

# UNIVERSITÉ FRANÇOIS – RABELAIS DE TOURS

*ÉCOLE DOCTORALE MIPTIS*  
LABORATOIRE INFORMATIQUE (EA 6300)

**THÈSE** présentée par :

**Anh Khoi NGO HO**

soutenue le : **19 Mars 2015**

pour obtenir le grade de : **Docteur de l'université François – Rabelais de Tours**

Discipline/ Spécialité : Informatique

## **Méthodes de classifications dynamiques et incrémentales**

### **Application à la numérisation cognitive d'images de documents**

**THÈSE dirigée par :**

**Mr. RAMEL Jean-Yves**

Professeur, Université François – Rabelais de Tours

**RAPPORTEURS :**

**Mr. ANQUETIL Éric**

Professeur, INSA de Rennes

**Mme. VISANI Muriel**

MdC (HDR), Université de La Rochelle

---

**JURY :**

**Mr. BEAUCHET-FILLEAU Patrick**

Invité, Arkhênum

**Mr. DOMENGER Jean-Philippe**

Professeur, Université Bordeaux 1

**Mme. EGLIN Véronique**

MdC (HDR), INSA de Lyon

**Mr. RAGOT Nicolas**

MdC, Université François – Rabelais de Tours

**Mme. VINCENT Nicole**

Professeur, Université Paris Descartes

Pour la Science, cette thèse, et le reste.

# Remerciements

Tout d'abord, je tiens à remercier mes encadrants, M Jean-Yves Ramel, M Nicolas Ragot et Mme Véronique Eglin, d'avoir suggéré une collaboration entre le LI (Laboratoire d'Informatique) de l'Université de Tours et le LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) de l'INSA de Lyon et de m'avoir accordé leur confiance pour la concrétiser par une cotutelle et surtout de m'avoir apporté leur soutien tout au long de ces années de travail.

Ces années ont été très enrichissantes pour moi tant sur le plan de la recherche que sur le plan relationnel. Un grand merci à M Jean-Yves Ramel pour son aide et sa confiance tout au long de cette thèse et aussi pour sa grande disponibilité. Je le remercie aussi pour son soutien et sa compréhension. Deux grands mercis à M Nicolas Ragot et Mme Véronique Eglin pour leur aide, leur enthousiasme, leur sympathie et leurs encouragements. Ils ont su me pousser à toujours mieux faire et ils m'ont ainsi aidé à surmonter les difficultés sur le chemin vers le doctorat.

Merci à M Éric Anquetil, Melle Muriel Visani, M Jean-Philippe Domenger, Mme Nicole Vincent, M Patrick Beauchet-Filleau d'avoir fait partie de mon jury de thèse et de m'avoir aidé, par leurs remarques très pertinentes, à améliorer la qualité de mes travaux.

Merci à tous les enseignants de l'Université de La Rochelle, tout particulièrement M Jean-Christophe Burie, M Jean-Marc Ogier, M Rémy Mullot, Melle Muriel Visani et Mme Karell Bertet qui ont su orienter mon avenir dans la bonne direction. Ce fut un grand plaisir pour moi de travailler avec eux.

Je souhaite également remercier les collègues de l'Université de Tours et l'INSA de Lyon pour tout leur soutien, de loin ou de près et spécialement Frédéric Rayar, Alina Mereuță, Octavio Razafindramanana, Vo Nhat Vinh, Pham The Anh, Ta Quang Chieu, Tanmoy Mondal, Abu-Aisheh Zeina, Alireza Alaei et Louisa Kessi.

Je voudrais remercier mes amis, les personnes extérieures aux LI et LIRIS, qui m'ont accueilli et ont créé un environnement idéal pour des années mémorables, spécialement les membres de l'association Touraine-Vietnam : Françoise et Jean-Jacques Rousselle, Dominique et Pierre-Philippe Leroux, Catherine Levêque, Nguyen Huong, Jocelyne Robin, Nadège Roy, ainsi que Nghiem Thu Trang, Tran Van Tinh, Nguyen Thanh Nhon, Le Tuan Anh, Thai Le Vinh, Nguyen Quoc Vuong, Nguyen Thi.

Merci également à tous ceux que j'oublie mais qui, d'une manière ou d'une autre, m'ont permis de terminer mon travail.

Enfin, merci à mes chers parents, à qui je dois tout dans la vie...

## Résumé

Cette thèse s'intéresse à la problématique de la classification dynamique en environnements stationnaires et non stationnaires, tolérante aux variations de quantités des données d'apprentissage et capable d'ajuster ses modèles selon la variabilité des données entrantes. Pour cela, nous proposons une solution faisant cohabiter des classificateurs one-class SVM indépendants ayant chacun leur propre procédure d'apprentissage incrémentale et par conséquent, ne subissant pas d'influences croisées pouvant émaner de la configuration des modèles des autres classificateurs. L'originalité de notre proposition repose sur l'exploitation des anciennes connaissances conservées dans les modèles de SVM (historique propre à chaque SVM représenté par l'ensemble des vecteurs supports trouvés) et leur combinaison avec les connaissances apportées par les nouvelles données au moment de leur arrivée. Le modèle de classification proposé (mOC-iSVM) sera exploité à travers trois variations exploitant chacune différemment l'historique des modèles. Le modèle mOC-iSVM.AP sélectionne les vecteurs supports selon leur « âge » dans l'historique ; le modèle mOC-iSVM.EP sélectionne les vecteurs supports issus des modèles suffisamment performants, le mOC-iSVM.nB sélectionne les vecteurs supports issus des  $n$ -meilleurs modèles de l'historique. Notre contribution s'inscrit dans un état de l'art ne proposant pas à ce jour de solutions permettant de traiter à la fois la dérive de concepts, l'ajout ou la suppression de concepts, la fusion ou la division de concepts, tout en offrant un cadre privilégié d'interactions avec l'utilisateur. Les expérimentations menées, à la fois en environnement stationnaire et non-stationnaire, fournissent des scores de classification très proches sinon meilleurs que ceux obtenus avec les classificateurs incrémentaux les plus performants du moment. La plupart des autres approches dynamiques étant applicables à des environnements particuliers. Dans le cadre du projet ANR DIGIDOC, notre approche a été appliquée sur plusieurs scénarios de classification de flux d'images pouvant survenir dans des cas réels lors de campagnes de numérisation. Notre modèle a montré son utilisabilité dans des situations où il s'avère nécessaire d'ajouter une catégorie de documents, de modifier la description d'un type de documents, de fusionner ou de diviser des catégories de documents de manière interactive. Ces scénarios ont permis de valider une exploitation interactive de notre solution de classification incrémentale pour classifier des images arrivant en flux afin d'améliorer la qualité des images numérisées.

**Mots clés :** classification dynamique, apprentissage incrémental, one-class SVM, environnement stationnaire et non-stationnaire, images de documents, numérisation.

## Résumé en anglais

This research contributes to the field of dynamic learning and classification in case of stationary and non-stationary environments. The goal of this PhD is to define a new classification framework able to deal with very small learning dataset at the beginning of the process and with abilities to adjust itself according to the variability of the incoming data inside a stream. For that purpose, we propose a solution based on a combination of independent one-class SVM classifiers having each one their own incremental learning procedure. Consequently, each classifier is not sensitive to crossed influences which can emanate from the configuration of the models of the other classifiers. The originality of our proposal comes from the use of the former knowledge kept in the SVM models (represented by all the found support vectors) and its combination with the new data coming incrementally from the stream. The proposed classification model (mOC-iSVM) is exploited through three variations in the way of using the existing models at each step of time. The mOC-iSVM.AP model selects the previous support vectors according to their « age »; the mOC-iSVM.EP model selects the support vectors according to their efficiency, and the mOC-iSVM.nB selects vectors from the  $n$ -best models in the history. Our contribution states in a state of the art where no solution is proposed today to handle at the same time, the concept drift, the addition or the deletion of concepts, the fusion or division of concepts while offering a privileged solution for interaction with the user. The experiments, at the same time on stationary and non-stationary environments, provide very good classification scores close or even better than those obtained with the most successful incremental classifiers at this moment. Furthermore, in contrary to our method, most of the other dynamic approaches are applicable only to particular environments. Inside the DIGIDOC project, our approach was applied to several scenarios of classification of images streams which can correspond to real cases in digitalization projects. Our model demonstrated its usability in situations where it turns out to be necessary to add a category of documents, to modify the description of a document type, to merge or to divide categories of documents. These different scenarios allow validating an interactive exploitation of our solution of incremental classification to classify images coming in a stream in order to improve the quality of the digitized images.

**Keywords** : dynamic classification, incremental learning, one-class SVM, stationary and non-stationary environments, document image classification, digitization.

# Table des matières

Remerciements .....	3
Résumé .....	4
Résumé en anglais .....	5
Table des matières .....	6
Liste des figures .....	11
Liste des tableaux .....	14
Liste des algorithmes .....	16
Liste des équations .....	17
I. Introduction .....	19
1.1. Contexte du travail de thèse : le projet ANR DIGIDOC .....	20
1.2. Problématique de la thèse .....	21
1.3. Notre contribution .....	24
1.4. Organisation du manuscrit .....	26
II. État de l'art de l'apprentissage continu .....	30
2.1. Limites des systèmes et de l'apprentissage statique .....	31
2.2. Systèmes reposant sur un apprentissage continu .....	32
2.2.1. Évolution des concepts avec un nombre de concepts fixe .....	34
2.2.1.1. Évolution des concepts en environnement stationnaire (renforcement de concept) .....	36
2.2.1.1.1. Approches basées sur les Séparateurs à Vaste Marge (SVMs) .....	37
2.2.1.1.2. Approches par arbre de décision .....	39
2.2.1.1.3. Approches connexionnistes (des modèles d'ARTMAP) .....	40
2.2.1.1.4. Approches par Système d'Inférence Floue (SIF) .....	41
2.2.1.1.5. Approches par ensemble de classificateurs .....	43
2.2.1.2. Évolution des concepts en environnement non-stationnaire .....	44
2.2.1.2.1. Approches basées sur la sélection des données .....	45
2.2.1.2.2. Approches basées sur la pondération des données .....	47
2.2.1.2.3. Approches par combinaison de classificateurs .....	49
2.2.2. Évolution de concepts avec changement du nombre de concepts .....	53
2.2.2.1. Ajout de concepts en environnement stationnaire .....	54

2.2.2.1.1.	Approches par Système d'Inférence Floue (SIF) .....	54
2.2.2.1.2.	Approche des modèles d'ARTMAP (Adaptive Resonance Theory).....	55
2.2.2.1.3.	Approche par combinaison de classificateurs (Learn++) .....	55
2.2.2.2.	Ajout de concepts en environnement non-stationnaire .....	56
2.2.2.3.	Fusion/division de concept.....	57
2.3.	Petit bilan terminologique .....	57
2.4.	Conclusion.....	59
III.	Contributions : systèmes dynamiques autour de l'architecture «Multi One-Class Incremental SVM» (mOC-iSVM).....	62
3.1.	Des SVMs binaires incrémentaux aux one-class SVMs incrémentaux .....	63
3.1.1.	Principes et notions élémentaires des SVMs.....	63
3.1.2.	Classification par SVM binaire .....	64
3.1.2.1.	Étape d'apprentissage.....	65
3.1.2.2.	Étape de classification.....	67
3.1.2.3.	Complexité .....	67
3.1.2.4.	Paramétrage.....	67
3.1.2.5.	Classification par SVM multi-classes .....	68
3.1.3.	Limites de la classification dynamique par SVM .....	68
3.2.	Introduction aux SVM one-class Incrémentaux (mOC-iSVM) .....	70
3.2.1.	Classification par SVM One Class.....	70
3.2.1.1.	Étape d'apprentissage.....	71
3.2.1.2.	Étape de classification.....	72
3.2.1.3.	Paramétrage.....	73
3.2.2.	Les classificateurs indépendants multi <i>One-Class</i> SVM incrémentaux (mOC-iSVM) dédiés à l'apprentissage continu.....	75
3.2.2.1.	Principe du multi one-class SVM incrémental standard: mOC-iSVM .....	76
3.2.2.2.	Comparaison du mOC-iSVM à une méthode par ensemble de SVM et au SVM multi-classes <i>1-vs-all</i> .....	78
3.2.2.3.	Évaluation de performances .....	79
3.2.2.4.	Optimisation des paramètres $\nu$ et $g$ .....	81
3.2.2.5.	Algorithme d'apprentissage du mOC-iSVM.....	82
3.2.3.	Évaluation du mOC-iSVM et de sa capacité d'incrémentalité en traitement « par lot ».....	84

3.2.3.1.	Environnement de développement et de tests .....	84
3.2.3.2.	Description du jeu de données de test .....	85
3.2.3.3.	Description du protocole de test.....	85
3.2.3.4.	Interprétation des résultats .....	86
3.2.4.	Évaluation du mOC-iSVM et de sa capacité d'incrémentalité « en ligne ».....	87
3.2.4.1.	Description du protocole testé.....	87
3.2.4.2.	Interprétation des résultats .....	88
3.3.	Adaptations du mOC-iSVM selon la nature de l'environnement (stationnaire ou non) .....	89
3.3.1.	Sélection des vecteurs supports selon l'âge : version mOC-iSVM.AP.....	90
3.3.1.1.	Motivation et principe .....	90
3.3.1.2.	Schéma de principe de l'apprentissage continu du mOC-iSVM.AP .....	92
3.3.1.3.	Algorithme d'apprentissage continu du mOC-iSVM.AP .....	94
3.3.2.	Sélection des vecteurs supports selon la performance : version mOC-iSVM.EP .....	95
3.3.2.1.	Motivation et principe .....	95
3.3.2.2.	Schéma de principe de l'apprentissage continu selon mOC-iSVM.EP ...	96
3.3.2.3.	Algorithme d'apprentissage du mOC-iSVM.EP .....	97
3.3.3.	Sélection des vecteurs supports selon les n-meilleurs classificateurs: version mOC-iSVM.nB .....	97
3.3.3.1.	Motivation et principe .....	97
3.3.3.2.	Schéma de principe de l'apprentissage continu du mOC-iSVM.nB.....	98
3.3.3.3.	Algorithme d'apprentissage continu du mOC-iSVM.nB.....	99
3.4.	Expérimentations et étude des performances des quatre versions de mOC-iSVM	100
3.4.1.	Objectifs .....	100
3.4.2.	Plan de l'étude comparative .....	100
3.4.3.	Évaluation de la capacité d'ajout de concepts en environnement stationnaire ....	103
3.4.3.1.	Description du jeu de données testé .....	103
3.4.3.2.	Description du protocole expérimental .....	103
3.4.3.3.	Interprétation des résultats .....	103
3.4.4.	Évaluation du mOC-iSVM et de ses variantes en environnement non-stationnaire avec changements brusques.....	108

3.4.4.1.	Description du jeu de tests .....	108
3.4.4.2.	Description du protocole expérimental .....	108
3.4.4.3.	Interprétation des résultats .....	109
3.4.5.	Évaluation du mOC-iSVM et de ses variantes en environnement non-stationnaire avec dérive progressive et réoccurrence de contexte .....	114
3.4.5.1.	Description du jeu de tests .....	114
3.4.5.2.	Description du protocole expérimental .....	116
3.4.5.3.	Interprétation des résultats .....	116
3.5.	Conclusion.....	127
IV.	Application à la numérisation cognitive .....	130
4.1.	Des scanners intelligents à la numérisation cognitive.....	131
4.2.	Panorama des méthodes de classification de documents .....	134
4.2.1.	Architecture de la chaîne de traitement.....	134
4.2.2.	Préparation des images.....	135
4.2.3.	De la représentation à la décision.....	137
4.2.3.1.	Approches perceptuelles .....	137
4.2.3.2.	Approches orientées structure et éléments de contenus.....	140
4.2.3.3.	Approches orientées indexation .....	143
4.2.4.	Traitement de flux de documents et limites actuelles .....	147
4.2.5.	Introduction de l'apprentissage continu pour la classification de document .	148
4.3.	Rappel des objectifs du projet DIGIDOC .....	149
4.3.1.	Modes de fonctionnement prévus du système.....	150
4.3.2.	Spécifications et cas d'usages retenus.....	151
4.3.2.1.	Cas d'usage et fonctionnalités d'un scanner cognitif.....	152
4.3.2.1.1.	Cas d'usage Utilisateurs .....	152
4.3.2.1.2.	Cas d'usage fonctionnels .....	153
4.3.2.2.	Classes d'images retenues pour la réalisation du prototype.....	155
4.3.3.	Interfaces et implémentation du prototype.....	157
4.4.	Exploitation des mOC-ISVM pour la numérisation cognitive.....	159
4.4.1.	Sélection des descripteurs d'images.....	159
4.4.1.1.	Liste exhaustive des descripteurs retenus.....	162
4.4.1.2.	Sélection de caractéristiques pertinentes sur une base réduite.....	168
4.4.1.3.	Sélection des descripteurs pertinents par SVM-RFE .....	170

4.4.2.	Réalisation des scénarios « orientés contenus » et résultats obtenus .....	171
4.4.2.1.	Description du protocole expérimental .....	171
4.4.2.2.	Apprentissage incrémental simple .....	172
4.4.2.3.	Apparition de nouveaux concepts au cours du temps .....	174
4.4.2.4.	Extension de Concepts .....	175
4.4.2.5.	Évolution de concept (Concept drift) .....	176
4.4.2.6.	Fusion et division de concepts .....	178
4.4.3.	Procédure en ligne du scanner cognitif .....	179
4.4.3.1.	Simulation du scanner cognitif par lot de données .....	180
4.4.3.2.	Expérimentations du mOC-iSVM et de ses variantes .....	181
4.5.	Conclusion.....	186
V.	Conclusion et perspectives .....	189
5.1.	Conclusion.....	189
5.2.	Perspectives.....	192
	Bibliographie.....	195

## Liste des figures

Figure 1 - Positionnement de notre contribution vis-à-vis de l'état de l'art. ....	26
Figure 2 - Quatre types d'évolution de concept. Dérive de concept soudaine (a), dérive de concept graduelle (b), réoccurrence d'un concept (c), dérive de concept virtuelle (d)....	35
Figure 3 - Apprentissage incrémental avec la méthode de Syed [Syed 1999]. ....	38
Figure 4. Illustration de la méthode de Rüping [Rüping 2001]. ....	38
Figure 5 - Méthode de Cauwenberghs [Cauwenberghs 2001].....	39
Figure 6 - Approches basées sur la sélection des données. ....	45
Figure 7 - Approches basées sur la pondération des données. ....	48
Figure 8 - Mise à jour par combinaison dynamique.....	50
Figure 9 - Mise à jour par restructuration des classificateurs. ....	52
Figure 10 - Relations entre les principaux termes de la littérature et notre terminologie .....	59
Figure 11 - Séparation dans un SVM binaire.....	65
Figure 12 - Principe de la séparation en deux classes par SVM binaire illustrant le principe de Syed : (a) l'ensemble des données complet et leur frontière de séparation (b) uniquement les vecteurs supports et leur frontière de séparation. ....	69
Figure 13 - Illustration de l'algorithme OC-SVM extrait de [Sato 2012].....	73
Figure 14 - Le résultat de SVM une classe en utilisant des paramètres différents. ....	74
Figure 15 - Comparaison des frontières de décision entre <i>one-class</i> SVM (ligne bleue pointillée) et SVM Binaire (ligne noire continue), [Chen 2001] .....	74
Figure 16 - Schéma de la procédure d'apprentissage du mOC-iSVM.....	77
Figure 17 - Classification d'une donnée (notée $n$ ) par mOC-iSVM .....	78
Figure 18 - Étude des performances ( $MAC_i$ et GMA) du mOC-iSVM et d'OC-SVM en trois étapes.....	87
Figure 19 - Courbe d'évolution de la performance du mOC-iSVM avec différentes dimensions de lot de données.....	89
Figure 20 - Principe de la procédure d'apprentissage du mOC-iSVM.AP avec $\alpha = \mathbf{1}$ .....	93
Figure 21 – Principe de la procédure d'apprentissage du mOC-iSVM.AP avec $f = 80\%$ .....	96
Figure 22 – Principe de la procédure d'apprentissage du mOC-iSVM.nB avec $n = 2$ .....	99
Figure 23 - Évolution de performances ( $GA$ ) de 5 classificateurs et de notre proposition de mOC-iSVM (et de ses variantes) pour 4 étapes d'apprentissage. ....	106
Figure 24 – Évolution du seuil définissant les classes de la base SEA au cours du temps....	108

Figure 25 - Illustration des données de l'ensemble d'apprentissage (à gauche) et de l'ensemble de test (à droite) correspondant à l'étape 1 : seuil de 8 (classe 1 en noir, classe 2 en rouge) et 10% de bruit ajouté dans l'ensemble d'apprentissage (à gauche). .....	109
Figure 26 - Evolution des performances ( $GA$ ) au cours du temps sur la base SEA.....	111
Figure 27 - Extraction des données par fenêtres d'échantillonnage fixe sur un damier tournant ( <i>Rotating CheckerBoard</i> ). .....	115
Figure 28 - Quelques résultats de la fenêtre d'échantillonnage (vitesse de dérive constante). Aux étapes $\alpha=0$ et $\alpha=\pi$ , on observe la même relation entre les régions noires et rouges (situation récurrente). .....	115
Figure 29 - Ensemble d'apprentissage (25 échantillons) et l'ensemble de tests (1024 échantillons) dans l'étape numéro 20 de la base CB Constant. ....	116
Figure 30 - Courbe d'évolution des performances ( $GA$ ) sur la base CB Constant .....	119
Figure 31 - Courbe d'évolution des performances ( $GA$ ) sur la base CB Exp .....	121
Figure 32 - Courbe d'évolution des performances ( $GA$ ) sur la base CB Pulse.....	122
Figure 33 - Courbe d'évolution des performances ( $GA$ ) sur la base CB Sinusoïdal.....	123
Figure 34 - Architecture extrait de [Chen 2007] .....	135
Figure 35 - Descripteur structurel de Gordo dans [Gordo 2009] .....	142
Figure 36 – Comparaison dynamique des documents avec la coloration de graphes. [Gaceb 2008].....	143
Figure 37 - Méthode combinant BoW et BoVW proposée par Augereau [Augereau 2014].	146
Figure 38 - Motifs de texture exploités dans [Sarkar 2006].....	147
Figure 39 - Schéma de principe du système cognitif .....	151
Figure 40 - IHM principale du scanner .....	158
Figure 41 - IHM de définition de protocoles.....	158
Figure 42 - IHM de définition de classes .....	159
Figure 43 - Six classes de documents orientées « Types de contenus » .....	169
Figure 44 – Distribution des valeurs des descripteurs retenus .....	170
Figure 45 - Apprentissage incrémental du mOC-iSVM ( $BA_i$ pour chaque classe et $GBA$ ). ..	174
Figure 46 - Apparition de nouveaux concepts au cours du temps .....	175
Figure 47 - Extension de Concept.....	176
Figure 48 - Concept drift de « Textes manuscrits I » (resp. « Textes imprimés I ») vers « Textes manuscrits II » (resp. « Textes imprimés II ») .....	178

Figure 49 - Fusion de « Textes imprimés I », « Textes manuscrits I », « Textes imprimés II », « Textes manuscrits II » vers « Textes imprimés » (I and II), « Textes manuscrits » (I et II).....	179
Figure 50 - Division de « Textes imprimés » (I and II), « Textes manuscrits » (I and II) vers « Textes imprimés I », « Textes manuscrits I », « Textes imprimés II », « Textes manuscrits II ».....	179
Figure 51 - Six classes de contenus.....	181
Figure 52 – Précision globale ( <i>GPR</i> ), sensibilité globale ( <i>GSN</i> ) et spécificité globale ( <i>GSP</i> ) mesurée pour un apprentissage un à un. Il est indiqué sur cette figure les périodes où les données arrivent et leur classe d'appartenance réelle. ....	183
Figure 53 - Courbe d'évolution de la performance ( <i>GBA</i> ) avec différentes dimensions de lots d'apprentissage.....	184
Figure 54 - Courbe d'évolution du nombre de SVs dans le mode « en ligne » .....	184
Figure 55 - Courbe d'évolution de la performance ( <i>GBA</i> ) des versions de mOC-iSVM dans la base de scanner.....	185

## Liste des tableaux

Table 1 - Distribution des exemples par classe de la base UCI ORHDD, [Kaynak 1995] .....	85
Table 2 - Tableau de synthèse des expérimentations conduites.....	102
Table 3 - Comparaison des résultats de mOC-iSVM.EP avec différentes valeurs de $f$ .....	104
Table 4 - Comparaison des résultats de mOC-iSVM.AP avec différentes valeurs de $\alpha$ .....	104
Table 5 - Comparaison des résultats de mOC-iSVM.nB avec différentes valeurs de $n$ .....	104
Table 6 – Performance ( $AC_i$ et $GA$ ) pour l’ajout de concept en environnement stationnaire.	107
Table 7 - Comparaison générale entre les différentes versions de mOC-iSVM. ....	107
Table 8 - Résumé des performances sur la base SEA .....	110
Table 9 - Comparaison des résultats de mOC-iSVM.AP avec différentes valeurs de $\alpha$ .....	112
Table 10 - Comparaison des résultats de mOC-iSVM.EP avec des différentes valeurs $f$ .....	113
Table 11 - Comparaison des résultats de mOC-iSVM.nB avec des différentes valeurs $n$ . ...	113
Table 12 - Résumé des résultats sur CB Dataset.....	118
Table 13 - Résultats du mOC-iSVM.AP avec différentes valeurs de $\alpha$ sur CB Const.....	124
Table 14 - Résultats du mOC-iSVM.AP avec différentes valeurs de $\alpha$ sur CB Pulse.....	124
Table 15 - Résultats du mOC-iSVM.AP avec différentes valeurs de $\alpha$ sur CB Sinus. ....	124
Table 16 - Résultats du mOC-iSVM.AP avec différentes valeurs de $\alpha$ sur CB Exp.....	125
Table 17 - Résultats du mOC-iSVM.EP avec différentes valeurs de $f$ sur CB Pulse. ....	125
Table 18 - Résultats du mOC-iSVM.EP avec différentes valeurs de $f$ sur CB Sinus. ....	125
Table 19 - Résultats du mOC-iSVM.EP avec différentes valeurs de $f$ sur CB Exp.....	125
Table 20 - Résultats du mOC-iSVM.EP avec différentes valeurs de $f$ sur CB Const.....	126
Table 21 - Résultats du mOC-iSVM.nB avec différentes valeurs de $n$ sur CB Constant.....	126
Table 22 - Résultats du mOC-iSVM.nB avec différentes valeurs de $n$ sur CB Exp. ....	127
Table 23 - Résultats du mOC-iSVM.nB avec différentes valeurs de $n$ sur CB Sinus.....	127
Table 24 - Résultats du mOC-iSVM.nB avec différentes valeurs de $n$ sur CB Pulse. ....	127
Table 25 - Synthèse qualitative des classificateurs étudiés dans ce chapitre et de nos quatre approches de classification Multi one-class. ....	128
Table 26 – Scanners sur le marché.....	132
Table 27 - Résultat de l’étude de Watanabe dans [Watanabe 1999].....	140
Table 28 - Quelques indications de performances obtenues par les méthodes citées dans cette section.....	143

Table 29 - Méthodes combinant BoW et BoVW par fusion simple fournie dans [Augereau 2014]. La base de tests utilisée, fournie par la société GestForm, est composée de 1985 documents répartis en 12 classes. 5 documents servent à l'apprentissage, le reste au test en validation croisée.....	146
Table 30 – Synthèse des descripteurs retenus .....	168
Table 31 – Résultat comparatif des approches.....	173
Table 32 – Distribution des données dans la base DIGIDOC pour les tests de performance du mOC-iSVM.....	182
Table 33 – Temps de calcul et performance des versions de mOC-iSVM dans la base de scanner.....	186

## Liste des algorithmes

Algorithme 1 - Algorithme d'apprentissage du mOC-iSVM.....	84
Algorithme 2 - Modification de l'algorithme d'apprentissage pour le mOC-iSVM.AP .....	94
Algorithme 3 - Modification de l'algorithme d'apprentissage pour le mOC-iSVM.EP .....	97
Algorithme 4 - Modification de l'algorithme d'apprentissage pour le mOC-iSVM.nB .....	100

## Liste des équations

Équation 1 .....	79
Équation 2 .....	79
Équation 3 .....	79
Équation 4 .....	80
Équation 5 .....	80
Équation 6 .....	80
Équation 7 .....	80
Équation 8 .....	81
Équation 9 .....	81
Équation 10 .....	81

# Introduction

# I.Introduction

**C***ette introduction présente le contexte général dans lequel se situe ces travaux de thèse, tant d'un point de vue plus applicatif (Projet ANR DIGIDOC) que d'un point de vue théorique (apprentissage et classification dynamique et incrémentale). L'objectif applicatif de cette thèse est d'améliorer ou de simplifier l'étape de numérisation des images de documents en proposant un moteur de « numérisation cognitive » (i.e. intelligent) qui pourrait à terme être intégré aux scanners. Le cœur du scanner cognitif prévu dans DIGIDOC est conçu comme un classificateur dynamique exploitant un module d'apprentissage incrémental et interactif capable de détecter les spécificités d'un document et de choisir les paramètres de numérisation adaptés à l'usage qui sera fait des images de documents. Nous présentons dans ce chapitre les enjeux de la thèse d'un point de vue théorique et expérimental ainsi que les principales contributions proposées pour la classification dynamique, qui seront détaillées dans les chapitres suivants.*

## 1.1. Contexte du travail de thèse : le projet ANR

### DIGIDOC

Le projet DIGIDOC<sup>1</sup> s'inscrit dans le contexte très riche de la numérisation du patrimoine culturel. Le travail de numérisation porté par de nombreux projets ces deux dernières décennies et effectué à l'échelle mondiale au sein de nombreuses institutions et bibliothèques a un double objectif : préserver les documents originaux, particulièrement les documents anciens, et diffuser leur contenu en fournissant des outils de recherche adaptés.

