IJDAR
DOI 10.1007/s10032-017-0286-6

ORIGINAL PAPER

A multi-one-class dynamic classifier for adaptive digitization of document streams

Anh Khoi Ngo Ho[1] · **Véronique Eglin**[1] · **Nicolas Ragot**[2] · **Jean-Yves Ramel**[2]

Received: 22 April 2016 / Revised: 26 April 2017 / Accepted: 4 May 2017

Abstract In this paper, we present a new dynamic classifier design based on a set of one-class independent SVM for image data stream categorization. Dynamic or continuous learning and classification has been recently investigated to deal with different situations, like online learning of fixed concepts, learning in non-stationary environments (concept drift) or learning from imbalanced data. Most of solutions are not able to deal at the same time with many of these specificities. Particularly, adding new concepts, merging or splitting concepts are most of the time considered as less important and are consequently less studied, whereas they present a high interest for stream-based document image classification. To deal with that kind of data, we explore a learning and classification scheme based on one-class SVM classifiers that we call mOC-iSVM (multi-one-class incremental SVM). Even if one-class classifiers are suffering from a lack of discriminative power, they have, as a counterpart, a lot of interesting properties coming from their independent modeling. The experiments presented in the paper show the theoretical feasibility on different benchmarks considering addition of new classes. Experiments also demonstrate that the mOC-iSVM model can be efficiently used for tasks dedicated to documents classification (by image quality and image content) in a context of streams, handling many typical scenarii for concepts extension, drift, split and merge.

Keywords Stream-based document images classification · Online document content and quality classification · Incremental learning · Concept drift · One-class SVM

1 Introduction

As a result of the rapid expansion of Big Data, many companies, public organisms, and libraries are rethinking the traditional approach for digitization, access, and management of their huge pools of information corresponding to document streams that cannot anymore be manually processed or verified [1]. Beyond political considerations, the commitment taken by the firm Google and more recently by other leaders of the IT world (Microsoft, Yahoo, and Amazon) brings very diverse reactions associated with economic but also scientific challenges. There is an emergency to digitize the written inheritance, to protect original versions, and to satisfy the increasing needs for consultation, distribution but also enrichment of these resources. Massive digitization and long-term data preservation are new challenges that raise new scientific questions relative to efficient quality controls and automatic content recognition during digitization processes [2].

So as to make those large new datasets available for consultation and uses, new technologies for intelligent acquisition and content enrichment must be invented. Google's offensive and the European reaction show how this societal question is important for the double point of view of data

✉ Véronique Eglin
veronique.eglin@insa-lyon.fr

Anh Khoi Ngo Ho
ngoho@univ-tours.fr

Nicolas Ragot
nicolas.ragot@univ-tours.fr

Jean-Yves Ramel
jean-yves.ramel@univ-tours.fr

[1] CNRS INSA-Lyon LIRIS - UMR 5205 CNRS, Université de Lyon, 69621 Lyon, France

[2] Laboratoire Informatique - LI EA 6300, Université François Rabelais Tours, Tours, France

Published online: 18 May 2017

conservation and knowledge dissemination.[1] In this context, the DIGIDOC[2] project was focused on the stage of document image acquisition to improve and simplify their later use (archiving, text recognition, documents indexing, etc.). The costs of digitization and the fragility of documents of old collections make practically impossible a second digitization. Consequently, it is crucial to improve the quality of digitization according to either the visual appearance of images (color appearance, resolution, contrasts, etc.) or the later uses and processing (OCR transcription, visualization, preservation, etc.). By considering both the uses of digitized documents and the specificity of their content, the DIGIDOC project aimed at creating a new intelligent digitizing environment.

Our contribution, in this context, is to propose a dynamic classification system for content and quality recognition that could be embedded into a scanner dedicated to document stream digitization. Classification into data streams is a recent and challenging issue [40], and not much work has been done on this topic in document analysis community [3,26,39]. The proposed system here is based on an adaptation of the method proposed in [46]. A set of independent one-class SVMs are used to improve their recognition efficiency according to new incoming images. The learning scheme has been extended to take into account user interactions considering that goals can change over the time: For example, the user can simply validate a digitalization or decide to do it again. He can also decide to create a new category of content when necessary, to suppress an outdated class, and finally to split or merge two or more classes that share similar descriptions.

In the domain of dynamic classification, two approaches dominate the literature: batch-incremental methods that gather examples in batches to train models and instance-incremental methods that learn from each example when it arrives. Most of papers in the literature choose one of these approaches without any clear justification. In the field of dynamic document classification, our proposition is the only one that proposes both online and batch mode to train the models.

In Sect. 2, we present the state of the art of dynamic document classification and how our proposition is going one step forward. Sect. 3 details our contribution based on multi-one-class incremental SVM, the novelties regarding learning schemes and the validation on theoretical benchmarks in Sect. 4. Section 5 details the integration of such method for stream-based document content and quality recognition and its experimental validation. Finally, Sect. 6 is discussing time consumption and general discussions are presented in the concluding Sect. 7.

2 Dynamic classification of documents

In the following sections, we use the terms class to design a group of characteristics logically grouped together and the term concept to refer to a class of elements that share the same properties and have something in common. The concept also refers to a more abstract or semantic reality compared to the notion of class.

2.1 Document streams are no more static…

One of the most difficult problems with supervised classification raises the difficulty to model the concepts to be identified (as part of different classes), since they are highly dependent on some hidden contexts. Consequently, any information about the number of concepts, their structure and their description should ideally be totally well-known beforehand. Of course, this is far from being possible but most often, *a priori* or arbitrary knowledge is injected in the learning scheme to simplify the task and make it more accurate. This is the reason why many supervised classification tasks are done under the assumption that the concepts are fixed and known before learning (this is also the case for the recent "deep learners").

It is only very recently that a great attention has been paid on content classification in the context of data streams where the concepts can change over the time [9,29]. Indeed, for many real processes, data classification is not so simple, even in a supervised mode. Even if the concepts can remain stable from a semantic point of view, representative examples in the stream (and thus apparent probability density function) can change over the time.

Inversely, it is also quite frequent that the concepts change over time (this phenomenon is called concept drift) [33–35]. The other common situations can be explicitly mentioned here as addition of new concepts, fusion or splitting, extension and deletion of concepts. These phenomena could occur because of lack of initial expertise, changes in final goals or final needs. They can especially occur while processing document image streams. For these situations, the users should have the possibility to define their own classes of documents with adapted digitization protocols and decide to change them during the use to take care of new situations.

In all such cases, it is mandatory to update dynamically the class models used for the classification. The system has also to take directly advantage (in an incremental way) of past experiences. These elements have been quite extensively analyzed in the literature, especially in the field of adaptive classifiers dealing with online and/or incremental learning [19,41,43,44].

[1] Europeana Project: http://www.europeana.eu/. Gallica Project: http://gallica.bnf.fr/. NYPL Digital Collection: http://digitalcollections.nypl.org/.

[2] (ANR-10-CORD-0020)—CONTenus et INTeractions (CONTINT) http://digidoc.labri.fr.

In document images area, three important surveys on document classification can be mentioned here: [3,4,6], but none of them are presenting real progress in the field of dynamic classification and learning. Chen in [3] gives a very complete vision on each stage of document processing and learning. Chen presents the need for "incrementally update class models" but does not give any further analysis. Baharudin in [4] provides a review of the theory and methods of document classification and text mining, showing the advantage and disadvantage of many methods of supervised learning. This review focuses only on the situations where all categories are predefined and cannot be dynamically created. Recently, Singh in [6] has presented a new review including semi-supervised and non-supervised approaches, but still dedicated to an "automatic classification of documents in predefined categories." Most of techniques presented in these surveys are only contributions in terms of data re-processing and re-learning for dynamic classification.

Conversely, another way to deal with an efficient adaptation to the context is to use dynamic classification systems able to learn through time and uses. Some specific scenarii of this scientific challenge have been studied in the literature of machine learning [11,17,59]. The study of the state of the art shows that the notions related to incremental learning or classification remain quite ambiguous, dealing with many criteria, alternately with data management (online, batch, offline), with models (adaptation, addition) or with concepts. In any cases, more or less significant constraints exist on the learning procedure guided by old and new knowledge. For instance, online learning is a mechanism that should process its inputs example by example in order of arrival, ideally, without keeping the entire training dataset in memory, but only a nearest sequence of dataset. Then depending on studies, concepts can remain fixed, or change, and new concept can be added or not.

Here, we intend to classify the different approaches of dynamic learning so as to show their behavior toward data and concept management in the context of document images classification. We propose to structure these approaches in two main parts: the first is dedicated to incrementality for stream-based data processing (in stationary and non-stationary environments) and the second to the news trends relative to dynamic multi-purpose classification for document images.

2.2 Incrementality for stream-based data processing

Stationary environments Many approaches dedicated to stream processing are based on an online learning procedure. In such situations, incremental learning means that the system is able to improve/enrich its modeling with new incoming data with a strict hypothesis of constancy of number of concepts/classes and stationarity of their probability density functions. Some propositions coming from the machine learning community based on variety of classifiers [7–12,15,16] have been adapted to deal with document images, like [13] or [14]. Song [13] adapts a SVM method that has been proposed in [7,15,16]. This method replaces the classical learning function of SVM, by an exceeding-margin technique. The main idea is to check whether new data exceed the margin or not defined by the actual SVM model.

Incremental learning terminology can also be used at concept or class level. In the field of machine learning, Polikar proposed in [17] a definition corresponding to these criteria: An incremental learning procedure should be able to learn additional information from new data and should not require access to the original data. It should preserve previously acquired knowledge (it should not suffer from catastrophic forgetting), and be able to learn new classes while they are introduced with new data. This last point increases a lot the difficulty of incremental learning when the number of concepts is changing. Indeed, it has a high impact on the structure of classical classifiers with their learning algorithms. Representative works on this topic are [18,19] based on ARTMAP (*Adaptive Resonance Theory*); [5,20] based on SOM (*Seft Organization Maps*); [21,22] based on FIS (*Fuzzy Inference System*); and [17,23,24] based on *ensemble of classifiers*. For these approaches, the key idea is to model a new class by adding a new prototype or new classifier, when new examples are far enough from all existing classes. In the document classification area, there are very few research papers falling under this specific axis. We can mention [25] based on SOM, [26] based on k-NN, [27] based on ensemble learning. Hamza in [25] extends the idea of IGNG (Incremental Growing Neural Gas, a neuronal classifier) proposed by Prudent in [5], for invoice document classification. Bouguelia in [26] proposed to solve both image classification and zone classification by using a very simple incremental k-NN. New classes are added by keeping elements and associated labels that are too far from existing training samples of known classes. Finally, Ristin in [27] proposed a variant of Random Forests where the decision nodes are based on *Nearest Class Mean* (NCM) classification.

