Discrete Applied Mathematics 172 (2014) 1–11

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

A hybrid classifier based on boxes and nearest neighbors

Martin Anthony [a,*], Joel Ratsaby [b]

[a] Department of Mathematics, The London School of Economics and Political Science, Houghton Street, London WC2A2AE, UK

[b] Electrical and Electronics Engineering Department, Ariel University of Samaria, Ariel 40700, Israel

ARTICLE INFO

Article history:
Received 28 November 2011
Received in revised form 30 July 2013
Accepted 16 August 2013
Available online 20 March 2014

Keywords:
Logical analysis of data
LAD methods
Generalization error
Machine learning
Learning algorithms
Large margin learning

ABSTRACT

In this paper we analyse the generalization performance of a type of binary classifier defined on the unit cube. This classifier combines some of the aspects of the standard methods that have been used in the logical analysis of data (LAD) and geometric classifiers, with a nearest-neighbor paradigm. We assess the predictive performance of the new classifier in learning from a sample, obtaining generalization error bounds that improve as a measure of 'robustness' of the classifier on the training sample increases.

1. Introduction

In this paper we study a method of classifying points of $[0, 1]^n$ into two classes. The classifiers we use combine the use of 'boxes' with a nearest-neighbor approach and for this reason we describe it as a *hybrid* classifier. Both classification by boxes and nearest-neighbor classification have been widely used. For instance, the use of boxes is integral to many of the standard methods used in the logical analysis of data (LAD); see [8,9], for instance.

The primary purpose of this paper is to quantify the performance of the hybrid classifiers by bounding their generalization error. In doing so, we obtain bounds that depend on a measure of how 'robust' the classification is on the training sample. In using real-valued functions to form the basis of the classification, we can also attach some degree of 'confidence' or 'definitiveness' to the resulting classifications, and this could be of some practical use.

In Section 2, we give some background by way of motivation. Chiefly, this is a description of some of the standard methods used in the logical analysis of data, especially as they apply to data in which the data points are not necessarily binary, but have real-valued components. In this context, unions of boxes are used as a key means of classification. (It should be noted that classification by unions of boxes has been more widely studied, not just in the context of LAD; see [10] for instance.)

Section 3 describes a type of 'hybrid' classifier which incorporates some of the features of the LAD (and other) techniques in that it uses unions of boxes. However, the classifiers combine this with a nearest-neighbor paradigm for classifying some regions of the domain. In this section, we define the classifiers, give an example, and discuss the rationale for this method of classification.

Section 4 provides the main theoretical results, giving bounds on the predictive performance (or generalization error) of the classifiers. The bounds we obtain are better if the classifier achieves 'definitively' correct classification of the sample points.

[*] Corresponding author.
E-mail addresses: m.anthony@lse.ac.uk (M. Anthony), ratsaby@ariel.ac.il (J. Ratsaby).

http://dx.doi.org/10.1016/j.dam.2014.02.018

2. Classification using unions of boxes

In standard logical analysis of data (LAD) for binary data, we have some collection of labeled *observations* (or data-points, or training examples) (x_i, b_i), for $i = 1, 2, \ldots, m$, where m is known as the sample size. The observations are the x_i and their labels are the b_i. The $x_i \in \{0, 1\}^n$ for which $(x_i, 1)$ appears among the observations are said to be the *positive observations*; and those for which $(x_i, 0)$ appears are the *negative observations*. We denote the sets of positive and negative observations by D^+ and D^- respectively, and the set of all m observations by D. The primary aim is to find some function $h : \{0, 1\}^n \to \{0, 1\}$, a *hypothesis* or *classifier*, that describes the classifications of the known observations well and therefore, as a result, would act as a reliable guide to how future, as yet unseen, elements of $\{0, 1\}^n$ ought to be classified. This is, indeed, a central issue generally in machine learning. The approach taken in LAD methods involves the use of Boolean functions constructed in precise algorithmic ways from the observations. In the standard LAD method for binary data [11], a disjunctive normal form Boolean function (a DNF) is produced. The terms of this DNF are called *positive patterns*. A (pure) positive pattern is a conjunction of literals which is true on at least one positive observation (in which case we say that the observation is *covered* by the pattern) but which is not true on any negative observation. The classifier is then taken to be the disjunction of a set of positive patterns. A more general technique combines the use of positive patterns with *negative* patterns, conjunctions which cover some negative observations. Points of $\{0, 1\}^n$ are then classified as follows: $x \in \{0, 1\}^n$ is assigned value 1 if it is covered by at least one positive pattern, but no negative patterns; and it is assigned value 0 if it is covered by at least one negative pattern, but no positive patterns. If a point x is covered by both types of pattern (which might well be the case, even if we have been careful to ensure that the observations themselves have only been covered by patterns of one type) then its classification is often determined by using a *discriminant*, which takes into account (perhaps in a weighted way) the number of positive and the number of negative patterns covering it.

