Pattern Recognition Letters 100 (2017) 59–66

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

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

João Paulo Papa [a,*], Gustavo Henrique Rosa [a], Luciene Patrici Papa [b]

[a] Department of Computing, São Paulo State University, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, Bauru, 17033-360, Brazil
[b] São Paulo Southwestern College, Av. Prof. Célso Ferreira da Silva, 1001, 14-01, Avaré, 18707-150, Brazil

ARTICLE INFO

Article history:
Received 9 April 2017
Available online 5 October 2017

Keywords:
Feature selection
Geometric Semantic Genetic Programming
Optimum-path forest

ABSTRACT

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

1. Introduction

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

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

A number of works modeled the problem of feature selection as a nature-inspired-based optimization task. Nakamura et al. [21] and Rodrigues et al. [30] proposed the Binary Bat Algorithm for feature selection purposes, being the optimization problem guided by the accuracy of the Optimum-Path Forest (OPF) [24–26] classifier over a validating set. [11] were one of the first to introduce the term *swarm feature selection*, where the well-known Particle Swarm Optimization (PSO) was used to select features in the context of hyperspectral remote sensing image classification. Non-wrapper approaches can be referred to as well, such as the work by [22], which employed evolutionary optimization for feature construction in benchmarking datasets and symbolic learning.

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

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

* Corresponding author.
E-mail address: papa@fc.unesp.br (J.P. Papa).

https://doi.org/10.1016/j.patrec.2017.10.002

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

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

Therefore, the main contributions of this paper are twofold:

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

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

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

2. Geometric semantic genetic programming

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

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

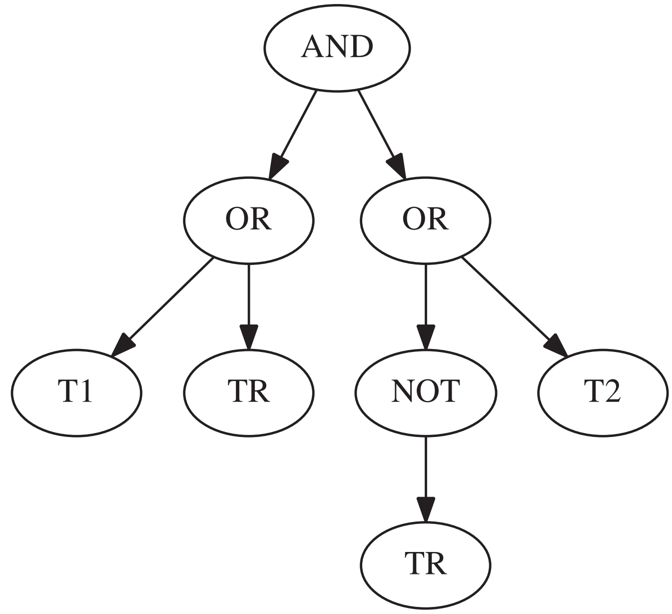

Fig. 1. Offspring generated by means of the semantic crossover defined in Eq. (1).

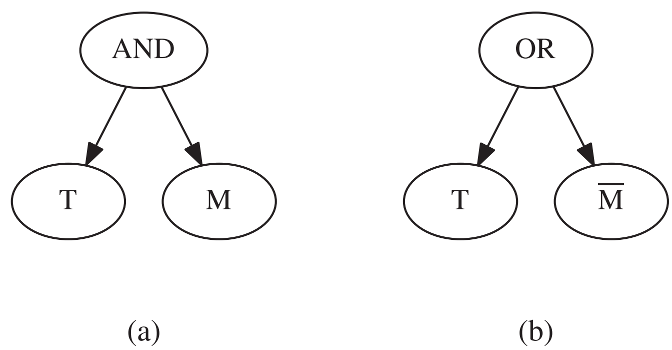

Fig. 2. Tree-like representation concerning the following expressions: (a) T AND M, and (b) T OR $\overline{M}$.

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

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

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

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

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

$$T_M = \begin{cases} T \text{ AND } M & \text{with probability } 0.5 \\ T \text{ OR } \overline{M} & \text{otherwise,} \end{cases} \tag{2}$$

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

[1] By logic function we mean an “OR” or “AND” operator, for instance.