European Journal of Operational Research 000 (2015) 1–14

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Computational Intelligence and Information Management

A pool-based pattern generation algorithm for logical analysis of data with automatic fine-tuning

Marco Caserta[a,*], Torsten Reiners[b]

[a] IE University and IE Business School, Maria de Molina, 31-B, 28006, Madrid, Spain
[b] Curtin University, Bentley, WA, 6102, Australia

ARTICLE INFO

Article history:
Received 2 October 2013
Accepted 30 May 2015
Available online xxx

Keywords:
Logical Analysis of Data
Data mining
Fine-tuning
bRKGA
Machine learning

ABSTRACT

In this paper, we address the binary classification problem, in which one is given a set of observations, characterized by a number of (binary and non-binary) attributes and wants to determine which class each observation belongs to. The proposed classification algorithm is based on the Logical Analysis of Data (LAD) technique and belongs to the class of supervised learning algorithms. We introduce a novel metaheuristic-based approach for pattern generation within LAD. The key idea relies on the generation of a pool of patterns for each given observation of the training set. Such a pool is built with one or more criteria in mind (*e.g.*, diversity, homogeneity, coverage, etc.), and is paramount in the achievement of high classification accuracy, as shown by the computational results we obtained. In addition, we address one of the major concerns of many data mining algorithms, *i.e.*, the fine-tuning and calibration of parameters. We employ here a novel technique, called biased Random-Key Genetic Algorithm that allows the calibration of all the parameters of the algorithm in an automatic fashion, hence reducing the fine-tuning effort required and enhancing the performance of the algorithm itself. We tested the proposed approach on 10 benchmark instances from the UCI repository and we proved that the algorithm is competitive, both in terms of classification accuracy and running time.

1. Introduction

Let us consider a binary classification problem, in which one is given a dataset composed of observations belonging to one of two classes, *e.g.*, positive or negative, where the class each observation belongs to is known. A typical data mining problem is the classification problem, *i.e.*, finding the class a new observation, not included in the dataset, belongs to. Binary classification finds a large number of applications, spanning from, *e.g.*, medical diagnosis (Alexe, Alexe, Axelrod, Hammer, & Weissmann, 2005; Hammer & Bonates, 2006), to credit risk rating (Hammer, Kogan, & Lejeune, 2006), from maintenance replacement (Ghasemi & Esameili, 2013), to fault diagnosis (Mortada, Yacout, & Lakis, 2013).

Owing to the fact that classification is such a relevant problem in the data mining field, effective techniques to classify data have been developed. The observations to be classified are characterized by a set of attributes, which are believed to affect the class each observation belongs to. However, the relationship between the value of the attributes and the class is unknown and, therefore, needs to be estimated via a training process. The classifier discovers such rules that map an object to a class, based on the values of the attributes.

Logical Analysis of Data (LAD) was introduced in Boros, Hammer, Ibaraki, and Kogan (1997), and Boros, Hammer, Ibaraki, Kogan, Mayoraz, and Muchnik (2000) and is a data analysis methodology that combines ideas from combinatorial optimization and Boolean functions and belongs to the family of supervised learning techniques. The LAD methodology relies on a “rule learning” mechanism and, therefore, is strongly connected with other popular classification rules presented in the machine learning literature. According to Fürnkranz (1999), many rule learning algorithms are based on a sequential covering procedure, in which the steps to follow are: “Learn a rule that covers part of the given training examples, remove the covered examples from the training set, and recursively learn another rule that covers some of the remaining examples until no examples remain.” LAD fits well within this sequential covering framework. For a more extensive discussion, we refer the reader interested in the connection between LAD and other “rule learning” techniques as well as on the “justifiability” of LAD to Boros, Crama, Hammer, Ibaraki, Kogan, and Makino (2011). Along the same line, Dembczynski, Kotlowski, and Slowinski (2010) establish a clear connection between LAD and other methods that fall within the framework of

[*] Corresponding author. Tel.: +34 915689600.
E-mail addresses: marco.caserta@ie.edu (M. Caserta), t.reiners@curtin.edu.au (T. Reiners).

http://dx.doi.org/10.1016/j.ejor.2015.05.078
0377-2217/© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

"rule ensamble algorithms," *e.g.*, algorithms that exploit boosting schemes (Cohen & Singer, 1999; Dembczynski et al., 2010), or based on the set covering machine (Marchand & Shawe-Taylor, 2002), or on the linear programming-boost framework (Kotlowski & Slowinski, 2009), or other linear programming-based approaches (Malioutov & Varshney, 2013).

Along the same line, LAD shares similarities with rough set methods. Rough set theory, proposed by Pawlak (1982, 1991), is a mathematical approach often employed to tackle classification problems (Bazan, Nguyen, Nguyen, Synak, & Wróblewski, 2000; Greco, Matarazzo, & Slowinski, 2001; Pawlak, 1998). In a fashion similar to what is done in LAD, rough set algorithms often rely on the use of discretization techniques, generation of reducts, *i.e.*, patterns in LAD, and the definition of a rough membership function. An extensive overview of rough set methods for data analysis is provided by Chikalov, Lozin, Lozina, Moshkov, Nguyen, Skowron, and Zielosko (2013), where three different methods for data analysis, *i.e.*, test theory, rough set, and LAD are thoroughly presented.

As pointed out by Malioutov and Varshney (2013), often what differentiates rule learning approaches is how the training procedure is managed, since each method might define a specific optimization objective (based, *e.g.*, on statistical theory, mathematical programming, etc.) In addition, these methods might vary in the way in which the rule learning problem is solved (*e.g.*, in a greedy fashion, using an exact method from mathematical programming, with brute-force approaches, via metaheuristics, etc.) In this regard, the LAD approach we propose in this paper can be characterized as: (a) a rule-based method that employs Boolean reasoning; (b) in which rules are built using a sequential covering-type of approach; (c) where the rule learning mechanism is formalized using mathematical programming; and (d) in which the optimization problem of (c) is solved using a pool-based metaheuristic.

