Expert Systems with Applications 42 (2015) 833–843

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Pattern generation for multi-class LAD using iterative genetic algorithm with flexible chromosomes and multiple populations

Hwang Ho Kim, Jin Young Choi *

Department of Industrial Engineering, Ajou University, Suwon, Republic of Korea

ARTICLE INFO

Article history:
Available online 6 September 2014

Keywords:
Multi-class classification
Logical analysis of data
Decomposition
Genetic algorithm
Classification accuracy

ABSTRACT

In this paper, we consider a pattern generation method for multi-class classification using logical analysis of data (LAD). Specifically, we apply two decomposition approaches—one versus all, and one versus one—to multi-class classification problems, and develop an efficient iterative genetic algorithm with flexible chromosomes and multiple populations (IGA-FCMP). The suggested algorithm has two control parameters for improving the classification accuracy of the generated patterns: (i) the number of patterns to select at the termination of the genetic procedure; and (ii) the number of times that an observation is covered by some patterns until it is omitted from further consideration. By using six well-known datasets available from the UCI machine-learning repository, we performed a numerical experiment to show the superiority of the IGA-FCMP over existing multi-class LAD and other supervised learning algorithms, in terms of the classification accuracy.

1. Introduction

Supervised learning is a machine-learning task for gaining information regarding data characteristics such as class or quality by learning from training data. Supervised learning analyzes training data and produces a model or function that can be used to give the correct result when new data (so-called test data) is given as input, without knowing the target *a priori* (Bishop, 2006; Jiawei & Kamber, 2001; Lugosi, 2002; Mitchell, 1997). Many algorithms using supervised learning have been developed and applied to classification problems in novel instances. These can be divided into binary classification and multi-class classification, depending on the number of classes K in a dataset (Berry & Linoff, 1997; Gehler & Nowozin, 2009).

For binary classification problems with $K = 2$, various classification methods have been suggested, such as decision trees (J48) (Safavian & Landgrebe, 1991), support vector machines (SVMs) (Schölkopf & Smola, 2002; Furey et al., 2000), and neural networks (NNs) (Hagan, Demuth, & Beale, 1996). Further, in the literature cited, their performance was evaluated through numerical experiments. However, it is known that they have some limitations regarding their ability to explain the causes of the classification results. To rectify this, an efficient method exhibiting high accuracy and explanatory power, called logical analysis of data (LAD), has recently been proposed and used in the medical, service, and manufacturing fields, among others (Hammer, Kogan, & Lejeune, 2012). LAD is an efficient data analysis methodology based on patterns containing hidden structural information of binary training data (Boros, Hammer, Ibaraki, & Kogan, 1997; Boros et al., 2000; Crama, Hammer, & Ibaraki, 1988; Hammer & Bonates, 2006; Han, Kim, Jeong, & Yum, 2011).

Meanwhile, for multi-class classification problems with $K \geqslant 3$, various algorithms have been provided, and we can categorize them in two ways: (i) the direct approach, and (ii) the decomposition approach. The direct approach deals with K classes directly by finding classifiers for K classes simultaneously, including k-nearest neighbors, naïve Bayes, and decision trees (Aly, 2005). However, there are disadvantages because its performance is influenced by noise data and suffers from high computational complexity in large datasets with many instances and classes. By contrast, the decomposition approach divides the original problem into several binary classification problems. These can be solved efficiently using binary classifiers (Tax & Duin, 2002). The decomposition approach follows two basic strategies: One versus All (OvA) and One versus One (OvO) (Galar, Fernández, Barrenechea, Bustince, & Herrera, 2011; Mayoraz & Alpaydin, 1999; Tax & Duin, 2002).

OvA decomposition splits K into two classes—dividing the class under consideration from the remaining $(K - 1)$ classes—eventually building K different binary classification problems. Then, for each problem, we find one classifier. For example, if we find the classifier f_1 from a dataset with four classes, as in Fig. 1, we let all instances in

* Corresponding author.
E-mail address: choijy@ajou.ac.kr (J.Y. Choi).

http://dx.doi.org/10.1016/j.eswa.2014.08.050
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

Class I stand as the positive set, and all instances in the remaining classes (II, III, and IV) stand as the negative set, as in Fig. 2. Then, by combining all classifiers $f_i (i = 1, 2, 3, 4)$, we derive the function

$$f(x) = \arg\max_i f_i(x), \tag{1}$$

to classify test data x as class i, with the index giving the largest value f_i, $\forall i$ (Tax & Duin, 2002).

The alternative strategy, OvO, builds $K(K-1)$ binary classification problems by considering all possible class pairs. With OvO, we find one classifier for each problem. For example, if we find the classifier f_{12} to identify Class I over Class II from a dataset with four classes, as in Fig. 1, we let Class I stand as the positive set and Class II as the negative set, as in Fig. 3. Then, by combining all classifiers $f_{ij} (i = 1, 2, 3, 4, j \neq i)$, we derive the function

$$f(x) = \arg\max_i \left(\sum_{j \neq i} f_{ij}(x) \right), \tag{2}$$

to classify test data x as class i, corresponding to the index giving the largest value $\sum_{j \neq i} f_{ij}$, $\forall i$ (Tax & Duin, 2002).