Il est à noter que le nombre de projets de numérisation est en forte croissance depuis quelques années. Au-delà des considérations politiques et des réticences nationales, l'engagement pris par la firme Google et plus récemment par d'autres leaders du monde informatique (Microsoft, Yahoo, Amazon,...) suscite des réactions très diverses qui témoignent des enjeux économiques mais aussi scientifiques qui en découlent. Il y a en effet urgence à numériser le patrimoine écrit afin d'une part, d'en préserver les versions originales, et d'autre part, de satisfaire les besoins croissants de consultation de ce patrimoine. Les besoins en consultation des ouvrages originaux ou anciens augmentent de façon exponentielle. Ces besoins impliquent des manipulations physiques qui mettent en péril le support lorsqu'elles dépassent la faible fréquence permise à ce jour dans les bibliothèques, archives et musées. Il est notable que moins de 10% des ouvrages précieux sont protégés par la reproduction sous forme de microfilms, eux-mêmes fragiles ou de fichiers numériques. Cette urgente nécessité est encore accrue par la destruction progressive, même sans manipulation, des papiers acides qui furent utilisés pendant presque un siècle, du XIXe siècle à la deuxième guerre mondiale. La numérisation est donc une obligation pour sauvegarder le patrimoine et elle doit être menée le plus rapidement possible. Préservation et diffusion du capital culturel vont de pair avec l'élaboration de techniques industrielles efficaces, économiquement viables, pour perpétuer les documents et les rendre accessibles, c'est à dire identifiables, « trouvables » et consultables par quiconque sous une forme efficace et attrayante. Ces nouveaux besoins, encore émergents et encore à inventer, risquent d'exploser avec l'évolution des technologies de diffusion de contenu mais aussi de création de contenu (recomposition d'ouvrage par exemple). L'offensive de Google sur ce point, et la réaction de

---

<sup>1</sup> Programme ANR : CONTenus et INTeractions (CONTINT) 2010. Référence projet : ANR-10-CORD-0020.

l'Europe, montrent toute l'importance du sujet, tant du point de vue de la préservation que de la diffusion au plus grand nombre.

Dans un contexte général où se multiplient les grands projets d'acquisition et de valorisation du patrimoine écrit, le projet DIGIDOC se focalise sur l'étape d'acquisition des images de documents pour améliorer et simplifier leur utilisation ultérieure (archivage, reconnaissance de texte, indexation de documents, etc.). Les coûts de numérisation et la fragilité des documents précieux rendent pratiquement impossible une deuxième numérisation; il est donc impératif de maîtriser la qualité de la numérisation en fonction de l'image et de l'usage ultérieur, dès la première numérisation. La prise en compte de l'usage des documents numérisés s'exprime nécessairement en termes de connaissances et contraintes métier. Le projet DIGIDOC vise donc à conditionner la phase de production des images en considérant à la fois des connaissances *a priori* sur les caractéristiques des documents à numériser et des connaissances sur l'utilisation qui en sera faite.

D'un point de vue pratique, le projet cherche à intégrer au sein des scanners un module d'analyse des images. Celui-ci fournira, en plus de l'image numérisée, un ensemble de descripteurs qui pourront être exploités afin de déduire la nature des contenus et leur qualité. Ainsi, en quantifiant l'adéquation entre le résultat de la numérisation d'un document et son usage ultérieur, il devrait être possible de refaire une numérisation dans la foulée avec de nouveaux paramètres du scanner prenant mieux en compte les spécificités du document et l'usage. Cette innovation vise ainsi à simplifier le paramétrage des scanners (numérisation et prétraitements) en déterminant, de manière semi-automatique, les meilleurs paramètres de numérisation des images. Par ailleurs, ces descripteurs images, types de contenus et de qualité, devraient permettre de faciliter l'ensemble des post-traitements, en identifiant très tôt dans la chaîne de numérisation les spécificités du document.

La particularité du module d'analyse est que celui-ci pourra s'enrichir et s'adapter au cours du temps, en s'appuyant sur les connaissances acquises au fil des numérisations. L'adaptation dynamique du système à la nature des données et aux besoins de l'utilisateur (ces besoins pouvant changer) est un des éléments centraux dont le scanner devra être doté. C'est sur cette adaptation que les contributions de la thèse ont porté, en particulier sur les aspects d'apprentissage et de classification incrémentale.

## 1.2. Problématique de la thèse

Dans le contexte du projet DIGIDOC, cette thèse traite de la problématique de l'apprentissage et de la classification incrémentale afin de produire un « moteur de numérisation » qui sera à terme intégré aux scanners pour aider à définir leur décision en fonction des documents numérisés, et de l'usage que l'on souhaite en faire. L'objectif est donc de mettre en place des mécanismes permettant au processus de numérisation de s'auto-adapter en fonction de la nature du document à traiter ainsi qu'en fonction des objectifs de numérisation définis par l'opérateur. Ces derniers sont directement liés aux traitements futurs auxquels les images seront soumises et aux applications qui les manipuleront : OCR, archivage, recherche par le contenu... Ainsi, à la différence des processus de numérisation habituels qui procèdent en une capture unique à haute résolution, notre ambition est de permettre des acquisitions multiples en fonction des objectifs d'utilisation et du contenu même du document. C'est pourquoi nous parlerons, dans la suite, de numérisation cognitive ou de scanner intelligent. Pour ce faire, l'utilisation de processus d'apprentissage et de classification est envisagée, pour identifier les spécificités des documents (types de contenus, qualité, défauts spécifiques notamment). La difficulté vient alors du fait que ces mécanismes devront être capables de supporter une mise à jour incrémentale et interactive des connaissances sur lesquelles ils se fondent, et des concepts représentés qui pourront également évoluer. Par exemple, on envisage des scénarios tels que :

- la numérisation de documents de même nature et destinés au même usage. Dans ce cas, le modèle associé (concept) se renforcera au fur et à mesure que de nouvelles images seront acquises et les performances permettant de les identifier augmenteront au cours du temps ;

- l'introduction d'un nouveau type d'image ou de défaut (et éventuellement du jeu de paramètres de numérisation associé) ;

- la possibilité de fusionner ou diviser des catégories qui existent déjà.

Ce travail s'inscrit ainsi dans le cadre général de l'apprentissage continu automatique et de la classification supervisée. L'apprentissage envisagé dans nos travaux est une étape requise par le système de classification qui doit posséder une réelle capacité d'évolutivité à travers le temps. La classification est un domaine très actif en recherche dès lors que l'on manipule des données en masse et qu'il est nécessaire de les identifier ou de les catégoriser. Les systèmes de classification de ces dernières décennies reposent souvent sur le principe de la création d'un ensemble de données d'apprentissage sous la supervision d'un expert qui contrôle et optimise le processus d'entraînement. Ensuite, un ensemble de données de test

totalemment indépendant permet de tester l'efficacité des modèles appris. La base d'apprentissage contient des échantillons déjà étiquetés des différentes classes qui doivent être reconnues par le système. Nous ne nous situons pas dans un contexte aussi linéaire rencontré dans les approches de classification supervisée usuelles, car dans notre cas, le classificateur doit être capable de s'adapter au fil du temps aux nouvelles données entrantes (avec toutes les variations qu'elles contiennent), reconfigurant ses modèles avec parfois très peu de données (car elles ne sont pas forcément disponibles en quantité suffisante à un instant donné) et aux situations parfois imprévues (données totalement inconnues ou non conformes aux modèles existants). Ce sont donc des environnements évoluant au fil du temps (évolutions régulières ou non, présentant des variations parfois brutales ou imprévisibles) que ce travail de thèse doit considérer en répondant à de nouveaux défis en terme d'apprentissage continu et de classification incrémentale. La mise en place de cet apprentissage incrémental est un véritable verrou scientifique et le mécanisme proposé doit apporter des réponses aux défis majeurs soulevés par l'apprentissage incrémental et interactif en **environnements stationnaires** et **non-stationnaires**. Il doit également garantir une amélioration progressive des performances de la classification **au cours du temps** grâce à la prise en compte de nouvelles données sans ré-exploiter systématiquement les plus anciennes selon un processus long et coûteux de réapprentissage complet.

Une solution envisageable et pertinente pour ce problème (et c'est l'approche défendue dans ces travaux) nous semble être d'accepter de faire appel à l'utilisateur que ce soit pour valider (implicitement) ou invalider et corriger explicitement les décisions du système, notamment lorsque l'incertitude est trop importante. C'est pourquoi nous parlons également de classification **interactive**. Par exemple, la définition d'une nouvelle catégorie de documents se fera de façon interactive avec l'aide de l'opérateur. Au final, différents scénarios d'apprentissage et de classification incrémentale dépendants de la décision de l'utilisateur par sa capacité à ajouter des nouvelles catégories d'informations (ajout de classes), à aider au renforcement de la classe, à intégrer une évolution voire un changement de description d'une classe, ou encore à guider le système dans les processus de division ou de fusion de classes requis parfois lorsque les données évoluent durablement, seront présentés lors des tests de faisabilité du module de classification supervisé que nous proposons en situation réelle en lien avec les données du projet DIGIDOC. Cette approche de la classification incrémentale ouvre la voie à des domaines d'applications très larges du traitement des données en flux.

Après avoir présenté les éléments fondamentaux de l'état de l'art en matière d'apprentissage et de classification (incrémentale et non incrémentale) au sein desquels s'inscrit la contribution majeure de cette thèse, nous présentons les éléments théoriques fondateurs de notre contribution en termes de classification dynamique et d'apprentissage incrémental. Nous prouvons l'efficacité du processus par des validations expérimentales conduites sur des benchmarks très diversifiés du domaine de l'apprentissage (*Machine learning*). Nous montrons ensuite l'utilisabilité du moteur incrémental sur une chaîne de numérisation en contexte réel impliquant l'utilisateur dans le processus de numérisation cognitive à travers toutes les interactions qu'il est capable de mettre en œuvre avec le scanner (et plus particulièrement le moteur de classification incrémentale).

### 1.3. Notre contribution

Pour répondre au défi majeur de cette thèse, nous proposons une solution performante de classification dynamique (en environnements stationnaires et non stationnaires) tolérante aux variations de quantités des données d'apprentissage et capable d'ajuster ses modèles selon la variabilité des données entrantes. Pour cela, nous proposons une solution faisant cohabiter des classificateurs *one-class* SVM indépendants ayant chacun leur propre procédure d'apprentissage incrémentale et par conséquent, ne subissant pas d'influences croisées pouvant émaner de la configuration des modèles des autres classificateurs. En effet, cette interdépendance s'avèrerait dommageable dans le processus de classification de notre étude où l'on ne connaît à l'avance, ni le nombre de catégories de contenus à traiter, ni les variations internes observables au sein d'une même catégorie. La solution proposée est donc une collection de classificateurs *one-class* incrémentaux que nous nommerons classificateur *multi-one-class* incrémentaux SVM (mOC-iSVM) et qui répondent à l'ensemble de nos demandes avec d'excellentes performances. L'originalité de la proposition repose sur l'exploitation des anciennes connaissances conservées dans les modèles de SVM (historique propre à chaque SVM représenté par l'ensemble des vecteurs supports trouvés) et leur combinaison avec les connaissances apportées par les nouvelles données au moment de leur arrivée. Le modèle de classification mOC-iSVM sera exploité à travers trois variations exploitant chacune différemment l'historique des modèles. Chacune de ces versions dérive du modèle mOC-iSVM standard : le mOC-iSVM.AP sélectionne les vecteurs supports selon leur « âge » dans l'historique ; le mOC-iSVM.EP sélectionne les vecteurs supports issus des modèles

suffisamment performants, le mOC-iSVM.nB sélectionne les vecteurs supports issus des  $n$  meilleurs modèles de l'historique.

Cette contribution s'inscrit dans un contexte où l'état de l'art ne propose pas à ce jour de solutions permettant de traiter à la fois l'incrémentalité de l'apprentissage en environnements stationnaires et non stationnaires, l'ajout de classes, la fusion/division de classes, tout en offrant un cadre privilégié d'interactions avec l'utilisateur. Afin de permettre, en supplément, la gestion de données entrantes en flux (et pas exclusivement en lots), nous proposons dans ce travail:

- Une contribution à inscrire dans le domaine de la classification dynamique, au sein de laquelle nous proposons une réponse portant à la fois sur la capacité du système à **prendre en compte l'évolution des concepts et sur l'évolution du nombre de concepts**. Nous proposons ici une solution unifiée répondant à ces nombreux enjeux de la classification dynamique.
- Une nouvelle approche de classification fondée sur le SVM qui s'est montré être un outil extrêmement efficace par sa capacité à résoudre des problèmes de classification multi-classes selon une approche incrémentale. Notre contribution est l'une des rares relevées à ce jour **reposant sur les SVM *one-class*** pour résoudre des problèmes de classification incrémentale multi-classes, avec la possibilité d'ajouter, fusionner/diviser dynamiquement de nouvelles classes.
- Une approche qui peut traiter à la fois des situations en environnement stationnaire et non-stationnaire avec des scores de classification très proches sinon meilleurs que ceux obtenus avec les classificateurs incrémentaux les plus performants du moment. La plupart des autres approches dynamiques sont applicables à des environnements particuliers (soit stationnaires, soit non-stationnaires mais rarement les deux à la fois). Notre contribution est, à notre connaissance, unique par sa **capacité à traiter de nombreux types d'environnements et de modifications de concepts**.
- Des expérimentations basées sur plusieurs scénarios de la classification de documents pouvant survenir dans des cas réels de traitement en flux : notre modèle a montré son utilisabilité dans des situations où il s'est avéré nécessaire d'ajouter une catégorie de documents, de modifier la description d'une classe, de fusionner ou de diviser des catégories entre elles. Les scénarios ont permis de valider une exploitation de notre solution pour traiter les données « en ligne » (ou en flux) et

également « en lot ». À notre connaissance, il n'existe pas de propositions de classification incrémentale répondant à un cahier des charges aussi large, notamment **sur une chaîne de numérisation ou de dématérialisation de documents traités en flux.**

## 1.4. Organisation du manuscrit

Nous avons choisi d'organiser le manuscrit de thèse en partant des aspects théoriques pour ensuite se focaliser sur leur applicabilité dans le domaine de la classification d'images de documents numérisés.

CHAPITRE 2. L'état de l'art proposé au chapitre 2 met en lumière différentes stratégies d'apprentissages continus permettant de répondre à des problématiques d'apprentissage en environnement stationnaire, non-stationnaire, de dérive de concepts, d'ajout de concepts. Nous essayons dans cette partie de montrer les spécificités des approches selon deux considérations : l'évolution des concepts et l'évolution du nombre de concepts. Ces axes de lecture, nous permettent de dépasser les difficultés liées à la variété et à la « polysémie » des termes et qualificatifs utilisés dans la littérature. Il est important à ce stade d'observer que les approches de classification incrémentale sont souvent dédiées à des situations cloisonnées (notées Figure 1) où les conditions d'évolution des concepts et de leurs nombres sont connues par avance.

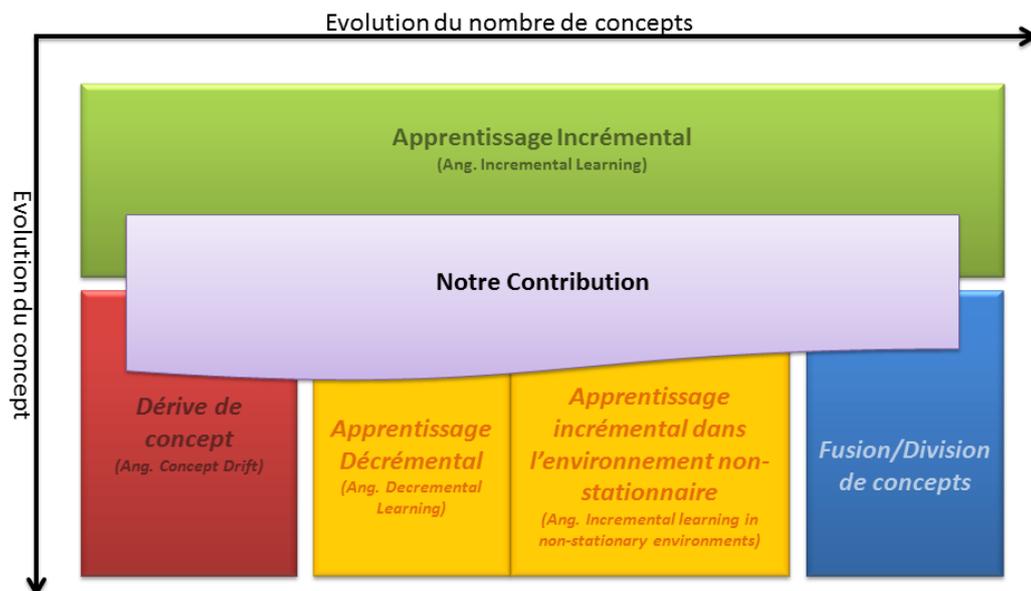


Figure 1 - Positionnement de notre contribution vis-à-vis de l'état de l'art.

CHAPITRE 3. Sur la base de cet état de l'art, nous proposons une méthode unifiée capable de résoudre des problèmes de classification incrémentale dans des situations où les concepts peuvent évoluer à la fois dans leur description (au fil du temps) et dans leur nombre. Cette approche repose sur une classification par combinaison de classificateurs indépendants *one-class* SVM. Le modèle proposé est un classificateur conçu à partir de plusieurs classificateurs *one-class* qui ont chacun la capacité d'évoluer indépendamment des autres (évolution de leur description) et qui autorise l'ajout de nouveaux classificateurs *one-class* sans impact sur les performances des autres. Ce modèle générique (qui est nommé mOC-iSVM, pour classificateur incrémental « *Multi One-Class Incremental SVM* ») est conçu pour permettre la génération de plusieurs versions spécialisées considérant différemment les vecteurs supports qui sont exploités dans les étapes successives d'apprentissage. Le chapitre 3 propose ainsi de démontrer la faisabilité du modèle standard générique du mOC-iSVM pour la classification dynamique en environnements stationnaires et illustre les performances de trois versions spécialisées qui sont développées pour répondre efficacement à des problèmes de classification relevant des environnements à la fois stationnaires et non-stationnaires (mOC-iSVM.AP: sélectionnant les vecteurs supports selon leur ancienneté ; mOC-iSVM.EP: sélectionnant les vecteurs supports selon les performances associées à leurs modèles ; mOC-iSVM.nB: sélectionnant les vecteurs supports selon les performances associées à leurs modèles mais se limitant aux n dernières étapes d'apprentissage).

CHAPITRE 4. Le chapitre 4 ouvre enfin la voie à l'application du modèle proposé dans un scénario d'usage lié à la numérisation en masses de documents par un scanner intelligent. L'intelligence du scanner est liée à la performance du modèle de classification embarqué qui, dans le cas présenté, doit être capable de reconnaître des catégories de documents au fur et à mesure de l'acquisition des images par le scanner. Différentes capacités du système (ajout de classes et évolution de concepts pour des environnements stationnaires et non stationnaires) sont testées sur une base d'images réelles utilisées dans le cadre du projet ANR DIGIDOC. Les interactions avec l'utilisateur sont fortement mises en avant dans cette partie car ce sont elles qui sont responsables de l'introduction des nouvelles connaissances et de la gestion du rejet.

Ce travail jette les bases d'une méthodologie nouvelle pour la classification des documents et ouvre la voie à de nouvelles applications très importantes pour la gestion des grandes masses de données traitées en flux, par lot ou à l'unité.



Première partie  
État de l'art de l'apprentissage continu

## II. État de l'art de l'apprentissage continu

**L'**objectif de cette partie est d'offrir une nouvelle lecture de différents mécanismes d'apprentissage existants afin de les mettre en regard et d'identifier ceux qui permettront d'élaborer le moteur du scanner cognitif. Pour rappel, celui-ci doit pouvoir adapter les paramètres de numérisation en fonction : des propriétés des documents à numériser (en particulier en fonction de leur contenu et de leur qualité) ; de l'usage que souhaite en faire l'utilisateur. Même si des catégories (ou classes) spécifiques de documents peuvent être définies a priori, avec des paramètres de numérisation associés, nous souhaitons que l'usage du scanner soit centré sur l'utilisateur et ses besoins et que ses performances puissent s'améliorer au cours du temps. Cela sous-tend deux choses :

- d'une part que le système soit capable, au fil du temps, d'améliorer sa capacité à distinguer les différentes catégories de documents préexistantes ;
- d'autre part, que le système ait la capacité d'identifier de nouvelles situations et de s'y adapter, que celles-ci soient intrinsèquement liées aux nouvelles données traitées ou bien qu'elles correspondent à un nouveau besoin de l'utilisateur.

Notre problématique se situe donc clairement dans le domaine de la classification et de l'apprentissage, en particulier autour de ce que l'on appelle couramment l'apprentissage en ligne et incrémental. Par souci de généralisation et pour éviter toute ambiguïté, nous utiliserons le terme d'apprentissage continu (attention ne pas confondre avec les méthodes continues/discrètes), qui s'oppose naturellement à l'apprentissage statique. Dans la suite, après avoir rappelé brièvement ce qu'est l'apprentissage statique et quelles en sont ses limites, nous présentons les notions principales autour des mécanismes d'apprentissage continu, en essayant de les structurer afin d'en avoir une vision la plus claire possible. En effet, la littérature est constituée d'exemples et d'outils dont la terminologie est très variable, la rendant peu lisible. Aussi, l'objectif premier n'est pas nécessairement de faire un état de l'art absolument exhaustif mais plutôt d'apporter une vision unificatrice des différents types d'approches avec un positionnement relatif de chacune d'elles, l'ensemble permettant de mieux appréhender nos contributions.

## 2.1. Limites des systèmes et de l'apprentissage statique

Le plus souvent, un système de classification, qu'il soit supervisé ou non, est considéré comme un **système statique** dont le modèle est élaboré par une procédure d'apprentissage. Ce **modèle** du problème constitue la représentation, sous forme de fonctions paramétriques par exemple, des **concepts** relatifs aux classes à identifier ainsi que des règles de décisions associées. Pour cela, des connaissances *a priori* peuvent être apportées par l'utilisateur (nombre de classes/clusters, étiquettes associées au concept de ces classes dans le cas supervisé, espace de représentation, etc.). Les autres connaissances nécessaires sont apportées par la base d'apprentissage. Cette base contient un ensemble de données (étiquetées dans le cas supervisé) représentatives de l'ensemble du problème à traiter. Le plus souvent, cette représentativité doit être statistiquement valide. Par conséquent, toutes les connaissances nécessaires à l'établissement du modèle sont supposées connues ou accessibles avant la procédure d'apprentissage. Le modèle peut donc être déterminé par la procédure d'apprentissage en une seule fois<sup>2</sup>. Une fois le modèle établi, celui-ci n'est plus sensé changer car les concepts modélisés sont considérés comme étant figés. Il sera donc utilisé tel quel pour les prises de décisions ultérieures concernant la classification de nouvelles données non étiquetées (phases de tests ou de production). C'est pourquoi, on considère que dans un système statique, à la fois le modèle et le processus d'apprentissage sont statiques. Si de nouvelles connaissances devaient être prises en compte (connaissances *a priori* ou nouvelles données exemples potentiellement capables d'enrichir le modèle) et que le modèle devait être mis à jour, il faudrait alors recommencer une procédure d'apprentissage complète, en spécifiant notamment une nouvelle base d'apprentissage représentative de ce qui serait considéré comme le nouveau problème à traiter. Le modèle initial est dans ce cas complètement ignoré, que ce soit pour l'apprentissage ou bien pour les classifications ultérieures, et remplacé par le nouveau modèle. Nous obtenons donc une entité différente à chaque fois, quelles que soient ses similarités avec la précédente. On parle de **ré-apprentissage**.

