup GI)\backslash(\chi^2\backslash OR)))\cap(GI\backslash(CC\cup\chi^2)\cup OR)$

## References


[1] J. Hovold, Naive Bayes spam filtering using word-position-based attributes, in: Proceedings of the 2005 Second Conference on Email and Anti-Spam (CEAS), 2005.

[2] I. Fahmi, in: Examining Learning Algorithms for Text Classification in Digital Libraries, Netherland, 2004. Master's thesis.

[3] R. Zheng, J. Li, H. Chen, Z. Huang, A framework for authorship identification of online messages: writing-style features and classification techniques, J. Assoc. Inf. Sci. Technol. (JASIST) 57 (3) (2006) 378–393.

[4] H. Zhang, T. Ma, G.B. Huang, Z. Wang, Robust global exponential synchronization of uncertain chaotic delayed neural networks via dual-stage impulsive control, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 40 (3) (2010a) 831–844, doi:10.1109/TSMCB.2009.2030506.

[5] H. Zhang, Z. Liu, G.B. Huang, Z. Wang, Novel weighting-delay-based stability criteria for recurrent neural networks with time-varying delay, IEEE Trans. Neural Netw. 21 (1) (2010b) 91–106, doi:10.1109/TNN.2009.2034742.

[6] H. Zhang, Z. Wang, D. Liu, Global asymptotic stability of recurrent neural networks with multiple time-varying delays, IEEE Trans. Neural Netw. 19 (5) (2008) 855–873, doi:10.1109/TNN.2007.912319.

[7] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.

[8] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (The Morgan Kaufmann Series in Data Management Systems), first, Morgan Kaufmann, 1999.

[9] G. Forman, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. Res. 3 (2003) 1289–1305.

[10] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[11] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Wozniak, F. Herrera, A survey on data preprocessing for data stream mining: current status and future directions, Neurocomputing 239 (2017) 39–57, doi:10.1016/j.neucom.2017.01.078.

[12] S.M.H. Bamakan, H. Wang, T. Yingjie, Y. Shi, An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization, Neurocomputing 199 (2016) 90–102, doi:10.1016/j.neucom.2016.03.031.

[13] F. Sebastiani, C.N.D. Ricerche, Machine learning in automated text categorization, ACM Comput. Surv. 34 (2002) 1–47.

[14] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97, 1997, pp. 412–420.

[15] D. Mladenic, Machine Learning on Non-homogeneous, Distributed Text Data, University of Ljubljana, Faculty of Computer and Information Science, 1998 Ph.D. thesis.

[16] Z. Zheng, X. Wu, R. Srihari, Feature selection for text categorization on imbalanced data, ACM SIGKDD Explor. Newslett. 6 (2004) 80–89.

[17] I. Sandin, G. Andrade, F. Viegas, D. Madeira, L.C. da Rocha, T. Salles, M.A. Gonçalves, Aggressive and effective feature selection using genetic programming, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012, June 10–15, Brisbane, Australia, 2012, pp. 1–8.

[18] P. Cunningham, Dimension reduction, 2007, (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.98.1478), Technical report UCD-CSI-2007-7, University College Dublin.

[19] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326, doi:10.1126/science.290.5500.2323.

[20] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28, doi:10.1016/j.compeleceng.2013.11.024.

[21] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[22] B. Xue, M. Zhang, W.N. Browne, A comprehensive comparison on evolutionary feature selection approaches to classification, Int. J. Comput. Intell. Appl. 14 (02) (2015).

[23] X.-w. Chen, M. Wasikowski, Fast: a ROC-based feature selection metric for small samples and imbalanced data classification problems, in: Proceedings of the 2008 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, 2008, pp. 124–132.

[24] H. Park, S. Kwon, H.-C. Kwon, Complete gini-index text (GIT) feature-selection algorithm for text classification, in: Proceedings of the Second International Conference on Software Engineering and Data Mining (SEDM), 2010, pp. 366–371.

