Artificial Intelligence 230 (2016) 108–133

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

A concept drift-tolerant case-base editing technique

Ning Lu [a], Jie Lu [a,*], Guangquan Zhang [a], Ramon Lopez de Mantaras [b]

[a] Faculty of Engineering and Information Technology, University of Technology Sydney, PO Box 123, Broadway, NSW 2007, Australia
[b] Artificial Intelligence Research Institute (IIIA-CSIC), Campus UAB, Bellaterra, Spain

ARTICLE INFO

Article history:
Received 18 June 2013
Received in revised form 18 August 2015
Accepted 21 September 2015
Available online 30 September 2015

Keywords:
Case-based reasoning
Case-base editing
Concept drift
Competence model

ABSTRACT

The evolving nature and accumulating volume of real-world data inevitably give rise to the so-called "concept drift" issue, causing many deployed Case-Based Reasoning (CBR) systems to require additional maintenance procedures. In Case-base Maintenance (CBM), case-base editing strategies to revise the case-base have proven to be effective instance selection approaches for handling concept drift. Motivated by current issues related to CBR techniques in handling concept drift, we present a two-stage case-base editing technique. In Stage 1, we propose a Noise-Enhanced Fast Context Switch (NEFCS) algorithm, which targets the removal of noise in a dynamic environment, and in Stage 2, we develop an innovative Stepwise Redundancy Removal (SRR) algorithm, which reduces the size of the case-base by eliminating redundancies while preserving the case-base coverage. Experimental evaluations on several public real-world datasets show that our case-base editing technique significantly improves accuracy compared to other case-base editing approaches on concept drift tasks, while preserving its effectiveness on static tasks.

1. Introduction

Case-based Reasoning (CBR) is a problem-solving strategy that uses prior experience to understand and solve new problems. Unlike other model-based learning methods, which store past experience as generalized rules and objects, CBR systems store past experience as individual problem solving episodes [1] and delay generalization until problem-solving time. Despite several reported advantages of CBR in the literature [2] (e.g., CBR performs well with disjointed concepts, or CBR, as a lazy learner, is easy to update), large-scale and long-term CBR systems suffer from effectiveness and competence issues [3], which has led to increased awareness of the importance of Case-base Maintenance (CBM).

Case-base Maintenance refers to the process of revising the contents of a CBR system, thereby improving the system's efficiency and competence [4]. While CBM extends beyond the case-base to include all knowledge containers, in this paper, CBM is restricted to maintenance of the case-base only. In CBM, methods of revising the case-base (also called case-base editing) involve reducing the size of a case-base or training set while endeavoring to preserve or even improve generalization accuracy [5]. Two distinct areas in case-base editing have been identified and investigated: 1) competence enhancement, which aims to remove noisy cases, thereby increasing classifier accuracy; 2) competence preservation, which corresponds to redundancy reduction, i.e., removing superfluous cases that do not contribute to classification competence. Although several hybrid methods exist that search for a small subset of the training set, and simultaneously achieve the elimination of noisy

[*] Corresponding author.
E-mail addresses: Ning.Lu@uts.edu.au (N. Lu), Jie.Lu@uts.edu.au (J. Lu), Guangquan.Zhang@uts.edu.au (G. Zhang), mantaras@iiia.csic.es (R. Lopez de Mantaras).

http://dx.doi.org/10.1016/j.artint.2015.09.009

and superfluous cases, competence enhancement and competence preservation methods can be combined to achieve the same objectives as hybrid methods [6].

The issue of concept drift refers to the change of distribution underlying the data [7,8]. More formally, the issue of concept drift can be framed as follows: If we denote the feature vector as x and the class label as y, then a data stream will be an infinite sequence of (x, y). If the concept drifts, it means that the distribution of $p(x, y)$ between the current data chunk and the yet-to-come data is changing. If we decompose $p(x, y)$ into the following two parts as $p(x, y) = p(x) \times p(y|x)$, we can say that there are two sources of concept drift: one is $p(x)$, which evolves with time t, and can also be written as $p(x|t)$, and the other is $p(y|x)$, the conditional probability of feature y given x [9]. Learning under concept drift is considered to be one of the most challenging problems in machine learning research [10]; it is consequently the subject of increased attention [11].