Les limites des approches statiques sont clairement visibles dès lors que toutes les connaissances ne peuvent être accessibles lors de l'apprentissage. Cela peut être pour des

---

<sup>2</sup> Attention, bien qu'un système soit statique, sa procédure d'apprentissage peut être itérative de façon à converger vers le modèle optimal au sens d'une fonction objectif.

raisons calculatoires (durée d'apprentissage, stockage complet des données exemples impossible), ou bien par un manque de connaissances sur le problème et les concepts sous-jacents (absence de connaissances sur une partie du problème, manque de représentativité des données d'apprentissage, évolution du problème de classification et des classes représentées au cours du temps, ou encore évolution de la représentation mentale que ce fait l'utilisateur du problème et donc de ses concepts). Dans tous ces cas, il faut mettre à jour le modèle et, dans le cas de systèmes statiques, l'utilisateur n'aura d'autres choix que de faire des réapprentissage complets (souvent coûteux d'un point de vue calculatoire) après avoir lui-même fusionné et synthétisé l'ensemble des connaissances disponibles sur le problème (connaissances *a priori* et base d'apprentissage). Le système ne peut donc pas bénéficier de l'expérience acquise au cours de son utilisation (apprentissages précédents).

Cette problématique a été assez largement traitée dans la littérature, notamment autour des méthodes d'adaptation de classificateurs mais également et surtout d'apprentissage en ligne et/ou incrémental. Cependant, l'étude de l'état de l'art montre que les notions afférentes au concept d'incrémentalité restent pour le moins ambiguës, se référant tantôt à la gestion des données, tantôt à celle des modèles, ou encore aux concepts des classes, avec des contraintes plus ou moins importantes sur la capacité de la procédure d'apprentissage à prendre en compte les nouvelles connaissances et à s'appuyer sur les anciennes. C'est la raison pour laquelle nous essayons ici d'apporter une vision plus globale et synthétique des différentes approches que l'on regroupe sous le terme d'**apprentissage continu**.

## 2.2. Systèmes reposant sur un apprentissage continu

La catégorisation des approches dynamiques existantes selon les termes usuels est particulièrement délicate, et cela pour deux raisons. Premièrement, ces approches sont très souvent identifiées ou décrites par rapport à une propriété qui est mise en avant, au détriment d'autres qui ne sont parfois même pas mentionnées. Or, ces propriétés des systèmes dynamiques ne sont pas exclusives les unes par rapport aux autres. En particulier, la fréquence de mise à jour du système et la façon dont sont gérées les nouvelles données d'apprentissage sont tout à fait complémentaires de la façon dont le modèle est modifié. Or, le mode de gestion des données d'apprentissage est souvent mis en avant pour qualifier les approches de la littérature, en particulier celui de « en ligne ». Même s'il est important, en particulier du point de vue de l'applicatif, cette terminologie (en) dit finalement assez peu sur l'aptitude du système à modéliser les concepts au cours du temps. Il n'est pas rare non plus de trouver des

méthodes qui travaillent sur différents types d'évolution de concepts alors qu'un seul sera mis en avant (l'ajout par exemple). D'une façon générale, selon les approches (ou les auteurs), une seule propriété semble le plus souvent caractériser les systèmes. C'est une des raisons pour lesquelles il est difficile de catégoriser l'approche de X ou encore de la différencier de celle de Y puisqu'un certain nombre des propriétés peuvent être implicites, ce qui induit une certaine confusion. Par ailleurs, la confusion se trouve accrue par le choix même des termes utilisés pour caractériser les approches. Ces « mots-clés » courts, ont souvent une portée générale et peuvent être compris et employés de différentes manières. C'est typiquement le cas du terme incrémental. Celui-ci est tantôt employé pour caractériser l'aptitude d'un système à construire son modèle au fur et à mesure que les données arrivent (la vision est donc plutôt centrée sur les données en ligne, le terme s'opposant alors au terme statique), tantôt utilisé pour décrire un système capable d'ajouter de nouvelles classes à celles qui existent déjà. De la même façon, les termes adaptatifs ou évolutifs sont tout à fait généraux et ne précisent pas quel type d'adaptation ou d'évolution s'effectue au niveau de la modélisation des concepts. C'est la raison pour laquelle, dans la suite, nous essayons donc de repositionner les principales approches existantes, ainsi que les qualificatifs qui leur ont été associées par rapport à une nouvelle typologie qui repose sur la notion de concept, aspect essentiel permettant de distinguer les systèmes dynamiques. C'est la raison pour laquelle nous proposons de structurer les approches autour de deux axes. Le premier considère l'évolution d'un concept, que celle-ci soit réelle (inhérente au problème ou à la représentation que s'en fait l'utilisateur) ou bien apparente (à cause d'un manque de connaissances, de données). Le second axe considère, quant à lui, l'évolution du nombre de concepts, ainsi que leurs interactions (fusion et division notamment). Bien entendu, l'origine dans ce « repère », *i.e.* l'absence d'évolution sur les 2 axes, représentera un système statique. Un bilan sera proposé dans la section 2.3 permettant de confronter la terminologie existante et notre typologie.

En reprenant notre typologie reposant sur les 2 axes d'études mentionnés ci-dessus, nous pouvons l'enrichir de la façon suivante, constituant ainsi le plan de ce chapitre :

- Évolution des concepts avec un nombre de concepts fixe :
  - Évolution apparente des concepts en environnement stationnaire ;
  - Évolution des concepts en environnement non-stationnaire ;
- Évolution des concepts avec changement du nombre de concepts :
  - Ajout de concepts en environnement stationnaire ;
  - Ajout de concepts en environnement non-stationnaire ;

- Division/fusion de concepts en environnement non-stationnaire.

### 2.2.1. Évolution des concepts avec un nombre de concepts fixe

Le nombre de concepts étant fixe, cette première catégorie se concentre sur l'évolution et donc la modification de la modélisation d'un ou plusieurs concepts. Comme évoqué brièvement plus haut, cette évolution peut être liée soit à une modification réelle des concepts au cours du temps, soit à une évolution apparente de ces derniers, selon la nature de l'environnement dans lequel le système évolue. Le premier cas survient lorsque l'environnement dans lequel on souhaite effectuer la modélisation est **non-stationnaire**, induisant une modification au cours du temps des propriétés intrinsèques de certains concepts. On parle alors de « vraie dérive de concept » (*real concept drift*). Il s'observe généralement par la disparition ou l'apparition de données présentant les dites propriétés. Ce phénomène est illustré sur la Figure 2 (a-b-c) présentant 3 sous-cas typiques [Hoens 2011] :

- la « dérive de concept soudaine » (*sudden concept drift*) (a) : elle correspond à un changement brusque (rapide) du concept à partir d'un instant  $t$  ;
- la « dérive de concept graduelle » (*gradual concept drift*) (b) : elle représente une modification du concept s'effectuant pendant une certaine période de temps ;
- la « réoccurrence d'un concept » (*reoccurring concept drift*) (c) : elle représente un cas particulier de changement de concept tel que le concept redevient à nouveau tel qu'il a été à un instant du passé. La modification peut être brusque ou graduelle.

Le deuxième cas se produit, quant à lui, dans des environnements **stationnaires**. L'apparente évolution des concepts provient alors essentiellement de l'incapacité dans laquelle s'est trouvé le système jusqu'alors de modéliser précisément l'intégralité d'un ou plusieurs concepts. On parle alors de « dérive de concept virtuelle » (*virtual concept drift*) (Figure 2, d). Cette incapacité elle-même provient essentiellement : soit de l'impossibilité du système à traiter toutes les données existantes (limitations mémoire et complexité calculatoire) ; soit de l'incomplétude des données observées jusque-là et donc de l'impossibilité d'observer le(s) concept(s) dans son (leurs) intégralité(s) à un moment donné. Ce phénomène se traduira donc le plus souvent par l'observation de données présentant soit des propriétés non observées jusque-là, soit par le renforcement (lié à la fréquence d'apparition) de propriétés existantes. D'une certaine façon, on peut donc considérer que le

cas stationnaire (limité à une extension ou un renforcement de « parties » de concepts) est un cas particulier du cas non-stationnaire (ce dernier incluant en plus l'affaiblissement voire la disparition de « parties » de concepts).

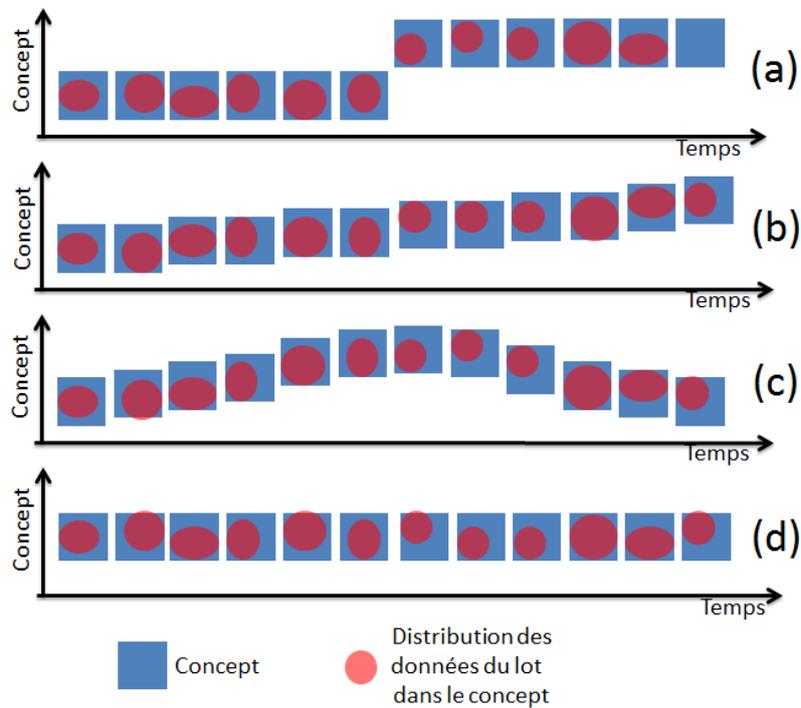


Figure 2 - Quatre types d'évolution de concept. Dérive de concept soudaine (a), dérive de concept graduelle (b), réoccurrence d'un concept (c), dérive de concept virtuelle (d)

Dans les deux situations, si l'on considère le problème de la modélisation, le résultat de l'évolution correspond à une modification de la description du concept dans l'espace de représentation. Du point de vue Bayésien (et donc dans le cadre de la classification supervisée), cela correspond à une modification de la distribution de la probabilité jointe de la classe et des observations. Dans le cas non supervisé, il s'agira plutôt de la modification de la couverture d'un ou plusieurs clusters. Mais l'évolution peut également porter sur le poids relatif du ou des concepts qui évoluent. C'est alors la probabilité *a priori* qui s'en trouvera modifiée dans le cadre Bayésien. Quoi qu'il en soit, cette évolution impactera le processus de décision, parfois au détriment d'un ou plusieurs autres concepts, aboutissant le plus souvent à une modification des frontières de décision (distribution *a posteriori* dans le cadre Bayésien).

En conclusion, on notera que bien que les deux situations soient relativement proches l'une de l'autre selon les éléments présentés ci-dessus, il subsiste un problème fondamental dès lors que l'on ne sait pas si l'on se trouve dans un environnement stationnaire ou non-stationnaire (ou de façon similaire lorsque l'on se trouve dans une période stationnaire d'un

environnement non-stationnaire). En effet, on se trouve alors confronté à ce que l'on appelle « **le dilemme stabilité/plasticité** » : alors que dans le cas stationnaire on veillera à conserver la meilleure stabilité, voire à ne pas oublier ce qui a été appris, la plasticité est au contraire essentielle pour s'adapter efficacement aux variations dans le cas non-stationnaire.

Dans les sous-sections suivantes, nous allons décrire plus précisément les différentes approches de la littérature permettant de gérer l'évolution de concepts dans les deux types d'environnements.

#### 2.2.1.1. Évolution des concepts en environnement stationnaire (renforcement de concept)

En environnement stationnaire, les concepts n'évoluent pas réellement au cours du temps. Seul l'échantillonnage des données qui les représentent peut être amené à changer, induisant par là même une apparente modification de la distribution du concept sous-jacent. Dès lors que ce cadre est connu, la problématique est donc très semblable à ce que l'on appelle classiquement l'apprentissage en ligne (qui ne représente en fait, comme évoqué plus haut, qu'un cas lié au mode de mise à jour du modèle). L'idée générale est donc de construire, à partir des données reçues au cours du temps, un modèle si possible identique à celui qui aurait utilisé toutes les données lors d'un apprentissage statique. Les systèmes devant fonctionner dans ce genre de situation doivent donc posséder la capacité de renforcer les connaissances déjà acquises par les nouvelles connaissances issues des données. C'est pourquoi certains auteurs utilisent également la notion d'apprentissage (ou système) incrémental pour ce genre d'approches.

Pour pouvoir apprendre au cours du temps des concepts d'un environnement stationnaire, on serait tenté de croire que les systèmes reposant naturellement sur un apprentissage en ligne, comme ceux basés sur les méthodes de descente de gradient, sont de bons candidats. En particulier, les réseaux de neurones artificiels (dont les perceptrons multi-couches) utilisant la rétro-propagation du gradient semblent tout à fait adaptés par nature. En fait, ce n'est sans doute pas tout à fait le cas, puisqu'il est bien connu que ces méthodes sont sensibles à l'ordre de présentation des données et que pour obtenir un bon résultat, en apprentissage statique, il est recommandé de faire plusieurs passes d'apprentissage en présentant les données dans un ordre aléatoire. On peut donc en déduire que leur usage, en apprentissage continu, peut poser des problèmes de stabilité (qui peut être amoindri par l'usage d'un *momentum* pour la régularisation) voire d'oubli, non souhaitable, si les données

reçues, pendant un certain temps, venaient à ne plus représenter une partie du concept. L'usage des moindres carrés récurrents semble alors une meilleure alternative. Cependant, dans les deux cas, il subsiste le problème du choix des hyper-paramètres (nombre de couches, de neurones par couche, etc.). En effet, en apprentissage statique, ces hyper-paramètres sont généralement obtenus grâce à une procédure de validation croisée permettant de choisir la meilleure configuration qui restera valable pour toute la modélisation. Il semble évident que ce genre de procédure n'est pas toujours applicable à cette fin en apprentissage continu, ou du moins qu'elle possède des limites claires, l'optimisation n'étant souvent possible que sur une fenêtre temporelle réduite. Bien que cette gestion des hyper-paramètres soit spécifique aux réseaux de neurones, leur impact sur les performances peut être très important puisqu'ils gèrent la structure même du classificateur. Sans en faire la cause unique, ces constatations sont certainement parmi les raisons qui font que les approches connexionnistes sont finalement assez peu nombreuses dans ce domaine. La recherche s'est donc également orientée vers d'autres méthodes d'apprentissage continu applicables à différents classificateurs. Finalement, les approches que nous présentons dans la suite, permettant de gérer explicitement l'apprentissage continu en environnement stationnaire, sont celles basées sur les classificateurs suivants : *Séparateurs à Vaste Marge (SVM)*, *Arbres de décision*, *Modèles connexionnistes ARTMAP*, *Systèmes d'Inférence Floue (FIS)*, et *Ensemble de classificateurs*.

#### 2.2.1.1.1. Approches basées sur les Séparateurs à Vaste Marge (SVMs).

Un certain nombre d'approches ont été proposées autour des SVMs : [Rüping 2001], [Ralaivola 2001], [Cauwenberghs 2001], [Laskov 2006], [Karasuyama 2010]. La première véritable approche d'apprentissage continu pour SVM, qualifiée d'incrémentale, a été proposée dans [Syed 1999]. Le principe de cette méthode repose sur le fait que le modèle d'un SVM, obtenu en utilisant seulement comme données d'apprentissage les vecteurs supports résultant de l'apprentissage, est similaire à celui construit en utilisant tout l'ensemble d'apprentissage. Ainsi, Syed propose une approche d'apprentissage, qualifiée d'incrémentale, permettant de ne pas conserver toutes les données d'apprentissage au cours du temps. Seuls les vecteurs supports sont conservés pour représenter les anciennes données. Ainsi, le système de Syed est conçu comme suit : étant donné un nouvel ensemble de données, le nouveau SVM est obtenu par la procédure d'apprentissage habituelle en utilisant comme base d'apprentissage les nouvelles données (qui représentent les nouvelles connaissances) ainsi que

les vecteurs supports provenant de l'étape d'apprentissage précédente (qui représentent les anciennes connaissances). La Figure 3 illustre la méthode de Syed.

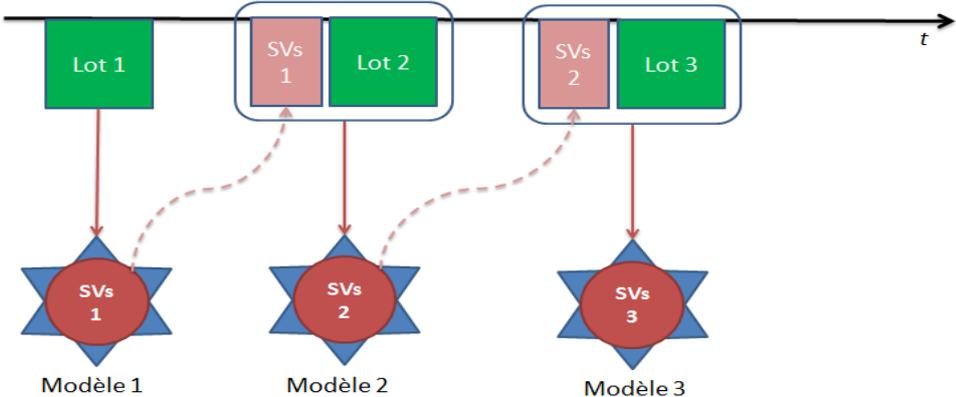


Figure 3 - Apprentissage incrémental avec la méthode de Syed [Syed 1999].

Le problème avec l'approche de Syed, lorsque l'on considère un environnement stationnaire, est sa trop grande plasticité [Rüping 2001]. Dans [Rüping 2001], l'auteur propose une solution en ajoutant des poids aux données d'apprentissage et donc aux vecteurs supports. De cette façon, on peut mettre un poids beaucoup plus faible aux nouvelles données qu'aux vecteurs supports. Ainsi, le système aura tendance à mieux conserver les anciennes connaissances qui pourraient être perdues au cours des apprentissages successifs. En particulier, la méthode tend à renforcer le poids des vecteurs supports dans la base d'apprentissage lorsque ceux-ci sont sous-représentés par rapport aux nouvelles données. La Figure 4 illustre cette méthode (la pondération des données est représentée par la taille relative des données et des vecteurs supports).

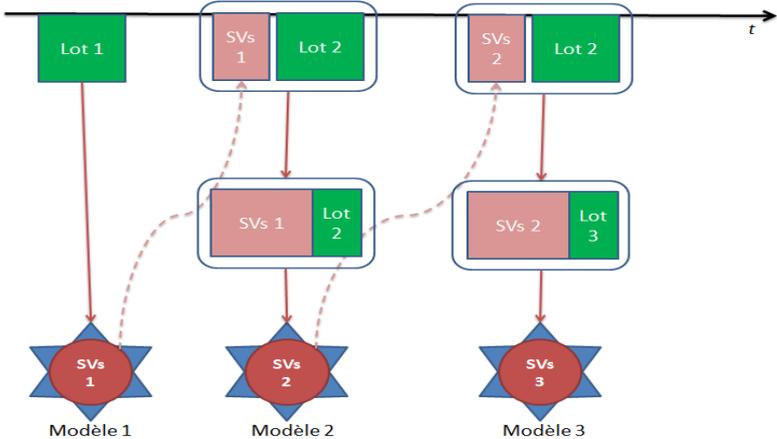


Figure 4. Illustration de la méthode de Rüping [Rüping 2001].

De leurs côtés, [Cauwenberghs 2001], [Laskov 2006], [Shilton 2005] et [Karasuyama 2010] remplacent la fonction d'apprentissage classique du SVM, par une méthode basée sur une marge supérieure (Ang. *Exceeding-margin technique*). L'idée principale est de vérifier si les nouvelles données qui arrivent dans le flux se trouvent au-delà de la marge définie par le modèle SVM. Cette vérification peut être basée sur les conditions de Karush-Kuhn-Tucker (KKT) qui servent habituellement à déterminer si une donnée fait potentiellement partie des futurs vecteurs supports ou non. Si c'est le cas, ces données seront regroupées avec les anciennes données pour lancer l'apprentissage (voir Figure 5). De façon similaire, Ralaivola dans [Ralaivola 2001] modifie la fonction d'apprentissage des SVMs de façon à prendre en compte un taux d'erreur estimé localement. Ce score est utilisé pour décider quand et où les vecteurs supports doivent être remplacés par un nouveau. En résumé, l'idée principale des travaux de Cauwenberghs, Laskov, Shilton, Karasuyama et Ralaivola est la suivante : en conservant au mieux les anciennes connaissances (représentées par les vecteurs supports), on essaie de ne sélectionner qu'assez peu de données parmi les nouvelles connaissances pour mettre à jour le système, au lieu d'utiliser toutes les nouvelles données. Ce principe a pour effet de réduire l'impact d'une distribution très différente dans les nouvelles données, par rapport à celle observée jusqu'alors. On augmente donc la stabilité au détriment de la plasticité, ce qui est souhaitable dans un environnement stationnaire.

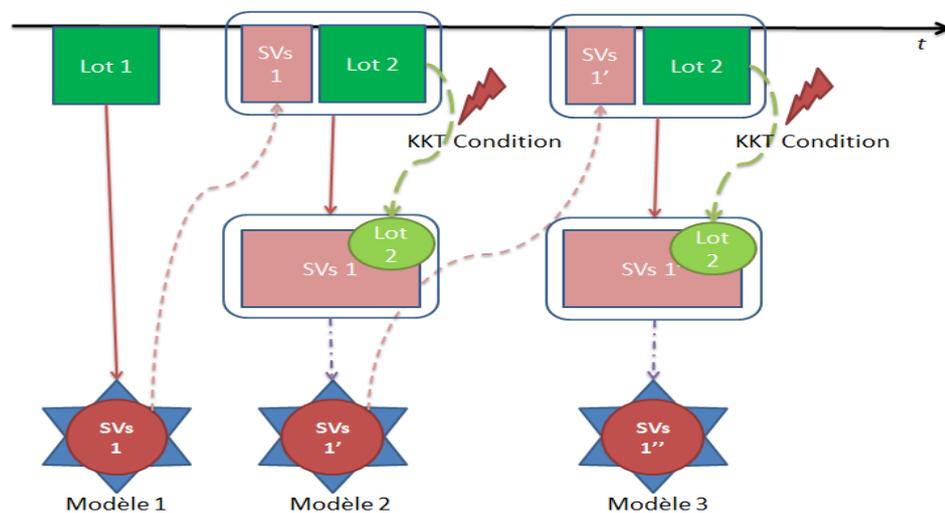


Figure 5 - Méthode de Cauwenberghs [Cauwenberghs 2001]

On notera que ces approches à base de SVM n'ont pas ou peu été testées pour faire de l'apprentissage continu sur plus de deux classes (usage de SVM multi-classes).

#### 2.2.1.1.2. Approches par arbre de décision

Dans [Domingos 2000] avec VFDT (*Very Fast Decision Tree*), l'auteur propose une méthode pour construire de manière incrémentale un arbre de décision de type C4.5 [Quinlan 1993] à partir d'un flux de données. Pour la construction de l'arbre, il faut identifier l'attribut qui doit être utilisé pour diviser une feuille, sans utiliser toutes les données disponibles. L'auteur prouve que l'attribut choisi en appliquant les *bornes de Hoeffding* sur un petit jeu de données, est le même que celui qui le serait en utilisant toutes les données. Cela permet la construction d'un arbre de décision presque identique à celui qui serait obtenu avec toutes les données. L'arbre est construit par un remplacement récursif des feuilles. La procédure est la suivante : pour le premier lot d'apprentissage, construire un arbre initial. Chaque nœud correspond à un attribut de séparation avec une valeur de décision. Les feuilles correspondent à l'étiquette d'une classe. Pour chaque exemple de chaque nouveau lot de données, s'il peut traverser l'arbre et atteindre une feuille avec la bonne étiquette, on ne fait rien. Sinon, on calcule pour chaque feuille la probabilité  $G$  de choisir un attribut correct grâce aux bornes de *Hoeffding*. Ensuite, on choisit la feuille qui a la valeur de  $G$  maximale pour qu'elle soit divisée selon cet attribut. L'arbre va ainsi continuer à se construire au cours du temps en fonction des nouvelles connaissances.

#### 2.2.1.1.3. Approches connexionnistes (des modèles d'ARTMAP)

Une des approches incrémentale les plus célèbres est sans doute celle de [Carpenter 1992] : Fuzzy ARTMAP. Cette méthode supervisée est une extension de l'approche neuronale non-supervisée ART (Adaptive Resonance Theory). ARTMAP peut générer un nouveau cluster pour chaque pattern trouvé dans la base et puis établir la connexion entre ce cluster et une classe. Si un pattern est suffisamment différent des autres patterns connus dans le système (selon un paramètre de vigilance), ce nouvel élément va être ajouté dans le système. La structure ARTMAP contient 3 modules ARTa, ARTb et inter-ART : chaque module est une carte auto-organisatrice avec des nœuds (clusters). ARTa reçoit les entrées et contient deux fois plus de nœuds que d'attributs. ARTb est une couche décrivant des concepts intermédiaires. Enfin, inter-ART contient les classes. Un ensemble de connexions  $W_{ij}$  relie les modules ARTa et ARTb. Un ensemble de connexions  $W^{ab}$  lie les modules ARTb et inter-ART. Quand une donnée arrive, elle stimule, via ARTa, les nœuds d'ARTb. Le plus activé (d'après la loi de Weber) va à son tour activer une classe dans inter-ART pour produire la classification. Cette activation ne peut se faire que si le nœud d'ARTb passe un test de vigilance basé sur un paramètre  $\rho$ , sinon cette connexion est désactivée et le système cherche le deuxième meilleur nœud dans ARTb. Si un nœud ARTb est désactivé alors qu'il ne le

devrait pas, le paramètre de vigilance  $\rho$  est mis à jour pour devenir plus souple. Si aucun nœud n'est activé, on ajoute dans ARTb un nouveau nœud J et on lui assigne la classe correcte de l'exemple dans inter-ART. Ce qui est intéressant dans cette structure est la séparation de la couche «classe» en deux couches différentes ARTb (qui contient 3es clusters) et inter-ART (qui contient les étiquettes): on transforme ainsi la projection de l'espace global d'une classe en un ensemble de sous-espaces (de clusters/prototypes). Ainsi, chaque modification de classe n'affecte que l'une des sous-espaces de la classe, en conservant les autres sous-espaces de cette classe. En ajoutant ou supprimant ces sous-espaces de la classe (lors de l'arrivée de chaque donnée), il est possible, finalement, d'influencer la décision. Un ajout de nœud correspond également à une extension de l'élargissement de l'espace du concept dans l'espace de représentation de classe, et une suppression de nœud est équivalente à une diminution de l'étendue de l'espace du concept de classe. Ce mécanisme modifie le concept en utilisant l'information de chaque exemple dans le flux de données. La difficulté dans cette approche est relative à la détermination du paramètre de vigilance qui contrôle la création de nouveaux nœuds. Ce paramètre devient parfois très sensible au bruit dans les données. En plus, on remarque que ce mécanisme a tendance à produire trop de clusters (nœuds) ce qui cause un sur-apprentissage [Polikar 2001], [Almaksour 2010]. Par contre, cette méthode est bien connue pour sa capacité à ajouter des classes (*cf.* section 2.2.2.1.2).