[25] P. Yang, W. Liu, B.B. Zhou, S. Chawla, A.Y. Zomaya, Ensemble-based wrapper methods for feature selection and class imbalance learning, in: Proceedings of the 2013 Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2013.

[26] L. Yin, Y. Ge, K. Xiao, X. Wang, X. Quan, Feature selection for high-dimensional imbalanced data, Neurocomputing 105 (2013) 3–11, doi:10.1016/j.neucom.2012.04.039. Learning for Scalable Multimedia Representation.

[27] S. Maldonado, R. Weber, F. Famili, Feature selection for high-dimensional class-imbalanced data sets using support vector machines, Inf. Sci. 286 (2014) 228–246, doi:10.1016/j.ins.2014.07.015.

[28] B. Tran, B. Xue, M. Zhang, Genetic programming for feature construction and selection in classification on high-dimensional data, Memet. Comput. 8 (2015) 1–13, doi:10.1007/s12293-015-0173-y.

[29] S. Ahmed, M. Zhang, L. Peng, Feature selection and classification of high dimensional mass spectrometry data: a genetic programming approach, in: Proceedings of the 2013 European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer, 2013, pp. 43–55.

[30] M.W. Aslam, Z. Zhu, A.K. Nandi, Feature generation using genetic programming with comparative partner selection for diabetes classification, Expert Syst. Appl. 40 (13) (2013) 5402–5412.

[31] W.L. Cava, J. Moore, A general feature engineering wrapper for machine learning using \epsilon -lexicase survival, in: Proceedings of the Twentieth European Conference on Genetic Programming – EuroGP 2017, April 19–21, Amsterdam, The Netherlands, 2017, pp. 80–95, doi:10.1007/978-3-319-55696-3_6.

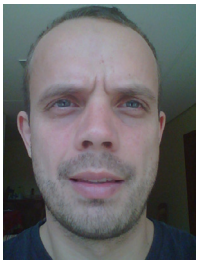
[32] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.

[33] E. Emary, H.M. Zawbaa, C. Grosan, A.E. Hassenian, Feature subset selection approach by gray-wolf optimization, in: Proceedings of the 2015 Afro-European Conference for Industrial Advancement, Springer, 2015, pp. 1–13.

[34] X. Zhang, W. Chen, B. Wang, X. Chen, Intelligent fault diagnosis of rotating machinery using support vector machine with ant colony algorithm for synchronous feature selection and parameter optimization, Neurocomputing 167 (2015) 260–279, doi:10.1016/j.neucom.2015.04.069.

[35] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626, doi:10.1109/TEVC.2015.2504420.
[36] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Text feature selection using ant colony optimization, Expert Syst. Appl. 36 (3) (2009) 6843–6853, doi:10.1016/j.eswa.2008.08.022.
[37] H.T. Ng, W.B. Goh, K.L. Low, Feature selection, perceptron learning, and a usability case study for text categorization, in: Proceedings of the Twentieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '97, 1997, pp. 67–73.
[38] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems), Cambridge, MA, USA, 1992.
[39] S.A. Danziger, R. Baronio, L. Ho, L. Hall, K. Salmon, G.W. Hatfield, P. Kaiser, R.H. Lathrop, Predicting positive P53 cancer rescue regions using most informative positive (MIP) active learning, PLoS Comput. Biol. 5 (9) (2009) e1000498.
[40] X. Glorot, Domain adaptation for large-scale sentiment classification: a deep learning approach, Learning 27 (1) (2011) 513–520.
[41] J.-H. Kim, Estimating classification error rate: repeated cross-validation, repeated hold-out and bootstrap, Comput. Stat. Data Anal. 53 (11) (2009) 3735–3745, doi:10.1016/j.csda.2009.04.009.
[42] T. Salles, L. Rocha, G.L. Pappa, F. Mourão, W. Meira Jr., M. Gonçalves, Temporally-aware algorithms for document classification, in: Proceedings of the Thirty-third International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, ACM, New York, NY, USA, 2010, pp. 307–314.
[43] D.D. Lewis, Evaluating and optimizing autonomous text classification systems, in: Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995, pp. 246–254.
[44] T. Weinbrenner, Genetic programming techniques applied to measurement data, 1997, Diploma Thesis, Institute for Mechatronics, Technical University of Darmstadt, Germany.

