Astronomy and Computing 11 (2015) 64–72

Contents lists available at ScienceDirect

Astronomy and Computing

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

Full length article

Effect of training characteristics on object classification: An application using Boosted Decision Trees

I. Sevilla-Noarbe *, P. Etayo-Sotos

Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Av. Complutense 40, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 8 February 2015
Accepted 24 March 2015
Available online 21 April 2015

Keywords:
Techniques: photometric
Catalogs
Supervised learning by classification

ABSTRACT

We present an application of a particular machine-learning method (Boosted Decision Trees, BDTs using AdaBoost) to separate stars and galaxies in photometric images using their catalog characteristics. BDTs are a well established machine learning technique used for classification purposes. They have been widely used specially in the field of particle and astroparticle physics, and we use them here in an optical astronomy application. This algorithm is able to improve from simple thresholding cuts on standard separation variables that may be affected by local effects such as blending, badly calculated background levels or which do not include information in other bands. The improvements are shown using the Sloan Digital Sky Survey Data Release 9, with respect to the *type* photometric classifier. We obtain an improvement in the impurity of the galaxy sample of a factor 2–4 for this particular dataset, adjusting for the same efficiency of the selection. Another main goal of this study is to verify the effects that different input vectors and training sets have on the classification performance, the results being of wider use to other machine learning techniques.

1. Introduction

Object classification in photometric images is an important first step in any analysis based on catalogs from such sources, as it constitutes a fundamental tool to build the set to be used for model comparison or parameter estimation. In particular, for cosmological analyses, a significant fraction of stars contaminating the galaxy sample can change the amplitude of the galaxy power spectrum. If this misclassified population (represented by the impurity fraction I) is spatially unclustered, the amplitude of the power spectrum is changed by a factor $(1-I)^2$ and errors must be increased to account for it, or a correction has to be applied. A well determined clustering amplitude is key for measuring effects such as the galaxy bias from a specific galaxy population (Coupon et al., 2012), understanding large-scale cosmological effects versus a systematic stellar contamination component (see for example Thomas et al. (2011) and Ross et al. (2011)) or distinguishing cosmological models with primordial non-Gaussianities (Giannantonio et al., 2014).

Star–galaxy classification has been addressed using many different morphology based cuts since the existence of the first photographic plate surveys (MacGillivray et al. (1976), Sebok (1979), Heydon-Dumbleton et al. (1989), Maddox et al. (1990)) and with more sophisticated techniques with the advent of digital imaging, machine learning methods (Odewahn et al. (1992), Weir et al. (1995), Miller and Coe (1996), Bertin and Arnouts (1996)) and exponentially increasing computational power. Most of the implementations have addressed the problem from the morphological point of view too. Multi-band imaging surveys, such as the Sloan Digital Sky Survey (SDSS) or the Canada–France–Hawaii Telescope Legacy Survey (CFHTLS), have opened up the possibility of adding color information as input variables (henceforth termed *features*) for the classifier. This is explored in Ball et al. (2006) for SDSS Data Release 3 (DR3) and in Hildebrandt et al. (2012) for CFHTLenS and to select a pure star sample for Milky Way studies using SDSS DR7 in Fadely et al. (2012). Recently, in Małek et al. (2013), the authors performed a study in classification using Support Vector Machines with VIPERS data as training set, highlighting the importance of adding infrared data to enhance the classification.

In this paper, we investigate the usage of AdaBoost Boosted Decision Trees as star–galaxy classifiers, and test their performance in galaxy selection against the standard SDSS morphological selection in SDSS Data Release 9. We use this popular flavor of decision trees to address this issue for the first time on optical catalog information, where we have broadened the scope of input features, to use color and morphological information simultaneously. Beyond optimizing the tree parameters, the goal is to study the influence of color and morphological information separately, and the influence

* Corresponding author. Tel.: +34 91 496 25 77.
E-mail address: ignacio.sevilla@ciemat.es (I. Sevilla-Noarbe).

http://dx.doi.org/10.1016/j.ascom.2015.03.010
2213-1337/

of different sizes and depth of training sets, which are required by any empirical-based classifier.

Decision Trees (DTs) have been explored thoroughly in the past for this purpose, as described in Suchkov et al. (2005) who were the first to apply a DT to separate objects from the SDSS-DR2. Later, in Ball et al. (2006) an axis-parallel decision tree was applied, using almost 500k objects from SDSS-DR3 with an extensive exploration of parameters using as input features the colors of the objects, for the range up to $r = 20$. In Vasconcellos et al. (2011) the authors broadened the scope of this work by comparing 13 different Decision Tree algorithms up to $r = 21$ and using SDSS DR7 as testbed, but limiting to morphological parameters.

Boosted Decision Trees, introduced in Freund and Schapire (1997), have been used very successfully in high energy physics (Roe et al., 2005) including particle classification in MiniBooNE (Yang et al., 2005), CMS data for identification of the Higgs particle (CMS-Collaboration, 2012), AMS (Aguilar et al., 2013) and Fermi (Ackermann et al., 2012). In optical astronomy, an application has been developed to extract photometric redshifts from imaging surveys (Gerdes et al., 2010), outperforming implementations based on neural networks. They have also been used for artifact identification in supernovae searches (Bailey et al., 2007).