Non-stationary environments Learning from non-stationary environment and concept drift problems represent the ability of a system to auto-adjust the class models to the modification of learnt concepts that can evolve over the time. This drift implies a modification in the probability density function that represents the concept. It can either correspond to an extension of the concept, either a reduction or a "translation" which mechanically entails old parts of the concept to be forgotten. Consequently, the plasticity/stability dilemma must be the heart of a well-adapted solution and forces to consider either explicit or implicit detection of changes. To reach such plasticity, most of existing methods are working

on a selection of examples to be used for each new training step. This can be done with a *sliding window* as in [28–30] by keeping a fixed number of newest instances. In such situation, plasticity is high, but stability is low: It is impossible to keep track of old concepts. It can also be done by *weighting examples* as in [31–33]. By extension, approaches based on ensemble of classifiers can either create new classifiers for each new dataset or delete oldest one by a *pruning classifiers* approach as in [34–36]. In the document analysis field, we can mention two main works: Nattee in [38] proposed an online classification of journal title pages based on the Winnow algorithm [28] (*sliding windows* approach) and Salles in [39] integrates a "temporal weighting function" (*weighting examples* approach) in a classification system dedicated to document images collected from digital libraries.

2.3 News trends relative to dynamic multi-purpose classification for document images

More recently, the combination between incremental learning and concept drift has been studied. Elwell in [40] and [41] combined incremental learning (with Learn++ [17]) and *pruning classifier* approach for concept drift. Bouillon in [42] also introduces *decremental learning* into FIS with incremental learning ability based on a *sliding window*. But no more exploration about the ability of extending, contracting, merging or splitting concepts has been precisely studied in dynamic learning, especially in the area of document classification.

The main difficulties for classifier dealing both with the number of concepts and the evolutions of the concepts are due to rigid architecture in terms of number of concepts and interdependency between parameters. Consequently, when a concept is changing, all parameters have to be adapted. This needs many representative examples, which are not really available when dealing with data streams. Today the most efficient approaches are dealing with sets of interdependent classifiers (potentially redundant) which makes the success of ensemble methods.

This brief state of the art shows the lack of efficient solutions dedicated to stream-based document images including combinations between incremental learning and concept evolution. Our objective here is to propose an original contribution to online/batch document classification. The proposition lies on the introduction of independent classifiers (one-class classifiers), each of them being specialized on a specific task: content recognition and quality evaluation. The development of multi-one-class classifiers associated with a dynamic learning procedure has led to very promising results for document stream processing. The following sections make the demonstration of their accuracies.

3 Solution for dynamic classification of documents: the multi-one-class incremental SVM classifier

Introduction to the mOC-iSVM As discussed above, one of the major problems with classifiers dealing with concept evolution (both the number of concepts and concepts themselves) is the need for adapting many (all) parameters that are most often interdependent or redundant. Most of actual approaches of machine learning integrate complex procedures and structures, with many interdependent parameters, especially because they rely on discriminant approaches, which make the modifications inside the concept representation space quite difficult. Simplifying this inter-dependency leads naturally to one-class classifiers that only use positive data considering a unique class to build their model. These approaches are less studied since they are known to be less accurate than discriminative ones. However, they also have a huge potential such as insensibility to imbalanced data, rejection process to detect new concepts or ambiguities (and thus potential merging of concepts), multi-class labeling, and individual feature space. Those elements, key for our research field, convinced us to develop our classification scheme based on one-class approach. Precisely, we decided to work with one-class SVMs because the principle of vector supports selection allows to handle nicely incremental learning [43,44], at the same time with a summary of the data representing concepts. The potential of one-class principle gives to the classifier a high ability to share, split, and merge information with successive models of the concepts. We explore here these possibilities lying on the definition of generic procedures for managing the learning data and the evolution of the concepts. These procedures have shown a great potential through experiments realized on real images from the DIGIDOC project and have proved their efficiency for classifying document images in streams.

Our proposed approach, called mOC-iSVM (multi-one-class incremental SVM), is composed by a set of one-class SVMs, each one modeling a concept. The basic generic proposition has been introduced in [46] with only batch learning. It has been experimented on to two simple scenarii: incremental learning of fixed concepts and addition of new concepts. Here, our contribution lies on the extension of the learning procedure that gives to the classifier more flexibility to process both online and batch data, and to handle new more complex scenarii like splitting or merging concepts interacting with the user's decision. Let's remind here the main principles of the approach.

Principle of one-class SVM Shölkopf in [49] is the first who extended SVM principle to one-class SVM (OC-SVM). The main idea of OC-SVM is to consider only positive examples of the considered class to model the decision boundary. The principle lies on projecting the data onto an hyper-plane (in a

higher dimension) whose distance to the origin is maximal. The result is a binary function that returns $+1$ if data fall into the region containing positive data (capturing also the training data) and -1 elsewhere. This function is controlled by the parameter $\nu \in]0; 1]$ that represents an upper bound on the fraction of data that may be outliers. To find the hyperplane w that separates positive data x_i ($i = 0, 1, 2, \ldots, n$) from the origin with a threshold ρ and nonzero slack variables $\varepsilon_i \geq 0, \forall i \in [1..n]$, the system must solve the following quadratic optimization problem:

$$\texttt{min}_{\rho,w,\varepsilon_i} \frac{1}{2}||w||^2 + \frac{1}{\nu.n}\sum_{i=1}^{n}\varepsilon_i - \rho \tag{1}$$

subject to:

$$(w.\Phi(x_i)) \geq \rho - \varepsilon_i \tag{2}$$

The decision function has the following form:

$$f(x) = \texttt{sgn}((w, \Phi(x_i)) - \rho) \tag{3}$$

Extension to incremental learning The basic dynamic learning principle that we used to adapt OC-SVM classifiers to incoming data (the stream) has been inspired from Syed's proposition dedicated to incremental binary SVM [43]. For each class, we learn a OC-SVM that incrementally evolves using the classical SVM learning procedure with both old support vectors and new incoming data having the expected label (corresponding to the class the SVM is modeling). By this way, the concept drift is managed directly by the SVM learning procedure. In addition, to limit the drawbacks of one-class system, we also include the use of negative data (when available) but only during the parameter selection process of the system, which is performed at each learning step. Using negative data for parameter selection impacts positively the balanced accuracy estimated on the recognition of new incoming data, while preserving the modeling properties of OC-SVM.

The other kinds of concept evolution (addition/removal of classes, splitting or merging) are handled as follows: when a data with an unknown label is appearing,[3] we simply have to create a new SVM to learn this concept, without any impact on other SVM models. When the user is deciding to merge two concepts, the two OC-SVMs are replaced by a single OC-SVM trained with support vectors from both old SVM models (representing the two concepts to be merged) and the new positive data (corresponding to the merging). If a split is required, then two OC-SVMs are built, each one using the support vectors from the old OC-SVM that is split, and new data corresponding to each of the two new classes.

Overall algorithm Let $T_1, T_2, \ldots, T_i$ be the sequence of data coming at time $t_1, t_2, \ldots, t_i$. $C_{\texttt{system}} = \{C1, C2, \ldots, Cj\}$ is the list of classes that have already been modeled by the system at t_i. This list is increasing over the time when data with previously unknown labels are appearing (i.e., new classes that do not belong to $C_{\texttt{system}}$). We also consider $SV_{\texttt{system}} = \{SV^{C1}_{\texttt{system}}, SV^{C2}_{\texttt{system}}, \ldots, SV^{Cj}_{\texttt{system}}\}$ the list of sets of support vectors (SVs) for each class in $C_{\texttt{system}}$. $SV^{Cj}_{\texttt{system}}$ is the set of all SVs used to model the class Cj. D_i^{Cj} and $D_i^{\bar{Cj}}$ are, respectively, the data from T_i with class j (positive examples) and data from other classes (negative examples), at time t_i. We call GridSearch(D_i^{Cj}, $D_i^{\bar{Cj}}$), the function that selects the best parameters γ, ν during the training at time t_i for class Cj. The parameter ν approximates the fraction of errors accepted over SVs. γ is the parameter for Gaussian kernel we are using here. These parameters can be re-estimated at each learning step t_i.[4] In the case of online mode, we modify the GridSearch function to adapt the situation of missing negative data $D_i^{\bar{Cj}}$ by only using the positive data D_i^{Cj} to find the best parameter. In the previous version [46] with the batch mode, this GridSearch function was not available and the system had to use both positive and negative data. Finally, we define the function $\texttt{OneClassSVM}((\gamma^{Cj}, \nu^{Cj}), D_i^{Cj})$ as the learning function for class Cj. The algorithm is given in Algorithm 1.

The principle of the algorithm is illustrated in Fig. 1 with four successive learning steps. The stream contains data that initially belong to two classes ("blue" and "red" concepts). Each step is associated with a labeled dataset. In this example, we have 4 sets: Lot 1, Lot 2, Lot 3 and Lot 4 arriving successively (timeline t). Each set contains examples belonging to different classes. In the first step, the system divides the data according to the class they belong to. A OC-SVM is learned on each subset of data to produce a model for each represented class. Each one contains corresponding support vectors. The multi-class system resulting from this learning step can then be used to process new incoming data from the stream for their classification, until next training set is available. At this second step, for each class, the system uses all support vectors from previous model with new data of the same class: SVs 1 in orange box corresponding to red class are used with red data from Lot 2. The new learnt models replace the previous ones. The mechanism is then going on according to the same principle. At time T_3, when some data are tagged with an unknown label ("violet"), they are used

[3] Please remind that we are dealing with supervised learning.

[4] The system could handle the adaptation of the parameters on each learning step to assure the best integration of concept changing. The system could also use the predefined parameters to economize the learning cost.

