Information Sciences 366 (2016) 1–11

Contents lists available at ScienceDirect

Information Sciences

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

Autonomously evolving classifier TEDAClass

Dmitry Kangin[a,*], Plamen Angelov[a], José Antonio Iglesias[b]

[a] *Data Science Group, Computing & Communications, Lancaster University, UK*
[b] *Carlos III University of Madrid, Spain*

ARTICLE INFO

Article history:
Received 28 November 2015
Revised 16 March 2016
Accepted 12 May 2016
Available online 24 May 2016

Keywords:
Classifiers
Evolving systems
TEDA
Fuzzy systems

ABSTRACT

In this paper we introduce a classifier named TEDAClass (Typicality and Eccentricity based Data Analytics Classifier) which is based on the recently proposed AnYa type fuzzy rule based system. Specifically, the rules of the proposed classifier are defined according to the recently proposed TEDA framework. This novel and efficient systematic methodology for data analysis is a promising addition to the traditional probability as well as to the fuzzy logic. It is centred at non-parametric density estimation derived from the data sample. In addition, the proposed framework is computationally cheap and provides fast and exact per-point processing of the data set/stream. The algorithm is demonstrated to be suitable for different classification tasks. Throughout the paper we give evidence of its applicability to a wide range of practical problems. Furthermore, the algorithm can be easily adapted to different classical data analytics problems, such as clustering, regression, prediction, and outlier detection. Finally, it is very important to remark that the proposed algorithm can work “from scratch” and evolve its structure during the learning process.

1. Introduction

Nowadays, the classical problems of machine learning are widely studied. However, many architecturally different algorithms still suffer from a number of limitations. If we take into account how they work, we can consider three different kinds of algorithms: offline, incremental and online learning. The offline algorithms are trained once, and then do not admit any modifications to the classifier parameters or structure. Hence, the performance of these algorithms is only restricted to the patterns observed during the training stage. Furthermore, the incremental algorithms admit modifications of (some of) the parameters, but they generally demand previous data samples (all or some part) to be stored in the memory. They do not allow modification of the classifier structure, and very often they are not sufficient for practical needs. Finally, the algorithms, which implement online learning, do not need to memorise all previous training samples. This kind of algorithms are usually computationally efficient, recursive procedure, and applicable for online real-time applications.

However, none of these three types of algorithms involve any kind of adaptation of the structure to take into account the newly emerging or dynamically evolving data patterns which were not present in the initial training data set/stream. For this reason, a recently emerged branch of machine learning aiming to address such problems: Evolving Intelligent Systems (EIS) [4]. The algorithm developed in this work is based on the paradigm of such systems that is a distinguishable feature of the proposed algorithm amongst the well-studied techniques. One of the most renowned machine learning challenges is the classification problem. Some of the versions of the problem statement are domain-specific, while others are defined

* Corresponding author. Tel.: +44 7437499029.
E-mail addresses: d.kangin@lancaster.ac.uk (D. Kangin), p.angelov@lancaster.ac.uk (P. Angelov), jiglesia@inf.uc3m.es (J.A. Iglesias).

http://dx.doi.org/10.1016/j.ins.2016.05.012

generally, and then adapted for a particular problem. Hereafter we provide a classifier which does not have any user- or problem-specific parameter.

One of the most renowned machine learning challenges is the classification problem. Some of the versions of the problem statement are domain-specific, while others are defined generally, and then adapted for a particular problem. Hereafter we provide a classifier which does not have any user- or problem-specific parameter.

Define some object set Ω, and a finite space of object classes, referred as C. Then define a subset, $\Omega_L \subset \Omega$ called learning set, for each of the following functions defined as: F_L: $\Omega_L \to C$. The problem is to build a function F: $\Omega \to C$, approximating the function F_L on the set Ω_L. The assumption is that F will be a good mapping for further objects, Ω_V where the index V denotes validation. $\Omega_V \cap \Omega_L = \emptyset,\ \Omega_V,\ \Omega_L \subset \Omega$.

