Applied Soft Computing 35 (2015) 558–582

Contents lists available at ScienceDirect

Applied Soft Computing

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

Integrating new classes on the fly in evolving fuzzy classifier designs and their application in visual inspection

Edwin Lughofer [a,*], Eva Weigl [b], Wolfgang Heidl [b], Christian Eitzinger [b], Thomas Radauer [c]

[a] Department of Knowledge-Based Mathematical Systems, Johannes Kepler University, Linz, Austria
[b] Machine Vision Group, Profactor GmbH, Steyr-Gleink, Austria
[c] Sony DADC, Austria

ARTICLE INFO

Article history:
Received 19 September 2014
Received in revised form 8 May 2015
Accepted 1 June 2015
Available online 8 July 2015

Keywords:
Evolving (fuzzy) classifiers
Integration of new classes on the fly
On-line visual inspection
All-pairs decomposition
Class imbalance
Expected change in classifier accuracy

ABSTRACT

In this paper, we address the problem of integrating new classes on the fly into on-line classification systems. The main focus is on visual inspection tasks, although the concepts proposed in this paper can easily be applied to any other on-line classification systems. We use evolving fuzzy classifiers (EFCs), which can adapt their structure and update their parameters in incrementally due to embedded on-line adaptable classifier learning engines. We consider two different model architectures — classical single model and an all-pairs approach that uses class information to decompose the classification problem into several smaller sub-problems. The latter technique is essential for establishing new classes quickly and efficiently in the classifier, and for reducing class imbalance. Methodological novelties are (i) making appropriate structural changes in the EFC whenever a new class appears while operating in a single-pass incremental manner and (ii) estimating the expected change in classifier accuracy on the older classes. The estimation is based on an analysis of the impact of new classes on the established decision boundaries. This is important for operators, who are already familiar with an established classifier, the accuracy of which is known. The new concepts are evaluated in a real-world visual inspection scenario, where the main task is to classify event types which may occur dynamically on micro-fluidic chips and may reduce their quality. The results show stable performance of established classifiers and efficient (low number of samples requested) as well as fast integration (steeply rising accuracy curves) of new event types (classes).

1. Introduction

1.1. Motivation

In modern industrial systems, on-line classification plays an essential role for various purposes such as classifying event types in business information systems and web mining applications [1], recognizing operator/user behavior [2], and predicting the current or future states of parts of the system (predictive maintenance) [3]. A particular variant of on-line classification is performed in visual inspection [4], where images are recorded on-line at the system to assure the quality of production items.[1] Thus, classification takes place at the image level, i.e., based on some extracted image characteristics (also termed as features), the task is to classify the item visualized in the image as “OK” or “not OK”. Typically, visual inspection is conducted at the end of a production chain line, before the items are packaged and sold to the customer. With a highly accurate inspection system, many customer complaints can therefore be avoided.

In some systems, the classification becomes more complex whenever larger variety of event types or system states may appear, indicating different fault candidates or pseudo-errors which must be recognized as such (i.e., not restricted to binary “OK” and “not OK” statements). For instance, it may be important to know the type of a fault in order to be able to provide an appropriate reaction and/or feedback to the system (e.g., varying the production parameters or, in severe cases, re-configuring and re-starting some system components). In this case, it is often impossible to integrate all event types a priori into the classifier, as collection and annotation of (a good bandwidth of) recorded samples — which is crucial in classifier training and important for high classifier accuracy — is time-consuming for the experts and thus costly for the company.

* Corresponding author. Tel.: +43 7236 3343 431.
E-mail addresses: edwin.lughofer@jku.at (E. Lughofer), eva.weigl@profactor.at (E. Weigl), wolfgang.heidl@profactor.at (W. Heidl), christian.eitzinger@profactor.at (C. Eitzinger), thomas.radauer@sonydadc.com (T. Radauer).

[1] http://en.wikipedia.org/wiki/Visual_inspection.

http://dx.doi.org/10.1016/j.asoc.2015.06.038
1568-4946/

In some systems, the dynamic nature of the process may result in new event types (classes) on demand and on the fly which the experts may not be completely aware of a priori (e.g., new types of user behavior). A concrete application example from the field of visual inspection is the observation of surfaces of production items on the basis of recorded images (as will be the handled in this paper, see also Section 6.1). There, new event types may arise as a *combination* of already known event types and integrated into the existing classifier (e.g. a blob and a scratch mixed up in one local region of an item (chip)). Such combinations may not necessarily trigger samples from the new event type (class) lying in extrapolation regions (w.r.t. already existing classes) as in case of outliers, but may have significant overlaps with older related events in the feature learning space (as their nature and appearance characterized by the features spanning the learning space is a mixture of old events).

Hence, automatic inclusion of new classes (event types) into the current classifier established on the classes seen so far is required, as otherwise the new class will never be returned by the classifier. This necessitates the evolution of new structural components and/or the updating classifier parameters and providing supplementary statistical information. Ideally, the inclusion should be such that the classifier built on the established classes remain stable, also in case of potential significant overlaps to the new class in case of combined or similar events (see above). This is an important aspect for operators, as they are already familiar with established classifiers that have known accuracy.