Il existe plusieurs variantes autour d'ARTMAP, notamment celles permettant de modifier la forme des clusters utilisés en interne pour mieux décrire les données. Alors que ces derniers sont des hyper-rectangles dans ARTMAP, [Williamson 1996] propose d'utiliser des gaussiennes. [Anagnostopoulos 2001] propose d'utiliser des hyper-ellipsoïdes. Dans [Su 2006] et [Tontini 1998], le module ARTb est remplacé par une couche type RBF (*Radial Basis Function*).

#### 2.2.1.1.4. Approches par Système d'Inférence Floue (SIF)

Les systèmes d'inférence floue dans [Mouchère 2005] sont des SIFs d'ordre un – dits de Takagi Sugeno. Ce classificateur se base sur un ensemble de  $r$  règles floues de la forme : Règle<sup>i</sup>: Si  $x$  appartient à  $C^i$ , alors  $y_k^i(x) = (l_1^i(x), \dots, l_c^i(x))$  où  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^c$  et avec :  $x$  le vecteur de caractéristiques d'une entrée ;  $y_k^i$  le vecteur d'appartenance aux  $c$  différentes classes ;  $C$  un cluster défini dans l'espace des entrées et caractérisé par un centre  $\mu^i$  et une zone d'influence définie par une matrice de covariance  $\Sigma^i$  ;  $l_c^i(x)$  représente le degré d'appartenance de  $x$  à la classe  $c$  selon la règle  $i$ . Quand  $x$  arrive, pour trouver la classe de  $x$ ,

son degré d'appartenance à chaque prototype est calculé par la distance de Mahalanobis  $d_{\Sigma^i}$  entre ce point  $x$  et le centre du prototype  $\mu^i$ :  $\beta^i(x) = \frac{1}{1+d_{\Sigma^i}(x,\mu^i)}$ . La sortie du système, pour chaque classe, est calculée par :  $y_k(x) = \sum_{i=1}^r \beta^i(x) \cdot y_k^i(x)$  avec  $r$  le nombre de règles. La décision finale est la classe  $k$  pour laquelle la sortie est maximale :  $\text{argmax}_k y_k(x); k = 1, \dots, c$ . L'apprentissage incrémental dans un SIF permet de mettre à jour les prototypes (leurs centres  $\mu$  et leurs matrices de covariance  $\Sigma$ ) à l'arrivée de chaque nouvelle donnée. Ce système est relativement plastique : les prototypes seront recentrés, déplacés ou déformés systématiquement de façon à s'adapter à chaque donnée. Cette plasticité peut être un problème pour des applications nécessitant une bonne stabilité comme celles de l'apprentissage de l'écriture. On notera également que ce type d'approches permet d'ajouter de nouvelles classes au cours de l'usage. Ces méthodes seront décrites dans la section 2.2.2.1.1. Les systèmes d'inférence floue sont, sous certaines conditions, équivalents à des réseaux de neurones à fonction radiale [Hunt 1996].

Les variantes de cette approche sont présentées très clairement dans [Bouillon 2012]. On peut les décrire brièvement ci-dessous :

- FLEXFIS (FLEXible Fuzzy Inference System) dans [Lughofer 2008] est un système d'inférences floues de Takagi-Sugeno d'ordre un avec une structure de conclusion linéaire. Cet algorithme utilise les principes de quantification vectorielle de manière incrémentale pour mettre à jour les prémisses des règles d'inférences. Le principal inconvénient de cette méthode de mise à jour des prémisses vient du fait qu'elle nécessite un seuil de distance qui contrôle l'équilibre entre la mise à jour des clusters existants et la création de nouveaux.
- EFuNN (Evolving Fuzzy Neural Networks) dans [Kasabov 2001]. est un système de classification floue, composé de règles associant une hyper-sphère de l'espace d'entrée à une hyper-sphère de l'espace de sortie. L'auteur propose qu'une nouvelle règle soit créée lorsqu'une nouvelle donnée d'apprentissage n'active pas de règle au-dessus d'un certain seuil (une nouvelle règle est également créée lorsqu'une donnée d'apprentissage active une règle mais que l'erreur dépasse un autre seuil).
- DENFIS (Dynamic Evolving Neural-Fuzzy Inference System) dans [Kasabov 2002] est un système d'inférence floue de type Takagi-Sugeno d'ordre un où la fonction d'appartenance floue est la fonction triangulaire. L'auteur utilise ce système pour faire de la prédiction dynamique de séries temporelles.

- eTS (Evolving Takagi-Sugeno) dans [Angelov 2004] est un système d'inférences floues de Takagi-Sugeno d'ordre un qui ajuste les règles d'inférences pour s'adapter au flux de données. L'ajustement du système se fait à plusieurs niveaux : a) les prémisses des règles existantes sont mises à jour à l'arrivée de nouvelles données ; b) de nouvelles règles sont créées si les nouvelles données ne correspondent pas aux prémisses des règles existantes ; c) les paramètres des conclusions des règles d'inférence sont également mis à jour ; d) la mise à jour des prémisses et la création de règles sont faites par une méthode de *clustering* basée sur la densité. L'avantage de cette solution est qu'elle ne nécessite pas l'ajustement de paramètres dépendant du problème.

Ces approches (notamment FLEXFIS, EFuNN, DENFIS) dépendent de seuils, qui sont des paramètres que l'on doit ajuster pour chaque problème [Bouillon 2012]. Par contre, ces méthodes sont bien connues pour leur faculté à permettre l'ajout de classes qui sera un point important dans la section 2.2.2.1.2.

#### 2.2.1.1.5. Approches par ensemble de classificateurs

Assez intuitivement, les approches à base de combinaison de classificateurs semblent également pouvoir répondre à la problématique de l'apprentissage continu en environnement stationnaire. En effet, dans certains de ces systèmes, un paradigme consiste à considérer que chaque classificateur de l'ensemble est un expert pour modéliser une « partie » des concepts. Ainsi, si l'on apprend plusieurs classificateurs au cours du temps, chacun va se focaliser sur une partie des concepts en fonction des informations portées par les données utilisées et les mémoriser. L'ensemble permettra donc de capitaliser les connaissances mémorisées au cours du temps et donc d'apprendre les concepts. Ces systèmes semblent donc adaptés à l'apprentissage en environnement stationnaire. Cependant, la plupart sont bien plus souvent mentionnés et adaptés pour les environnements non-stationnaires. Nous les décrivons donc en détail, plus loin dans la section 2.2.1.2.3.

Seules deux approches sont explicitement citées pour résoudre le problème de l'apprentissage en environnement stationnaire sans modification du nombre de concepts. La première est celle présentée par Minku dans [Minku 2009]. Elle repose sur l'apprentissage par corrélation négative (*Negative Correlation Learning* : NCL) proposé initialement dans [Liu 1998] et appliqué à un ensemble de réseaux de neurones. Cet apprentissage utilise un paramètre de contrôle permettant d'avoir un compromis entre la performance en termes de

classification (via la moyenne de l'erreur quadratique) et la diversité de l'ensemble (via une mesure de pénalité). Ainsi, chaque réseau de neurones appris de cette façon maximise sa différence de modélisation avec les autres pour mieux capturer la diversité des concepts modélisés. Pour l'apprentissage continu, Minku propose deux variantes. La première, FSNCL (Fixed Size NCL), modifie l'ensemble des réseaux de neurones existant à chaque nouveau lot de données reçu. Cet ensemble est alors appris à partir des poids précédents pour s'adapter aux nouvelles données. Le nombre de réseaux de neurones dans l'ensemble est fixe, seuls les poids changent. On remarquera ici que, bien que l'auteur présente cette approche comme étant une solution aux environnements stationnaires, elle semble bien plus adaptée aux environnements non stationnaires. En effet, comme évoqué plus haut, les poids des réseaux s'adaptent systématiquement aux nouvelles données, ce qui peut engendrer une perte d'information si certains « neurones » ne sont pas activés pendant un certain temps. Ainsi, seul le paramètre gérant la vitesse d'apprentissage et donc la plasticité/stabilité, peut limiter cet impact. La seconde version, GNCL (Growing NCL), apprend de la façon suivante : à chaque fois qu'un nouveau lot de données arrive, un nouveau réseau de neurones est ajouté à l'ensemble. Ce réseau est appris sur les nouvelles données à partir de poids initiaux nuls. L'inconvénient de cette approche est que le nombre de réseaux de neurones peut devenir très grand assez rapidement.

Notons toutefois que certaines approches dédiées à l'apprentissage en environnement stationnaire permettent également de faire évoluer le nombre de concepts. La plus connue est sans doute l'approche Learn++ proposée par Polikar dans [Polikar 2001] qui sera décrite en détails dans la section 2.2.2.1.3.

### 2.2.1.2. Évolution des concepts en environnement non-stationnaire

Pour parler de ce cas, la littérature utilise abondamment le terme de **dérive de concept** (*Concept drift*). Malheureusement, du moins en français, ce terme est lourd de sens et d'a priori et c'est la raison pour laquelle nous ne souhaitons pas ici le reprendre pour notre propre typologie. En effet, la notion de dérive sous-tend à la fois une modification lente, continue et un oubli inexorable de la situation précédente. Or la difficulté en environnement non-stationnaire vient justement du fait que, selon les problèmes, les évolutions peuvent être tantôt très rapides, tantôt lentes, parfois avec oubli, parfois sans (il arrive même parfois que des connaissances réapparaissent après avoir momentanément disparu). C'est bien dans ces

cas complexes que le dilemme stabilité/plasticité prend tout son sens. Il faut également noter que ces approches ne sont pas qualifiées «d’approches incrémentales». Dans la littérature, [Tsymbal 2004] répertorie trois différents types d’approches proposant une solution à ce problème : *les approches basées sur la sélection des données, celles basées sur la pondération des données, et enfin celles basées sur les ensembles de classificateurs.*

#### 2.2.1.2.1. Approches basées sur la sélection des données

La prise en compte de l’évolution des concepts en environnement non-stationnaire peut se faire grâce à l’usage de « fenêtres temporelles » (*sliding windows*) comme l’approche FLORA [Widmer 1996]. Le principe consiste à mettre à jour le modèle à chaque moment  $t$  en utilisant les données d’apprentissage les plus récentes, ce qui est défini par une fenêtre temporelle de taille à déterminer (soit sur l’échelle du temps, soit en nombre de données). Cette approche peut soit faire un réapprentissage de type «*batch*» (sur les données sélectionnées par la fenêtre temporelle)<sup>3</sup>, soit faire une mise à jour du modèle si une méthode d’apprentissage en ligne le permet. Dans ce cas, l’oubli est géré automatiquement par cette méthode d’apprentissage. Ce type de méthodes consiste généralement en 3 étapes [Bifet 2007] : 1) détecter les changements de concepts en utilisant des tests statistiques sur différentes fenêtres ; 2) si il y a un changement observé, choisir les exemples représentatifs et récents pour adapter les modèles ; 3) mettre à jour les modèles. La dimension de la fenêtre est déterminée *a priori* par l’utilisateur. La Figure 6 illustre cela avec 5 lots de données traités successivement avec une fenêtre d’analyse de taille fixe et égale à 2 lots. Chaque fenêtre recouvre la précédente en partageant un lot de données. À chaque étape, on apprend un nouveau modèle qui correspond à un ensemble de classes mises à jour.

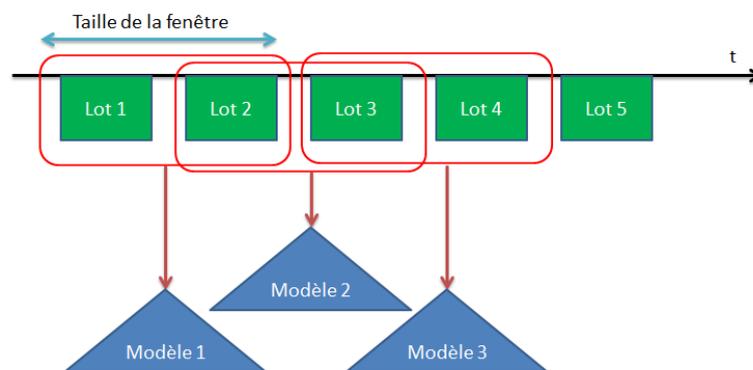


Figure 6 - Approches basées sur la sélection des données.

<sup>3</sup> On comprendra aisément pourquoi ce type d’approche n’est alors pas du tout adapté à l’apprentissage en environnement stationnaire.

Le point clé de ces approches repose sur la détermination de la taille de la fenêtre. La plupart des méthodes utilisent une fenêtre de taille fixe configurée pour chaque problème réel. Cependant, il existe des méthodes qui visent à détecter automatiquement la taille de la fenêtre d'analyse :

- Dans [Bifet 2007] avec ADWIN (*ADaptive WINdow*), l'auteur teste un ensemble de tailles de fenêtres en divisant chaque fenêtre en sous-fenêtres de taille minimale (*large enough*). Si les sous-fenêtres ont une distribution suffisamment différente (*distinct enough*), cette taille est considérée comme un bon choix. La définition de *large enough* ou *distinct enough* est donnée par un test statistique.
- Dans [Lazarescu 2004], les auteurs proposent d'exploiter à chaque étape deux modèles, chacun appris avec une taille de fenêtre différente : S (une taille standard prédéfinie) et 2S. La première fenêtre avec une petite taille S, sert à détecter la dérive du concept avec un test statistique comme ci-dessus, la deuxième fenêtre, avec une grande taille (2S), sert à mettre à jour le modèle si une dérive est détectée.
- Dans [Last 2002] avec OLIN (On Line Information Network), les auteurs proposent de changer dynamiquement la dimension des fenêtres par rapport aux performances obtenues sur un jeu de données de validation. À chaque étape, les nouvelles données sont séparées en deux parties : l'une pour l'apprentissage, et l'autre pour la validation. Un ensemble de fenêtres de taille différente va être appliqué séparément pour apprendre et tester. La taille qui donne le meilleur résultat sur les données de validation est choisie pour l'étape actuelle. Pour qu'elle soit réellement exploitable, cette approche nécessite que les phases d'apprentissage se fassent sur des lots de données de taille suffisamment grande.
- Dans [Klinkenberg 2004] l'auteur applique cette méthode aux SVM. À chaque étape, il calcule la performance (en terme de taux d'erreur) pour différentes tailles de fenêtres (taille No1 = le dernier lot, taille No2 = les 2 derniers lots, taille No3 = les 3 derniers lots...). La taille qui donne les meilleures performances est choisie.
- Une extension de VFDT (déjà présenté dans 2.2.1.1.2) qui s'appelle CVFDT (Concept-adapting Very Fast Decision Tree) dans [Hulten 2001] applique cette idée avec des fenêtres temporelles de taille fixe pour s'adapter dans un environnement non-stationnaire. Plus précisément, dans son expérimentation, il utilise 3 fenêtres temporelles T0, T1, T2: T0 = 10000 exemples, T1 = 9000 exemples, T2 = 1000 exemples. T2 est pour la détection de dérive de concept tandis que T1 est utilisé pour apprendre. T0 est la taille maximale de stockage dans le système.

- Plus récemment, dans [Bifet 2009] avec HWT (*Hoeffding Window Tree*) et HAT (*Hoeffding Adaptive Tree*), l'auteur utilise la détection de dérive de concept par la méthode ADWIN (*cf. ci-dessus*) dans les VFDT (présentés en 2.2.1.1.2).

Ce qui est intéressant avec cette approche par fenêtre glissante, c'est la possibilité de l'appliquer facilement sur la plupart des approches statiques et ainsi les transformer en approches dynamiques. On notera également que certains auteurs proposent aussi une méthode de détection de dérive de concept qui ajuste automatiquement la fenêtre d'apprentissage. Cependant, cette approche nécessite de maintenir une mémoire des données, au moins partielle.

#### 2.2.1.2.2. Approches basées sur la pondération des données

Ces approches reposent sur une pondération des données d'apprentissage permettant de prendre en compte l'importance des observations (anciennes ou nouvelles). Ces approches sont particulièrement adaptées à des classificateurs qui permettent naturellement d'exploiter ces poids soit lors de la prise de décision (*K-nn* par ex.), soit lors de la procédure d'apprentissage (SVM, RN) pour mettre à jour ou réapprendre le système. La Figure 7 illustre un exemple très simple de cette approche en assignant un poids de 1 à toutes les données d'un nouveau lot et un poids de 0.5 à toutes les données des anciens lots. Comme dans le cas des approches basées sur la sélection des données, ces approches nécessitent de mémoriser les données (de façon standard, il n'y a pas de méthode permettant de supprimer des données d'apprentissage de l'historique). En pratique, sur des flux importants, leur usage semble donc délicat sans adaptation pour limiter les coûts de stockage. En plus, ces approches ne sont pas facilement applicables à tous les systèmes classiques, notamment les réseaux de neurones et les SVMs. C'est pourquoi elles sont beaucoup moins exploitées dans la littérature.

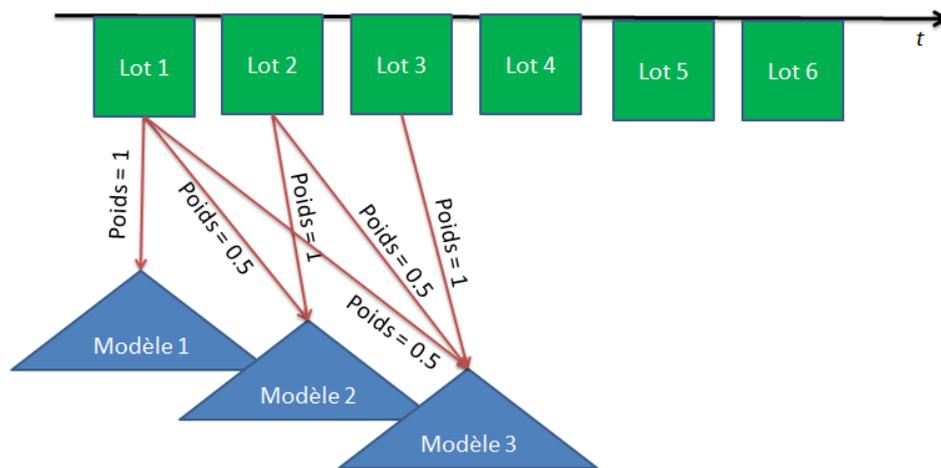


Figure 7 - Approches basées sur la pondération des données.

On peut citer [Alippi 2007], [Alippi 2008], [Alippi 2009] comme un exemple de ces approches, basé sur un  $k$ -NN (*k-Nearest Neighbors*). Dans le  $k$ -NN traditionnel, le poids est proportionnel à la distance entre l'exemple de requête et l'exemple déjà classé. Dans [Alippi 2007] et [Alippi 2008], l'auteur propose d'assigner simplement 0 à l'exemple de la base ancienne et 1 à l'exemple actuel. De façon plus complexe, dans [Alippi 2009], l'auteur propose d'assigner un plus grand poids pour les observations de l'apprentissage qui portent des changements. Ce niveau de changement est calculé sur la base de la similarité entre la situation au moment  $i$  précédent ( $C(i)$ ) et le moment  $t$  actuel ( $C(t)$ ). La clé de cette approche est d'introduire une mesure de « changement »  $C(t)$  calculée par un test statistique (qui se nomme CI-CUSUM). S'il s'agit d'une dérive de concept lente, toutes les données (anciennes et nouvelles) se voient attribuer le même poids ; en revanche, s'il s'agit d'une dérive de concept rapide, les anciennes données ont un poids plus petit que les nouvelles (cela peut aller jusqu'à un poids de 0 pour toutes les anciennes données si la dérive est brutale). Ce poids sera utilisé dans le processus de décision pour calculer le nombre de plus proches voisins.

Dans [Klinkenberg 2004], l'auteur développe une version pour les SVM. Chaque lot de données est associé à un âge correspondant au rang d'apparition du lot (son âge). Le poids attribué aux données dépend de l'âge du lot auquel elles appartiennent : plus le lot est récent, plus le poids est grand ; plus le lot est vieux, plus le poids est faible. Dans la procédure d'apprentissage, ce poids est utilisé pour accentuer les erreurs commises (et donc forcer l'algorithme à faire moins d'erreurs sur les données avec un poids fort) sur les exemples anciens (poids plus faible) et les nouveaux exemples (poids plus fort). Dans son article,

l'auteur note que cette approche donne des résultats moins bons que celle utilisant la fenêtre temporelle sur sa base d'expérimentation.

#### 2.2.1.2.3. Approches par combinaison de classificateurs

On peut finalement évoquer les travaux portant sur la combinaison de classificateurs, très largement utilisée pour ce type de problème. Le principe est simple : lorsque les données arrivent, le système prend une décision de classification basée sur des poids relatifs à chaque classificateur de l'ensemble. Ces poids sont mis à jour régulièrement, en fonction des nouvelles données d'apprentissage. De nouveaux classificateurs peuvent également être ajoutés à l'ensemble lorsque le besoin s'en fait sentir. Un point fort de ces approches est qu'elles peuvent gérer efficacement la réoccurrence de concepts car les informations issues des données au cours du temps sont toujours « enregistrées » dans les classificateurs, même si ces derniers peuvent momentanément être « désactivés » par un poids faible. On peut diviser ces approches en trois grandes catégories (proposées dans [Kuncheva 2004]): *mise à jour par une combinaison dynamique ; mise à jour par la mise à jour de la base d'apprentissage ; mise à jour par restructuration des classificateurs.*

Passons en revue ces différentes catégories :

- ***Mise à jour par une combinaison dynamique*** : chaque classificateur est associé à un poids qui est diminué ou augmenté selon la performance de chaque classificateur sur le nouveau lot de données. Initialement, le poids est mis à 1 pour tous les classificateurs. La méthode MVA (*Major Vote Algorithm*) proposée par Littlestone dans [Littlestone 1994] diminue le poids des classificateurs qui ont mal prédit (même si la prédiction de l'ensemble est bonne), tandis que *Winnnow*, dans [Littlestone 1988], ne met à jour tous les classificateurs que si la prédiction de l'ensemble est fautive : le poids des classificateurs qui font une prédiction correcte est augmenté ; il est diminué pour les autres. La Figure 8 donne un exemple simple de cette approche. On notera que dans les deux approches ci-dessus, le nombre de classificateurs de l'ensemble est fixe, ainsi que le nombre de concepts. Ces méthodes semblent plus adaptées aux environnements stationnaires que non-stationnaires car même si les classificateurs sont désactivés au cours du temps, ils sont tous conservés. Sans compter les problèmes de stockage et de complexité calculatoire liés à l'usage au cours du temps, cette quantité de classificateurs peut réduire la capacité d'adaptation (notamment en termes de vitesse d'adaptation). En revanche, par rapport aux approches par sélection de données, il n'est pas nécessaire de conserver les données, les

connaissances portées par celles-ci étant résumées dans les classificateurs. Les besoins en termes de stockage devraient donc être moindres.

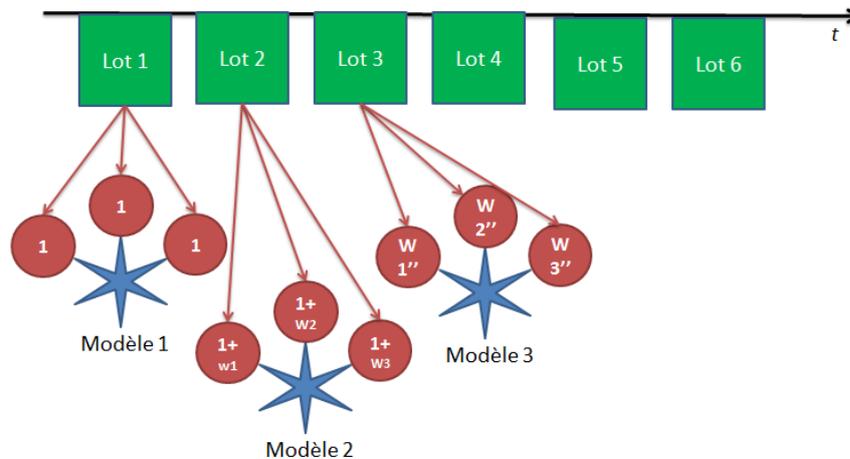


Figure 8 - Mise à jour par combinaison dynamique.

- **Mise à jour par mise à jour de la base d'apprentissage** : le principe de ces méthodes est de mettre à jour les classificateurs (tous ou quelques-uns) par un réapprentissage régulier. Les deux méthodes les plus intéressantes à mentionner sont le *online bagging* et le *online boosting* présentées dans [Oza 2001], versions dérivées des méthodes classiques de *bagging* et *boosting*. Chacune duplique chaque nouvelle donnée d'apprentissage afin de pouvoir s'adapter réellement à chacune d'elles.
  - Le *bagging* [Breiman 1994] divise la base d'apprentissage en  $M$  sous-bases d'apprentissage par un tirage aléatoire et apprend séparément un classificateur à partir de chacune des sous-bases. Il y a donc un total de  $M$  classificateurs créés,  $M$  étant un nombre fixé. La décision finale est prise par un vote majoritaire. La version online travaille sur chacune des données qui arrivent au système. Chaque donnée est dupliquée  $f$  fois ( $f$  est trouvé par une loi de Poisson). Toutes ces copies sont ajoutées dans la base d'apprentissage initiale qui sera à son tour divisée en  $M$  sous-bases pour réapprendre  $M$  nouveaux classificateurs qui remplacent les précédents.
  - En ce qui concerne le *boosting*, *Adaboost*, présenté dans [Freund 1997], a pour principe d'améliorer l'apprentissage en augmentant/diminuant le poids de chaque exemple d'apprentissage selon les erreurs de classification commises. Pour chaque nouveau lot de données  $D$ , *Adaboost* initialise le poids de chacun des  $N$  exemples de  $D$  par un poids unique de  $1/N$ . Le système apprend un classificateur  $X$  sur

celles-ci. Après ce premier apprentissage, le classificateur  $X$  est évalué sur ce même lot. L'erreur (somme des poids de tous les exemples mal classés) est ainsi calculée. Si elle est grande ( $> 1/2$ ), *Adaboost* supprime ce classificateur  $X$ , sinon, il le conserve dans le système. Les poids des exemples de  $D$  sont ensuite mis à jour de la façon suivante : les exemples corrects voient leur poids diminuer (ces poids sont multipliés par  $p_{corr} = \frac{1}{2(1-Err)}$ ) ( $Err$  est la somme des poids des exemples mal classés), et les exemples incorrects voient leurs poids augmenter (ces poids sont multipliés par  $p_{incorr} = \frac{1}{2*Err}$ ). Dans l'ensemble d'apprentissage, on ajoute  $f$  copies de chaque nouvel exemple qui arrive,  $f$  étant déterminé par une distribution Poisson. Le poids des données d'apprentissage est ensuite mis à jour comme dans le *boosting* originel en prenant en compte le nombre de données rajoutées par la loi de Poisson. Un nouveau classificateur est appris à partir de cette base d'apprentissage modifiée, comme dans l'algorithme de base, et ce jusqu'à convergence. La prise de décision se fait également de la même façon.