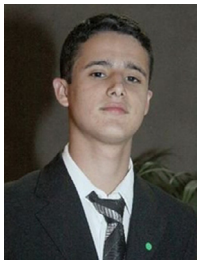
**Felipe Viegas** is a Ph.D. student at Federal University of Minas Gerais. He holds a M.S. in Computer Science in Federal University of Minas Gerais (2015), and a B.S. from Federal University of São João Del Rei, Brazil (2012). His research interests include information retrieval and machine learning.



**Leonardo Rocha** is an Assistant Professor at Computer Science Department at Federal University of São João Del Rei, Brazil. He holds a Ph.D. in Computer Science from Federal University of Minas Gerais, Brazil (2009), a M.S. in Computer Science in Federal University of Minas Gerais (2005), and a B.S. from Federal University of Minas Gerais, Brazil (2003). His research interests include Information Retrieval, Data Mining, Machine Learning, Data Mining and Recommender Systems, having published about 80 papers in these areas.

**Marcos Gonçalves** is an Assistant Professor at Computer Science Department at Federal University of Minas Gerais. He holds a Ph.D. in Computer Science from Virginia Tech (2004), a M.S. in Computer Science in University of Campinas, Brazil (1997), and a B.S. from Federal University of Ceará, Brazil (1995). He has served as referee on different journals (TOIS, TKDE, IP&M, etc.) and at several conferences (SIGIR, CIKM, JCDL, etc.). His research interests include information retrieval, digital libraries and text mining in general, having published more than 100 papers in these areas.

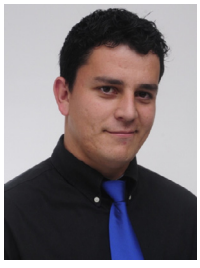
**Fernando Mourão** is Data Scientist at Seek AI Labs, Brazil. He holds a Ph.D. (2014), a M.S. (2008) and a B.S. (2007) in Computer Science from Federal University of Minas Gerais, Brazil. His research interests include Information Retrieval, Data Mining, Machine Learning, Data Mining and Recommender Systems, having published about 50 papers in these areas.



**Giovanni Sá** is a Software Developer and System Analyst, and develops Public Health Systems in São João del-Rei, Brazil. He holds a Bachelor in Computer Science from Federal University of São João del-Rei, Brazil (2014). Over his undergraduate formation, he has developed several researches on the field of Data Mining, focusing on Feature Selection and Classification. His research interests also includes Natural Language Processing, Sentiment and Social Media Analysis, having published a few important papers in these areas.

**Thiago Salles** holds a M.Sc. degree in Computer Science from Federal University of Minas Gerais. He is a Ph.D. student at the same university. His research focus is on machine learning methods for automatic data classification. He has 8+ years experience with machine learning theory and practice, with published work on the major conferences of the area.

**Guilherme Andrade** holds a degree in Computer Science from the Federal University of São João del Rei (2012) and a Master's Degree in Computer Science at Federal University of Minas Gerais (2014). He currently is a Ph.D. candidate in Computer Science from the Computer Science Department (DCC) at the Federal University of Minas Gerais, working on research in the areas of high performance computing in heterogeneous architectures.



**Isac Sandin Ribeiro** is a Senior System analyst at RV Tecnologia e Sistemas. He holds a M.S. in Computer Science in Federal University of Minas Gerais, Brazil (2015) and a B.S. from Federal University of São João del Rei, Brazil (2012). His research interests include information retrieval, digital libraries and text mining in general, having published articles in conferences like WSDM and IEEE and a derivative work published in ACM SIGIR.