Neurocomputing 133 (2014) 342–357

Contents lists available at ScienceDirect

Neurocomputing

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

# A niching genetic programming-based multi-objective algorithm for hybrid data classification

Marconi de Arruda Pereira [a,b], Clodoveu Augusto Davis Júnior [c], Eduardo Gontijo Carrano [d], João Antônio de Vasconcelos [d,e]

[a] Departamento de Computação (DECOM/CEFET-MG), CEFET-MG Av. Amazonas, 7675, 30510-000 Belo Horizonte, MG, Brasil
[b] Electrical Engineering Graduate Program (PPGEE/UFMG), UFMG Av. Antonio Carlos, 6627, 31270-010 Belo Horizonte, MG, Brasil
[c] Departamento de Ciência da Computação (DCC/UFMG), UFMG Av. Antonio Carlos, 6627, 31270-010 Belo Horizonte, MG, Brasil
[d] Departamento de Engenharia Elétrica (DEE/UFMG), UFMG Av. Antonio Carlos, 6627, 31270-010 Belo Horizonte, MG, Brasil
[e] Evolutionary Computing Laboratory (LCE/UFMG), UFMG Av. Antonio Carlos, 6627, 31270-010 Belo Horizonte, MG, Brasil

ARTICLE INFO

*Article history:*
Received 3 May 2013
Received in revised form
5 December 2013
Accepted 6 December 2013
Communicated by Bo Shen
Available online 22 January 2014

*Keywords:*
Classification rules
Spatial data mining
Genetic programming
Multi-objective algorithm

ABSTRACT

This paper introduces a multi-objective algorithm based on genetic programming to extract classification rules in databases composed of hybrid data, i.e., regular (e.g. numerical, logical, and textual) and non-regular (e.g. geographical) attributes. This algorithm employs a niche technique combined with a population archive in order to identify the rules that are more suitable for classifying items amongst classes of a given data set. The algorithm is implemented in such a way that the user can choose the function set that is more adequate for a given application. This feature makes the proposed approach virtually applicable to any kind of data set classification problem. Besides, the classification problem is modeled as a multi-objective one, in which the maximization of the accuracy and the minimization of the classifier complexity are considered as the objective functions. A set of different classification problems, with considerably different data sets and domains, has been considered: wines, patients with hepatitis, incipient faults in power transformers and level of development of cities. In this last data set, some of the attributes are geographical, and they are expressed as points, lines or polygons. The effectiveness of the algorithm has been compared with three other methods, widely employed for classification: Decision Tree (C4.5), Support Vector Machine (SVM) and Radial Basis Function (RBF). Statistical comparisons have been conducted employing one-way ANOVA and Tukey's tests, in order to provide reliable comparison of the methods. The results show that the proposed algorithm achieved better classification effectiveness in all tested instances, what suggests that it is suitable for a considerable range of classification applications.



## 1. Introduction

Rule mining is one of the steps for knowledge discovery in databases (KDD) [15]. It has been studied in a wide range of practical applications in which such information can be useful. The objective of rule mining is to identify rules that can be used for classifying or identifying data. In the literature, several approaches are being employed for extracting non-trivial knowledge from databases. Amongst these approaches, it is possible to cite unsupervised learning algorithms [11] for classification rule building, such as *k*-nearest neighbors (KNN) [47,46,21] and Bayesian classifiers [3,6,38,40] and also supervised algorithms, like decision trees (DT) [8,40,22], artificial neural networks (ANN) [26,2,29,19,20] and support vector machines (SVM) [42,9,27,30]. Furthermore, there are works that propose evolutionary-based tools for handling with classification problems [16], such as genetic algorithms (GA) [35], genetic programming (GP) [4], artificial immune systems [1], ant colony algorithms [34] and particle swarm optimization.

Algorithms that are capable of directly handling hybrid databases, without data preprocessing, have not been found in the literature. A database is said hybrid when it is composed of conventional attributes (e.g. numerical, textual, and logical) and unconventional attributes (e.g. geographical). Usually, the algorithms that deal with hybrid data adopt some alternative structure for representing the unconventional attributes as conventional ones. For instance, the works [49,25,48], which deal with geographic data, employ particular schemes for representing the geographic entities. In general cases, these alternative representations are not desirable, since the

*E-mail addresses:* marconi.arruda@gmail.com, marconi_arruda@yahoo.com.br (M. de Arruda Pereira), clodoveu@dcc.ufmg.br (C.A. Davis Júnior), egcarrano@ufmg.br (E. Gontijo Carrano), jvasconcelos@ufmg.br (J.A. de Vasconcelos).

0925-2312/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.12.048

performance of the classification algorithm and the interpretation of the results become highly dependent on the representation scheme adopted. Actually, there are many geographic databases that use a pattern model to represent and store the attributes [12,13]. These databases use functions that are well known [12], but just a few classification algorithms explore this rich set of functions [5]. Moreover, the existing methods require data to be preprocessed, and they are not able to mix conventional and unconventional functions in the same classification rule.