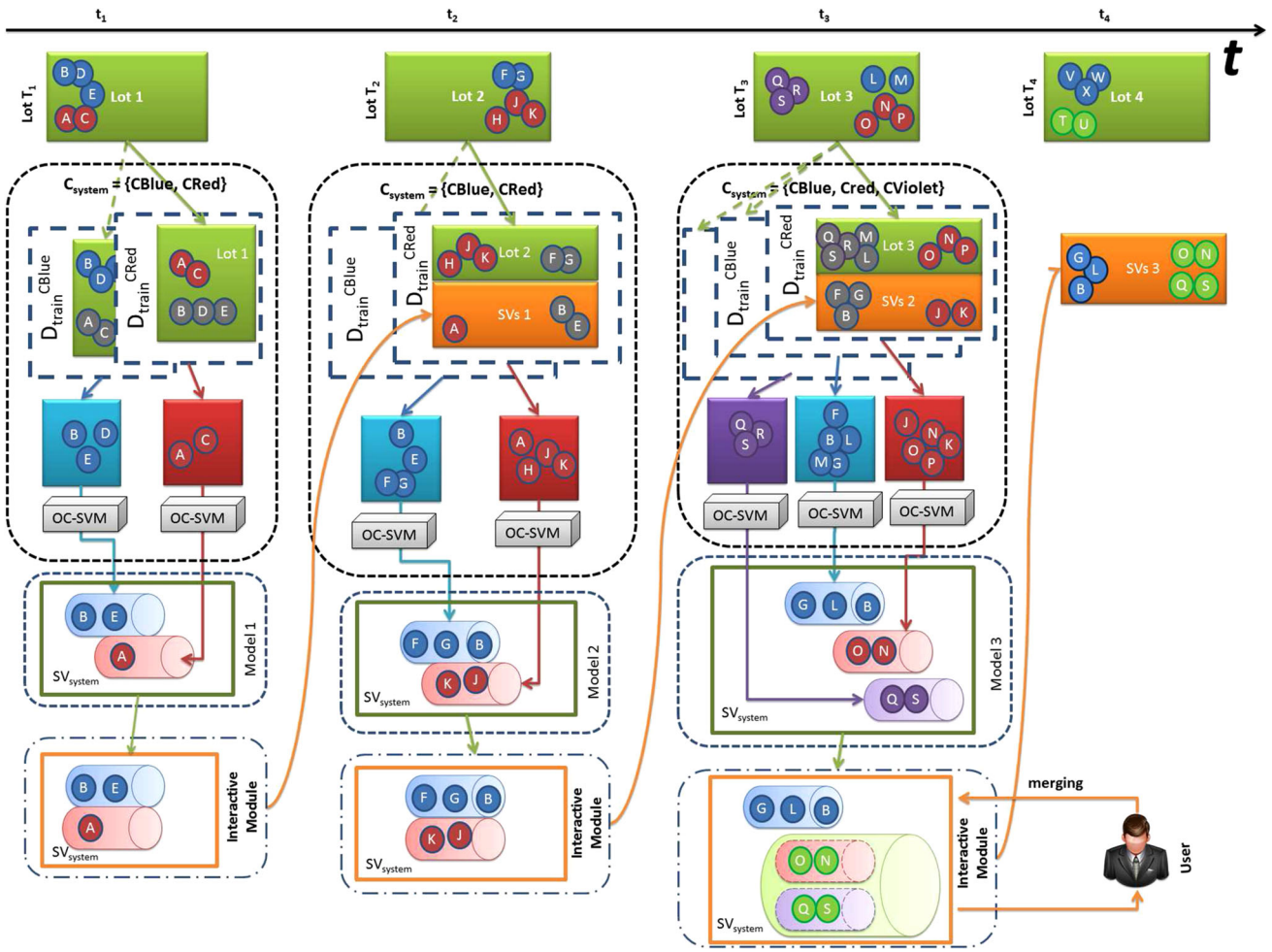

Fig. 1 Graphical illustration of mOC-iSVM algorithm with a stream divided into 4 learning steps and containing 3 classes: "*red*," "*blue*" and "*violet*"

to learn a new OC-SVM dedicated to this new class which is integrated into the system and then handled in the same way as the others. To adapt the basic algorithm in [46] to new complex scenarii (concepts splitting, merging or deleting) with user interaction, an interactive module is added to the system; see Fig. 1. This module, besides allowing the correction of the classification results provided by the current mOC-iSVM, has two main tasks: firstly, transform the class model by transferring the old models into the new created ones (two or more old models into one new model or inversely, one old model into two or more new models) and secondly, activate or deactivate the remaining class models. In case of concepts splitting or merging, old models must be deactivated and in case of class suppression, this solution simply deactivates the deleted model. The user can even reactivate or reuse these old deactivated models if necessary. At time T_4, the user decides to create a class merging (between "blue" and "violet") via this interactive module.

4 Theoretical validation of mOC-iSVM on batch and online learning.

In this part, we evaluate the ability of the system to achieve good accuracy while doing successive batch learning steps and online learning, whereas the number of concepts is increasing. The evaluation is done on a classical benchmark, not related to end-user experiments described in Sect. 5, to allow the comparison with other state-of-the-art algorithms. The dataset used is the Optical Handwritten Digits DataSet (UCI Machine Learning Repository). The choice for this dataset is linked to existing state of art results with other incremental algorithms. This dataset is composed of ten different classes (digits) represented by 5,620 examples (3823 for training and 1797 for testing) and 64 attributes. It has been created by Kaynak in 1995 [50].

Algorithm 1 mOC-iSVM algorithm

INIT STEP:
for each Cj in $C_{\texttt{system}}$ **do**
  $D_i^{Cj} = \emptyset$
end for
STEP 1: ADD NEW CLASS IF EXISTS
for each `data` in T_i **do**
  //if the class label of data is not in $C_{\texttt{system}}$ then name it $Cj+1$
  if `label(data)` $\notin C_{\texttt{system}}$ **then**
    //increase the number of classes
    $j = j + 1$;
    //add new class to list of class
    $Cj \leftarrow$ label(data)
    $C_{\texttt{system}} = C_{\texttt{system}} \cup Cj$
    $D_i^{Cj} = \{\emptyset\}$
    //create new list of SVs for the new class Cj
    $SV_{\texttt{system}}^{Cj} = \{\emptyset\}$
    $SV_{\texttt{system}} = SV_{\texttt{system}} \cup SV_{\texttt{system}}^{Cj}$
  end if
end for
STEP 2: CREATE LEARNING SETS FROM T_i
$\forall k \leq j$, $D_i^{Ck} = \{\texttt{data} \in T_i / \texttt{label(data)} = Ck\}$
$\forall k \leq j$, $D_i^{\bar{C}k} = \{T_i - D_i^{Ck}\}$
STEP 3: UPDATE MODELS
for each Cj in $C_{\texttt{system}}$ / $D_i^{Ck} \neq \emptyset$ **do**
  //add SVs of class Cj in training data of class Cj
  $D_{\texttt{train}}^{Cj} = SV_{\texttt{system}}^{Cj} \cup D_i^{Cj}$;
  //select the best parameters γ and ν
  $(\gamma^{Cj}, \nu^{Cj}) = \texttt{GridSearch}(D_{\texttt{train}}^{Cj}, D_i^{\bar{C}j})$;
  //run the classic One-Class SVM to find new SVs of class Cj
  $SV_{\texttt{system}}^{Cj} = \texttt{OneClassSVM}((\gamma^{Cj}, \nu^{Cj}), D_{\texttt{train}}^{Cj})$;
end for

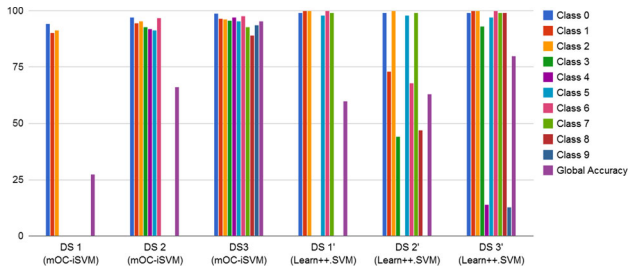

Fig. 2 Accuracy of mOC-iSVM and Learn++.SVM based on Erdem et al's [24] protocol

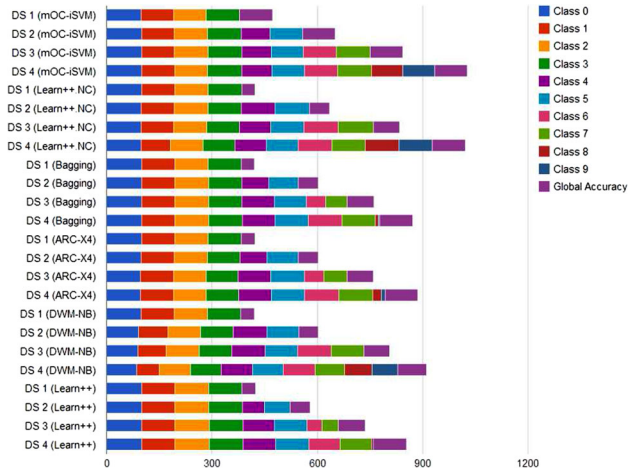

Fig. 3 Accuracy of mOC-iSVM, Learn++.NC, Bagging, ARC, DWM, Learn++ based on Muhlbaier et alś [23] protocol

4.1 Experiments with batch learning

We are first evaluating the performances of our method in its ability to add new classes of digits along with the sets of incoming data. In [46], we have already shown the very good results of our approach compared to classical non-incremental one-class SVM (OC-SVM) algorithms and other incremental approaches like Learn++.SVM (Learn++ with SVM) [24], Learn++.NC (Learn++ with New Class) [23], Bagging, ARC (ARCing), and DWM (Dynamic Weighted Majority) [35] approaches. We remind here these results, based on the same experimental protocols that the ones proposed by Muhlbaier in [23] for Learn++.NC and the one of Erdem [24] for Learn++. SVM.

Erdem, in [24], divided the training dataset into three different parts corresponding to three successive incremental training steps (T_i). Each part is mutually exclusive and contains, respectively, 30, 35 and 35% of the training data, drawn randomly. The data subsets from [24] are not available, so, in our experiment, classes are appearing in a different order, with similar proportion of data at each step. Since our system is based on independents OC-SVM, this way of preparing data might not have impact on results.

The experiments detailed in [23] were using 90% of full dataset as training set. Authors divided these training data into 4 different parts corresponding to 4 successive incremental training steps (22.5% each one), drawn randomly. This second experiment aims at comparing the accuracy of the mOC-iSVM with Learn++, Learn++.NC, Bagging, ARC, DWM.

For both cases, the test set used at each step is fixed (original test set in [24]; 10% of full dataset in [23]). The accuracy for each class, defined as the proportion of true results (both true positives and true negatives) relatively to the total number of examples, is the performance metric used. For alternative approaches, we report the results with same metric, mentioned in the above papers. Please notice that for mOC-iSVM, we used the grid search at each step for the parametrization of the one-class SVM.

Through the results depicted in Figs. 2 and 3, we can observe that the mOC-iSVM classifier is able to add new classes during the training and has the ability to improve its performances step by step. In comparison with Learn++.SVM (Fig. 2), the accuracy of mOC-iSVM is higher on most classes whenever they are appearing (see step 2, classes 3 and 8 for Learn++.SVM compared to classes 3 and

6 for mOC-iSVM; and at step 3 the new classes 4 and 9 for Learn++.SVM compared to classes 7 and 9 for mOC-iSVM; the class 3, introduced at same time step corroborate this analysis). Moreover, these good results remain throughout the different steps while for Learn++.SVM, because of inter-dependencies of binary SVMs, the addition of new classes has some impact in the accuracy: It is decreasing for the class 1 and 6 at step 2. Also, at the end, while all learning data have been processed, the overall accuracy is far better for mOC-iSVM than others methods (10% better).

In the case of Mulhbaier's protocol in [23] (Fig. 3), when new classes are appearing (class 4 and 5 in step 2; class 6 and 7 in step 3; class 8 and 9 in step 4), mOC-iSVM classifier can again learn very efficiently and quickly the new classes. It leads to an accuracy better than all other approaches. Particularly, Bagging, ARC, and Learn++ have lower performances with the introduction of new classes (see step 4 with the new classes 8 and 9). In the final step (see step 4), when all data have been processed and all classes have been learned, mOC-iSVM classifier has the same accuracy than Learn++.NC. Both are much more efficient than Bagging, ARC, DWM and Learn ++ that give recognition rates only between 75 and 82%.