1.2. State of the art

In past studies, machine learning classifiers applied in industrial systems were often static [5], i.e., trained once based on pre-recorded samples or historic data-bases, but did not allow change of their parameters and structures over time when installed within the on-line process in order to react properly onto dynamic changes, see [6,7]. In other (application) cases, the classifiers are indeed adaptable using recursive parameter update schemes and integrated structural evolution and pruning techniques; thus, they are called incremental adaptive [8–10] and/or evolving classifiers [11]. However, they do not support inclusion of a new class during classifier adaptation on the fly. In principle, a re-training of state-of-the-art classifiers — once samples from a new class have been identified — is indeed possible (by also including these samples in the re-training stage), but usually this slows down the whole on-line interaction process: when the processing speed of the production chain is high — as is often the case in visual inspection scenarios [12] — re-training is often too slow to cope with the real-time operation [13]. Furthermore, selecting samples for appropriate re-training is often not trivial [14] (especially, in a way to avoid catastrophic forgetting [15]).

In visual inspection scenarios (the focus of our study) the situation is similar to the more general case. Current inspection systems rely on classifiers which are usually set up in a *batch off-line development phase*, either based on analytic approaches leading to hard-coded rules (old-school: low flexibility, often too specialized) [12,4] or using machine learning classifiers based on image data and extracted features collected in on-line processes (new school) [16,17]; however, these are kept static (non-adaptable) in the on-line/in-line classification process [18,19]. In [20,21], dynamic image classifiers were proposed, but without the possibility to define a new class during classifier adaptation; they only provide methods for robust and fast updates of the their parameters and structures based on known classes. In [22], image classifiers are indeed used within a wider scope employing ensemble methods and incremental fusion techniques, however the combiners (fusion operators) depend structurally on class certainty vectors with fix length (used as inputs), which cannot be changed on the fly. Classifiers are usually updated in active learning cycles [23] to minimize on-line interactions and annotation efforts [24]. Active learning selects those samples which are expected to improve the performance of the classifier most and is thus preferable to random (blind) selection. Annotating all the samples is inefficient and impossible in high-frequency throughput systems.

The machine learning literature describes several methods for incremental updating classifiers with respecting the possible inclusion of new classes, such as incremental discriminant analysis [25] (which expands within-class and between-class scatter matrices), on-line incremental learning vector quantization (LVQ) [26] (which creates new clusters or moves existing clusters away from new class samples), and on-line instance-based learners [27] (which update the case base properly with samples from the new class). These would be possible options in visual inspection applications. However, they completely lack any interpretability (which may become important when decisions must be explained to operators — a challenge planned to be investigated and developed further on in our framework), and furthermore they provide global decision boundaries over all classes. Such boundaries are usually counterproductive when samples from a new class are introduced, as these are often "overwhelmed" by older classes that have much more support since they are introduced at the beginning of the stream learning process: a fact that cannot be controlled, as data stream samples arrive in a specific order (in the same order as recorded from the process), and on which the learning system has no influence [28].

Another approach which is able to integrate new (object) classes on-the-fly is presented in [29]; this approach acts on the basis of an unsupervised learning stage extracting several density components. This also means that a new class/event is only recognizable as such, when it has some distinct appearance to the other classes (e.g. lying in extrapolation regions or in drifted distributions). This assumption is much weaker in our approach (see subsequent section), as we are operating in a supervised (active) learning framework, where the operator explicitly specifies a new class. The active sample selection is conceived in a way that new samples from new classes falling in-between other classes or even in class overlap regions (often caused by a combination of events forming a new class) are also selected.

1.3. Our approach

In this paper, we extend state-of-the-art evolving fuzzy classifiers (EFCs) (a comprehensive survey can be found in [30]) to incorporate new classes on the fly: we trigger appropriate structural changes in the EFC whenever a new class appears, while operating in a single-pass incremental manner, see Sections 3.4 and 4.3. An important aspect of this work is the study of the expected changes in the decision boundaries on already established classes (Section 3.5), which serves as basis for estimating the expected change in classifier accuracy (Section 5). This is important for operators, because they are already familiar with established classifiers with known accuracy. We provide such novel extensions to EFC for (classical) single-model architecture (first used in [31–33]), see Section 3, and for the recently introduced all-pairs architecture [34,35]), see Section 4, which divides the sample space into several smaller sub-problems by decomposing it into several binary classification problems (one for each class pair).

This decomposition increases the likelihood of balanced classes, and reduces the likelihood that newly introduced classes are "overwhelmed". Furthermore, it reduces the complexity of the decision boundaries and speeds up the learning process, as analyzed in [34]. In particular, the complexity reduction is a central advantage for our purposes, as decision boundaries are thus easier to learn