Neurocomputing 000 (2018) 1–12

Contents lists available at ScienceDirect

Neurocomputing

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

A taxonomic look at instance-based stream classifiers

Iain A.D. Gunn[a,*], Álvar Arnaiz-González[b], Ludmila I. Kuncheva[a]

[a] School of Computer Science, Bangor University, Dean Street, Bangor LL57 1UT, UK
[b] Escuela Politécnica Superior, Universidad de Burgos, Burgos 09006, SPAIN

ARTICLE INFO

Article history:
Received 23 February 2017
Revised 29 July 2017
Accepted 24 January 2018
Available online xxx

Communicated by Dr Xiaofeng Zhu

Keywords:
Machine learning
Stream classification
Instance selection
Prototype generation
Concept drift

ABSTRACT

Large numbers of data streams are today generated in many fields. A key challenge when learning from such streams is the problem of concept drift. Many methods, including many prototype methods, have been proposed in recent years to address this problem. This paper presents a refined taxonomy of instance selection and generation methods for the classification of data streams subject to concept drift. The taxonomy allows discrimination among a large number of methods which pre-existing taxonomies for offline instance selection methods did not distinguish. This makes possible a valuable new perspective on experimental results, and provides a framework for discussion of the concepts behind different algorithm-design approaches. We review a selection of modern algorithms for the purpose of illustrating the distinctions made by the taxonomy. We present the results of a numerical experiment which examined the performance of a number of representative methods on both synthetic and real-world data sets with and without concept drift, and discuss the implications for the directions of future research in light of the taxonomy. On the basis of the experimental results, we are able to give recommendations for the experimental evaluation of algorithms which may be proposed in the future.

1. Introduction

Storing large data sets can be problematic, especially in stream learning, where data is continuously arriving. This issue is more relevant than ever in an era of "big data", where important problems involve data streams which cannot be stored in full [1,2]. Many techniques have been suggested for forming reduced reference sets for instance-based classifiers, in particular the nearest-neighbour classifier [3,4]. However, as we argued in a previous contribution [5], the taxonomy developed for describing offline algorithms for data editing is inadequate to describe algorithms for streaming data.

In summary, the offline taxonomy fails because many approaches developed for offline editing are inherently unsuitable for streaming data. For example, in the offline case, there are methods which only add instances to the reference set, never removing them; methods which only remove instances from the reference set (starting with all the training data), and never re-add them; and methods which both add and remove instances as they run. These are distinguished taxonomically as "incremental", "decremental", and "mixed" methods. Clearly, editing methods which can only add or only remove examples are unsuitable for dealing with unbounded data streams. Only "mixed" methods can be used in the streaming case, so the offline taxonomic distinction is useless for the streaming case. In the streaming case, all methods now being "mixed", the key taxonomic question of interest is the choice of processes by which instances are added to and removed from the reference set in response to the stream of arriving data, as it is here that the nature of the streaming problem forces a great difference in approach from the offline case.

In addition to the need for editing, a second key issue with stream learning is that data streams may typically be "non-stationary", that is, subject to "concept drift". We also found previously [5] that the established taxonomy developed to describe algorithms designed to deal with concept drift [6] cannot sensibly be used to classify instance-based algorithms. The existing taxonomy in this case used separate concepts of "Data Management" and "Memory" which could not be applied to lazy learners, for which memory simply *is* data retention.

This paper expands our previous study [5] on instance selection methods for drifting data streams. In addition to augmenting and refining the taxonomy of such methods, we carry out a numerical experiment to compare the performance of some modern algorithms, in light of the taxonomy. The present work also gives

* Corresponding author at: Department of Computer Science, Middlesex University, The Burroughs, London NW4 4BT, UK.
E-mail addresses: i.gunn@mdx.ac.uk (I.A.D. Gunn), alvarag@ubu.es (Á. Arnaiz-González), l.i.kuncheva@bangor.ac.uk (L.I. Kuncheva).

https://doi.org/10.1016/j.neucom.2018.01.062
0925-2312/

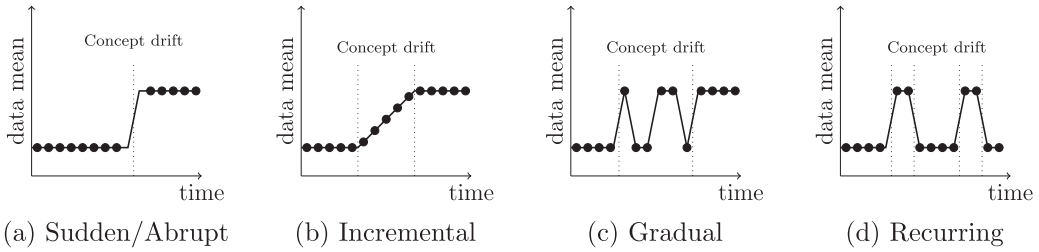

Fig. 1. Main concept drift types illustrated schematically as if for one-dimensional data. Adapted from Gama et al. [6].

greater consideration to prototype generation methods, typified by the learning vector quantisation (LVQ) family [7].