4.2 Experiments with online learning

In this section, we illustrate the performances of the mOC-iSVM classifier with online learning: each time a new example is available, a new training is performed. We are still using the same dataset as previously with original test set (1797 examples). A part of the learning set (3000 examples randomly chosen among the 3832) is used for online learning. The full experiment is repeated five times with different learning datasets (different examples and different order of presentation in the stream). In this online mode, new classes can appear at anytime during the process. Moreover, at each step, we compare the online learning with a batch learning realized every 50 and 100 data. The parameters of the mOC-iSVM are determined as previously (with a grid search at every learning step) for the batch learning procedure. But for online learning mode, we are using fixed parameters for OC-SVMs in such way: The first 50 data are processed in batch mode (with grid search), and then, these parameters are kept fixed during online learning on remaining data.[5]

The performance measure used on this experiment (and next ones) is the Balanced Accuracy (BA [57] where TP is true positives, FN false negatives, TN true negatives and FP false positives) for each class:

$$\begin{aligned} \mathrm{BA} &= \frac{\mathrm{sensitivity} + \mathrm{specificity}}{2} \\ &= \frac{0.5 \times \mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{0.5 \times \mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \end{aligned} \tag{4}$$

The Global Balanced Accuracy (GBA) is also used as a performance indicator for all one-class classifiers taken together. This way of evaluating the classifier accuracy takes into account imbalanced data in the test set and consequently allows to evaluate with a higher precision the compromise between detection of true positives (sensibility) and rejection of true negatives (specificity).

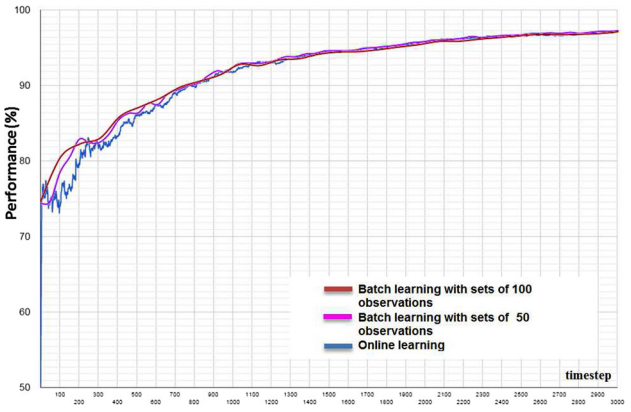

Fig. 4 Result of mOC-iSVM classifier in online mode compared to batch modes with varying training set sizes (50 and 100 observations per batch)

Figure 4 presents the results, averaged over the five tests carried out. We can observe that the mOC-iSVM is sensitive to the amount of training data. This phenomenon has been detailed and explained by Sato in [45]. In the early steps, the accuracy is generally lower than the one observed with the batch mode and the results are less stable. This phenomenon is then smoothed up to 300 observations. Beyond this threshold, the stability is maintained and the accuracy becomes similar for all approaches from the 800th example. This figure clearly illustrates the incremental learning capacity of mOC-iSVM: Thanks to its efficient selection of support vectors, we can see that the performances are always increasing (except some minor local variations). We can also notice that on such data, the parameters of OC-SVM fixed after only 50 data are not criminal since online method is in any case converging toward batch methods with regular re-estimation of parameters. Finally, we can also see that the performance of the mOC-iSVM is not much dependent on the choice of the data batch size.

[5] We could have done some grid search optimization at different times, as in batch mode, by keeping some training examples, for example, each of the 50 data. A windowing technique could have also been used. Consequently, the results obtained can be easily improved.

5 Application on document stream classification: scenarii for a smart digitization process

5.1 Embedding mOC-iSVM inside a smart scanner

The goal of these new series of experiments is to evaluate the capacities of our dynamic system to perform two identification tasks: first to identify the quality of a digitized document; second to categorize a document based on its content and user needs, taking into account possible evolution of document categories (apparition of new classes, merging, splitting, and drifting). A specific digitization protocol (scan parameters) is associated with these categories. It aims at providing a better or complementary digitization of documents according to their content/quality and to the user needs. The classification is performed on a data stream composed of document images continuously produced by a scanner. Our objective is to illustrate the ability of the mOC-iSVM approach to classify the incoming data in situations where the learning steps are very frequent and realized on small data quantities, and where the user plays an important role to guide and to control (accept/modify) the system decisions (i.e., results of classification). The user has also to determine the frequency of learning steps on valid data. Each time a document is processed by the scanner, a default scanning protocol is used to scan the image and to provide a "normalized" description on which classification tasks are performed.

At the beginning of the process of images digitization, the scanner is working in a fully supervised mode. This means that the classifier provides recognition scores for all classes that already exist in the scanner and the consequent role of the user is to validate the correct class assignments and the digitization protocol associated with them (the protocol and parametrization of the scanner will not be detailed in this paper).

During the digitization process, the user will have the possibility to interact in different ways with the system depending on the targeted application (content recognition or quality recognition). Each interaction will have feedback with the learning process, either in specifying defined actions (of adding, subtracting, splitting or merging classes), or in "just" validating the recognized image.

More precisely, the user has the possibility to order four different actions: He can create new classes (+) or suppress (-) one existing class if it is no more useful. He can also order to split (when an existing class has to be divided into two or more different classes) or merge classes (when a class has to extend its frontiers by merging with other existing class models). This last situation can be encountered in the only case of content recognition (see Fig. 5).

In a digitization scenario, the frequency of learning is flexible: The system can either wait a for a given number of digitized images (called batch mode in Fig. 5), or it can give rise to learning after each validated scan (called online mode in Fig. 5). After the first learning steps, the classier improves its accuracy and is able to decide autonomously the current document category (or the image quality if the classifier is dedicated to image quality identification). Each time, if the system returns an incorrect suggestion, the user can interact with it by correcting the category assignment or by creating a new one associated with the new digitized image. Particularly, for the content recognition, he can also split or merge

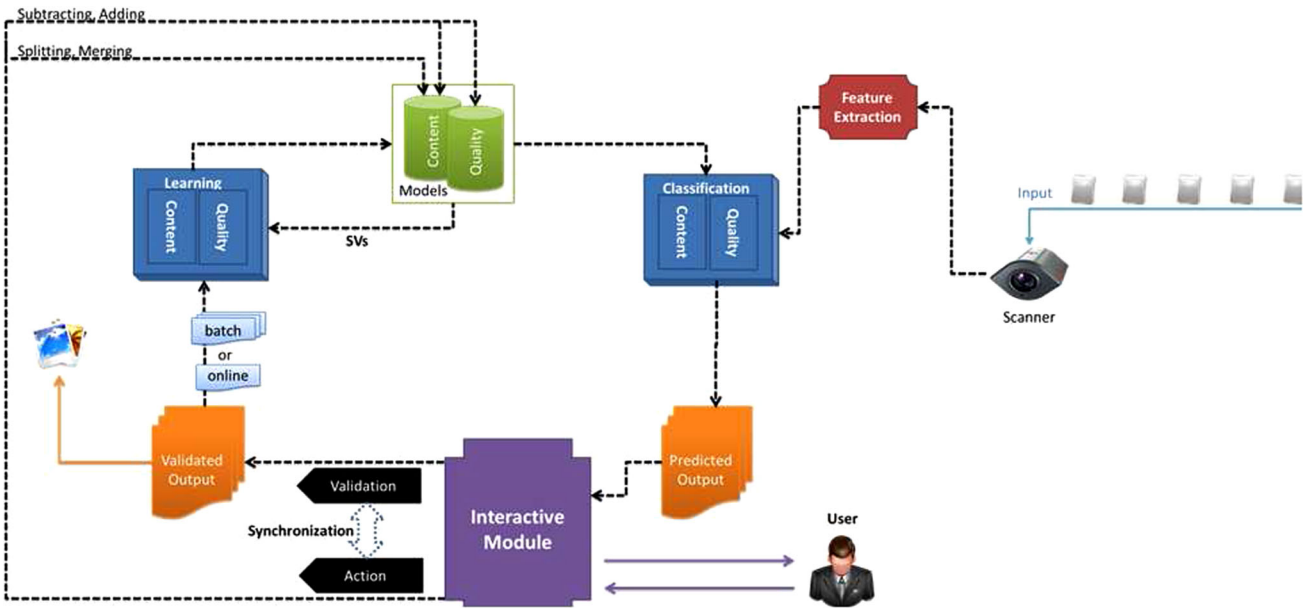

Fig. 5 Overall flowchart of the system for digitization of document quality and content

some existing categories before performing class assignment to adapt the system into his goals of processing.

5.2 Identifying document quality from a stream

5.2.1 Dataset and feature description

The database used for these experiments contains three classes of document images based on three different levels of quality depending on level of blur: distinct, blurred, and very blurred (see Fig. 6). This categorization in three levels of quality is studied in [51] who detailed the relation existing between the different qualities of a document image and the expert definition of the quality for different purposes: archiving, storing, visualizing and analyzing. The first level "*distinct*" means that the quality of a document image is satisfying for an industrial task (content recognition by OCR, content segmentation, etc.). The second level "*blurred*" means that the quality of a document image can be satisfying for human reading but not for automatic processing such as OCR. The last level "*very blurred*" means that the human eyes can identify that the document image is blurred. With the initial document image considered to be "distinct" (best quality), we can get the second and the third level by applying a simple Gaussian filter with $\sigma = 0.8$ and $\sigma = 1.2$, respectively [51]. We applied this principle on images with different contents (printing, handwriting, graphic) each one existing at the three levels of quality.

The classification of image quality relies on simple gradient measures. Vinsonneau, in [51], has shown that this simple method can provide very good performances compared to many others complex methods like Haar method [52], Defocus Blur Estimation [53], Binarization of Fourrier [54].