On remarquera ici que ces 2 méthodes d'apprentissage ne correspondent pas à la notion d'apprentissage incrémental usuel dans le sens où toutes les données d'apprentissage doivent être conservées et l'ensemble de classificateurs est réappris intégralement à chaque étape.

- ***Mise à jour par restructuration des classificateurs*** : le principe de ces méthodes est de mettre à jour le système par ajout ou suppression d'un classificateur (d'où la notion de restructuration du système global introduite par Kucheva). Dans [Street 2001] avec SEA (*Streaming Ensemble Algorithm*), l'auteur propose d'ajouter un classificateur pour chaque nouveau lot de données arrivant. Après cet ajout, une évaluation de performances est lancée sur tous les classificateurs de l'ensemble pour détecter le plus mauvais, qui sera supprimé. Dans [Wang 2003], l'auteur utilise une idée similaire consistant à ajouter un classificateur appris sur un nouveau lot, mais ici, la suppression du classificateur s'effectue non pas selon sa performance mais selon son âge (on supprime le plus vieux classificateur). La Figure 9 illustre un exemple simple dans lequel un classificateur est ajouté à chaque nouveau lot (classificateurs 1, 2, 3, 4), et le plus vieux classificateur est supprimé pour mettre à jour le modèle (classificateur 1) dans l'étape 4 (lot 4). Dans [Kolter 2007], l'approche DWM (*Dynamic Weighted Majority*) utilise à la fois une combinaison dynamique en pondérant les classificateurs, ainsi que la possibilité de

restructurer le système en ajoutant/supprimant des classificateurs. La procédure ajoute un classificateur (de poids 1) appris sur une nouvelle donnée quand l'ensemble commet une erreur de classification (vote majoritaire) sur cette donnée. Les classificateurs de l'ensemble qui commettent cette erreur voient également leur poids diminuer. Lorsqu'un classificateur se retrouve avec un poids très faible, il est supprimé du système. On notera que les approches ci-dessus ne peuvent gérer efficacement la réoccurrence d'un concept disparu puisque des classificateurs sont supprimés. Cependant, l'approche de Kolter le permettrait simplement en laissant évoluer les classificateurs avec un faible poids qui pourrait à nouveau croître. On notera que ces méthodes sont à la base des approches les plus utilisées en ce moment dans les approches basées sur des ensembles de classificateurs pour fonctionner en environnement non-stationnaire.

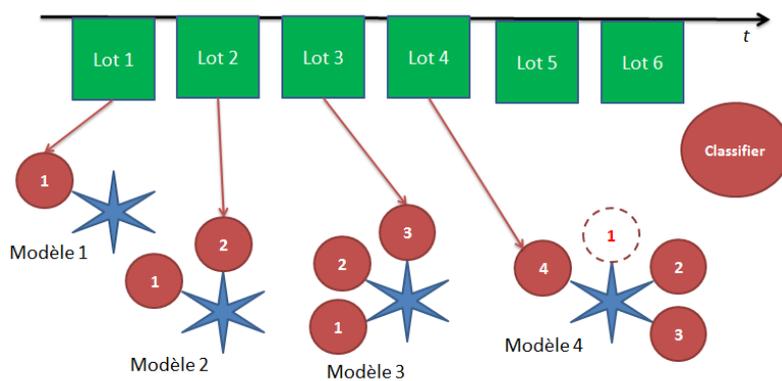


Figure 9 - Mise à jour par restructuration des classificateurs.

Bien entendu, il existe d'autres approches permettant de gérer l'apprentissage en environnement non-stationnaire. Nous verrons notamment dans la partie 2.2.2.1.3 celles permettant de modifier le nombre de concepts.

En dehors de la façon dont sont gérées les procédures d'apprentissage et de décision pour prendre en compte la non-stationnarité de l'environnement, un des points clés de toutes ces approches repose sur la détection de cette non-stationnarité. En effet, si certaines approches considèrent comme *a priori* que l'environnement est non-stationnaire, d'autres s'attachent à déterminer précisément quand une dérive survient et à quelle vitesse. Dans l'environnement non-stationnaire, on peut distinguer les méthodes par approche passive (*passive approach*) et par approche active (*active approach*). Une approche dite active propose une étape de détection de la dérive de concept avant de mettre à jour le modèle. Si le changement est détecté, on l'apprend, sinon on renforce le modèle. Pour détecter la dérive de concept, le système doit faire un calcul statistique sur les données qui arrivent. C'est pourquoi

les approches actives travaillent souvent avec des lots de données (hors-ligne) en intégrant une fenêtre temporelle pour déterminer si deux fenêtres successives possèdent ou non des propriétés différentes en termes de représentation des concepts. Les autres approches passives (comme celles basées sur la combinaison de classificateurs ou sur la pondération des données) considèrent, quant à elles, que la dérive de concept existe à toutes les étapes. Le système apprend donc à chaque lot (ou chaque exemple) arrivé.

### 2.2.2. Évolution de concepts avec changement du nombre de concepts

Alors que les approches précédentes se focalisaient sur la modification des paramètres du modèle utilisé pour prendre en compte la modification des concepts, on se concentre d'avantage, dans cette partie, sur des propriétés supplémentaires que doivent posséder les approches pour pouvoir gérer l'évolution du nombre de concepts. Bien souvent, les modifications doivent être apportées à un niveau supérieur, *i.e.* structurel, imposant l'usage de systèmes reposant sur des structures flexibles.

Un des cas les plus traités correspond à la possibilité d'ajouter de nouveaux concepts. Les origines d'un tel besoin peuvent être multiples. En apprentissage non-supervisé par exemple, cela peut provenir d'une nouvelle donnée (ou plusieurs) que l'algorithme estime trop « éloignée » des clusters existants et nécessitant donc la création d'un nouveau cluster. Un cas similaire peut se produire en apprentissage supervisé si le système observe des données non étiquetées qu'il estime ne correspondre à aucune des classes existantes/attendues. On parle alors de rejet de distance dans [Mouchère 2006]. Ce type de rejet sera, en apprentissage supervisé, facilité s'il est possible d'avoir une notion de distance aux concepts existants. Ceci n'est pas évident avec les approches classiques purement discriminantes, mais l'est beaucoup plus si l'on introduit une modélisation générative ou à base de prototypes (l'apprentissage non supervisé par exemple permet d'obtenir ce type de modèles). On notera toutefois qu'il s'agit là d'un cas particulier réservé à certains types de problèmes supervisés dans lesquels le système n'a pas de connaissance *a priori* sur l'ensemble des classes à modéliser. Un cas de figure plus classique survient lorsque les données d'apprentissage, reçues jusqu'alors depuis le flux, ne décrivaient pas un concept que l'on savait exister. Finalement, que ce soit en supervisé ou non, l'ajout d'un concept peut être une conséquence de l'interaction avec l'utilisateur qui souhaite le voir créer (suite à une proposition éventuelle du système dans l'un des cas précédents). En dehors de ce cas bien

précis d'ajout de concept, l'évolution peut également se faire en supprimant, divisant ou fusionnant des concepts. On notera cependant que les approches traitant explicitement de ces cas sont rares en apprentissage supervisé.

La plupart des approches décrites ci-dessous sont essentiellement des extensions des méthodes présentées plus haut, lorsque la nature du classificateur permet, d'un point de vue structurel, de jouer sur l'évolution du nombre de concepts modélisés. C'est pourquoi on retrouve essentiellement des approches autour des systèmes à base de règles et des ensembles de classificateurs. Dans les sections suivantes, nous passons en revue les différents cas selon le fait que l'environnement soit stationnaire ou non.

### 2.2.2.1. Ajout de concepts en environnement stationnaire

Cette catégorie d'algorithmes, souvent qualifiés d'incrémentaux, a notamment été étudiée par Polikar dans [Polikar 2001]. Selon lui, ces approches doivent avoir les propriétés suivantes : a) elles doivent être capable d'apprendre des connaissances supplémentaires à partir des nouvelles données ; b) elles ne doivent pas nécessiter l'accès aux données d'origine (c'est-à-dire aux données qui ont été utilisées pour apprendre le classificateur actuel) ; c) elles doivent préserver les connaissances déjà acquises ; et d) elles doivent être en mesure de modéliser de nouvelles classes susceptibles d'être introduites avec de nouvelles données. La contrainte forte ici est donc que le système doit être non régressif, c'est-à-dire qu'il doit toujours conserver les aptitudes de classification acquises au cours du temps. C'était déjà le cas pour les approches d'apprentissage continu en environnement stationnaire, sans évolution du nombre de concepts. Cela s'applique également à présent aux concepts en eux-mêmes, qui, dès lors qu'ils sont appris, ne peuvent plus disparaître. D'où le qualificatif d'incrémental.

#### 2.2.2.1.1. Approches par Système d'Inférence Floue (SIF)

Ces approches sont des extensions directes de celles présentées dans la section 2.2.1.1.4. L'avantage de la structure à base de SIF est que l'on peut assez facilement rajouter des règles au système puisque chaque règle modélise une région de l'espace par un prototype et que la conclusion donne la contribution de ce prototype dans la modélisation de chacune des classes. Ainsi, lorsqu'un exemple arrive, s'il ne correspond à aucune des zones déjà modélisées, les activations des règles seront toutes faibles. Cet exemple peut alors être utilisé pour construire un nouveau prototype et donc une nouvelle règle qui, potentiellement peut modéliser une nouvelle classe. Par extension, si un exemple correspond à une classe inconnue jusqu'alors (il porte une nouvelle étiquette), une nouvelle règle peut être créée

automatiquement pour décrire cette classe à partir de cet exemple. Dans les deux cas, l'ensemble des conclusions devra être mis à jour pour prendre en compte l'introduction de ce nouveau label. Les travaux de [Almaksour 2009], [Almaksour 2010], [Almaksour 2011] sont des exemples représentatifs de cette approche. Les différences entre les versions reposent essentiellement sur les changements des formules d'adaptation de prototypes et des conséquences.

#### 2.2.2.1.2. Approche des modèles d'ARTMAP (Adaptive Resonance Theory)

L'approche ARTMAP (détaillée en 2.2.1.1.3) permet l'ajout de concepts grâce à la couche ARTb. Le principe est le suivant. Si un exemple porte une nouvelle étiquette, aucun nœud ne peut prédire la bonne étiquette. Le système va alors ajouter un nouveau nœud dans la couche ARTb et lui associer une nouvelle étiquette dans inter-ART. Cette méthode est remarquable car c'est une des premières du genre qui permet d'ajouter des classes avec des résultats avérés [Connolly 2008]. La faiblesse de cette méthode est mentionnée dans [Polikar 2001] et [Almaksour 2009] qui montrent que cette méthode est très sensible à la valeur du paramètre de vigilance qui est difficile à contrôler.

#### 2.2.2.1.3. Approche par combinaison de classificateurs (Learn++)

Learn++ proposée par Polikar [Polikar 2001] est sans doute une des approches dites incrémentales les plus connues. Elle est basée sur la structure *Adaboost* décrite plus haut et hérite de ses avantages (section 2.2.1.2.3). Dans *Adaboost*, chaque nouveau classificateur ajouté à l'ensemble est formé à partir d'un ensemble d'exemples qui contient notamment ceux qui sont mal classés par l'ensemble de classificateurs courant. Si la règle d'*Adaboost* est optimisée pour améliorer la précision du classificateur, celle de Learn++ est optimisée pour l'apprentissage incrémental de nouveaux exemples, en particulier lorsque ceux-ci correspondent à une nouvelle classe. L'idée est simple : à chaque fois qu'un nouveau concept apparaît, les données qui le représentent seront mal classées puisqu'aucun classificateur ne modélise ce concept. Le système ajoute alors un classificateur comme dans *Adaboost*, se concentrant sur ces exemples représentant cette nouvelle étiquette. Le changement proposé dans Learn++ concerne aussi la mise à jour des poids (l'initialisation reste la même). Learn++ va calculer l'erreur de classification totale sur tout le nouveau lot de données : si l'erreur est grande, on supprime ce nouveau classificateur ; si l'erreur est faible, le classificateur est enregistré dans l'ensemble.

Il existe plusieurs variantes de Learn++. On peut citer par exemple : Learn++.SVM [Erdem 2005] ; *Learn++.UDNC* [Ditzler 2010] permettant la gestion de bases d'exemples déséquilibrées (*unbalanced database*) ; Learn++.NC [Mulhbaier 2009] qui a été développé pour l'apprentissage de nouvelles classes (les classes existantes étant supposées stationnaires). Cette approche repose sur les mêmes principes que Learn++. Elle diffère par la modification du système de vote. Alors que dans Learn++, tous les classificateurs participent au vote final, dans Learn++.NC un vote pondéré dynamique est utilisé pour déterminer quels classificateurs devraient (ou ne devraient pas) voter. Ce mécanisme renforce les nouveaux classificateurs qui contiennent des informations importantes (par exemple l'apparition d'une nouvelle classe) pour mieux concurrencer les anciens classificateurs. Enfin, Learn++.MF ([Polikar 2010]), pour *Missing Features*, permet aux classificateurs d'apprendre à partir de différents sous-espaces de caractéristiques pour gérer implicitement les valeurs de caractéristiques manquantes qui peuvent être prises en compte à partir des autres membres de l'ensemble appris, eux, sur des valeurs disponibles.

#### 2.2.2.2. Ajout de concepts en environnement non-stationnaire

Comme pour l'environnement stationnaire, la littérature associée aux environnements non-stationnaires évoque essentiellement la notion d'apprentissage incrémental (Incremental Learning in Non Stationary Environment). [Elwell 2009] est un exemple typique pour lequel le système peut ajouter de nouveaux concepts. Cette approche, appelée *Learn++.NSE with Controlled Forgetting*, est une variante de Learn++ adaptée aux environnements non stationnaires. Dans Learn++, de nouveaux classificateurs sont ajoutés à chaque lot, tant que des informations relatives aux concepts n'ont pas été modélisées par l'ensemble. Alors que cela permet d'avoir une meilleure stabilité et performance du système au cours du temps, cette approche n'est pas envisageable en environnement non-stationnaire. En effet, d'une part le nombre de classificateurs risque d'augmenter sans cesse (problème de complexité) et d'autre part, cela va entraîner une inertie (stabilité) qui sera préjudiciable au système qui nécessite plutôt, dans ce type d'environnement, une bonne plasticité. [Elwell 2009] propose alors d'utiliser la « restructuration des classificateurs » (voir 2.2.1.2.3) de l'une des façons suivantes : remplacer les classificateurs en fonction de leur âge, comme dans [Wang 2003] ou les remplacer en fonction de leur performance, comme dans [Street 2001]. Cela permet de supprimer les classificateurs obsolètes. On remarque donc que, au sens de la définition donnée par [Polikar 2001], ce type de système n'est plus vraiment incrémental puisque des connaissances (associées aux classificateurs) disparaissent. Dans [Elwell 2011], l'auteur

propose une autre variante appelée Learn++.NSE. Ici, on ne supprime plus les classificateurs mais on choisit lesquels vont participer au vote final. La modification concerne donc l'étape de mise à jour des poids des classificateurs : quand un nouveau lot de données arrive, le système doit tester séparément la performance de tous les classificateurs (nouveaux et anciens) sur ce lot. Si le nouveau classificateur n'obtient pas une performance satisfaisante, on réapprend ce classificateur à partir de ce lot mais avec un poids plus élevé sur les données mal classées. Si les anciens classificateurs n'obtiennent pas non plus une bonne performance, on les pénalise en diminuant leur poids pour le vote final. Ainsi, un classificateur n'est jamais supprimé et si sa modélisation redevient pertinente plus tard, pour un nouveau lot, son poids va à nouveau augmenter. D'un certain point de vue, on retrouve ici la notion d'incrémentalité.

En dehors de ces approches par combinaison de classificateurs, certains auteurs ont également souhaité relâcher la contrainte forte sur la non-régression d'un système afin de permettre l'oubli de concepts devenus caduques. C'est le cas notamment de l'approche dite *décrémentale (Decremental Learning)* de Bouillon dans [Bouillon 2013]. Une des limites de cette approche est qu'elle ne permet pas de gérer la réoccurrence de concepts.

#### 2.2.2.3. Fusion/division de concept

Il s'agit sans doute du cadre d'apprentissage continu le plus souple. En effet, outre l'ajout et la suppression de nouveaux concepts, on envisage également la fusion de concepts (resp. leur division) lorsque deux concepts que l'on croyait distincts se trouvent finalement représenter la même chose (resp. lorsqu'un concept, souvent multimodal, a besoin d'être considéré comme plusieurs concepts distincts). Dans ces cas, pouvoir détecter les ambiguïtés entre concepts (*i.e.* leur confusion) ou leur hétérogénéité (apparition de modalités bien distinctes) peut permettre de détecter quand une fusion/division est nécessaire. Là encore, si cela est faisable en mode non supervisé (que ce soit au travers d'approches hiérarchiques [Widyantoro 2002], [Ribert 1999], [Chen 2002] ou par partitionnement via l'usage de critères de partitionnement), l'opération est beaucoup plus compliquée en apprentissage supervisé si l'utilisateur n'apporte aucune information à ce sujet. Selon nos recherches, ce type d'apprentissage continu n'a d'ailleurs pas ou peu été traité dans ce mode supervisé.

### 2.3. Petit bilan terminologique

Tout au long de ce chapitre, nous avons proposé de distinguer cinq situations d'*apprentissage continu* :

- L'Évolution apparente des concepts en environnement stationnaire ou renforcement du concept (sans modification du nombre de concepts) – présentée dans 2.2.1.1
- L'Évolution réelle des concepts en environnement non-stationnaire (toujours sans modification du nombre de concepts) – présentée dans 2.2.1.2
- L'Ajout de concepts en environnement stationnaire – présentée dans 2.2.2.1
- L'Ajout de concepts en environnement non-stationnaire – présentée dans 2.2.1.1 2.2.2.2
- La Fusion/Division de concepts – présentée dans 2.2.2.3

Nous avons introduit ces termes afin de palier l'ambiguïté existante dans la terminologie que l'on retrouve dans la littérature. En effet, nous avons déjà évoqué tout au long de ce chapitre les problèmes liés aux termes utilisés fréquemment comme : apprentissage en ligne, adaptation, apprentissage incrémental, dérive de concept, apprentissage en environnement non-stationnaire, apprentissage décrémental. Ceux-ci se réfèrent tantôt au mode de gestion des données, tantôt à une propriété particulière ou au contraire générale de l'environnement, ou bien encore à une particularité du classificateur en terme de gestion des concepts. En dehors de ces termes, il faut noter également que d'autres auteurs ont introduit d'autres termes, notamment pour distinguer les méthodes d'apprentissage incrémentales. Ainsi, Zhi-Hua Zhou dans [Zhou 2002] distingue trois types d'algorithmes incrémentaux : E-IL (*Example-Incremental Learning*) ; C-IL (*Class-Incremental Learning*) et A-IL (*Attribute-Incremental Learning*). Les méthodes associées à l'E-IL correspondent à une vision centrée sur les données et donc aussi à la notion d'apprentissage en ligne. Polikar dans [Polikar 2001] les qualifie de fausses approches incrémentales. Elles font partie de la catégorie relative à l'Évolution apparente des concepts. En revanche, les méthodes C-IL correspondent essentiellement aux algorithmes permettant l'ajout de nouveaux concepts identifiés (une partie de ces approches correspond à l'ajout de concepts en environnement stationnaire). Enfin, les approches A-IL se focalisent sur le changement de l'espace de représentation. Elles peuvent donc appartenir à toutes les catégories. On remarquera que Zhou est l'un des seuls à utiliser un terme permettant d'envisager explicitement l'apprentissage continu sous cet angle de l'espace de représentation, à l'exception de l'approche Learn++.MF mentionnée plus haut. Une autre terminologie utilisée dans [Lughofer 2011] distingue les systèmes adaptatifs (correspondant à l'évolution apparente des concepts) et les systèmes évolutifs (ajout de concepts en environnement stationnaire). Enfin, de façon équivalente, on

retrouve dans [Almaksour 2009] l'apprentissage de paramètres (systèmes adaptatifs) et l'apprentissage de structures (système évolutif).

On notera finalement que le terme d'apprentissage décrémental se réfère à la fois à la capacité d'ajouter/oublier des concepts dans un environnement non-stationnaire comme dans [Bouillon 2013], mais aussi à une adaptation en environnement non-stationnaire sans utiliser les anciennes données et sans pouvoir ajouter de classes (comme dans [Cauwenberghs 2001], [Karasuyama 2010]).

La Figure 10 montre une synthèse sous forme graphique de ces liens entre les différents termes et en reprenant nos deux axes évoqués plus haut.

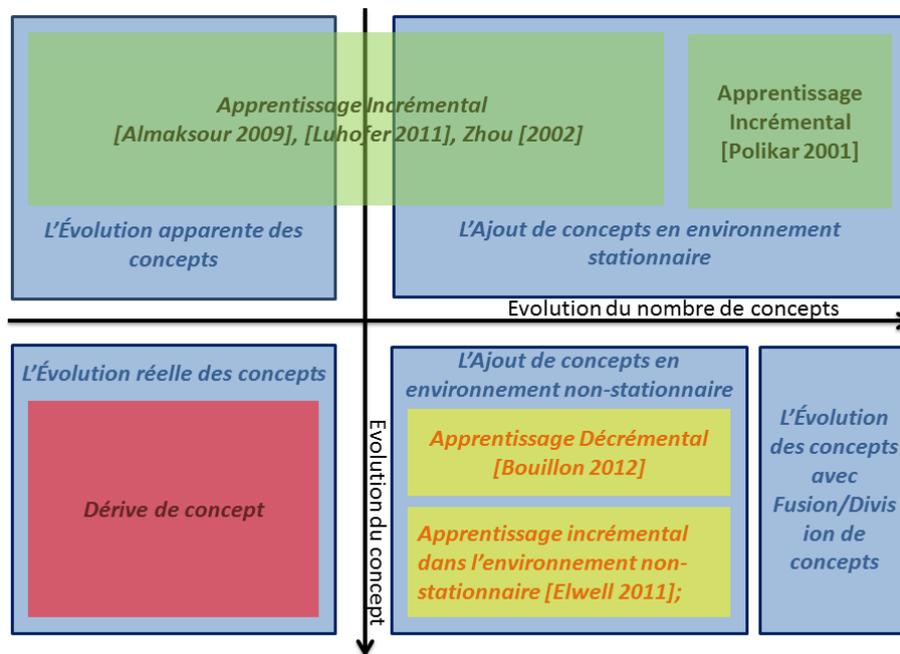


Figure 10 - Relations entre les principaux termes de la littérature et notre terminologie

## 2.4. Conclusion

Dans ce chapitre, nous avons mis en évidence la variété des méthodes d'apprentissage incrémentales, de la terminologie associée et de leur domaine d'application (environnement stationnaire ou non-stationnaire).

Il ressort de cette étude que chaque méthode proposée est dédiée à un nombre limité de problématiques spécifiques (évolution du concept, ajout de concepts...). Nous proposons, dans la suite, d'essayer de mettre en place une méthode plus générale pouvant s'appliquer dans la plupart des environnements que nous avons mentionnés dans cet état de l'art. Pour

cela, nous nous basons sur l'usage d'une combinaison de classificateurs *one-class* construits à partir de SVM. Nos propositions seront d'abord testées sur les benchmarks habituels utilisés en apprentissage incrémental. Elles seront ensuite adaptées pour produire le module de numérisation cognitive relatif au projet DIGIDOC.

## Deuxième partie

Contributions : systèmes dynamiques autour  
de l'architecture « Multi One-Class  
Incremental SVM» (mOC-iSVM)

### III. Contributions : systèmes dynamiques autour de l'architecture «Multi One-Class Incremental SVM» (mOC-iSVM)

**L'**état de l'art sur l'apprentissage continu montre qu'il existe peu de solutions qui tiennent compte à la fois de l'évolution des concepts, de leur nombre et de leurs interactions comme la fusion et la division de concepts. Par ailleurs, on constate également que, même si les SVMs sont largement utilisés dans des systèmes dynamiques à base d'ensembles de classificateurs, leur usage n'est pas sans limites. En particulier, lorsque les concepts ne sont définis que par un ensemble d'interactions entre plusieurs classificateurs discriminants, l'évolution des concepts, et en particulier la détection automatique de nouveaux concepts, la fusion ou la division de concepts semble plus délicate, du fait même de leur entremêlements dans des modèles discriminants. A contrario, alors que les classificateurs dits one-class semblent plus naturellement adaptés à ces types de problèmes, leur usage reste très modeste, sans doute à cause de la difficulté inhérente à leur paramétrisation ainsi qu'à leurs performances souvent moins élevées en classification que leurs homologues discriminants. C'est pourquoi notre choix s'est tourné vers l'usage d'one-class SVM, afin de tirer parti de leurs avantages pour notre problématique, tout en essayant d'aller au-delà de leurs limitations. Nous montrerons dans la suite que leur usage dans le cadre de l'apprentissage continu est tout à fait intéressant.

## 3.1. Des SVMs binaires incrémentaux aux one-class SVMs incrémentaux

### 3.1.1. Principes et notions élémentaires des SVMs

Dans cette partie, nous rappelons brièvement les principes essentiels sur lesquels reposent les SVMs. Les machines à vecteurs supports (SVM) ont été proposées par Vapnik [Vapnik 1999] avec l'hypothèse que l'espace de description initial pouvait être projeté dans un espace de plus grande dimension afin de faciliter la séparation des classes dans cet espace. Une bonne séparation est définie par un hyperplan qui a la plus grande distance aux données d'apprentissage des classes. La distance entre l'hyperplan et la donnée la plus proche détermine la marge du classificateur. La fonction de décision peut être spécifiée par un petit sous-ensemble de données qui définit la position de l'hyperplan dans l'espace. Ces points sont appelés les vecteurs supports (SVs). Par sa capacité à projeter les données dans un espace optimal, le SVM est devenu un des classificateurs les plus exploités.

Le principe des SVMs repose sur deux notions fondamentales : la marge maximale et la fonction noyau. Grâce à elles, un problème de discrimination non-linéaire peut être reformulé en un problème d'optimisation quadratique pour lequel on connaît des méthodes de résolution.