In this paper, we address the problem of a general purpose classifier, which needs to meet the following requirements:

- *Ability of incremental learning:* the classifier should be one-pass. Thus, each sample is proceeded once and there is no need to re-learn any previous samples once the new training data appear.
- *Online design:* the classifier should proceed the samples at the time of their arrival, the algorithm should not rely on memorised context of all data samples. There is no need to store all the given samples, but only some descriptor data of them, ideally with fixed size upper bound.
- *Evolving structure:* the classifier should be able to start learning “from scratch” and its parameters and structure should be self-adapting on-line. In addition, it is non-parametric.
- *Speed and memory efficiency:* The algorithm should be computationally effective so it can be applied to online real-time (or near real-time) applications.

In order to meet all these requirements, we propose a novel and efficient classifier named TEDAClass in which the novel TEDA framework [2] is used within a system of evolving fuzzy AnYa-type rules [6]. This combination enables to build easily interpretable model for classification. The most important contribution of this paper is the novel classifier TEDAClass. In addition, we also present the derivations of incremental calculation formulae, which are necessary for updating the incremental fuzzy rules of the classifier. The novel classifier has been successfully tested on several benchmark problems.

The reminder of the paper is organised as follows. Section 2 describes the related works. Section 3 gives a brief description of the previously proposed TEDA approach. Section 4 contains necessary novel derivations of the incremental formulae within TEDA framework, necessary for further algorithm description, and Section 5 formulates the classifier itself. The experimental data is presented in Section 6. Finally, Section 7 contains future work and concluding remarks.

2. Related work

There are many different well-studied algorithms of classification. The most relevant algorithms include Support Vector Machines (SVMs) [44], Neural Networks (Single- and Multi-layer neural networks [12,39], Radial Basis Function networks [14]), Bayesian networks [37], fuzzy rule-based models [7] or decision trees [38]. But from a wide variety of classification methods, we particularly pay attention to fuzzy logic approaches, which offer well-developed apparatus for evolving system development [8]. This kind of methods gives an essential contribution to this article. Here we review the state-of-the art algorithms in the prism of evolving systems development.

SVMs [44] provide rigorous and sparse solution for various classification problems, and are especially well renowned for the results obtained via the so-called kernel trick. They can also be implemented incrementally. If we discard some of the support vectors according to some rule, it is possible to obtain Evolving Systems based on SVM.

The evolving systems use a complex and varying combination of simple models which are adjustable “from scratch”, and capable of changing its own structure during the system learning [9,26,41]. Many evolving fuzzy systems (EFSs) [31] and evolving connectionist systems (ECSs) [27] have been developed during the last two decades. Amongst the branch of evolving fuzzy classifiers, which is a part of evolving fuzzy systems, the following methods can be mentioned: *eClass* [7], *AutoClass* [10], DEC [11], FLEXFISClass [5,32], self-evolving classifiers [19]. Also, several important works [30,33,36] has appeared in a few recent years.

Evolving Spiking Neural Networks [28,42] is one of the evolving approaches based on Artificial Neural Networks. However, reasonable efforts should be usually undertaken to learn it properly without falling into a local extremum (for example, in case of [28], the genetic algorithms are utilised for the optimisation).

Nowadays, fuzzy rule-based evolving systems are predominantly of the type of Takagi-Sugeno [43] or Mamdani [35]:

Mamdani:

$$\text{IF } (\text{ant}_i(\vec{x}))\text{ THEN } (y_i \text{ IS } Y_i), \tag{1}$$

Takagi-Sugeno:

$$\text{IF } (\text{ant}_i(\vec{x}))\text{ THEN } (y_i = \vec{x}^T\Theta_i), \tag{2}$$

where y_i is an outcome of the ith rule, $i \in [1 \ldots N]$, Θ_i is a design matrix for linear regression, $\vec{x}$ is an input data vector, $\text{ant}_i(\vec{x})$ is the antecedent of the fuzzy rule.

In both cases, the antecedent is expressed as

$$\text{ant}_i(\vec{x}) : x^1 \text{ IS } L_i^1 \text{ AND } x^2 \text{ IS } L_i^2 \text{ AND } \cdots \text{ AND } x^n \text{ IS } L_i^n, \tag{3}$$