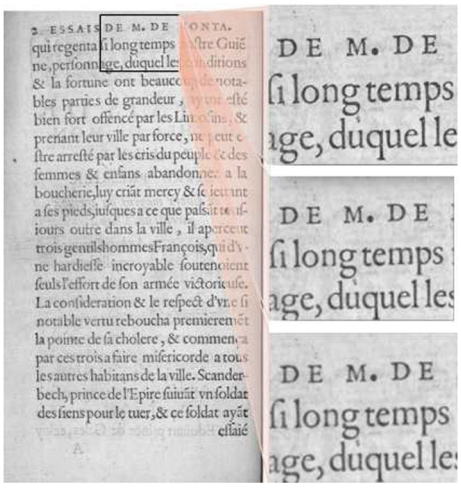

Fig. 6 Three levels of quality (from top to bottom: distinct, blurred, and very blurred)

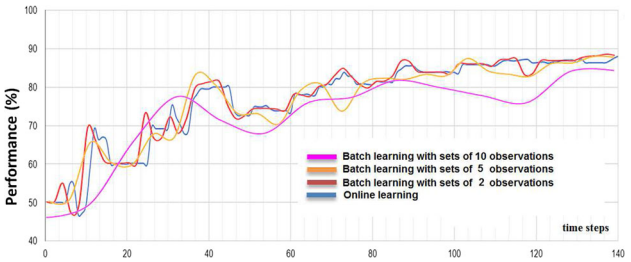

Fig. 7 Comparison between Global Balanced Accuracy (GBA) obtained from different sizes of learning sets and with online learning on document quality dataset

5.2.2 Experiments with online and batch processing of a data stream

In this scenario, we illustrate the capacity of the mOC-iSVM classifier to determine the right document quality according to the gradient features. In those experiments, we used a series of 140 images for learning and observe the classification results on a test set of 450 images at each learning step. The images from learning set are processed as a stream, simulating a situation of real acquisition creating vibrations in the arrival of the image to be scanned in a predefined frequency. The incremental classification scenario is performed in batch mode as well as in online mode (at each time step an image is captured with a fixed digitization protocol and its gradient features vector is extracted for classification and learning). The system starts with a short series of images whose classification (quality level) is validated by an operator. After this initializing period, the training procedure starts to run and updates the models corresponding to the first classes of quality. After a while, the training procedure automatically continues to retrain old models with new incoming images that are validated. This can be done in both online or batch modes.

Figure 7 shows the performances with 3 different sizes of training sets (batch) applied on the stream: with 2 images, 5 images and 10 images per set.[6] We can observe that the Global Balanced Accuracy (GBA) is almost independent to the size of the learning sets. There are some minor differences on the evolution of accuracy: online mode seems more efficient than others in absolute values, but from the evolution point of view, they are the same. Between the 50th and the 60th steps, all curves go down probably because of a temporary change in apparent probability distribution function of classes (examples might be concentrated on a part of probability distribution function of the classes). Consequently, the system fails to recognize correctly the classes. But thanks to the selection of best performing SVs, the system is rapidly improving its accuracy and is going on in such way. In the

[6] Online results are also presented.

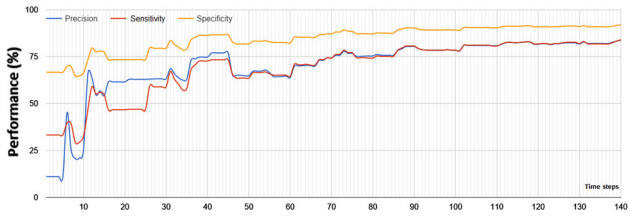

Fig. 8 Result of the mOC-iSVM classifier according to global precision, global sensitivity (or global recall), and global specificity with online learning on document quality dataset

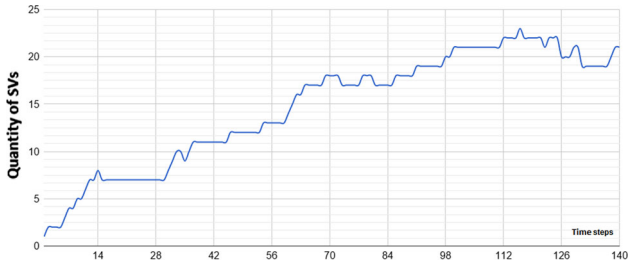

Fig. 9 Evolution of the quantity of SVs with online learning on document quality dataset

last step, the difference between all curves does not exceed 5%.

Figure 8 is giving global precision, sensitivity, and specificity for the same experiment (averaged over the 3 classes). Overall, the system is achieving a high specificity (i.e., high rejection rate of true negatives). The sensitivity (i.e., true positive detection or recall) and precision are starting with quite low values but are increasing more and quickly (the sensitivity is nearly double after 25 iterations). We can see also that in the 60 first steps, the results are quite unstable (the curves go up and down suddenly) but after a while, results become stable with higher performances.

Considering the quantity of support vectors (SVs) that are used in the SVM models (see Fig. 9), we can notice that it increases from 0 to around 20 to achieve around 85% of GBA (Global Balanced Accuracy). This means that only 20 SVs are needed to describe the overall 140 learning observations. If more SVs are added (after 100th step), the GBA is no more increasing. We can also observe that the quantity of selected SVs can decrease when new incoming examples can replace more than 2 other SVs (while preserving a high performance). Indeed, around the 126th step, the curve seems to lie down when the system is using more than 20 SVs (Fig. 9). Finally, we can mention that, in the 45th step (a step where the average performances are decreasing in Figs. 7 and 8): the best fitted SVs might be replaced by less efficient ones. Then, until the system can find out other good SVs in the 60th step, the accuracy is no more increasing.

These observations tend to put focus on one weakness of mOC-iSVM classifier: once good SVs are replaced by others, they cannot be reuse later again if needed. What is an advantage for flexibility could become a drawback for stability. Nevertheless, if other good representatives are encountered again, the system can quickly and efficiently select them to improve the performances. Let's notice that based on such observations, some mOC-iSVM variants have been studied to achieve a better trade-off between stability and plasticity, especially in non-stationary environment [47,48].

5.3 Categorization of a stream of document images based on their content

5.3.1 Dataset and feature description

The database used for these experiments contains six classes of documents, each one with a different content (see Fig. 10): Handwritten musical scores (490 instances; class *C1*), Printing I (230 instances; class *C2*), Printing II (99 instances; class *C3*), Handwriting I (379 instances; class *C4*), Handwriting II (103 instances; class *C5*), Maps (279 instances; class *C6*). The classes Printing I and Printing II are composed of documents whose origin is different (historical and recent documents) making images different considering background, noise and defects, fonts, etc. This distinction was made to produce 2 subclasses for Printing. This is also true for Handwriting. For all of the 6 classes, the content is relatively homogeneous. Particularly, printed and handwritten images with graphical elements have been removed to allow a better homogeneity.

The document images are described by a set of 41 features that have been selected for their ability to characterize the large variety of contents, and to ensure satisfying intra-class consistency. Considering that the documents are mainly composed by contrasted strokes and "binary" content (objects/foreground over background), most of descriptors are scalar values that express the distribution of objects pixels, their entropy, density, compactness. The analyzed documents contain evident differences in the amount of elements corresponding to their content (printed or handwritten text, line strokes, drawings, etc.). Those singularities are estimated by different features. So as to work with the best adapted features among the 41, we used the SVM-RFE method presented in [55] to select the most efficient ones.

Fig. 10 Dataset of document images with 6 classes of content

We finally kept 26 features among the 41: 6 geometrical blob features; 10 features based on the variation of luminance; 1 feature measuring the density of information; 1 feature representative to the complexity; 1 texture feature; 5 direction-based features; and finally 2 color features. Because the selected images of our dataset mainly contain gray-level images, the color features are used as complementary luminance features.

5.3.2 *Experiments with online and batch processing of a data stream*

In this scenario, we illustrate the capacity of mOC-iSVM to classify each image depending on its content, with the online and batch mode. A similar protocol as before is used here. We are using a stream of 130 images as learning dataset that are considered one by one or in batch mode (2 instances, 5 instances, 10 instances at each learning step). Each image belongs to one of the six classes of content, and they are introduced successively in the following order: *C1*; next *C2*; next *C3*; next *C4*; next, images from *C1* to *C4*; next *C5* and *C6*; and finally images from *C1* to *C6*. A test set with 1450 images is used for observing the evolution of accuracy for each classifier corresponding to each of the 6 classes.

Figure 11 gives the global precision, global sensitivity/recall and global specificity in case of online learning while introducing the different classes. We can see several phenomena. First, the average specificity remains stable while introducing new class models which means that the models are not affected by appearance of new classes. Moreover, each time a new class is encountered and the next learning step leads directly to a significant increase in recognition on tested data for that new considered class (see precision and sensitivity while learning *C2*, *C3*…). Furthermore, each classifier performance is also improved at each new learning step even if the new tested data do not belong to its own class. Indeed, a data can be wrongly accepted as *C1* by the *C1* one-class SVM until the creation of the *C2* classifier that will prevent ambiguities. When all models are known (at step 70th), we observe that the average precision, sensitivity, and specificity rates are increasing very slowly over the time with small oscillations until stabilization.

Figure 12 illustrates the comparative performance of mOC-iSVM with different sizes of learning sets (1[7], 2, 5 and 10 documents by set). The performance is measured using Global Balanced Accuracy (GBA). As previously, we can see that the variations of GBA do not exceed 5% while using different sizes of training sets. Consequently, the influence of this parameter is minor. This proves that the combination between new incoming data (even in much reduced quantity) and ancient SVs used all together for the training phases is a very efficient solution guaranteeing both older knowledge preservation and adaptation to new incoming data.

As observed in Fig. 9 for quality categorization, we can notice in Fig. 13 that after a given amount of SVs (around 30), their increase is not necessary followed by an increase in accuracy: GBA remains nearly constant after the 70th time step (less than 1% of difference). Theoretically, it could be interesting and useful to be able to determine such threshold to limit number of SVs used and so to limit the complexity of the system and maintain its speed. Consequently, it might be possible to reduce the frequency of training steps.

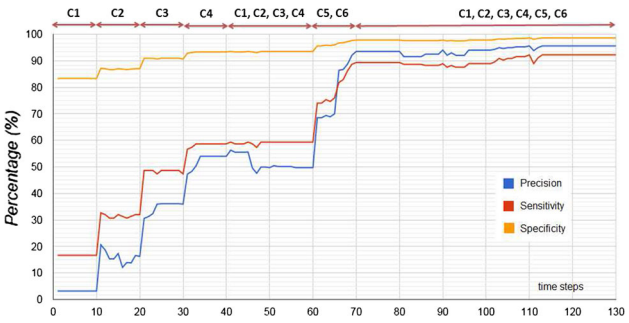

Fig. 11 Performances of the mOC-iSVM classifier based on global precision, global sensitivity (or global recall) and global specificity while doing online learning on document content dataset

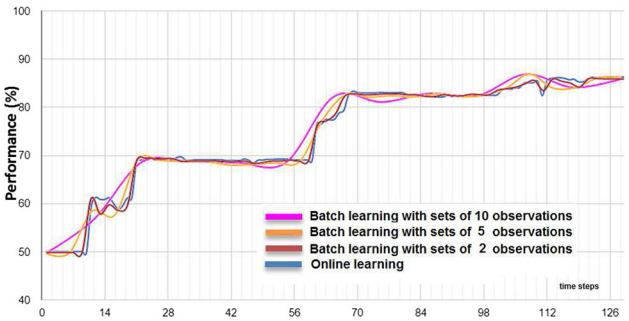

Fig. 12 Comparison between accuracies (Balanced Accuracy) obtained with different sizes of learning sets on document content dataset

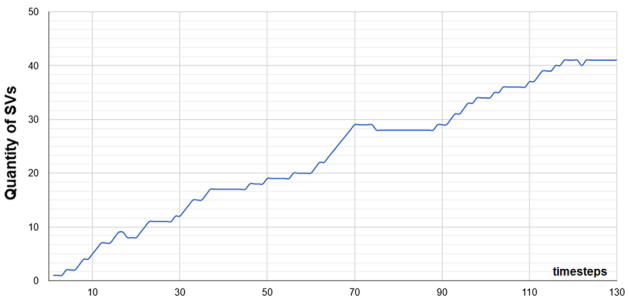

Fig. 13 Evolution of the number of SVs with online learning on document content dataset

[7] Or online learning.