These standard LAD techniques apply when the data is binary. However, many applications involve numerical data, in which $D \subseteq [0, 1]^n \times \{0, 1\}$. The LAD methods have been extended to deal with such cases; see [9], for instance. The approach is first to *binarize* the data, so that observations $x \in [0, 1]^n$ are converted into binary observations $x^* \subseteq \{0, 1\}^d$, where, generally, $d \geq n$. The standard way to do so is to use *cutpoints* for each attribute (that is, for each of the n geometrical dimensions). For each coordinate (or dimension) $j = 1, 2, \ldots, n$, let $u_1^{(j)}, u_2^{(j)}, \ldots, u_{k_j}^{(j)}$ be, in increasing order, all the distinct values of the jth coordinate of the observations in D. For each j, let

$$\beta_i^{(j)} = \frac{u_i^{(j)} + u_{i+1}^{(j)}}{2}$$

for $i = 1, \ldots, k_j - 1$. For $j = 1, 2, \ldots, n$ and $i = 1, 2, \ldots k_j - 1$, and for each $x \in D$, we define $b_i^{(j)}(x)$ to be 1 if and only if $x_j \geq \beta_i^{(j)}$. Let x^* be the resulting binary vector

$$x^* = (b_1^{(1)}(x), \ldots, b_{k_1}^{(1)}(x), \ldots, b_1^{(n)}(x), \ldots, b_{k_n}^{(n)}(x)) \in \{0, 1\}^d,$$

where $d = \sum_{j=1}^n k_j$. The set $D^* = \{x_i^* : 1 \leq i \leq m\}$ is then a binarized version of the set D of observations, and standard LAD techniques can be applied.

There are a number of ways, however, in which the binarization just described could be non-optimal and, usually, some cutpoints can be eliminated; see the approaches taken in [8,9]. In [5], variants on these approaches are discussed, the aim being to find 'robust' cutpoints; that is, cutpoints which define hyperplanes geometrically at least at a certain distance from the data points. Suppose, then, that a (reduced) set $C^{(j)}$ of K_j cutpoints (a subset of the corresponding $\beta_i^{(j)}$) is selected for coordinate j, and suppose the members of $C^{(j)}$ are

$$a_1^{(j)} < a_2^{(j)} < \cdots < a_{K_j}^{(j)}.$$

Let $d = \sum_{j=1}^n K_j$. An element $x \in [0, 1]^n$ will be 'binarized' as $x^* \in \{0, 1\}^d$ where x^* is

$$(b_1^{(1)}(x), \ldots, b_{K_1}^{(1)}(x), \ldots, b_1^{(n)}(x), \ldots, b_{K_n}^{(n)}(x)),$$

where $b_i^{(j)}(x) = 1$ if and only if $x_j \geq a_i^{(j)}$. Let the Boolean literal $u_i^{(j)}$ be given by $\mathbb{I}[x_j \geq a_i^{(j)}]$, where $\mathbb{I}[P]$ has value 1 if P is true and value 0 otherwise. Then a positive pattern is a conjunction of some of the Boolean variables $u_i^{(j)}$. By definition of $u_i^{(j)}$, $u_i^{(j)} = 1$ implies $u_{i'}^{(j)} = 1$ for $i > i'$, and any j. So a typical positive pattern can be written in terms of these Boolean variables as

$$\bigwedge_{j=1}^n u_{r_j}^{(j)} \bar{u}_{s_j}^{(j)},$$

where $s_j > r_j$. (Here, $\wedge$ denotes the Boolean conjunction, the 'and' operator.) Geometrically, this positive pattern is the indicator function of the 'box'

$$[a_{r_1}^{(1)}, a_{s_1}^{(1)}) \times [a_{r_2}^{(2)}, a_{s_2}^{(2)}) \times \cdots \times [a_{r_n}^{(n)}, a_{s_n}^{(n)}).$$