Note that our study considers the various algorithmic approaches for forming a reference set from streaming data, not the variety of instance-based classifiers which might use such a reference set. We do not compare alternative classifiers: we simply use the nearest-neighbour (1NN) classifier. (Differences between classification rules may be taxonomically considered as for the offline case.)

The rest of the paper is organised as follows. The formal problem of classification of a stream subject to concept drift, and related terminology, are introduced in Section 2. Data-editing methods are introduced in Section 3. Our refined taxonomy of instance-based methods for the concept drift problem is presented in Section 4. The algorithms included in the experiment, and some other representative algorithms, are discussed in Section 5. Our experimental set-up and results, with discussion, are presented in Section 6. The conclusion Section 7 contains recommendations for future experimental practice.

2. The concept-drift problem

The streaming version of the classification problem is typically posed thus:

- One data point $\mathbf{x} \in \mathbb{R}^d$ is received at time t.
- The class label of the point is not available at time t. The point is labelled by the classifier.
- The true label is then revealed before the next data point is classified.

The model can easily be altered to a batch-input form, in which a set of N points $X \subset \mathbb{R}^d$ is considered to arrive all at once at time t, and all N points must be labelled before the true labels are revealed and the next input batch arrives.

"Concept drift" is the generally accepted term for change in the probability distributions related to the problem, and the management of this problem is essential in streaming learning [8]. Occurrences of concept drift have been described in terms of the behaviour of the stream at the onset of the drift: see Fig. 1 for an illustration of this idea. The terminology is taken from Bose et al. [9] and Gama et al. [6]. Concept drift may be sudden, or the underlying distribution may pass continuously and relatively slowly through intermediate states ("incremental drift"). The original concept may then be gone forever, or it may recur, briefly (in "gradual drift"), or indefinitely, in which case it is called a true "recurring concept". In general, it is to be expected that some algorithms deal better than others with certain forms of drift. For example, algorithms which explicitly maintain a library of former concepts have been so engineered in order to perform better when the stream includes recurring concepts, but can only be disadvantaged by this apparatus when applied to a stream containing only sudden, irrevocable concept shifts. (This approach of storing former concepts for re-use is typified by the FLORA3 algorithm [10], one of the first algorithms to explicitly address recurring concepts. It is part of the FLORA family of algorithms [11,12], dating back to 1989.)

Whether such a collection of former concepts is maintained or not, an algorithm for handling concept drift will have both a learning mechanism of some sort and a *forgetting* mechanism of some sort, the latter being essential to ensure the classifier does not become stuck in some setting after seeing a large amount of data which exceeds its capacity for learning. Some methods use explicit *change detection* strategies, which allow the algorithm to make a suitable increase in learning and forgetting rates when a concept shift is detected. We refer the reader to the survey of Gama et al. [6] for a good recent review of concept-drift adaptation methods.

Concept drift is of interest to the extent that it affects adversely the future performance of the classifier and requires action: the occasional outlier or short abnormal event should simply be treated as noise and ignored.

3. Prototype selection and generation

One key distinction among data-editing methods must be introduced before the entire taxonomy is presented. This is the distinction between "prototype selection" and "prototype generation" families, which have been treated very similarly in taxonomies of their offline members [3,4], but which we have argued [5] need very different treatment in their online incarnations.

The process of editing training data for use with the nearest-neighbour classifier, or similar instance-based classifiers, consists of replacing the set of training data, S, with a *smaller* reference set of what are called "prototypes". The meaning of "prototype" depends on the approach (selection or generation) chosen for the data editing[2]. In prototype selection, the reduced set of prototypes, S' is a subset of S (along with the labels of the objects). In prototype generation, the prototypes are allowed to be different points in the same space (or to be extended as other structures such as hyper-rectangles or hyper-ellipses). Generated prototypes in the original space can be created by various procedures for relabelling, merging or re-positioning members of an initial subset of S, such as by finding cluster centres.

Prototype generation is potentially the richer of the two data-editing approaches. In prototype generation, the entire space is available for the positioning of prototypes, allowing the approximation of any classification boundary with a specified precision. If instead the prototypes are constrained to be chosen from the finite set of points constituting the training data, the set of possible boundaries is correspondingly reduced.

The learning vector quantisation (LVQ) family of methods are exemplars of the Prototype Generation approach which are particularly popular and successful in the online case. Many algorithms of the LVQ type have been developed, and a taxonomy of

[2] Synonyms of prototype generation in the literature are prototype *construction, extraction, reduction* and *replacement*.

Please cite this article as: I.A.D. Gunn et al., A Ttaxonomic Llook at Iinstance-based Sstream Cclassifiers, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.01.062