5.4 Experiments considering different kind of concept evolution

All results following are obtained using the previous document content dataset with a 5-fold stratified cross-validation. To simulate the data stream, the full learning set for each cross-validation is split into 12 subsets that are corresponding to 12 successive learning steps. Each set is mutually exclusive and contains, respectively, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 20, and 20% of the training set, drawn randomly. Consequently, small number of data per class are coming first to evaluate the ability of the learning procedure to work with few examples (from 2 to 8 data). The test set used at each step is constant (remaining part of dataset left apart by cross-validation). The different scenarii proposed in the next sections simulate a data stream in which different situations of evolution of classes are encountered. Of course, depending on the scenario used, the labels for each document image are adapted to meet the requirements of the experiments (see below).

5.4.1 Standard incremental learning scenario (fixed number of concepts)

The aim of this first scenario is to evaluate the incremental ability of the system to learn over the time, with a fixed number of classes. At the end of the last step, the performances are compared to the one of the static systems, learnt using full training set. This static approach is a classical multi-class SVM composed with a set of one-class SVM and using the max rule for class assignment. The mOC-iSVM is learnt using a grid search at each learning step, with negative data, in order to improve their discriminative accuracy. The system will handle the adaptation of all parameters of system given by the grid search function when a concept is changing during time. As mentioned before, this does not impact properties of one-class SVM. We are doing the same with the static system.[8]

The results presented in Table 1 show the accuracy (Global Balanced Accuracy—GBA) of mOC-iSVM achieved after the final learning step, compared with the one of the static one-class SVM. We can observe that GBA values are similar demonstrating that the dynamic learning process is not degrading the accuracy. This was already observed in [46] on digits benchmark.

Figure 14 shows the temporal evolution of Balanced Accuracy (BA) for each class, as well as GBA. This last one is globally increasing, even if for some classes, and at given steps, we can observe a decrease in BA. This phenomenon can be explained by the progressive introduction, at each step, of examples that are making the classes more difficult to distinguish. Then only more representative examples, coming later, can help to remove ambiguities.

[8] Experiments we have performed have shown that using the negative information available for parameter selection, in case of one-class SVM can improve their performances of at least 10%.

Table 1 Performance comparison between mOC-iSVM after all training steps and static multi-one-class SVM (learnt in one step with all data)

Approaches	Static One-Class SVM (%)	mOC-iSVM (%)
Musical scores	97.94	98.33
Printing I	72.90	72.31
Printing II	73.78	73.78
Handwriting I	88.78	88.51
Handwriting II	99.42	99.46
Map I	82.70	83.00
GBA	89.25	89.23

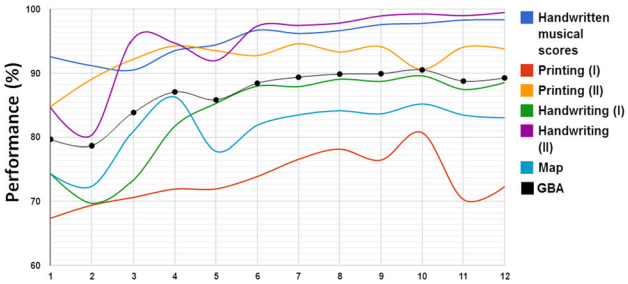

Fig. 14 Accuracy (BA and GBA) of mOC-iSVM over the learning steps considering fixed number of concepts

5.4.2 Incremental scenario with addition of concepts

This second scenario is used to evaluate the ability of the system to model new incoming classes, and its impact on existing classes. Here we consider that the introduction of a new concept is provided by the user giving some examples with the new label.

The document stream is composed as follows. Musical scores class is coming since 1st step. Handwriting I is coming only from 2nd step. Then Handwriting II is appearing from 3rd step, Printing I from 4th step, Printing II from 5th step and, finally, Maps from 6th step. The next steps (from 6th to 12th) are the same as in the 1st scenario (Sect. 5.4.1). So as to compare results with this first scenario, we have to consider that less training examples are used for some classes. Indeed, examples in learning sets that belong to classes that have not yet been encountered are completely removed from learning procedure (for example, the Maps examples belonging to steps 1–5 in first scenario are not used for training in this scenario).

The results are depicted in Fig. 15. We can observe that the first classes to be learnt do not suffer at all from confusions with other classes. Thus their accuracy is very high. Of

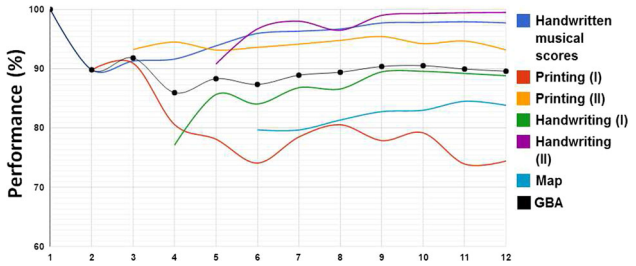

Fig. 15 Accuracy (BA and GBA) of mOC-iSVM over the learning steps considering an increasing number of concepts

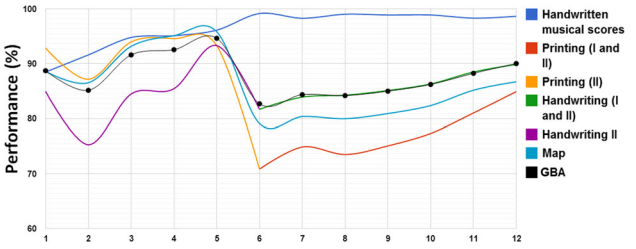

Fig. 16 Accuracy (BA and GBA) of mOC-iSVM over the learning steps considering concept extension from Handwriting II (resp. Printing II) to Handwriting (resp. Printing)

course, when new classes are added, the accuracy decreases but without being much less than in classical scenario. For example, in 2nd step, the BA of Musical Scores is only 3% less than in previous scenario and increases just after. Similarly, Printing I is starting with a higher accuracy that decreases next up to 6th step where all classes have been added. All other classes are associated with a regular increasing in BA, and in 12th step, we obtain the same GBA than previously, even if less data have been used for learning.

To conclude this part, we can propose alternatively to use an automatic detection of new concepts thanks to the introduction of a rejection rule, which might be easier thanks to the use of one-class classifiers (a big advantage over discriminative approaches): if a sample is considered to belong to none of the existing classes, then we can assign it a new label. This possibility is not studied here.

5.4.3 Incremental scenario with extension of concepts

The 3rd and 4th scenarii are designed to simulate concept extensions. For instance, in our classification problem, we have two kinds of handwritten documents and two kinds of printed documents. Each one could be considered to belong to the same kind of documents, i.e., to the same super-class Handwriting (and Printing, respectively). To simulate this evolution in 3rd scenario, the system is first trained (steps 1-5) on the first subclasses (Handwriting I and Printing I) with a generalized label (Handwriting and Printing), both of them with other document classes, as in first scenario. From 6th step, examples from the two other subclasses (Handwriting II and Printing II) are also coming with the same label (Handwriting and Printing). Doing so, the concept of Handwriting (resp. Printing) is extended from Handwriting I (resp. Printing I) to Handwriting II (resp. Printing II). Of course, the test set used here is adapted to reflect these modifications (examples of Handwriting II and Printing II are not considered in steps 1-5 and are considered to belong to the same class as Handwriting I and Printing I after 6th step). The 4th scenario is the same, but considering firstly the examples belonging to Handwriting II and Printing II instead of Printing I and Handwriting I.

Only results on the 4th scenario are presented here (the 3rd one is presenting a more regular increase in accuracy and thus is less interesting in terms of classifier behavior). Since subclasses Handwriting II and Printing II are quite easy to model (different from other classes), the BA is quite high before the extension (see the curves of Handwriting and Printing in Fig. 16). But at the 6th step, when new subclasses (Handwriting I and Printing I) are coming, the loss in BA is significant, even for the Map class which is impacted by this extension (notice that it is not the case for musical scores). Next, the increase in BA during the following steps (6th to 12th) is regular and at 12th step, the final BA of Handwriting (resp. Printing) is close to the average BA of its subclasses, showing that the loss of accuracy has been nearly caught up, even if less data have been used (for subclasses coming at 6th step, i.e., Handwriting I and Printing I).

5.4.4 Incremental scenario with concept drift

Concept drift is a very typical case when considering processing of data stream. This phenomenon occurs when the probability density function of a class is changing over the time. The previous scenario (concept extension) could be considered as being a specific case of concept drift. Nevertheless, concept drift often implies that the initial probability density function is forgotten and replaced by a new one, which is not true with concept extension. To simulate classical concept drift, we are using again the two Handwriting and Printing subclasses. As previously, we have two scenarii. The first one considers the drift from Handwriting I (resp. Printing I) to Handwriting II (resp. Printing II). The second one is based on the reverse drift. In both cases, the drift that occurs at step 6 is abrupt since it corresponds to a shift from one distribution to another. Figure 17 only gives results corresponding to the first case.

In comparison with other approaches dedicated to such cases, our approach does not detect explicitly the drift but the one-class models are automatically adapted to the changes observed with a high plasticity. Indeed, between the 6th and the 7th steps, the maximum BA achievable on target classes is already nearly obtained. We can also notice again that

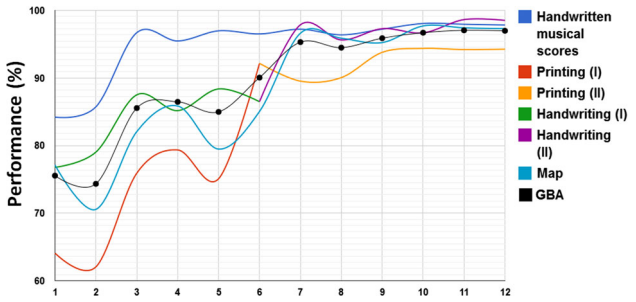

Fig. 17 Accuracy (BA and GBA) of mOC-iSVM over the learning steps considering concept drift from Handwriting I (resp. Printing I) to Handwriting II (resp. Printing II)

Musical Scores class is not impacted by the drift. The novelty here comes from the increase in accuracy for the Map class which was not true before. One explanation could be that the concept drift eliminates some confusions between Map and Handwriting I/Printing I.

5.4.5 Incremental scenario with split and merge

As far as authors know, splitting and merging concepts are specific scenarii of concept evolution that have not been studied so far with dynamic supervised learning. Here, we consider first the merging of Handwriting I (resp. Printing I) with Handwriting II (resp. Printing II) at learning step 6 (see Fig. 18). Next, we consider the splitting of a Handwriting super-class (resp. Printing super-class) into two subclasses corresponding to Handwriting I and II (resp. Printing I and II). This is also performed during the 6th step.