At present, approaches for handling concept drift can be generally divided into three categories [12]: 1) instance selection (window-based), where the key idea is to select the instances that are most relevant to the current concept. Many case-base editing strategies in CBR – including this research – that delete noisy, irrelevant or redundant cases are also a form of instance selection. Since this research focuses on case-based techniques in dynamic environments where concept drift may occur, we will give a more detailed review of CBM methods in Section 2; 2) instance weighting, where each instance is assigned a weight to represent the decreasing relevance of existing training examples. These instances can be weighted according to their "age", or their competence with regard to the current concept [13]. Research into this category [14,15] mainly focuses on exploring a proper weighting schema; 3) ensemble learning (learning with multiple models) is reported to be the most popular and successful approach for dealing with concept drift [16]. It utilizes multiple models by voting [17, 18] or selecting the most relevant model to construct an effective predictive model [19]. Generally, there are two ensemble frameworks: 1) horizontal ensemble [20,21], which builds on a set of buffered data chunks; 2) vertical ensemble [9,22], which builds on the most recent data chunk only. More recently, an aggregate ensemble framework, which could be seen as a hybrid approach of the two, has been proposed [23]. There is also research that maintains ensembles with different diversity levels [24,25]. The key idea behind this kind of research is that "before the drift, ensembles with less diversity obtain lower test errors. On the other hand, it is a good strategy to maintain highly diverse ensembles to obtain lower test errors shortly after the drift independent of the type of drift" [26].

Although concept drift has been an active research area in machine learning, little effort has been made in respect of CBM-related research. In real world applications, the case-base accumulates with usage over time, and the case distribution as well as the decision concepts underlying the cases may be subject to continuous change. This phenomenon poses additional challenges for existing case-base editing methods that implicitly assume that existing cases are drawn from a fixed yet unknown distribution, i.e., stationary assumption [22]. When concept drift occurs, past cases may not reflect current concepts; as a consequence, current case-base editing methods may preserve harmful cases. In addition, when class boundaries shift or novel concepts emerge, new cases representing novel concepts are more likely to be treated as noise and removed by competence enhancement algorithms, because they conflict with past concepts. This may seriously delay or even prohibit a case-based learner from learning new concepts. Finally, redundancy reduction is particularly challenging in domains with evolving decision boundaries. Most current competence preservation methods preserve only cases that lie on the decision boundaries. We argue that this is too aggressive and is inappropriate, particularly in the concept drift environment, for the following reasons: First, it will destroy the original case distribution, thus affecting the result of any change detection method which directly compares the case distribution. Second, it makes a learner too sensitive to noise, i.e., incorrectly retaining noisy cases as novel concepts, at the center of class definitions, will dramatically affect classification boundaries. Last, because "boundary cases distinguish one concept from another, while typical cases characterize the concept they belong to [27]", preserving a certain number of typical cases may also help to improve a CBR system, e.g., by improving case explanation. Motivated by the above-mentioned issues, while recognizing that it is not possible to know in advance whether there will be concept drift, this research proposes a novel case-base editing method that addresses both competence enhancement and competence preservation, and works well in both static and changing environments.

In this paper, we present a new case-base editing technique which takes both competence enhancement and competence preservation into consideration. For competence enhancement, we develop a Noise-Enhanced Fast Context Switch (NEFCS) algorithm to prevent noise from being included during case retention and to speed the context switch process in the face of concept drift. By taking advantage of our previous research on concept drift detection [28], our NEFCS algorithm minimizes the risk of discarding novel concepts. For competence preservation, we invent a Stepwise Redundancy Removal (SRR) algorithm that uniformly removes superfluous cases without loss of case-base competence. Experimental evaluations based on public real-world datasets show that our case-base editing technique demonstrates significant improvements in time-varying tasks and exhibits good performance on static tasks compared with the most common case-base editing methods.

This paper is organized as follows. Section 2 reviews the established literature on case-base editing techniques, as well as the research in CBR for tackling concept drift. Section 3 presents our proposed two-stage case-base editing technique. Section 4 evaluates each proposed algorithm and the overall technique for handling concept drift, using real datasets of both static and concept drift tasks. Section 5 summarizes our conclusions.