Neurocomputing 73 (2010) 2837–2845

Contents lists available at ScienceDirect

Neurocomputing

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

Building binary-tree-based multiclass classifiers using separability measures

Ana Carolina Lorena [a,*], André C.P.L.F. de Carvalho [b]

[a] Centro de Matemática, Computação e Cognição, Universidade Federal do ABC, 09.210-170 Santo André, SP, Brazil

[b] Departamento de Ciências de Computação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo—Campus de São Carlos, Caixa Postal 668, 13560-970 São Carlos, SP, Brazil

ARTICLE INFO

Available online 25 June 2010

Keywords:
Supervised machine learning
Multiclass classification problems
Hierarchical decomposition

ABSTRACT

Various popular machine learning techniques, like support vector machines, are originally conceived for the solution of two-class (binary) classification problems. However, a large number of real problems present more than two classes. A common approach to generalize binary learning techniques to solve problems with more than two classes, also known as multiclass classification problems, consists of hierarchically decomposing the multiclass problem into multiple binary sub-problems, whose outputs are combined to define the predicted class. This strategy results in a tree of binary classifiers, where each internal node corresponds to a binary classifier distinguishing two groups of classes and the leaf nodes correspond to the problem classes. This paper investigates how measures of the separability between classes can be employed in the construction of binary-tree-based multiclass classifiers, adapting the decompositions performed to each particular multiclass problem.

1. Introduction

Multiclass classification using machine learning (ML) techniques consists of inducing a function $f(\mathbf{x})$ from a data set composed of pairs $(\mathbf{x}_i, y_i)$, where each $\mathbf{x}_i$ is a data item and $y_i \in \{1, \ldots, k\}$, where $k > 2$, corresponds to the desired output or class of $\mathbf{x}_i$. Some popular learning techniques, like support vector machines (SVMs) [1], were originally proposed to solve binary classification problems ($k=2$).

Two approaches have been followed in the literature for dealing with multiclass problems using binary classifiers:

- adaptation of the internal operations of the algorithm used to induce the classifier;
- decomposition of the original multiclass classification problem into a set of binary (two-class) classification problems.

In this paper, we investigate the second approach. In this approach, the decomposition may be performed hierarchically, generating a binary-tree-based multiclass classifier. In general, the introduction of a hierarchy in a multiclass application can reduce the complexity involved in its solution. The idea is to perform more general discriminations first, which are successively refined until the final classification is obtained.

According to this strategy, the binary predictors and the problem classes are represented as nodes in a graph or tree. The root node usually contains a predictor that divides all problem classes into two groups. These groups are also recursively divided into two parts each, until one unique class remains.

The hierarchical structure adopted may influence the quality of the solution in the multiclass problem. Thus, the classes associated with each node may have a strong influence on the final classification accuracy. This work investigates the use of separability measures to define the binary partitions contained in the hierarchy, allowing to define the tree structure according to the characteristics of each multiclass data set. The aim was to investigate how different separability criteria could be used for defining the binary partitions of classes in a hierarchy. The ability of each of these criteria in obtaining suitable hierarchical structures was proven in a controlled set of experiments involving several benchmark multiclass data sets.

This paper is structured as follows: Section 2 reviews the main existent approaches for generalizing binary learning techniques to solve multiclass problems. Section 3 explains how the binary-tree-based classifiers are built in this work. Section 4 describes the experiments performed in the evaluation of the obtained trees and analyses the results obtained in these experiments. Section 5 concludes this paper and provides suggestions for future works.

* Corresponding author.
E-mail addresses: ana.lorena@ufabc.edu.br (A.C. Lorena), andre@icmc.usp.br (A.C. de Carvalho).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.03.027

2. Multiclass approaches

As previously mentioned, there are two main approaches for the generalization of binary ML techniques to deal with multiclass problems. The first modifies the ML technique training algorithm, creating multiclass versions of this algorithm. Nevertheless, such modifications usually are not trivial and may lead to costly algorithms [2,3]. Consequently, the second alternative, which decomposes the original problem into several binary subproblems, is more frequently employed.

This section presents several decomposition strategies investigated in the literature. They can be broadly divided into two groups: code-matrix based and hierarchical.

2.1. Code-matrix strategies

The code-matrix based approach unifies various decomposition strategies [4], which can be generally represented by a code-matrix $\mathbf{M}$. Each row of this matrix has a binary code associated with one of the classes and each column of $\mathbf{M}$ is associated with a binary classifier, defining a binary partition of the k classes for this particular classifier. Thus, $\mathbf{M}$ has $k \times l$ dimension, in which l denotes the number of binary classifiers used. Each element of $\mathbf{M}$ assumes values in the set $\{-1,0,+1\}$. An element m_{ij} with the value +1 indicates that the class associated to row i assumes positive label in the classifier f_j's induction. The value -1 designates a negative label and the value 0 indicates that data from class i do not participate on the classifier f_j induction. Binary classifiers are trained to learn the labels represented in the columns of $\mathbf{M}$. Fig. 1 presents examples of code-matrices for a problem with four classes.