La première notion que nous traiterons est celle de marge maximale. La marge est la distance entre la frontière qui sépare les classes et les exemples les plus proches (appelés vecteurs supports). Dans les SVMs, la frontière de séparation qui est choisie est celle qui maximise la marge. Ce choix est justifié par la théorie de Vapnik-Chervonenkis. Le problème est donc de retrouver la frontière séparatrice optimale, à partir d'un ensemble d'apprentissage. Ce problème peut être résolu par une optimisation quadratique qui est définie de la façon suivante :

Soit  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ , vecteur  $g \in \mathbb{R}^n$ , matrice réelle symétrique  $H \in \mathbb{R}^{n \times n}$ , matrice réelle  $A \in \mathbb{R}^{m \times n}$ , vecteur réel  $b^m \in \mathbb{R}$ . L'optimisation quadratique repose sur une fonction objective à minimiser, de la forme suivante :

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n H_{ij} x_i x_j + \sum_{j=1}^n g_j x_j$$

sous les contraintes :

$$\sum_{j=1}^n A_{ij} x_j - b_i \leq 0 ; \forall i \in \mathbb{N}.$$

La deuxième notion est celle relevant de la fonction noyau du SVM. Dans les cas où les exemples ne sont pas linéairement séparables (la plupart des cas), on transforme l'espace de représentation des exemples en un espace de plus grande dimension (potentiellement de dimension infinie), dans lequel il est possible de trouver un séparateur linéaire. Une fonction noyau qui respecte le théorème de Mercer<sup>4</sup> (cela permet de ne pas avoir à connaître explicitement la transformation à appliquer pour le changement d'espace) permet de transformer un produit scalaire exprimé dans un espace de grande dimension (trop coûteux), en une simple évaluation ponctuelle d'une fonction. Cette technique est connue sous le nom de *kernel trick*.

Pour la classification des exemples à deux classes du SVM, on utilise des caractéristiques connues de ces exemples. Ces  $n$  caractéristiques sont représentés par un vecteur  $x \in \mathbb{R}^z$ . La classe associée à ce vecteur est représentée par  $y \in \{-1, 1\}$ , où une des classes possibles est représentée par -1 et l'autre par 1. Le problème est de déterminer la valeur de  $y$  en se servant de  $x$ . Pour cela, il faut avoir un ensemble d'exemples (dit ensemble d'apprentissage) pour lesquels la classe est déjà connue et qui sont utilisés pour construire une règle de décision permettant de prédire la classe d'exemples non étiquetés (dit ensemble de test). Cette règle doit posséder de bonnes capacités de généralisation afin de pouvoir classer correctement des données ayant des caractéristiques légèrement différentes de celles de l'ensemble d'apprentissage.

### 3.1.2. Classification par SVM binaire

À partir de ces définitions, le problème du SVM discriminant à deux classes peut être décrit de la façon suivante :

---

<sup>4</sup> Théorème de Mercer : Soit  $K$  une fonction noyau continue symétrique de type positif. Alors : les valeurs propres de  $T_K$  sont des réels positifs ; il existe une base hilbertienne  $(e_i)$  de  $L^2[a, b]$  constituée de fonctions propres ; les fonctions propres correspondant à des valeurs propres strictement positives sont continues sur  $[a, b]$  ;  $K$  admet la représentation où la convergence est absolue et uniforme.

Supposons qu'on dispose d'un ensemble d'apprentissage de  $n$  exemples  $(x_i, y_i); x_i \in \mathbb{R}^z$  le vecteur de caractéristiques d'entrée,  $y_i \in \{-1,1\}$ , le vecteur de classification (sortie),  $i=1, \dots, n$ . Séparer les données en 2 classes revient à chercher un hyperplan de la forme  $w * x + b = 0$  tel que  $w * x + b > 0$  pour tout  $x$  appartenant à la classe 1 (les exemples positifs) et  $w * x + b < 0$  pour tout  $x$  appartenant à la classe -1 (les exemples négatifs) avec  $w = (w_1, \dots, w_m) \in \mathbb{R}$  le vecteur des coefficients de l'hyperplan et  $b \in \mathbb{R}$  un scalaire appelé le biais. Il existe une infinité d'hyperplans qui peuvent servir de séparateurs. Si  $dist(x_i, w, b)$  représente la distance entre le point  $x_i$  et l'hyperplan  $w * x + b = 0$ , la marge  $M$  est définie ainsi :  $M = \min_{x_i} \{ dist(x_i, w, b) \}, i = 1, \dots, n$ . Le meilleur hyperplan est celui qui possède la plus grande marge  $M$ . Le SVM doit trouver l'hyperplan qui sépare les exemples des deux classes avec la plus vaste marge possible, puis utiliser cet hyperplan pour classer de nouveaux exemples (voir Figure 11).

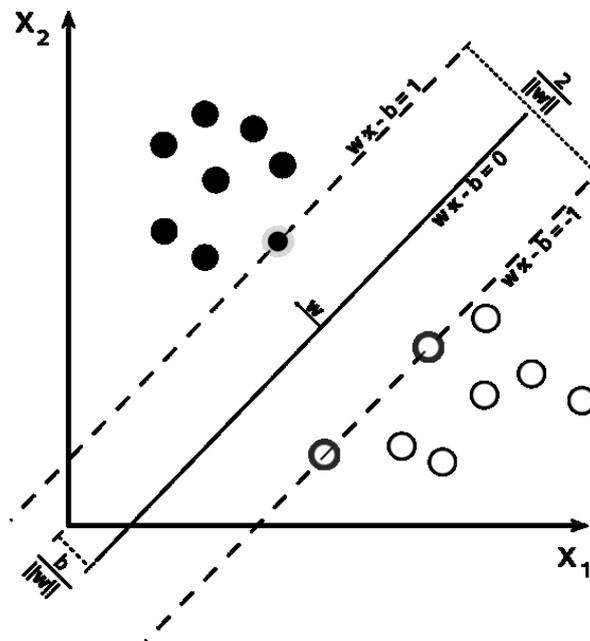


Figure 11 - Séparation dans un SVM binaire.

Deux étapes doivent alors être conduites : l'étape d'apprentissage et l'étape de classification.

### 3.1.2.1. Étape d'apprentissage

L'hyperplan  $w * x + b = 0$  possède une marge  $M$  donnée par l'expression :  $M = \frac{1}{\|w\|}$ . Plus  $\|w\|$  est petit, plus la marge est grande. On peut prouver aussi que plus  $\frac{1}{2} \|w\|^2$  est

petit, plus la marge est grande. La recherche du meilleur hyperplan peut donc s'écrire sous la forme du problème d'optimisation dit **primal**. La formule du **problème primal** est la suivante:

$$\text{Minimiser } \frac{1}{2} \| w \|^2$$

sous les contraintes :  $y_i(w * x_i + b) \geq 1 ; \forall i$ .

Pour assouplir les contraintes, on introduit des variables « ressort » non-négatives  $\xi_i$  qui correspondent à des exemples mal classés et on pénalise le non-respect de la contrainte par le paramètre de coût  $C$ . La formule du **problème primal** devient :

$$\text{Minimiser } \frac{1}{2} \| w \|^2 + C \sum_{i=1} \xi_i$$

sous les contraintes :  $y_i(w * x_i + b) \geq 1 - \xi_i ; \forall i$ .

On passe du **problème primal** au **problème dual** en introduisant des multiplicateurs de Lagrange pour chaque contrainte. Le **problème dual** s'exprime alors sous forme d'un problème d'optimisation quadratique :

$$\text{Maximiser } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j$$

sous les contraintes :  $\sum_{i=1}^n \alpha_i y_i = 0 ; \forall i, 0 \leq \alpha_i \leq C$

Supposons que la fonction  $\phi$  est une fonction qui transforme l'espace des caractéristiques des exemples en un espace de plus grande dimension  $\mathcal{H}$ . Plutôt que de trouver la transformation directe de cette fonction  $\phi : X \mapsto \mathcal{H}$  (qui est plus coûteuse), on peut choisir simplement une fonction noyau  $k : x \mapsto h$  (nombres réels) pour traduire la répartition des exemples dans cet espace  $\mathcal{H}$  avec la formule:  $k(x_1, x_2) = \phi(x_1) \cdot \phi(x_2)$ . Lorsque  $k$  est bien choisi, on n'a plus besoin de calculer la représentation de tous les exemples dans cet espace pour trouver  $\phi$ . Le noyau  $k$  peut être polynomial, par exemple  $k(x_1, x_2) = (x_1 * x_2 + 1)^p$ , ou une fonction à base radiale  $k(x_1, x_2) = e^{(\|x_1 - x_2\|)^2 / 2\sigma^2}$ . La formule est alors redéfinie comme suit :

$$\text{Maximiser } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

sous les contraintes :  $\sum_{i=1}^n \alpha_i y_i = 0 ; \forall i, 0 \leq \alpha_i \leq C ; i = 1, \dots, n$

### 3.1.2.2. Étape de classification

La classe associée à un exemple présenté en entrée est donnée par la fonction de décision suivante:

$$f(x, \alpha) = \text{sign}\left(\sum_{i=1}^n y_i \alpha_i k(x_i, x_j) + b\right)$$

Le résultat de cette équation est la valeur de décision du classificateur pour un exemple  $x$  et représente la distance de cet exemple à l'hyperplan séparateur dans  $\mathcal{H}$  ( $\alpha_i$  sont des vecteurs supports). Cette valeur est une mesure de confiance sur la décision pour un exemple. Plus la valeur absolue est grande, plus la confiance que l'on a dans le classificateur est grande [Vlachos 2004]. Cette valeur a aussi les propriétés suivantes : elle n'est pas comparable lorsqu'elle est issue de différents noyaux, elle n'est pas comprise entre  $[-1,1]$  et ne peut être interprétée de façon probabiliste. Cependant, des travaux, permettant de faire le lien entre cette valeur et une probabilité, ont été réalisés dans [Platt 2000], [Sollich 1999]. Les plus utilisés sont ceux de Platt qui appliquent une fonction sigmoïde sur ces valeurs.

### 3.1.2.3. Complexité

Des travaux ont montré que la complexité des SVMs est polynomiale en  $n$ , le nombre de données d'apprentissage, et  $d$  la dimension de l'espace de représentation d'origine [Hasan 2006].

$$dn^2 \leq \text{complexité} \leq dn^3$$

### 3.1.2.4. Paramétrage

Dans les SVMs binaires, il faut déterminer le paramètre  $C$  qui représente le coût pour accepter des exemples mal classés, ainsi que les paramètres de la fonction noyau (dépendant de chaque noyau). La méthode généralement employée est fondée sur une recherche par grille : chaque ensemble de valeurs de paramètres sur la grille est testé sur une partie au moins de l'ensemble d'apprentissage pour trouver le meilleur paramètre  $C$ . Dans [Hsu 2010], l'auteur explique l'intérêt d'utiliser la recherche par grille. L'inconvénient des autres méthodes non exhaustives est qu'elles ne garantissent pas de trouver une solution réellement optimale au problème. Le coût de la recherche par grille s'est également avéré supportable comparativement à d'autres méthodes d'approximation plus exactes. Enfin, ce type de recherche est très facile à paralléliser car tous les paramètres sont indépendants.

Cette recherche par grille peut être fait simplement avec une base de validation ou, combinée à une validation croisée n-fois.

### 3.1.2.5. Classification par SVM multi-classes

Pour résoudre un problème multi-classes, on combine des SVMs binaires selon les approches un contre tous (*1-vs-all*) ou un contre un (*1-vs-1*). Dans le cas *1-vs-1*, si  $k$  est le nombre de classes, on utilise  $k(k - 1)/2$  classificateurs ; chaque classificateur binaire permettant de discriminer deux classes du problème et tous les couples sont envisagés. Dans le cas *1-vs-all*, on utilise  $k$  classificateurs. Chaque classificateur binaire permet de discriminer une classe du problème contre toutes les autres classes du problème. La décision finale est déduite en comparant les valeurs de décision données par ces classificateurs sur l'exemple interrogé. On utilise souvent le *max* de ces valeurs ou bien on effectue une transformation sous la forme d'un indicateur de probabilité.

### 3.1.3. Limites de la classification dynamique par SVM

La première véritable approche d'apprentissage incrémental avec SVM a été présentée par Syed dans [Syed 1999]. Le principe de cette méthode repose sur le fait que le modèle d'un SVM obtenu en utilisant uniquement les SVs (obtenus au fil du temps par apprentissages successifs) est similaire à celui construit en utilisant l'ensemble des données d'entraînement cumulées. Naturellement la quantité de données manipulées à chaque étape d'apprentissage est considérablement réduite lorsqu'elles sont remplacées par les SVs sélectionnés pour leur pouvoir séparateur. La Figure 12 illustre ce principe : à gauche, toutes les données sont utilisées pour définir la séparation, tandis qu'à droite, seuls les SVs sont utilisés, les deux conduisant à une même séparation à vaste marge.

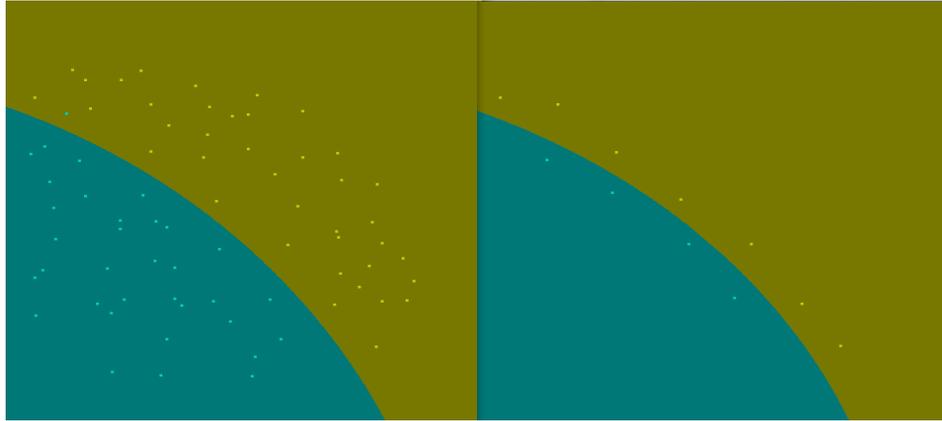


Figure 12 - Principe de la séparation en deux classes par SVM binaire illustrant le principe de Syed : (a) l'ensemble des données complet et leur frontière de séparation (b) uniquement les vecteurs supports et leur frontière de séparation.

Le système de classification proposé par Syed repose sur un principe d'apprentissage incrémental: étant donné un nouvel ensemble de données, le modèle d'une classe formé par SVM repose à la fois sur les nouvelles données (qui représentent les nouvelles connaissances) et sur les SVs provenant de l'étape d'apprentissage précédent (et qui représentent donc les anciennes connaissances).

Les verrous à lever en apprentissage incrémental sont de plusieurs ordres. Tout d'abord on peut citer le problème, commun à de nombreux classificateurs en environnements stationnaires, de la représentativité des frontières de décision. Dans [Rüping 2001], l'auteur montre en quoi le principe d'apprentissage incrémental à partir des SVs pose problème. En effet, les SVs ne maintiennent pas toutes les informations sur la distribution des classes. Il est donc possible de perdre en performance en ne conservant que les SVs d'une étape à la suivante, notamment lorsque les nouvelles observations sont distribuées différemment. Dans ce cas, les SVs retenus vont avoir tendance à s'adapter à ce changement de distribution, les autres SVs étant alors perdus. Il en résulte une évolution dans la description du modèle, peu adaptée aux environnements stationnaires.

Une distribution des données à un instant donné peut avoir un effet néfaste sur les performances du classificateur. Ainsi en renforçant la construction du modèle courant d'un classificateur avec des connaissances antérieures, on peut réduire l'influence d'une distribution hasardeuse ou inattendue des données à un instant donné [Cauwenberghs 2001], [Laskov 2006], [Shilton 2005], [Karasuyama 2010], [Ralaivola 2001]. Dans ces travaux, les auteurs proposent de n'employer qu'une partie des nouvelles données (et pas le lot entier) afin de limiter l'effet de leur distribution sur l'étape d'apprentissage à venir. La stabilité de ces

méthodes est également très liée aux hypothèses de stabilité des données en environnement stationnaire, ce qui n'est pas souvent le cas dans les situations réelles (environnements non-stationnaires).

Trouver un mécanisme adaptable à la fois aux environnements stationnaires et non-stationnaires constitue un enjeu important pour la classification dans les flux de données. Les systèmes comme Learn++.SVM [Muhlbaier 2009] sont réputés pour leur capacité à ajouter incrémentalement de nouvelles classes en environnement stationnaire. Est-il possible de proposer un modèle de classification capable de limiter les oublis à travers le temps, d'ajouter ou supprimer des classes, et d'être adapté aux environnements stationnaires et non-stationnaires ? La solution que nous préconisons pour répondre à ces enjeux multiples (et parfois contradictoires) réside dans la construction d'une multitude de classificateurs *one-class* SVM disposant chacun d'une procédure d'apprentissage incrémental séparée. C'est cette indépendance sur la modélisation des concepts qui va permettre de faire face à une grande diversité de situations.

## 3.2. Introduction aux SVM one-class Incrémentaux (mOC-iSVM)

### 3.2.1. Classification par SVM One Class

Shölkopf dans [Shölkopf 1999] est le premier à avoir présenté le principe de la classification par SVM *one-class* (OC-SVM). Son intérêt a été initialement soulevé dans [Chen 2001] : tous les exemples positifs présentent des ressemblances selon un certain point de vue, tandis que les données négatives peuvent individuellement être différentes mais pour une grande multiplicité de raisons. Autrement dit, ce qui est différent peut différer d'une multitude de façons : un document manuscrit ressemble à un autre manuscrit (exemple dit « positif ») pour son contenu « manuscrit », et ce qui ne l'est pas (les exemples dits « négatifs ») peut différer pour différentes raisons (documents imprimés, graphiques, cartes, photos, *etc.*). Dans tous les cas, les méthodes de classification discriminantes se heurtent à la difficulté de modéliser des frontières complexes rendant compte de la séparation entre des données positives et des données négatives dont il n'est pas aisé d'avoir un échantillon représentatif : l'hyper-surface représentant la « classe A » n'est pas nécessairement celle correspondant à l'espace entier privé de l'hyper-surface correspondant à l'ensemble des classes « non-A ». Toutes les méthodes discriminantes fondées sur une différenciation entre

classes ont cette même limite et ce phénomène est accentué lorsque les données sont issues d'un flux.

Notre questionnement relève alors de la possible indépendance entre la description de la classe A vis-à-vis de toutes les autres, cherchant ainsi à lever un niveau de contraintes important en lien avec la bonne représentativité des données négatives. L'hypothèse fondamentale associée à ce type de problème est qu'il existe une façon de regrouper dans un cluster unique des données positives indépendamment du reste.

Shölkopf présente deux façons pour résoudre le problème : soit en trouvant un hyperplan séparateur, soit par une hyper-sphère regroupant les données de la classe. Nous nous appuyerons sur cette version pour bien illustrer nos idées. L'idée principale d'un classificateur SVM *one-class* consiste ainsi à n'utiliser que des exemples positifs de la classe considérée pour modéliser la frontière de décision. Cela revient à adapter une sphère (dans un espace transformé) incluant un maximum d'exemples positifs. Cette sphère tente de capturer la région dans laquelle les exemples positifs sont regroupés, dans le but de les séparer du « reste du monde ». Le principe du classificateur SVM *one-class* travaille sur la projection des exemples positifs dans un espace de dimension supérieure afin de trouver la sphère la plus petite, qui contient un maximum d'exemples.

Le principe du SVM *one-class* est décrit comme suit :

### 3.2.1.1. Étape d'apprentissage

Supposons qu'on dispose d'un ensemble d'apprentissage de  $n$  exemples de forme  $(a_n, 1)$ . L'ensemble des vecteurs  $X = \{x_1, x_2, \dots, x_n\}$  sont des exemples d'entrée. Notons  $\phi$  la transformation qui transforme l'espace des caractéristiques des exemples en un espace de plus grande dimension  $\mathcal{H}$ . Le paramètre  $\nu \in (0,1)$  représente une limite supérieure sur la fraction des exemples qui peuvent être des exemples aberrants (*outliers*). La variable du ressort  $\xi$  représente l'écart de la fonction objectif pour assouplir les contraintes. On suppose idéalement que les exemples positifs sont regroupés sous forme d'une hyper-sphère de centre  $c$  et de rayon  $R$ . La recherche de la meilleure sphère peut se résoudre par une fonction de minimisation de la quantité suivante (**problème primal**) :

$$\text{minimiser } R^2 + \frac{1}{\nu n} \sum_{i=1} \xi_i$$

sous les contraintes:

$$\| \phi(x_i) - c \|^2 \leq R^2 + \xi_i; \xi_i \geq 0; \forall i \in [1..n]$$

Comme dans le cas du SVM binaire, on peut remplacer le calcul de la transformation  $\phi$  par une fonction noyau plus simple à calculer  $k: x \mapsto h$  (nombres réels) pour traduire la répartition des exemples dans cet espace  $\mathcal{H} : k(x_1, x_2) = \phi(x_1) \cdot \phi(x_2)$ . Et on passe d'un **problème primal** à résoudre à un **problème dual** en introduisant des multiplicateurs de Lagrange pour chaque contrainte. Le **problème dual** se présente sous la forme d'un problème d'optimisation quadratique. Il consiste à minimiser la quantité suivante :

$$\text{Minimiser } \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) - \sum_i \alpha_i k(x_i, x_i)$$

sous les contraintes:

$$\sum_i \alpha_i = 1; \forall i, 0 \leq \alpha_i \leq \frac{1}{vn}$$

### 3.2.1.2. Étape de classification

La classe associée à un exemple est prédit par la fonction de décision qui représente le score du classificateur:

$$f(x) = \text{sign}(R^2 - \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) + 2 \sum_i \alpha_i k(x_i, x) - k(x, x))$$

La valeur de décision représente donc la distance à la frontière de la classe (modélisée par le rayon  $R$ ). Plus cette valeur est grande, plus l'exemple est proche du centre [Chen 2001]. Si elle est négative, on s'éloigne de la sphère. La Figure 13 illustre le principe du OC-SVM. L'algorithme OC-SVM détermine la frontière de décision la plus proche autour des données (points noirs) cibles. Cette frontière est exploitée pour classifier les nouvelles données comme *outliers* (s'ils tombent en dehors des frontières). Concrètement, le problème consiste à trouver la plus petite hypersphère (en rouge) qui peut renfermer un pourcentage  $(1 - \nu)$  prédéfini d'exemples. Cette optimisation peut se reformuler par un changement d'espace défini par un hyperplan (en bleu) le plus éloigné de l'origine et qui assure qu'un pourcentage prédéfini  $(1 - \nu)$  des exemples est de l'autre côté de l'origine. Le  $\nu$  présente la fraction entre les exemples bien classés (les non-SVs) et les exemples mal classés (les *outliers*), renseignant sur la généralité/spécificité du modèle.

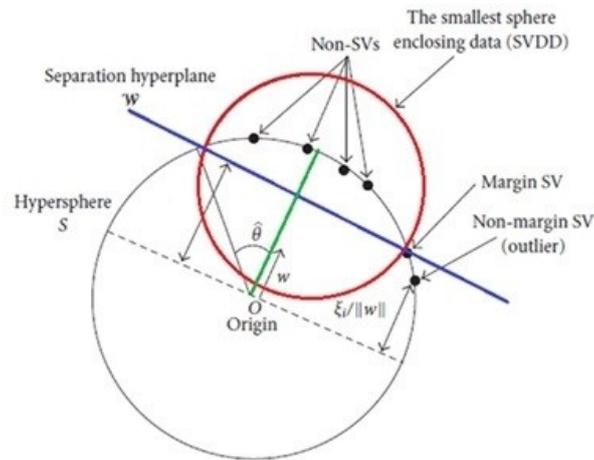


Figure 13 - Illustration de l'algorithme OC-SVM extrait de [Sato 2012]

### 3.2.1.3. Paramétrage

La fonction OC-SVM nécessite deux types de paramètres pour fonctionner :  $\nu$  la limite supérieure sur la fraction des exemples qui peuvent être des exemples aberrants ( $\nu \in (0,1)$ ) et les paramètres du noyau. Comme dans le cas du SVM binaire, on exploite généralement une recherche par grille (*grid search*) durant le processus d'apprentissage pour déterminer les meilleurs paramètres du classificateur *one-class*. Cependant, dans le cas des *one-class*, la recherche de bons paramètres est connue pour être un problème délicat. En effet, autant dans le cas binaire l'usage de la mesure de performance permet de trouver de bonnes solutions, autant ici cela est plus difficile car nous ne disposons *a priori* que d'exemples positifs dans la base d'apprentissage. La Figure 14 illustre cette difficulté sur un *one-class* à noyau gaussien. En se basant uniquement sur les exemples positifs, la deuxième solution semble la meilleure (la frontière inclut tous les exemples). Cependant, peut-être qu'en pratique la première est meilleure pour avoir une plus grande généralisation sur l'acceptation des positifs. *A contrario*, si on souhaite limiter les faux positifs, peut-être vaut-il mieux choisir une frontière plus étroite comme dans les solutions de droite. C'est à cause de cette difficulté (et de leur conséquence : les performances des *one-class* sont souvent inférieures à celles de leurs homologues discriminants) que des données négatives sont assez souvent utilisées, lorsqu'elles sont disponibles, pour trouver les paramètres des *one-class*.

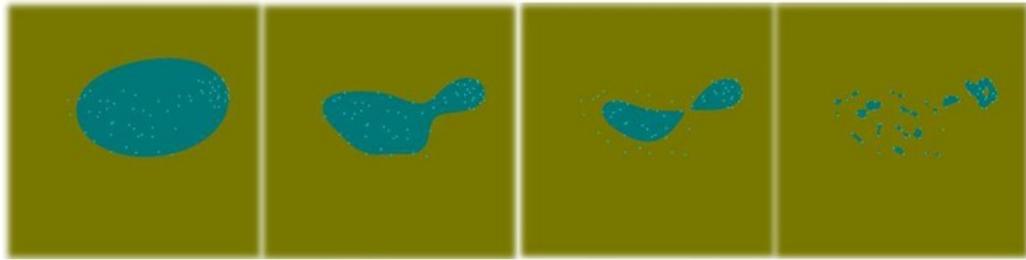


Figure 14 - Le résultat de SVM une classe en utilisant des paramètres différents.

### 3.2.1.4. Intérêt des OC-SVM pour la classification dynamique

La Figure 15 (extraite de [Chen 2001]) rappelle les différences conceptuelles entre les deux approches de classification : discriminante ou *one-class*. En classification par SVM binaire, les SVs sont utilisés pour décrire la frontière de décision (ligne noire continue) séparant les exemples positifs (points bleus) et les exemples négatifs (croix rouges). Par conséquent, la frontière de décision obtenue par le SVM binaire aura tendance à classer (mal) de nombreuses données issues d'autres distributions ou d'autres concepts que ceux utilisés initialement (triangles verts). Contrairement au SVM binaire, la frontière de décision du SVM *one-class* (pointillés bleus) est également la frontière de la classe. Cette propriété est due à l'absence d'informations négatives lors de la détermination de la frontière de décision. Cette modélisation permet donc de détecter plus facilement les modifications de concepts ou l'apparition de nouveaux concepts. Par ailleurs, en utilisant un ensemble de classificateurs indépendants, la modification d'un concept (classe) n'a pas de répercussion directe sur la modélisation des autres.