In a similar effort to improve classification accuracy and explanatory power, LAD for multi-class classification has also been proposed (Moreira, 2000; Mortada, Yacout, & Lakis, 2013; Herrera & Subasi, 2013). We can divide multi-class LAD into OvA and OvO as well, depending on the decomposition type. Moreira (2000) and Mortada (2010) studied OvO-type LAD algorithms for multi-class classification problems. Moreira (2000) proposed two methods for breaking down a multi-class classification problem into binary-class problems using OvO. The first method, called a multi-layered 2-class LAD, used the typical OvO approach, which does not require any modification of the structure of the original LAD algorithm. The second method modified the architecture of the pattern generation and theory formation steps in the original LAD algorithm, and introduced the decomposition matrix used for recording multi-class patterns generated through iterations of LAD. In comparing these two methods, the second approach is less accurate, but more intuitive. Moreover, because both are enumeration-based techniques, combining and enumerating random binary attributes (called "literals") as a way to find all patterns in the solution space, they suffer from high computational complexity and lengthy execution times.

To improve the classification accuracy of the second OvO-type method, Mortada (2010) proposed a multi-class LAD algorithm, integrating the second approach by Moreira (2000) with a pattern generation method using the mixed 0–1 integer and linear programming (MILP) model by Ryoo and Jang (2009). Mortada (2010) showed that the classification accuracy of the proposed method is higher than the LAD models proposed by Moreira (2000). Furthermore, Herrera and Subasi (2013) proposed an algorithmic approach using MILP to build OvA-type LAD patterns in a multi-class dataset efficiently. However, its classification accuracy is worse than with OvO (Tax & Duin, 2002). This is because the negative set using OvA is the mixed set of class-specific properties for all the remaining classes, making it difficult to represent the characteristics pertaining exclusively to the positive set.

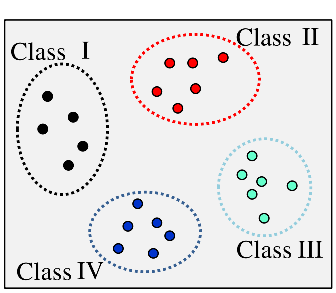

Fig. 1. Example of a dataset containing four classes.

Overall, these MILP-based approaches face three challenges. First, such approaches generate only a single pattern per iteration by finding an optimal solution to the considered MILP, and then delete some data covered by the generated pattern. In generating patterns, the next iteration is applied to the remaining dataset until the data is removed. Using this procedure, we observe that deleting data covered by one pattern can affect the generation of other patterns in later iterations. In other words, were it not removed, the deleted data might have been useful for providing relevant information for identifying subsequent patterns. Second, if most generated patterns cover only a slight amount of data, it will take extensive time to find the multi-class LAD patterns in large-sized data. In the worst case, if each pattern covers only one data item, it would need to be repeated for each observation. Finally, it would take too much time and effort to develop and solve MILP models for large-sized data.

Motivated by these observations, we propose a new pattern generation algorithm for multi-class LAD, namely the "iterative genetic algorithm with flexible chromosomes and multiple populations" (IGA-FCMP). This paper is devoted to explaining our proposal. The algorithm is designed to improve the performance of generated patterns by using the following control parameters: (i) the number of patterns to select at the termination of a genetic procedure; and (ii) the number of times that an observation is covered by some patterns before it is deleted. More specifically, we designed two types of IGA-FCMP using OvA- and OvO-type decompositions for multi-class LAD, and we shall demonstrate the superiority of their performance over existing multi-class LAD, including MILP-based LAD, and other supervised learning algorithms. We shall do so with a numerical experiment showing that our algorithm increases the efficiency of OvO and the accuracy of OvA simultaneously.

The rest of this paper is organized as follows. Section 2 introduces the notations and basic concept behind the LAD framework. Section 3 suggests a new pattern generation method for multi-class LAD using an iterative genetic algorithm with special structures, especially with OvA- and OvO-type decomposition methods. We explain how we tested the performance of the proposed algorithm with a numerical experiment in Section 4. Finally, in Section 5, we discuss the direction of future work in this area.

2. Basic concept of LAD

In this section, we explain the notations and summarize the basic idea behind the LAD framework. There are four steps (see Fig. 4): data binarization, support-set generation, pattern generation, and theory formulation (Boros et al., 2000).

2.1. Data binarization

All the input data for the LAD framework must be binary. Thus, binarization is usually performed as the first step of LAD. Non-binary input variables require a data conversion process. In LAD, we use several cut points (cp) for the binarization of numerical data. In general, cp is defined as the median value of the observation O^+ from the positive set Ω^+ and the observation o^- from the negative set Ω^-. For each cp, we define a binary attribute and assign its value as follows. If the value of an observation is larger than cp, the observation is 1. Otherwise, it is 0. We represent them by using a literal, either a Boolean variable b or its complement. A term t is a conjunction of literals such as $t = b_1 b_2 \bar{b}_3 \bar{b}_4$. The degree G_t of a term t is the number of literals in it.