In Figs. 18 and 19, we can observe that the accuracy is decreasing when the splitting or merging is done which is quite evident.[9] But just after, it is starting to increase again quickly, step by step, showing that models can be adapted to the new structure of concepts. Here again, we can notice that the final BA of the classes becomes nearly the same as the one that could be expected regarding previous scenarii. We can also notice that the effect on subclasses is not always the same. For example, in case of splitting, both subclasses of Printing are suffering from the changes, whereas in case of Handwriting, Handwriting II keeps same accuracy than its super-class and only Handwriting I is decreasing in accuracy. In case of Handwriting merging, the accuracy is nearly averaged over the two subclasses, whereas for Printing, the accuracy remains similar to the one of the less accurate subclass (Printing I). Of course, this could be explained by the intrinsic ambiguities between subclasses, which gives some ideas on how automatic detection of splitting/merging could be proposed to the user, according to feature space, instead of considering semantic point of view. Alternatively, this could help to find better feature sub-space adapted to a better discrimination (or less ambiguities) between one-class models.

[9] Please notice that in this figure, as well as others, interpolated curves are shown to provide a better rendering of a stream simulation (considering our protocol a more exact representation would have been dot or step plots). Consequently, there is apparent loss of accuracy between steps (between steps 5 and 6 here, for example). In fact, in our scenarii such loss only occurs at the new step (step 6 here) and not before.

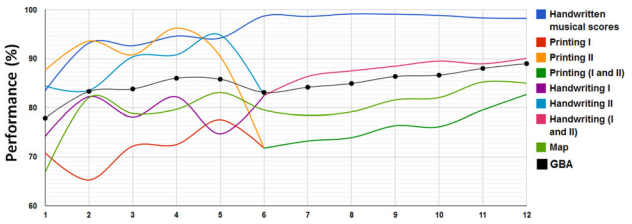

Fig. 18 Accuracy (BA and GBA) of mOC-iSVM over the learning steps considering merging of Printing I (resp. Handwriting I) with Printing II (resp. Handwriting II). Printing and Handwriting are corresponding super-classes

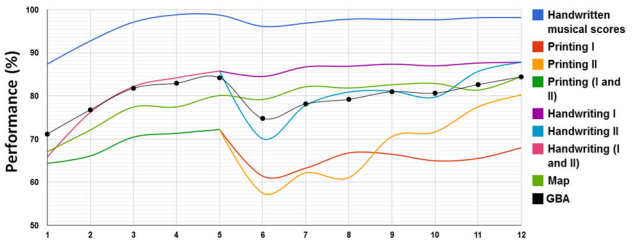

Fig. 19 Accuracy (BA and GBA) of mOC-iSVM over the learning steps considering splitting from Printing (resp. Handwriting) into Printing I and II (resp. Handwriting I and II)

6 Discussion on runtimes of learning

For the implementation, we used LibSVM [58] in MATLAB to implement our proposition on a PC with a Intel core i7 2.40GHz CPU and 8GB of RAM. For the kernel function, we used RBF (Radial Basis Function). Firstly, let talk about the quality of the models. Figures 20 and 21 present the evolution of the quantity of SVs in the models of the system. It shows that the final models have almost the same quantity of SVs whether they were trained in online or batch mode.

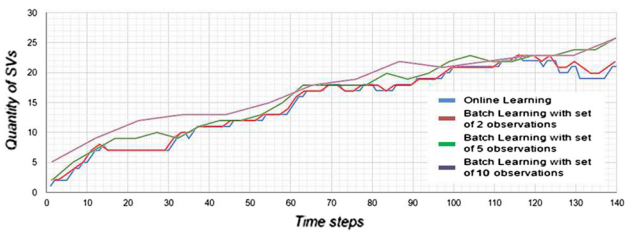

Fig. 20 Evolution of quantity of SVs on document quality dataset

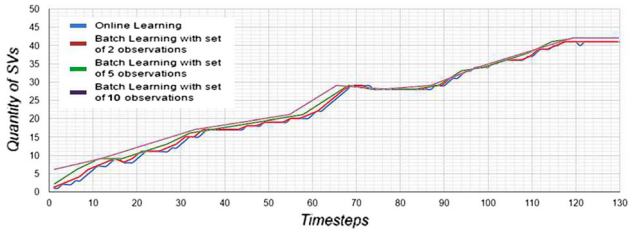

Fig. 21 Evolution of quantity of SVs on document content dataset

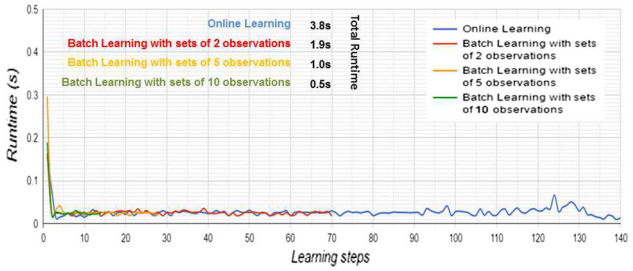

Fig. 22 Runtime of each learning step on document quality dataset

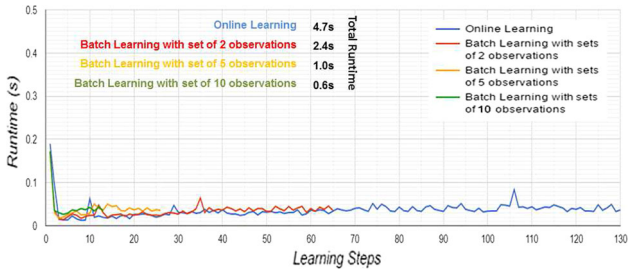

Fig. 23 Runtime of each learning step on document content dataset.

In addition to the conclusions of the previous experiment (Sects. 5.2, 5.3), in which the performance of learning is independent of the choice between online and batch mode, we can conclude that the quality of models is almost identical and independent to the learning configuration. The main difference comes only from the runtime.

Theoretically, the runtime of mOC-iSVM is highly dependent on the number of data in the training set and on their parameters. In our incremental process, at each learning step, the system identifies automatically the best parameters through the *GridSearch* function. Consequently, the runtime of each learning step could vary. In this , we show the runtime achieved by our system configured to settings used in previous experiments. When using fixed parameters (as the experiments with online learning in Sects. 5.2, 5.3), these runtimes do not include the time for searching the best parameters. On the contrary, when the system optimizes its own parameters via the grid search (as experiments in Sect. 5.4), the runtime depends on the search space of parameters given to *GridSearch* function (it could increase highly for a wide range of value).

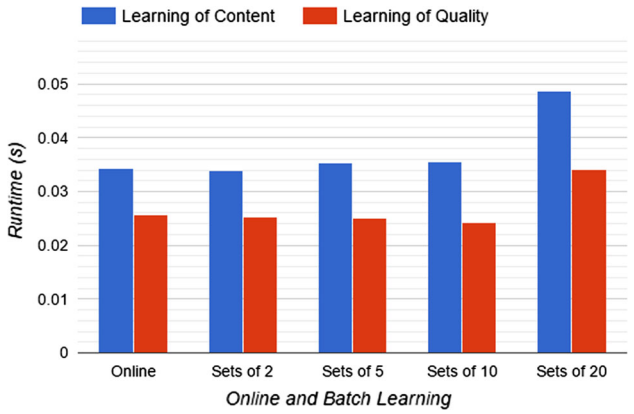

Fig. 24 Average runtime of a learning step: comparison between online and batch learning

The evolutions of runtime for each learning step for both online and batch learning on the quality and content dataset (Figs. 22, 23) show that the runtime is almost stable through time with a very small increase caused by the evolution of the models and their SVs (the difference between the 50 first steps and the 50 last steps is smaller than 0.0030s). The high runtimes in the first steps are influenced by the creation of models. In our experiments, the average runtime of each learning step is around 0.025s for quality dataset and 0.035s for content dataset.[10] The runtime of each learning step in online learning is similar to the batch mode with sets of 2, 5, and 10 observations (difference lower than 0.001s), Fig. 24. From the very large sets with 20 observations, the difference becomes more important (higher than 0.01s). The total runtime for a stream processing depends also on the quantity of performed learning steps, and the consequence is that even if the learning time of the online mode is less than the batch one, the total runtime in the online mode could be still much higher than batch ones.[11]

7 Conclusion

In this paper, we have proposed a new dynamic classification scheme based on a set of independent one-class SVMs (mOC-iSVM). This new proposition has been validated on very popular benchmarks devoted to concepts evolution learning. It shows the very good results compared to the best incremental classifiers of the state of the art. This classifier and its dynamic supervised learning procedure are dedicated to classification tasks into data streams especially when concepts are evolving over the time (concept drift, extension,

[10] One book of 400 pages in online learning will have an approximate cost of 10 and of 14 s for, respectively, quality and content learning.

[11] Particularly, the time needed for reading files (models of SVM, features, etc.) has a huge impact.

splitting/merging, new concepts). For a better illustration, we developed the study in the context of a classification task over a stream of digitized document. This contextualization is due to the DIGIDOC project; expected to define the functionalities of a self-adaptive scanners able to suggest automatically (under the supervision of an operator) the best parameters of the scanner to obtain better digitized images according to get the best suited images depending the objectives (quality and content). In such context, the operator can accept suggestions from the scanner or correct it including the possibility to change concepts associated with protocols (creating new classes, merging them, etc.).

Thanks to the selection of convenient support vectors and the independent concepts modelization with one-class models, the classifier can easily learn, even with few data, its new configuration and can be adapted very quickly to concept evolution. Indeed, the system presents a high plasticity in many situations like addition of new concepts, extension/reduction of concepts, concept drift, splitting, and merging of concepts. The wide spectrum of experiments done here has shown that the system can be used with online learning procedure or with batch learning. In all cases, the accuracy obtained after a given number of learning steps is the same as the one that is obtained with a classical static learning procedure using directly, in one step, all data encountered in the stream. During the digitization process, some classes can be positively or negatively affected due to some confusions that can be introduced or removed in the feature space representation. This is especially true when changes are proposed by the user when very few data are available.

The learning scheme presented here can be improved in several ways. First, to get a trade-off between plasticity and stability, we intend to keep an history of support vectors used over the time and to give them some weights according to their "activation" during data stream processing. We also intend to take benefits of one-class independent modeling to help the user who interacts with the system, especially in case of concept evolution. For example, rejection rules can be efficiently used to detect new concepts. Confusions between classes can also be used to propose some merging of concepts. We also expect to use dedicated feature sub-space for each one-class model, in order to limit the lack of discriminative power of such one-class classifiers and to achieve an accuracy much closer to the accuracies of discriminative approaches. Finally, we intend to evaluate the approach on other benchmarks and also on real data streams.

Acknowledgements This research has been carried out under the DIGIDOC project with financial support of the ANR (French National Agency for Research).