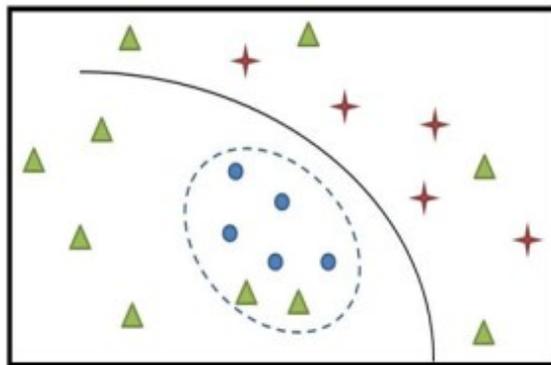


Figure 15 - Comparaison des frontières de décision entre *one-class* SVM (ligne bleue pointillée) et SVM Binaire (ligne noire continue), [Chen 2001]

L'absence d'informations négatives inhérente à la construction du modèle établi par un classificateur *one-class* SVM conduit généralement à des résultats moins précis que ceux qui sont obtenus par classificateurs binaires SVM ([Manevitz 2001]) si l'on considère un problème stationnaire pour lequel une base d'apprentissage représentative est disponible. Il est également bien connu que le processus de sélection des paramètres est complexe ([Hsu 2010]).

Cependant, dans un contexte de classification dynamique où les données sont issues d'un flux et traitées par lots, l'exploitation de plusieurs classificateurs OC-SVM est très pertinente du fait de leur indépendance mutuelle qui facilite la modification des concepts (ajout, division, fusion) au fil du temps. Le OC-SVM permet de créer une frontière de décision par la prise en compte exclusive des informations positives ce qui permet, lors de l'arrivée de données d'une nouvelle classe, de générer un modèle indépendant conçu uniquement à partir des données de cette classe. On peut par ailleurs remarquer que ce système de classification est capable de détecter automatiquement une nouvelle classe non encore modélisée, ce qui est le plus souvent impossible avec des classificateurs purement discriminants. Cette détection peut se prendre par l'interprétation des scores de classification de chaque classificateur *one-class* connu du système. La classification par *one-class* SVM peut enfin présenter d'autres intérêts comme la classification multi-classes, l'exploitation d'un espace de caractéristiques individuelles par classe, etc.

Dans les sections suivantes, nous présenterons une première version incrémentale de classificateurs OC-SVM, que nous nommerons *multi one-class incrémental SVM* (mOC-iSVM) et qui sera mise au service d'un processus de classification dynamique. Nous présenterons ensuite plusieurs variantes conçues pour s'adapter à des environnements non-stationnaires et stationnaires.

### 3.2.2. Les classificateurs indépendants multi *One-Class* SVM incrémentaux (mOC-iSVM) dédiés à l'apprentissage continu

Les propriétés du OC-SVM sont favorables aux contextes dynamiques et facilitent l'apprentissage incrémental : adaptation à l'évolution du nombre de classes, adaptation aux environnements stationnaires et non-stationnaires. Nous présentons ici la version générique d'un classificateur mOC-iSVM conçu comme une combinaison de classificateurs indépendants.

### 3.2.2.1. Principe du multi one-class SVM incrémental standard: mOC-iSVM

L'algorithme d'apprentissage du classificateur multi-classe est basé sur une combinaison de classificateurs SVM *one-class* qui apprennent incrémentalement les nouvelles descriptions des classes en choisissant les nouveaux SVs à partir des modèles précédents (reposant sur les anciens SVs) et des nouvelles données entrantes. Ce mécanisme peut être comparé à l'approche de Syed ([Syed 1999]) : chaque OC-SVM est appris/modifié à chaque étape, en utilisant la procédure d'apprentissage classique de chaque SVM *one-class*. La base d'apprentissage à chaque étape est constituée des SVs issus des apprentissages précédents combinés aux nouvelles données portant la même étiquette que la classe à modéliser (données positives). Lorsque des données ayant une étiquette inconnue apparaissent, il suffit de créer un nouveau SVM *one-class* pour apprendre le nouveau concept, sans aucune incidence directe sur les autres modèles du système.

Ce principe est illustré par la Figure 16. Celle-ci présente 4 étapes d'apprentissage successives associées à un flux. Ce flux contient des données appartenant initialement à deux classes (données « bleues » et données « rouges »). Chaque étape est associée à un lot de données. Dans cet exemple on en compte 4 : Lot 1, Lot 2, Lot 3 et Lot 4 qui arrivent au fur à mesure (ligne du temps  $t$ ). Chaque lot contient  $x$  exemples appartenant aux différentes classes. Dans la 1<sup>ère</sup> étape, le système sépare les données appartenant à chacune des classes. Un OC-SVM est appris sur chaque sous-ensemble de données pour produire un modèle pour chaque classe représentée. Chaque modèle contient les vecteurs supports de la classe correspondante. L'ensemble des modèles des classes constitue donc le modèle multi-classes global du système capable de traiter (classer) les données du flux qui arriveront ensuite.

Pour le test, on considère ici, comme dans les benchmarks que nous utiliserons ci-après pour l'évaluation, qu'à chaque étape d'apprentissage un test est associé à un lot de tests indépendant. Le modèle issu de l'apprentissage est donc utilisé pour étiqueter les exemples du lot de tests, ce qui permet d'évaluer les performances du système lors de cette étape. Ainsi, les modèles de l'étape 1 serviront à classer les données arrivant jusqu'à l'étape 2. Pour cela, pour chaque donnée à classer, chaque classificateur *one-class* SVM produit un score d'appartenance à la classe qu'il représente et une décision finale est prise en agrégeant ces résultats (en violet sur la Figure 17). Nous proposons ici d'utiliser par défaut le *max* des valeurs de décision :  $C = \operatorname{argmax} f^i(x)$  avec  $f^i(x)$  la fonction de décision qui représente

le score du classificateur  $i$  sur l'exemple  $x$ . On peut également décider d'utiliser un seuil de confiance (pouvant conduire éventuellement à un rejet).

Lors de la deuxième étape, à l'arrivée du second lot de données étiquetées, le système récupère également tous les vecteurs supports issus du modèle précédent (SVs 1 - rectangle orange). Cet ensemble constitué des SVs précédents et des nouvelles données va servir pour la seconde étape d'apprentissage. Celle-ci se passe de la même manière qu'à l'étape 1 : le système apprend séparément les modèles représentatifs des classes en présence en n'utilisant que les données/SVs positifs correspondants. Les nouveaux modèles remplacent les précédents. Le mécanisme se poursuit ensuite de la même façon. Lors d'une étape d'apprentissage, lorsque que des données sont étiquetées avec un label inconnu (données violettes à l'étape 3), elles servent à apprendre un nouveau OC-SVM dédié à cette nouvelle classe. Celui-ci est intégré au système et géré ensuite de la même façon que les autres.

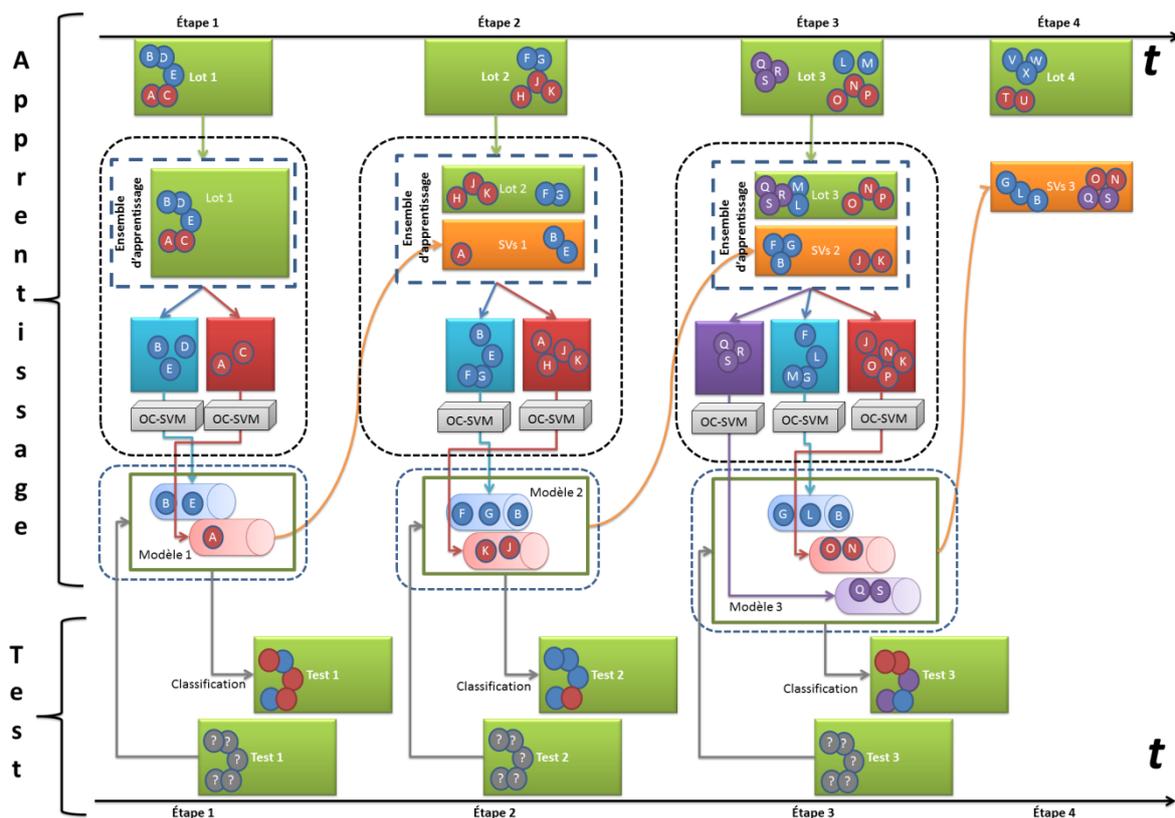


Figure 16 - Schéma de la procédure d'apprentissage du mOC-iSVM

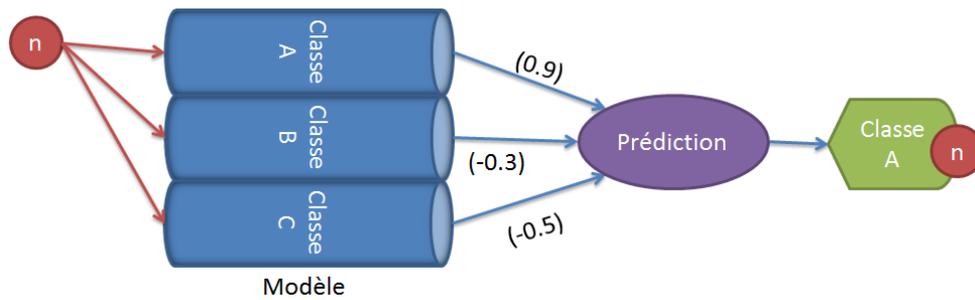


Figure 17 - Classification d'une donnée (notée n) par mOC-iSVM

### 3.2.2.2. Comparaison du mOC-iSVM à une méthode par ensemble de SVM et au SVM multi-classes *1-vs-all*

Nous allons préciser dans cette section quelles sont les différences importantes entre le mOC-iSVM et d'une part un ensemble de SVMs binaires fusionnés (par exemple par un vote majoritaire), et d'autre part un SVM multi-classe *1-vs-all* qui a une structure très proche du mOC-iSVM. Rappelons d'abord que la performance d'une approche par ensemble de classificateurs vient de la combinaison des informations de chaque classificateur via un vote majoritaire (ou une loi de combinaison spécifique). Cela peut devenir une faiblesse dans le cadre du traitement d'un flux. En effet, la modélisation étant répartie sur l'ensemble, il faudra vraisemblablement mettre à jour un grand nombre de classificateurs de l'ensemble à chaque étape, ce qui nécessite à la fois du temps de calcul mais aussi une quantité de données d'apprentissage suffisante pour effectuer les changements adéquats au sein des classificateurs du système. Ces phénomènes ont été observés dans la littérature et les expérimentations présentées plus loin le montrent également.

Ce problème est moins présent dans le mOC-iSVM et le SVM multi-classes *1-vs-all* car le modèle d'une classe est « enregistré » dans un classificateur unique. Dans ce type de configuration, le nombre de classes est égal au nombre de classificateurs du système. Les performances de ces deux systèmes sont donc dépendantes des performances individuelles de chaque classificateur. La différence entre le mOC-iSVM et le SVM multi-classes *1-vs-all* est que, dans ce dernier, les performances de chaque classificateur sont fortement dépendantes à la fois des informations des exemples de sa classe mais également des exemples des autres classes (les données négatives). Cette interdépendance peut aboutir à des frontières difficiles à modéliser ou peu pertinentes lorsque les contre-exemples sont peu nombreux ou insuffisamment représentatifs de l'ensemble des classes. Dans le cas du mOCi-SVM, de par la

contrainte sur la construction du modèle, chaque exemple négatif est une aide pour fixer les paramètres et définir les frontières mais celles-ci n'en restent pas moins dépendantes avant tout des exemples positifs et contraintes par ces derniers, limitant les phénomènes de « sur-généralisation ».

### 3.2.2.3. Évaluation de performances

Nous utilisons différents types d'évaluation de performances:

- Pour la classification sur un problème à plusieurs classes (ensemble de *one-class*), nous pouvons, pour nous comparer avec d'autres approches classiques comme dans [Mulhabaier 2009], utiliser la règle de décision finale définie plus haut ( $Classe = \operatorname{argmax} f^i(x)$  avec  $f^i(x)$  la fonction de décision qui représente le score du classificateur  $i$  sur l'exemplaire  $x$ ) pour qu'un exemple ne puisse être classé que dans une classe. Dans ce cas, le taux de reconnaissance de chaque classe (*Accuracy* de  $i$  :  $AC_i$ ) est défini par l'Équation 1

Équation 1

$$AC_i = \frac{\# \text{ Vrais positifs de la classe } i + \# \text{ Vrais négatifs de la classe } i}{\# \text{ de données de test}}$$

Le taux de reconnaissance du système complet (*Global Accuracy* ou  $GA$ ) est obtenu par :

Équation 2

$$GA = \frac{\sum_i AC_i}{i}$$

Cette mesure peut être agrégée pour obtenir la performance sur un flux en effectuant la moyenne sur  $n$  étapes (*Stream Accuracy* :  $SAC$ ). Cette mesure permet notamment de comparer notre approche avec les résultats dans [Elwell 2011] sur des problèmes binaires.

Équation 3

$$SAC = \frac{\sum_1^n GA}{n}$$

- Nous pouvons également ne pas utiliser la règle de classification du *max* et accepter qu'un même exemple puisse ne pas être classé du tout (rejet). Cette façon

d'évaluer les performances nous sera notamment utile pour évaluer notre système dans 3.2.3. Pour cela, nous définissons les performances de chaque one-class indépendamment (*Monoclass Accuracy* ou  $MAC_i$ ) :

Équation 4

$$MAC_i = \frac{\# \text{Vrais positifs de classe } i + \# \text{Vrais négatifs de classe } i}{\# \text{ de données de test}}$$

La façon de calculer  $MAC_i$  est identique à la précédente (Équation 1) mais la répartition des données dans chaque classe est différente pour chaque *one-class*. C'est pourquoi nous utilisons une autre dénomination. Le taux de reconnaissance global du système (*Global Monoclass Accuracy* ou  $GMA$ ) se calcule de la même façon que ci-dessus:

Équation 5

$$GMA = \frac{\sum_i^i MAC_i}{i}$$

- L'inconvénient de la mesure précédente est qu'elle est sensible à la proportion de données de chaque classe dans la base de test. Sur les benchmarks classiques, les classes sont le plus souvent équilibrées et les mesures  $MAC_i$  et  $GMA$  ne posent pas plus de problème. En revanche, sur des problèmes plus réels, nous n'avons pas forcément la même garantie. Il vaut mieux alors utiliser un taux de reconnaissance équilibré pour chaque classe (*Balanced Accuracy* :  $BA_i$ ) [Brodersen 2010] :

Équation 6

$$BA_i = \frac{\text{sensibilité} + \text{spécificité}}{2}$$

$$BA_i = \frac{0.5 \times \# \text{Vrais positifs de la classe } i}{\# \text{Vrais positifs de la classe } i + \# \text{Faux négatifs de la classe } i} + \frac{0.5 \times \# \text{Vrais négatifs de la classe } i}{\# \text{Vrais négatifs de la classe } i + \# \text{Faux positifs de la classe } i}$$

La performance globale sur toutes les classes (*Global Balanced Accuracy* :  $GBA$ ) est donnée par :

Équation 7

$$GBA = \frac{\sum_i^i BA_i}{i}$$

- Nous utilisons aussi la précision globale (*Global PRecision* : *GPR*), la sensibilité globale ou le rappel global (*Global SeNsibility* : *GSN*) et la spécificité globale (*Global SPecificity* : *GSP*) sur toutes les classes.

Équation 8

$$GSN = \frac{\sum_i^i \text{sensibilité}_i}{i} = \frac{\sum_i^i \frac{\#Vrais\ positifs\ de\ la\ classe\ i}{\#Vrais\ positifs\ de\ la\ classe\ i + \#Faux\ négatifs\ de\ la\ classe\ i}}{i}$$

Équation 9

$$GSP = \frac{\sum_i^i \text{spécificité}_i}{i} = \frac{\sum_i^i \frac{\#Vrais\ négatifs\ de\ la\ classe\ i}{\#Vrais\ négatifs\ de\ la\ classe\ i + \#Faux\ positifs\ de\ la\ classe\ i}}{i}$$

Équation 10

$$GPR = \frac{\sum_i^i \text{précision}_i}{i} = \frac{\sum_i^i \frac{\#Vrais\ positifs\ de\ la\ classe\ i}{\#Vrais\ positifs\ de\ la\ classe\ i + \#Faux\ positifs\ de\ la\ classe\ i}}{i}$$

#### 3.2.2.4. Optimisation des paramètres $v$ et $g$

Pour sélectionner les meilleurs paramètres, nous pouvons utiliser l'approche de recherche par grille classique (*cf.* section 3.2.1.3 et 3.1.2.4) à chaque étape d'apprentissage. Pour cette recherche, nous avons vu précédemment que nous pouvions, soit utiliser des critères d'optimisation basés sur le taux de vrais positifs seul, soit utiliser un critère prenant aussi en compte les faux positifs/négatifs. Ainsi, on peut utiliser : la sensibilité (ou le rappel) qui mesure la proportion d'exemples positifs correctement identifiés ; la spécificité qui mesure la proportion d'exemples négatifs correctement identifiés ; la précision qui mesure la proportion d'exemples positifs parmi les exemples classés comme positifs... On peut également combiner ces mesures. Dans notre cas, lorsque des données négatives sont disponibles dans le lot, nous choisissons d'utiliser deux critères : soit la moyenne arithmétique de la sensibilité et de la spécificité ; soit le compromis entre la sensibilité et la spécificité (i.e. lorsque la sensibilité égale la spécificité). Si les exemples négatifs ne sont pas disponibles, on utilisera la sensibilité seule.

Cette optimisation des paramètres devrait normalement se faire à chaque étape de l'apprentissage (méthode GSES-*Grid Search at Each Step*). Cependant, pour limiter la complexité et le temps de calcul dans les processus d'apprentissage longs (beaucoup d'étapes d'apprentissage), nous pouvons aussi utiliser la recherche par grille appliquée uniquement aux premières étapes du processus d'apprentissage (méthode GSFS - *Grid Search at Firsts Steps*). On réutilise ensuite ces paramètres dans les étapes suivantes en faisant **l'hypothèse de leur généricité**. Bien entendu cette dernière version n'est pas optimale et peut être dérivée (*Grid Search Every X Steps, ...*) mais nous ne traitons pas ces cas.

### 3.2.2.5. Algorithme d'apprentissage du mOC-iSVM

Le pseudo-algorithme d'apprentissage du *Multi One-Class Incremental SVM* (mOC-iSVM) est décrit ci-dessous :

Soit les notations suivantes :

- $T^1, T^2 \dots T^i$  sont des lots de données qui arrivent aux temps  $t^1, t^2 \dots t^i$ .
- $C_{\text{system}} = \{C1, C2 \dots Cj\}$  est la liste des classes que connaît le système à un instant donné. Cette liste peut augmenter si de nouvelles classes arrivent au cours du temps (données portant de nouvelles étiquettes).
- $SV_{\text{system}} = \{SV_{\text{system}}^1, SV_{\text{system}}^2 \dots SV_{\text{system}}^i\}$  est l'historique des ensembles de vecteurs supports enregistrés dans le système aux temps  $t^1, t^2 \dots t^i$ .
- $SV_{\text{system}}^i = \{SV_{\text{system}}^i C1, SV_{\text{system}}^i C2 \dots SV_{\text{system}}^i Cj\}$  est l'ensemble des vecteurs supports associés à chaque classe de  $C_{\text{system}}$  au temps  $t^i$ . Ainsi,  $SV_{\text{system}}^i Cj$  est l'ensemble des vecteurs supports de la class  $Cj$  au temps  $t^i$ .
- $M_{\text{system}} = \{M_{\text{system}}^1, M_{\text{system}}^2 \dots M_{\text{system}}^i\}$  est l'historique des modèles enregistrés au cours du temps (de  $T^1$  à  $T^i$ ).
- $M_{\text{system}}^i = \{M_{\text{system}}^i C1, M_{\text{system}}^i C2 \dots M_{\text{system}}^i Cj\}$  est l'ensemble des modèles pour chaque classe de  $C_{\text{system}}$  au temps  $t^i$ .  $M_{\text{system}}^i Cj$  est le modèle de la classe  $Cj$  au temps  $t^i$ .
- $\nu, g$  sont les paramètres de la fonction de *one-class SVM* :  $\nu$  est le paramètre équivalent à la fraction du nombre d'erreurs sur le nombre de vecteurs supports,  $\nu \in (0,1]$  ;  $g$  est le paramètre du noyau Gaussian,  $g \in (0, \infty)$ .
- $D_{\text{train}}$  est un lot de données pour faire l'apprentissage du système à un instant donné.

- $D^i$  sont les données reçues au temps  $T^i$ . Ces données sont étiquetées avec des labels  $C_k$ .
- $D^i C_j$  sont les données de  $D^i$  appartenant à la classe  $C_j$ . Cet ensemble ne contient que des données positives par rapport à  $C_j$ .
- $D^i \bar{C}_j$  sont les données de  $D^i$  appartenant aux classes autres que  $C_j$ . Elles correspondent aux données négatives de  $C_j$  (notées :  $\bar{C}_j$ ).
- Grid-Search ( $D_{\text{train}}$ ,  $D_{\text{validation\_positive}}$ , [ $D_{\text{validation\_negative}}$ ]) est la fonction de recherche des paramètres optimaux qui utilise un lot de données d'apprentissage et un lot de données positives (et négatives de façon optionnelle).
- Train( $(v, g)$ ,  $D_{\text{train}}$ ) est la fonction d'apprentissage qui utilise les données  $D_{\text{train}}$  avec des paramètres  $v$  and  $g$ .
- Test( $M_{\text{system}}^m C_j$ ,  $D_{\text{test}}$ ) est la fonction de test qui utilise les données de test avec le modèle  $M_{\text{system}}^m C_j$ . La sortie de cette fonction est une mesure de performance : *GA, GBA, GMA etc...*

### //ETAPE 1 : Détecter des nouvelles classes

// Si dans le lot courant, une étiquette  $C_k$  n'existe pas dans  $C_{\text{system}}$

Pour tout  $C_k$  dans  $D^i$

Si  $C_k$  not-in  $C_{\text{system}}$

// Ajouter une nouvelle classe dans l'ensemble des classes

$C_{\text{system}} = C_{\text{system}} \cup C_k$ ;

// Ajouter un espace pour les vecteurs supports de cette classe

$SV_{\text{system}}^i C_k = \{\emptyset\}$ ;

$SV_{\text{system}}^i = SV_{\text{system}}^{i-1} \cup \{SV_{\text{system}}^i C_k\}$ ;

// Augmenter le nombre de classes total dans le système.

$j = j + 1$ ;

Fin

**// ETAPE 2 : Créer des données d'apprentissage (positives et négatives) pour chaque classe  $C_j$  dans le lot de données reçu à  $T^i$**

$D^i \rightarrow \{D^i C_1, D^i C_2 \dots D^i C_j\}$

$D^i \rightarrow \{D^i \bar{C}_1, D^i \bar{C}_2 \dots D^i \bar{C}_j\}$

**//ETAPE 3 : Mettre à jour le système (pour chaque classe de  $D^i$ )**

Pour chaque  $D^i C_j$  non vide //on met à jour le modèle de  $C_j$

**//ETAPE 3-1 : Sélection des SVs anciens de cette classe dans le modèle précédent**

$$SV_{\text{best}} = SV_{\text{system}}^{i-1} C_j$$

**//ETAPE 3-2 : Regrouper les SVs sélectionnés et les données positives de cette classe**

**dans  $D_{\text{train}}$**

$$D_{\text{train}} C_j = SV_{\text{best}} C_j \cup D^i C_j;$$

**//ETAPE 3-3 : Sélection des paramètres**

$$(v, g) = \text{GridSearch}(D_{\text{train}} C_j, D^i C_j, D^i C_j)$$

**//ETAPE 3-4 : Déterminer les nouveaux SVs de la classe  $C_j$**

$$[SV_{\text{system}}^i C_j, M_{\text{system}}^i C_j] = \text{Train}((v, g), D_{\text{train}} C_j);$$

**// ETAPE 3-5 : Enregistrer des SVs dans l'historique du système**

*//Constituer le modèle final  $M_{\text{system}}^i$  grâce à l'ensemble des classificateurs one-class*

$$M_{\text{system}}^i = \{M_{\text{system}}^i C_1, M_{\text{system}}^i C_2 \dots M_{\text{system}}^i C_j\}$$

$$SV_{\text{system}} = SV_{\text{system}} \cup SV_{\text{system}}^i C_j$$

$$M_{\text{system}} = M_{\text{system}} \cup M_{\text{system}}^i C_j$$

Fin

**//ETAPE 4 : Mettre à jour le modèle actuel**

*// Mettre à jour le modèle utilisé actuel*

$$M_{\text{actual}} = M_{\text{system}}^i$$

Algorithme 1 - Algorithme d'apprentissage du mOC-iSVM

### 3.2.3. Évaluation du mOC-iSVM et de sa capacité d'incrémentalité en traitement « par lot »

L'objectif de cette expérimentation est de prouver la capacité du mOC-iSVM à capturer de nouvelles informations venant des données à travers le temps et à les intégrer dans la modélisation de ses classificateurs. Il s'agit ainsi de montrer que le mécanisme d'apprentissage incrémental conduit à des performances de classification aussi bonnes que celles obtenues avec un apprentissage complet portant sur la totalité des données.