In these strategies, a new data item $\mathbf{x}$ is classified by evaluating the predictions of the l classifiers, which generate a vector $\mathbf{f}(\mathbf{x})=(f_1(\mathbf{x}),\ldots f_l(\mathbf{x}))$. This vector is then compared to the rows of $\mathbf{M}$. The data item is attributed to the class (row) with the closest code according to some distance measure. This process is also named decoding.

Several strategies have been employed for the decomposition of the multiclass problem. Among the most common, one can cite the one-against-all (OAA) [5], the one-against-one (OAO) [6] and the use of error correcting output codes (ECOC) [7].

In the OAA strategy (Fig. 1a), given a problem with k classes, k binary classifiers $f_i(\mathbf{x})$ are induced. Each classifier is trained to distinguish a class i from the others. This technique can be represented by a $k \times k$ matrix, in which the diagonal elements have the value +1 and the others, the value -1.

In the OAO decomposition (Fig. 1b), $k(k-1)/2$ binary classifiers are generated. Each classifier discriminates a pair of classes (i,j), in which $i \neq j$. The code-matrix in this case has dimension $k \times k(k-1)/2$ and each column corresponds to a binary classifier for a pair of classes. In a column representing the pair (i,j), the value of the element in the row i is +1 and the value of the member in the row j is -1. All other elements in the column have the value 0, indicating that instances from the other classes do not participate in this classifier induction.

In an alternative decomposition strategy, Dieterich and Bariki [7] proposed the use of error correcting output codes (ECOCs) to represent the k classes of the multiclass problem. One particular type of ECOC is produced by an exhaustive method and has $2^{k-1}-1$ columns (Fig. 1c). The code of the first class is composed of +1 values. For each other class i, in which $i>1$, the code is composed of alternate runs of 2^{k-i} negative (-1) and positive $(+1)$ labels.

A common criticism to the OAA, OAO and ECOC strategies is that all of them decompose the multiclass problem a priori, without considering the properties and characteristics of the data sets [4,8]. The tree definition process presented in this work deals with this problem by considering information from the data set to build the tree structure.

2.2. Hierarchical strategies

An alternative approach to solve multiclass problems with binary predictors is to dispose these classifiers in a hierarchical structure. These structures are composed of nodes and ramifications. Internal nodes correspond to binary classifiers, the ramifications represent the possible outputs of these classifiers and the leaf nodes represent the problem classes.

For the classification of a new instance, the nodes and ramifications are traversed according to the binary classifications produced until a leaf node is reached. In order to define the binary partitions of classes in the hierarchy, which is equivalent to decompose the multiclass problem into a set of binary problems, several alternatives might be followed.

A common type of hierarchical structure is a tree, in which, apart from the root node, each node has just one parent. Fig. 2 illustrates examples of trees for a problem with four classes. The trees require training $k-1$ binary classifiers for a problem with k classes, the lowest number among the decomposition strategies presented so far. For the test phase, in the best case, depending on the tree structure, it is possible to classify a data item in the first node of the tree. In the worst case, the $k-1$ classifiers have to be consulted. Therefore, this structure can accelerate the test phase.

For a problem with $k \geq 3$ classes, there are $\prod_{i=3}^{k} 2i-3$ distinct tree structures [9]. Two possible structures for a problem with four classes are illustrated in Fig. 2. The tree structure may influence the classification accuracy results. Therefore, it is necessary to be careful in the definition of the binary partitions of the classes in each node of the tree. Usually, a specific criterion is recursively applied to subsets of classes, dividing them into two until a single class remains.

Schwenker [10,11] used the concept of confusion classes to define the binary partitions of classes in a tree. This concept leads to subsets of classes. Each subset presents a high similarity between its examples. To determine these subsets, Schwenker [11] recursively applied the k-means algorithm [12], with k equal to 2. In a digit recognition problem, the classification accuracy results of the produced tree, using SVMs as base classifiers, were similar to those of the strategies OAA and OAO.

Takahashi and Abe [13] proposed that nodes in the initial levels of the tree should divide the most separated classes. They

a
$$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$$
b
$$\begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix}$$
c
$$\begin{pmatrix} +1 & +1 & +1 & +1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 & -1 & -1 \\ -1 & +1 & -1 & +1 & -1 & +1 \end{pmatrix}$$

Fig. 1. (a) OAA, (b) OAO and (c) ECOC code-matrices for a problem with four classes.