References

1. Helbing, D.: Thinking ahead: essays on big data, digital revolution, and participatory market society. p 194, Springer (2015)
2. di Lenardo, I, Kaplan, F.: Venice Time Machine : Recreating the density of the past; Digital Humanities 2015, Sydney, June 29–July 3 (2015)
3. Chen, N., Blostein, D.: A survey of document image classification: problem statement, classifier architecture and performance evaluation. Int. J. Doc. Anal. Recognit. **10**(1), 1–16 (2007)
4. Baharudin, B., Lee, L.H., Khan, K.: A review of machine learning algorithms for text-documents classification. J. Adv. Inf. Technol. **1**(1), 4–20 (2010)
5. Prudent, Y., Ennaji, A.: A New Learning Algorithm For Incremental Self-Organizing Maps, ESANN 2005, pp. 27–29. Bruges, Belgium (2005)
6. Singh, U., Hasan, S.: Survey paper on document classification and classifiers. Int. J. Comput. Sci. Trends Technol. **3**(2), 83–87 (2015)
7. G. Cauwenberghs, T. Poggio; Incremental and decremental support vector machine learning. In NIPS 2000, **13** (2001)
8. Karasuyama, M., Takeuchi, I.: Multiple incremental decremental learning of support vector machines. IEEE Trans. Neural Networks **21**(7), 1048–1059 (2010)
9. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of KDD 2000, ACM Press, New York, USA, pp. 71–80 (2000)
10. Su, M.C., Lee, J., Hsieh, K.L.: A new Artmap-based neural network for incremental learning. Neurocomputing **69**(16–18), 2284–2300 (2006)
11. Lughofer, E.: Flexfis : a robust incremental learning approach for evolving Takagi–Sugeno fuzzy models. IEEE Trans. Fuzzy Syst. **16**(6), 1393–1410 (2008)
12. Minku, L., Li, F., Inoue, H., Yao, X.: Negative Correlation In Incremental Learning. Journal Natural Computing: An International Journal Archive, Kluwer Academic Publishers Hingham, MA, USA **8**(2), 289–320 (2009)
13. Song, S., Qiao, X., Chen, P.: Hierarchical text classification incremental learning. Neural Inf. Process. LNCS **5863**, 247–258 (2009)
14. M.N. Kapp, R. Sabourin, P. Maupin; Adaptive incremental learning with an ensemble of support vector machines. In: 20th International Conference on Pattern Recognition, pp. 4048–4051 (2010)
15. Laskov, P., Gehl, C., Kruger, S., Muller, K.-R.: Incremental support vector learning: analysis, implementation and applications. J. Mach. Learn. Res. **7**, 1909–1936 (2006)
16. Shilton, A., Palaniswami, M., Ralph, D., Tsoi, A.C.: Incremental training of support vector machines. IEEE Trans. Neural Netw. **16**(1), 114–131 (2005)
17. Polikar, R., Udpa, L., Udpa, S., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. IEEE Trans. Syst. Man And Cybern. (C) Spec. Issue Knowl. Manag. **31**(4), 497–508 (2001)
18. Connolly, J.-F., Granger, E., Sabourin, R.: Supervised Incremental Learning with the Fuzzy ARTMAP. IAPR Workshop on Artificial Neural Networks in Pattern Recognition, LNAI **5064**(2008), pp 66–77 (2008)
19. Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B.: Fuzzy Artmap: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. Neural Netw. **3**, 698–713 (1992)
20. Shen, F., Hasegawa, O.: Self-organizing Incremental Neural Network and Its Application; Artificial Neural Networks (ICANN'10) (2010)

21. Almaksour, A., Anquetil, E.: Fast incremental learning strategy driven by confusion reject for online handwriting recognition. In: 10th International Conference On Document Analysis And Recognition (ICDAR'09), Spain (2009)
22. Almaksour, A., Anquetil, E., Quiniou, S., Cheriet, M.: Evolving fuzzy classifiers application to incremental learning of handwritten gesture recognition. In: International Conference On Pattern Recognition (ICPR'10), Istanbul, Turkey (2010)
23. Muhlbaier, M., Topalis, A., Polikar, R.: Learn++.NC: combining ensemble of classifiers combined with dynamically weighted consult-and-vote for efficient incremental learning of new classes. IEEE Trans. Neural Netw. **20**(1), 152–168 (2009)
24. Erdem, Z., Polikar, R., Gurgen, F., Yumusak, N.: Ensemble Of SVMs For Incremental Learning; Simulation, **3541**, pp. 246–256, Springer (2005)
25. Hamza, H., Belaïd, Y., Belaïd, A., Chaudhuri, B.B.: An end-to-end administrative document analysis system. In Document Analysis Systems (DAS'08), pp. 175–182 (2008)
26. Bouguelia, M.R., Belaïd, Y., Belaïd, A.: Document image and zone classification through incremental learning. In: 20th IEEE International Conference On Image Processing (ICIP'13), pp. 4230–4234 (2013)
27. Ristin, M., Guillaumin, M., Gall, J., Gool, L.V.: Incremental Learning of NCM Forests for Large-Scale Image Classification; Computer Vision and Pattern Recognition (CVPR'14), pp. 3654–3661 (2014)
28. Littlestone, N.: Learning quickly when irrelevant attributes abound: a new linear threshold algorithm. Mach. Learn. **2**, 285–318 (1988)
29. Bifet, A.: Adaptive Stream Mining: Pattern Learning And Mining From Evolving Data Streams. IOS Press Inc, Amsterdam (2010). http://www.iospress.nl/book/adaptive-stream-mining-pattern-learning-and-mining-from-evolving-data-streams/
30. Lazarescu, M., Venkatesh, S., Bui, H.: Using multiple windows to track concept drift. Intell. Data Anal. IOS Press Amsterdam **8**(1), 29–59 (2004)
31. Alippi, C., Roveri, M.: Just-in-time adaptive classifiers in non-stationary conditions, pp. 1014–1019. IJCNN, IEEE, New York (2007)
32. Alippi, C., Boracchi, G., Roveri, M.: Just in time classifiers: managing the slow-drift case, pp. 114–120. IJCNN, IEEE, New York (2009)
33. R. Klinkenberg; Learning drifting concepts: example selection vs. example weighting. Intell. Data Anal. Special Issue On Incremental Learning Systems Capable Of Dealing With Concept Drift, **8**(3) pp. 281–300 (2004)
34. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams classifiers. In: Proceeding of The 9th ACM SIGKDD International Conference, ACM Press, New York, pp. 226–235 (2003)
35. Kolter, J., Maloof, M.: Dynamic Weighted Majority (DWM): An Ensemble Method For Drifting Concepts; JMLR'08, pp. 2755–2790 (2008)
36. Street, W.N., Kim, Y.S.: A streaming ensemble algorithm (SEA) for large-scale classication. In Proceeding of The 7th International Conference On Knowledge Discovery And Data Mining, ACM Press, pp. 377–382 (2001)
37. Oza, N.C.: Online Ensemble Learning; PhD Thesis, University Of California, Berkeley (2001)
38. Nattee, C., Numao, M.: Geometric method for document understanding and classification using online machine learning. In: Proceeding Of The 6th International Conference On Document Analysis And Recognition, Seattle, USA, pp. 602606 (2001)
39. Salles, T., Rocha, L., Pappa, G.L., Mouro, F., Meira, Jr. W., Gonalves, M.: Temporally-aware algorithms for document classification. In: Proceeding of the 33rd International Conference on Research and development in Information Retrieval (SIGIR'10), ACM, New York, NY, USA, pp. 307–314 (2010)
40. Elwell, R., Polikar, R.: Incremental learning in nonstationary environments with controlled forgetting. In: International Joint Conference On Neural Networks (IJCNN 2009), Atlanta, GA, pp. 771–778 (2009)
41. Elwell, R., Polikar, R.: Incremental learning of concept drift in non-stationary environments. IEEE Trans. Neural Netw. **22**(10), 1517–1531 (2011)
42. Bouillon, M., Anquetil, E., Almaksour, A.: Decremental learning of evolving fuzzy inference systems: application to handwritten gesture recognition. Mach. Learn. Data Min. Pattern Recognit. LNCS **7988**, 115–129 (2013)
43. Syed, N., Liu, H., Sung, K.: Incremental learning with support vector machines. In: Proceeding of The Workshop On Support Vector Machines IJCAI'99, Stockholm, Sweden (1999)
44. Rüping, S.: Incremental learning with support vector machines; ICDM01, pp. 641–642 (2001)
45. Sato, J.R., Jane, R., Janaina, M.-M.: Measuring abnormal brains: building normative rules in neuroimaging using one-class support vector machines. Front. Neurosci. **6**, 178 (2012). doi:10.3389/fnins.2012.00178
46. Ngo-Ho, A-K., Ragot, N., Ramel, J-Y., Eglin, V., Sidere, N.: Document Classification in a Non-stationary Environment: a One-Class SVM Approach; ICDAR13, Washington DC, USA (2013)
47. Ngo-Ho, A.-K., Ragot, N., Ramel, J.-Y., Eglin, V., Sidere, N.: Multi one-class incremental SVM for both stationary and non-stationary environment. In: 16th Confrence Francophone sur l'Apprentissage Automatique. Saint-Etienne, France (2014)
48. Ngo-Ho, A.-K., Ragot, N., Ramel, J.-Y., Eglin, V., Sidere, N.: Multi one-class incremental svm for document stream digitization. In: 12th IAPR International Workshop on Document Analysis Systems. Santorini, Greece (2016)
49. Scolkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution; technical report, microsoft research, MSR-TR-99-87 (1999)
50. Kaynak, C.: Methods of combining multiple classifiers and their applications to handwritten digit recognition; Msc. Thesis, Institute Of Graduate Studies In Science And Engineering, Bogazici University (1995)
51. Vinsonneau, E., Domenger, J-P., Cherif, A.: Mesure de la Netteté Sur Une image Seule Dans Des Documents Anciens, CIFED 2014, France (2014)
52. Tong, H., Li, M., Zhang, H., Zhang, C.: Blur detection for digital images using wavelet transform. In: IEEE International Conference on Multimedia and Expo. (ICME04), vol. 1, IEEE, pp. 17–20 (2004)
53. Zhuo, S., Sim, T.: Defocus map estimation from a single image. Pattern Recogniti. **44**(9), 1852–1858 (2011)
54. Lelegard, L., Bredif, M., Vallet, B., Boldo, D.: Motion Blur Detection in Aerial Images Shot with Channel-Dependent Exposure Time; International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS), vol. 38, part 3A, Saint-Mand, France, pp. 180–185 (2010)
55. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**(1–3), 389–422 (2002)
56. Chen, Y., Zhou, X.S., Huang, T.: One-class SVM for learning in image retrieval. In: IEEE International Conference on Image Processing (ICIP'2001), pp. 34–37 (2001)
57. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution; ICPR'10, pp. 3121–3124 (2010)
58. Chang, C.-C., Lin, C.-J.: LibSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011)
59. Zhou, Z.-H., Chen, Z.-Q.: Hybrid decision tree. Knowl. Based Syst. **15**(8), 515–528 (2002)