The paper is structured as follows: in Section 2, BDTs and the specific implementation we have used are detailed. In Section 3, we describe the dataset employed, data features chosen, training, evaluation and test sets. In Section 4 we detail the approach for the optimization of the tree parameters for our specific problem, i.e., obtaining high purity galaxy samples. We show our results for the best parameter set in Section 5 and we compare the performances for different training sets and feature selection. Then we end with some conclusions and possible lines of future work.

2. Boosted Decision Trees

A Decision Tree is a structured classifier which makes step-by-step choices based on a single *feature* describing the data. A series of sequential cuts is devised to separate the data into one of two categories: signal and background. The value of the cuts, the feature used and the order in which they are applied, are established using a training set. The process continues through these *nodes* until a final node *(leaf)* is reached.

The training process starts at a root node with an arbitrary choice of feature and value of the cut. The separation into signal and background is done according to this criterion and a separation power θ is evaluated. In this case, we use the *Gini index* to determine the performance of this particular choice:

$$G = p \cdot (1 - p) \tag{1}$$

where p is the purity of the selected sample (whether it be signal or background). Using the index P for the parent node and the indices s and b for the signal and background daughter nodes, we determine the best choice of feature *and* value of the cut which maximizes:

$$\theta = abs(G_P - (G_s + G_b)). \tag{2}$$

Every input feature is scanned, using a predetermined number of cuts for each (parameter *ncuts*), to look for the best pair at each node. Thus the configuration of the tree continues until a minimum number of data points in a particular node is reached (parameter *nevmin*) or if the number of consecutive nodes reaches a predetermined maximum (parameter *maxdepth*).

Decision Trees are known to be a powerful but unstable learning method, i.e., a small change in the training sample can translate into a large change in the tree and the result of the classification. In addition, a theoretically 'perfect' classification can be achieved if the tree is allowed to develop fully so that each leaf only contains signal or background data points, therefore separating fully the dataset. Of course, this is only an accurate description of the *training* set, which most probably will not be descriptive of new data, as it has incorporated all the noise inherent to that specific data (overfitting).

Boosting is a way of enhancing the classification performance and increasing the stability with respect to statistical fluctuations in the training sample, as well as to avoid the overfitting problem. If a training data point is misclassified in a leaf, a weight is assigned to that data point, according to:

$$w = \frac{1 - \epsilon}{\epsilon} \tag{3}$$

where ϵ is the misclassification rate of the tree. The weight w is assigned to all such data points and a second tree is generated anew, with the original dataset using these weights instead (well classified values keep a weight value $w = 1$). The process is iterated tens or hundreds of times (parameter *ntrees*), with all the resulting trees combined into a 'forest' to provide significantly enhanced classification power. This is the so-called *AdaBoost* technique (Freund and Schapire, 1997). With this forest of trees at hand, the classification of a single data point is performed based on the majority vote of the classifications done by each tree.

We have used the Toolkit for Multivariate Analysis framework (Speckmayer et al., 2007), provided with the ROOT analysis package (Brun and Rademakers, 1997), widely used in high energy physics with great success. This framework has been used in other astrophysical applications such as the ArborZ photometric redshift code described in Gerdes et al. (2010). It is specially designed for processing the parallel evaluation and application of different multivariate classification techniques, among which are AdaBoost Boosted Decision Trees.

A first test was performed on a training sample based on SDSS DR7 data (Etayo-Sotos and Sevilla-Noarbe, 2013) using several of the methods described in the package, with some standard, default values. The results are shown in Fig. 1 via the Receiver Operator Characteristic (ROC) curve which measures the true positive rate versus the false positive rate of the classifier for different thresholds. The BDTD method (which is a Boosted Decision Tree with a prior step of input feature decorrelation) turns out to have the best performance for this problem and training set. The decorrelation step takes care of linear correlations between the input features (vector $\mathbf{x}$) by computing the square root S of their covariance matrix and constructing a new input feature vector $\mathbf{x'} = S^{-1}\mathbf{x}$. The other standard methods which were compared are:

- k-Nearest Neighbors (*kNN*): a method which searches for the k closest training events in feature space.
- Fisher Discriminant (*BoostedFisher*): a linear discriminant analysis in which an axis in feature hyperspace is determined so that signal and background are as separated as possible.
- Neural Network (*MLP*): a multi-layer standard perceptron implementation of this classic technique, in which a non-linear mapping of the input feature vector is done onto a one-dimensional space as well. This is done through a complex mesh of cells which react to the input variables and modify their final classification accordingly.

This result, coupled with the success of this specific implementation in recent particle physics literature, pushed us to choose this machine learning algorithm for our tests.

Random Forests are a particularly successful technique too in the field of classification and regression in astronomy (see, e.g., Carrasco Kind and Burner (2013)). They have better generalization properties as they can account for some scatter from the training set to the application set. On the other hand, AdaBoost BDTs can outperform slightly if the training set is representative enough. In