Genetic programming is a class of optimization algorithms that generates a population of individuals and evolves them along a generational process [24]. It employs genetic operators inspired by nature, like the ones employed in traditional GAs (selection, crossover, mutation, etc.), but they differ in their structure. If a proper evaluation mechanism is adopted, genetic programming becomes able to deal with complex classification problems, including the possibility of handling with multiple objectives simultaneously.

This paper proposes a multi-objective genetic programming algorithm that has been specially designed to classify conventional and hybrid data in real world databases. An improved niche strategy [18] and a population archive are used to increase the ability of extracting suitable rules for all problem classes present in the database. The algorithm is also very flexible; in such a way that the user can choose the function set that should be employed to build the rules, regardless the attribute types (numerical, textual, geographical, etc.). This characteristic makes the proposed approach adaptable to virtually any kind of problem, without structural changes in the algorithm. A preliminary version of the proposed algorithm can be found in [36], in which the flexibility of the algorithm is illustrated in spatial data mining problems.

The efficiency of the algorithm is measured using the following data sets that are detailed in Section 3:

- Hepatitis and wine, both from UCI machine learning repository. These data sets are composed of regular attributes (numerical and textual) and the instances are classified into two and three classes respectively.[1]
- Digital gases analysis (DGA) problem, where three real data sets of power transformers are classified into three classes [35].
- City Development, in which cities are classified into three classes. This data set is composed of conventional (categorical/numerical) and geographical (points, lines and polygons) attributes.[2]

The algorithm proposed here is employed to maximize classification effectiveness and to minimize rule size. It has been conceived to handle a wide range of real world classification problems. Furthermore, the characteristics of the algorithm suggest that it is well suited for dealing with unbalanced data sets (i.e., data sets in which the number of samples in some class is considerably higher than in the others). Problems with unbalanced data sets are notoriously harder to solve [33], especially when more than two classes are involved [31].

Finally, it is important to emphasize that all problems considered in this work have been modeled in such a way that other classical algorithms could be used and their performances could be assessed to establish a fair comparison with the proposed algorithm. For that purpose, three different classification assessment strategies to select and to test the obtained rules have been proposed.

[1] Wine: http://archive.ics.uci.edu/ml/datasets/Wine, Hepatitis: http://archive.ics.uci.edu/ml/datasets/Hepatitis.

[2] http://www.geominas.mg.gov.br/ (accessed 2011).

This paper is structured as follows. The proposed algorithm is described in Section 2. Results achieved by the proposed algorithm for the considered problems are presented in Section 3, along with comparisons to the results of three classical approaches. Finally, Section 4 presents the conclusions that could be drawn from the study.

## 2. Proposed algorithm

### 2.1. Algorithm modeling

The possible solution or individual in our MOGP (multi-objective genetic programming) algorithm is represented by a Boolean predicate, defined in the same manner as the WHERE clause of the structured query language's (SQL) SELECT statement [14]. The adequacy of the individual to the problem is assessed by using a vector of objective functions, which determines its selection probability.

The classification problem is modeled as a bi-objective optimization problem, in which the objectives are (1) to maximize the effectiveness of the extracted rules, and (2) to minimize the complexity (size) of the rules. The main goal of this model is to generate low complexity (small) rules that correctly classify the samples, since smaller rules are easier to understand and tend to avoid overfitting [41,45,50]. The proposed algorithm, during its execution, can only generate valid rules. Since the first objective is often more important than the second one, an external mechanism, which controls the chance of accepting longer solutions during the algorithm execution, is proposed. In this mechanism, the chance of accepting long solutions is kept high at the first iterations and it decreases along the evolutionary process. At the end of the run, the two objectives are equally balanced.

The algorithm is divided in two phases: (1) rule extraction and (2) data classification. The first phase (see Algorithm 1) generates rules using the training set and predefined sets of functions and terminals that are necessary to the genetic programming algorithm. The best rules found in the first phase are used on the second one, in which three different classification strategies can be used. Details about these classification strategies are presented in Section 2.8.

**Algorithm 1.** Pseudo code of the rule extraction algorithm (multi-objective maximization problem).

Input: **Set of functions** *function_set*, **Set of terminals** *terminal_set*, **Population Size (integer)** *n*, **Number of Generations (integer)** *max_number_of_generations*, **Data** *training_data*, **Classes** *classes*.
Output: **Subpopulations of non-dominated solutions**

1. Generate *n* individuals using *function_set* and *terminal_set*;
2. Identify the number of distinct classes in *classes*. Call this quantity *dc*;
3. Create *dc* subpopulations;
4. **for each** individual *i* **do** /* *this loop looks for the best class to label each individual* */
5. *best_fitness*:= 0; *individual_class*:= null;
6. **for each** class *cl* **in** *classes* **do**
7. Using the *training_data*, calculate the fitness of individual *i* considering this individual belonging to class *cl* and store these results *f*;