LAD is based on four basic steps: (i) data binarization, (ii) support feature selection, (iii) pattern generation, and (iv) theory formation. (For a detailed introduction on methodology and applications of LAD, see also Chikalov et al. (2013).) It is well understood that one of the key features of LAD is concerned with the pattern generation process. As mentioned in Hammer, Kogan, Simeone, and Szedmk (2004), patterns are fundamental blocks in LAD as well as in many other rule induction algorithms, *e.g.*, C4.5 rules (Quinland, 1993), AQ17-HCI (Wnek & Michalski, 1994), RIPPER (Cohen, 1995), SLIPPER (Cohen & Singer, 1999), RuleFit (Friedman & Popescu, 2008), and ENDER (Dembczynski et al., 2010). All these algorithms, thus, place special emphasis on identifying a small subset of patterns. It is interesting to notice, though, that on the one hand, empirical evidence shows that the way in which patterns are built within LAD has a strong bearing on the classification accuracy. On the other hand, due to the large number of patterns that can be constructed from a dataset, the algorithm used to build such patterns strongly determines the practical usability of LAD, especially when it comes to dealing with very large datasets.

It is worth pointing out that LAD and many of the classical rule learning techniques share the same advantage, *i.e.*, knowledge represented by rules is generally easier to interpret by people. In addition, as highlighted by Boros et al. (2011), LAD patterns are "justifiable," thus enhancing the ability to motivate the reasons behind a certain decision and, consequently, allowing for human insight into what is learned. However, these methods face the same problem, *i.e.*, the search space of literals can become intractable. Consequently, the proposed pool-based approach for pattern generation can be extended to other rule learning methods.

The goal of this paper is to present a novel metaheuristic to generate patterns within LAD, which allows to construct patterns with a predefined criterion in mind, while limiting the computational time required by the pattern generation phase. We introduce the concept of *pool of patterns*, where each pattern generated by a metaheuristic scheme is added to the pool only if some criteria are satisfied (*e.g.*, diversity, coverage, homogeneity, etc.) The contribution of the paper is twofold: On the one hand, a methodological contribution is made, since it is the first time that a metaheuristic is used for generating a pool of patterns with pre-specified characteristics within LAD and, on the other hand, a further contribution, based on empirical evidence, is made in the direction of accuracy and computational running time, thus allowing to extend the use of LAD to large datasets.

In addition, in this paper we propose a solution to one of the major drawbacks of a number of data mining algorithms, *i.e.*, the calibration and fine-tuning of the algorithmic parameters, via the use of a novel technique to automatically fine-tune such parameters. In the literature, it is quite common to find data mining techniques that require extensive calibration to reach satisfactory classification results. However, the way in which the calibration process should be conducted is very seldom described and, even when guidelines about the fine-tuning phase are provided, such phase still requires a large amount of time and effort on the side of the researcher. In this paper, we propose the use of a novel approach that allows to automatically fine-tune all the parameters required by the algorithm. To the best of the authors' knowledge, it is the first time that an automatic fine-tuning technique is used in the context of a LAD algorithm.

LAD requires the dataset to be in binary format. Therefore, the first step transforms any non-binary value into a set of binary attributes. Such binarization process is carried out introducing a set of cutpoints for each non-binary attribute, as illustrated in Boros et al. (1997, 2000). Let us consider a non-binary attribute a_j and a set of cutpoints c_{jk}, with $k = 1, \ldots, n_j$. We binarize attribute a_j introducing a set of n_j binary values a_{jk} whose value is 1 if $a_j \geq c_{jk}$ and 0 otherwise. It is worth observing, though, that the number of binary values needed to convert a numerical variable into a set of binary values can be quite large. Consequently, the next step of LAD is devoted to finding a minimum size support set that allows to distinguish any two observations belonging to different classes, *i.e.*, a set of cutpoints that is sufficient to preserve the information contained in the original non-binary variables. This phase, the features selection phase, is modeled using a Set Covering problem, which belongs to the class of $\mathcal{NP}$-hard problems and, therefore, gives rise to the first difficulty in the LAD process. Finding the minimum size support set might prove to be too difficult and, therefore, the associated set covering problem is often solved heuristically (Boros et al., 2000).

Once a minimum size support set is identified, the next step concerns the creation of patterns, *i.e.*, subcubes having a nonempty intersection with one of the two subsets of the data, and an empty intersection with the other subset. Once a pool of positive and negative patterns has been produced, the last step of LAD is concerned with the creation of a theory. The key assumption in this step is that an observation covered by some positive patterns but none (or just a few) of the negative patterns should be classified as positive. Thus, we finally build a discriminant function to compute a weighted score for each new observation. The value of such a score determines how this new observation is classified.

In the LAD literature, special attention has been given to the pattern generation phase. It is well understood that the prevalence, or absence, of patterns of one class provide strong indications about the nature of new, unclassified, observations (Alexe, Alexe, & Hammer, 2006; Alexe & Hammer, 2006; Boros et al., 2000; Hammer et al., 2004). Consequently, special emphasis has been placed on creating patterns that are "good predictors" of the class each new observation belongs to. Broadly speaking, two main approaches for pattern generations emerge from the literature. On the one hand, the traditional approach is enumeration-based or constructive in nature, *e.g.*, bottom-up and top-down approaches (Boros et al., 1997; Boros et al., 2000; Hammer et al., 2004). These types of approaches are computationally quite expensive and pose some limitations in terms of the degree, *i.e.*, the length, of the patterns produced. A second approach for pattern generation relies on mathematical modeling. In their