#### 3.2.3.1. Environnement de développement et de tests

Pour implémenter notre approche mOC-iSVM, nous avons utilisé la librairie Matlab LibSVM [Chang 2011]. Pour les tests, nous avons utilisé une machine avec un processeur Intel(R) Core(TM) i5 CPU m460, 2.53 GHz duo, mémoire RAM de 4.00Go sur un système d'exploitation 64 bits Windows 7 (2009-P1).

### 3.2.3.2. Description du jeu de données de test

La base utilisée dans ces expérimentations est la base *UCI Optical Recognition of Handwritten Digits Dataset* (ORHDD)<sup>5</sup> [Kaynak 1995]. En effet, cette base, outre le fait d'être un benchmark classique, a également été utilisée par d'autres auteurs pour évaluer des systèmes incrémentaux. Nous la réutiliserons donc plus tard pour situer nos approches par rapport à la littérature. On notera au passage que cette base correspond à un problème stationnaire.

Dans cette base, l'auteur a utilisé les programmes de prétraitements mis à disposition par le NIST<sup>6</sup> pour extraire des images bitmaps normalisées de chiffres manuscrits issues d'un formulaire pré-imprimé. Le nombre de classes de chiffres est 10 (nommées de 0 à 9 ci-dessous). Sur un total de 43 scripteurs, on en retient 30 pour constituer la base d'apprentissage et 13 pour la base de test. Chaque image, de taille 32x32 pixels, est divisée en 8x8 blocs de dimension 4x4 pixels ne se chevauchant pas. Ce découpage génère une matrice d'entrée de 64 attributs correspondant au nombre de pixels noirs dans chaque bloc (une valeur entre 0 et 16). Ceci réduit la dimension et garantit une meilleure invariance aux petites déformations. Ceci réduit la dimension et garantit une meilleure invariance aux petites déformations.

Les expériences sont menées avec 3823 images pour l'ensemble d'apprentissage et 1797 pour l'ensemble de test. La distribution des exemples par classe  $y$  est également indiquée :

	Classe 0	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7	Classe 8	Classe 9	Total
L'ensemble d'apprentissage	376	389	380	389	387	376	377	387	380	382	3823
L'ensemble de tests	178	182	177	183	181	182	181	179	174	180	1797

Table 1 - Distribution des exemples par classe de la base UCI ORHDD, [Kaynak 1995]

### 3.2.3.3. Description du protocole de test

Une expérimentation repose sur un processus en trois étapes d'apprentissage et de test. On utilise le même ensemble de tests pour les trois tests de l'expérimentation. En revanche, l'ensemble d'apprentissage (3823 exemples) est divisé en trois lots correspondant à 3 étapes d'apprentissage successives. On réalise l'évaluation globale du système en répétant l'expérimentation selon une validation croisée en 5 plis). En ce qui concerne les paramètres de

<sup>5</sup> <https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits>

<sup>6</sup> National Institute of Standards and Technology - [www.nist.gov](http://www.nist.gov)

l'*one-class* SVM, tant pour l'approche classique que pour notre approche incrémentale, nous avons utilisé une fonction *noyau* RBF classique. Les paramètres ( $\nu$  et  $g$ ) sont sélectionnés par une recherche par grille appliquée à chaque étape (méthode GSES-*Grid Search at Each Step*). Pour l'évaluation, nous avons utilisé les taux de reconnaissance monoclasses :  $MAC_i$  (cf. Équation 4) et  $GMA$  (cf. Équation 5).

Afin d'évaluer le caractère incrémental de notre approche, nous la comparons à des *one-class* SVM classiques (*i.e.* avec réapprentissage complet à chaque étape sur l'ensemble des lots présentés jusqu'ici).

#### 3.2.3.4. Interprétation des résultats

Globalement, on peut observer, à la Figure 18, que le résultat du mOC-iSVM est presque équivalent à celui du multi-OC-SVM (mOC-iSVM conduit à des taux de reconnaissance de 96.94% dans la dernière étape alors que le multi-OC-SVM obtient 97.13%). A titre de comparaison, un test réalisé avec un classificateur *k-nn* (sur toutes les données) utilisant la distance Euclidienne conduit à un taux de reconnaissance de 98% ( $k = 1$ ).

À chaque étape, le mOC-iSVM peut capturer des informations supplémentaires à travers les nouveaux exemples tout en préservant les connaissances déjà acquises. Cela se manifeste par une augmentation des performances à chaque étape (91.29% → 95.52% → 96.94%). Les performances relatives à chaque classe augmentent différemment à chaque étape : les scores associés aux classes 4 et 3 augmentent très rapidement (plus de 7%), tandis que celui de la classe 2 n'augmente que de 4%. Le taux de reconnaissance dans l'étape finale n'est pas le même non plus pour chaque classe : alors que la classe 8 n'obtient que 95% de reconnaissance environ (score le plus faible), la classe 0 obtient, elle, près de 99% de reconnaissance. Dans tous les cas, le comportement est similaire pour le système incrémental et non incrémental. Ce résultat est donc encourageant.

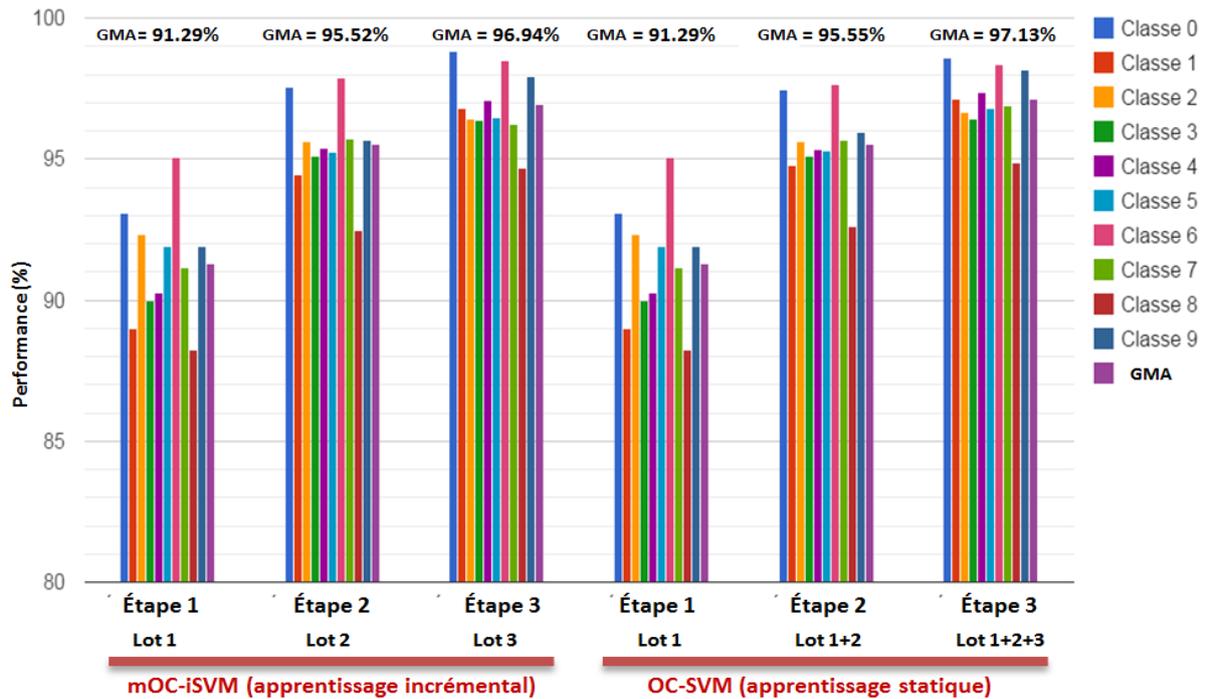


Figure 18 - Étude des performances ( $MAC_i$  et GMA) du mOC-iSVM et d'OC-SVM en trois étapes.

### 3.2.4. Évaluation du mOC-iSVM et de sa capacité d'incrémentalité « en ligne »

L'objectif de cette expérimentation est de prouver la capacité du mOC-iSVM à capturer les évolutions des observations d'entrée en ne considérant que les informations une par une. Cette version dite *on-line* permet de simuler un processus de reconnaissance incrémental sur une ligne continue de traitement de données où il peut être nécessaire de considérer les données au fur et à mesure de leur arrivée, non plus par lot mais individuellement. Le jeu de données considéré est le même qu'à la section précédente (*UCI Optical Recognition of Handwritten Digits Dataset*, [Kaynak 1995]) mais le protocole va subir des modifications importantes.

#### 3.2.4.1. Description du protocole testé

On reprend dans ces expérimentations l'ensemble de tests fixe de 1797 exemples, et une partie de l'ensemble d'apprentissages (3000 exemples tirés au hasard sur les 3832 exemples). Les données d'apprentissage seront présentées une par une et un apprentissage sera réalisé à chaque fois. L'expérimentation complète est refaite cinq fois avec une base d'apprentissages différents (exemples et ordre des exemples). Par ailleurs, nous comparons

l'apprentissage en ligne avec des apprentissages par lots de 50 et 100 données. Les paramètres ( $v$  et  $g$ ) sont déterminés par une recherche par grille appliquée à chaque étape (méthode GSES-*Grid Search at Each Step*) dans le cas des lots de 50 et 100 données. Dans le cas en ligne, on applique la méthode GSFS (*Grid Search at First Steps*) en se basant sur les 50 premières étapes pour fixer les paramètres ( $v$  et  $g$ ). Pour l'évaluation, nous utilisons la mesure *GBA*; Équation 7.

#### 3.2.4.2. Interprétation des résultats

Comme on peut le constater à la Figure 19 qui présente une moyenne de dix expériences obtenue sur tests réalisés sur des lots aléatoires, les performances des classificateurs one-class sont très sensibles à la quantité de données d'apprentissage. Cela a également été montré dans les travaux de Sato dans [Sato 2012]. Dans le mode en-ligne, dans les premières étapes, les scores sont globalement inférieurs à ceux observés dans le traitement par lot. Les résultats sont par ailleurs plus instables (dans les 300 premiers exemples) présentant des écarts plus grands. Cela s'observe en particulier dans les premières étapes (cas des premiers lots de 50 et 100 exemples), où les performances moyennes sont plus stables.

On peut constater que la capacité d'apprentissage incrémental du mOC-iSVM reposant sur la sélection de vecteurs supports performants, est vérifiée au fil du processus avec une augmentation des performances et une diminution des écarts types. Finalement, les deux approches « par lot » et *on-line* finissent par converger vers des résultats de classification identiques au bout d'un certain temps.

On peut souligner que de façon générale, les expérimentations que nous avons conduites nous ont permis de conclure que le mOC-iSVM en mode *on-line* était globalement moins stable au démarrage du processus d'apprentissage (lorsque la quantité de données est très faible) mais que les performances ne cessent de croître pour rejoindre les meilleurs scores du mOC-iSVM en traitement par « lot ». Cela nous permet d'envisager des applications dans des chaînes de dématérialisation et de numérisation en ligne que nous détaillerons au chapitre 4 de ce manuscrit.

On peut enfin faire le constat que les performances du classificateur mOC-iSVM sont assez indépendantes du choix de la taille du lot (même si on observe un léger bénéfice en faveur des lots les plus grands, ce qui n'est pas surprenant). Les performances du classificateur mOC-iSVM sont, dans tous les cas, totalement superposables à celles obtenues

par un multi one-class SVM intégrant en une seule fois l'ensemble des données d'apprentissage.

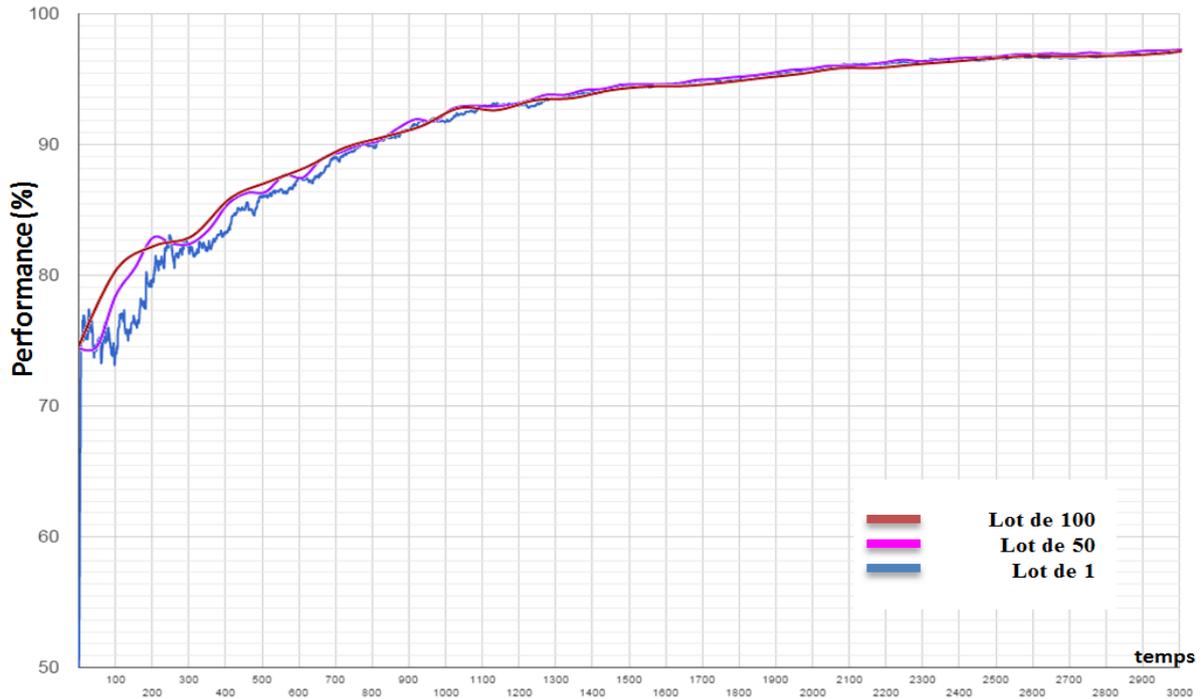


Figure 19 - Courbe d'évolution de la performance du mOC-iSVM avec différentes dimensions de lot de données

### 3.3. Adaptations du mOC-iSVM selon la nature de l'environnement (stationnaire ou non)

Pour améliorer les performances du mOC-iSVM à la fois en environnement stationnaire et non-stationnaire, il nous a semblé opportun de mettre en place une méthode capable de s'adapter rapidement à l'évolution des concepts, tout en préservant une bonne capacité d'apprentissage au fur et à mesure de l'arrivée des données. Ces nouvelles formes de classificateurs issus de la famille des mOC-iSVM devront ainsi profiter de la richesse de l'historique des vecteurs supports qui définissent les classificateurs one-class individuellement. La question de la performance et de l'adaptation des classificateurs peut alors se reformuler de la façon suivante : « Comment ne sélectionner que les meilleurs vecteurs supports candidats (SVs) parmi les anciens SVs issus des modèles précédents pour garantir la meilleure adaptation aux données ? », autrement dit « comment proposer les meilleures configurations de modèles à partir de l'historique des modèles ? ».

Pour assurer ces adaptations au contexte, différentes propositions ont été faites dans le domaine. On peut notamment citer les travaux relevant de la combinaison de classificateurs, comme ceux de Wang et Street dans [Wang 2003] et [Street 2001]. Les auteurs ont proposé de configurer les classificateurs SVM à partir d'une **sélection portant, d'une part, sur l'âge des SVs** et, d'autre part, **sur la performance des SVs**. Il faut noter que chaque approche a son avantage, dépendant de la nature de l'environnement et du contexte. Le meilleur choix est finalement un compromis (ou une combinaison) entre les deux techniques. Dans cette section, nous allons exploiter les deux approches de construction des modèles reposant sur une sélection des vecteurs supports en fonction de leur âge et en fonction des performances induites.

### 3.3.1. Sélection des vecteurs supports selon l'âge : version mOC-iSVM.AP

#### 3.3.1.1. Motivation et principe

Comme nous allons le montrer, une des limitations du mOC-iSVM est liée à l'incapacité des classificateurs one-class à sélectionner de façon ajustée les bons vecteurs supports issus des configurations antérieures. Pour les environnements non-stationnaires, cela peut conduire à diminuer la précision du système en conservant des vecteurs supports qui ne sont plus représentatifs des nouvelles données. Afin d'éviter d'introduire involontairement des données erronées ou des données mal adaptées à l'évolution de l'environnement au moment de l'apprentissage, nous avons choisi de partir de l'hypothèse que les derniers SVs trouvés sont potentiellement les représentatifs des dernières variations de l'environnement. Cette approche de sélection des supports vecteurs s'apparente à ce qui est communément nommé *age-pruning selection*.

Nous avons exploré ce principe de la sélection par l'âge et conçu un modèle de classificateurs que nous nommerons mOC-iSVM.AP par la suite pour *Multi One-Class Incremental SVM with Age Pruning*. Le principe repose sur le fait que chaque SV enregistré est associé à un âge (que nous noterons  $\alpha$  par la suite) correspondant à l'ordre de l'apparition du SV dans le système. Seuls les SVs qui ont un âge antérieur à une valeur fixée (population de vecteurs supports plus récente) seront utilisés dans la procédure d'apprentissage du nouveau modèle associé aux nouvelles données du lot. L'avantage de cette solution en

environnement non-stationnaire, est que l'on peut partiellement ou totalement limiter l'influence des anciens vecteurs supports sur des données d'apprentissage récentes.

Le mécanisme de sélection des vecteurs supports selon leur âge est aussi une technique qui permet d'améliorer les performances générales d'un système en environnement stationnaire en utilisant l'âge le plus grand, c'est-à-dire celui qui correspond aux périodes des premiers apprentissages. En d'autres termes, tous les SVs qui ont été sélectionnés au fil du temps peuvent être ré-exploités pour réactualiser le modèle des classificateurs multi-classes.

Il a été montré que les méthodes discriminantes de classification incrémentale étaient sensibles à la distribution des données pour chaque classe considérée. En effet, elles reposent souvent sur le principe d'une adaptation rapide aux nouvelles données conduisant à un oubli important des informations des modèles précédents. Dans le mOC-iSVM, la distribution des données dans chaque lot est également prise en compte dans l'élaboration des modèles one-class. Les variations de la distribution des données au fil du temps influencent donc la composition des modèles et les décisions du système. Avec la sélection des SVs selon leur âge, l'amplitude de l'historique (largeur de la fenêtre temporelle) est ainsi contrôlée ce qui peut être déterminant pour augmenter (ou non) la capacité d'oubli du système. Bien entendu, tous les SVs anciens ne seront pas nécessairement repris dans le nouveau modèle : c'est la procédure d'apprentissage qui se chargera de faire le tri.

Le choix du facteur âge est d'autant plus important dans un environnement non-stationnaire, que s'il est mal déterminé, il peut conduire à la prise en compte de vecteurs supports totalement inappropriés qui auront tendance à modifier les frontières de décisions des classificateurs et à conduire à des erreurs d'étiquetage. En ne tenant compte que des changements les plus récents ( $\alpha=1$ ), le mOC-iSVM semble assurer une adaptation quasi immédiate au nouveau contexte. Nous vérifierons ces hypothèses lors des expérimentations ci-après.

Le choix de l'âge est une difficulté de cette approche. En effet, on peut vite s'apercevoir que son intérêt n'est pas systématique et est très lié à la nature du contexte. En effet, dans un environnement non-stationnaire réoccurrent (les données traitées présentent des variations intra-classe fortes mais, au fil du temps, elles peuvent reprendre « l'allure » de données traitées plus tôt), la méthode de sélection des vecteurs supports selon leur *âge* peut ne plus pouvoir réutiliser certaines informations utiles provenant de SVs plus anciens si leur âge est plus grand que l'âge choisi pour effectuer l'élagage courant. Par conséquent, si l'âge choisi est petit, le système augmente sa capacité d'adaptation au contexte le plus récent mais

va également limiter sa capacité à reconnaître des données similaires à des informations plus anciennes.

En environnement stationnaire, le choix de l'âge n'est pas facile non plus, notamment pour des raisons de complexité. En effet, si l'âge est grand, le système va mécaniquement augmenter son temps d'exécution et son coût de stockage. Il convient donc d'identifier en amont, dans la mesure du possible, la nature du contexte et de l'environnement des données pour choisir la valeur de l'âge seuil adéquat. Dans les expérimentations que nous ferons dans les sections suivantes, nous proposons d'utiliser un âge faible pour l'environnement non-stationnaire et un âge plus grand pour l'environnement stationnaire, tentant ainsi d'augmenter l'efficacité du système selon l'environnement rencontré. On s'aperçoit qu'introduire une technique de recherche automatique de l'âge optimal peut s'avérer utile pour perfectionner cette approche et optimiser les résultats. Des travaux portant sur des approches dites de « fenêtres temporelles » (en considérant que la sélection de l'âge est un type de fenêtre glissante sur les modèles anciens) pourraient être considérés ici. Une autre approche plus systématique consiste à tester pour chaque étape l'ensemble des vecteurs supports issus des modèles antérieurs et considérer comme critère de sélection de l'âge celui qui conduit à la première diminution de performance (en environnement non-stationnaire) ou à aucune augmentation significative des performances (en environnement stationnaire).

### 3.3.1.2. Schéma de principe de l'apprentissage continu du mOC-iSVM.AP

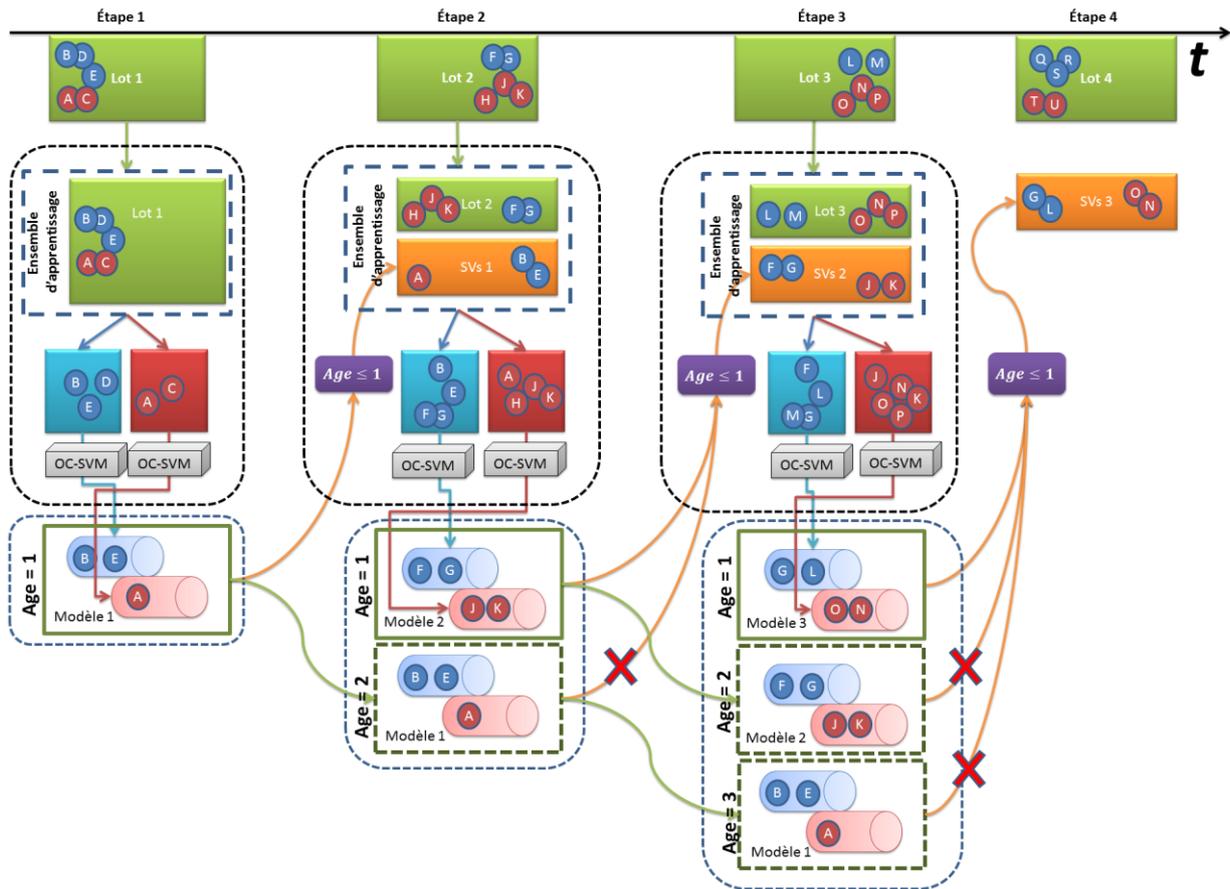


Figure 20 - Principe de la procédure d'apprentissage du mOC-iSVM.AP avec  $\alpha = 1$

Le schéma de la Figure 20 présente 4 étapes (4 moments de la vie d'un classificateur mOC-iSVM.AP s'intéressant à des données correspondant à deux classes de contenus : la classe des données « bleues » et la classe des données « rouges »). Chaque étape est associée à un nouveau lot de données exploitable pour l'apprentissage. On en compte 4 en tout : Lot 1, Lot 2, Lot 3 et Lot 4 qui arrivent au fur à mesure (de la ligne du temps  $t$ ). Chaque lot contient dans cet exemple cinq données représentatives des deux classes (*Rouge* et *Bleu*). L'âge standard dans ce schéma est fixé à 1 ( $\alpha = 1$ ). Dans la 1<sup>ère</sup> étape, à l'arrivée du premier lot de données, le système fonctionne comme précédemment : on sépare les exemples de chaque classe. Les exemples de la classe *Rouge* (resp. *Bleu*) sont traités avec la fonction OC-SVM classique durant l'apprentissage afin de produire le modèle de la classe *Rouge* (resp. *Bleu*). Chaque modèle de classe (tubes rouges/bleus) contient les vecteurs supports de cette classe. Le modèle 1 (regroupant les modèles indépendants de chaque classe) va servir à la classification des données de test qui pourront être introduites à partir de ce moment. Dans la deuxième étape, lors de l'arrivée du second lot de données, le système reprend tous les

vecteurs supports issus du modèle précédent (Modèle 1) pour constituer une base de vecteurs supports (SVs 1 - rectangle en orange). À la différence du mOC-iSVM classique, cette base de vecteurs supports est filtrée selon le critère d'âge («  $Age \leq 1$  »). Seuls les vecteurs supports qui ont un âge inférieur ou égale à l'âge limite ( $\alpha = 1$ ) peuvent alimenter la base d'apprentissage. Cet ensemble de SVs filtrés est regroupé avec les nouvelles données (Lot 2) pour conduire la seconde étape d'apprentissage. L'apprentissage s'effectue de la même manière que pour l'étape 1 : le système apprend séparément les modèles représentatifs des classes *Rouge* et *Bleu* avec les exemples de chacune des deux classes. Le résultat de cet apprentissage (Modèle 2) est mis en place dans le système pour remplacer le modèle précédent (Modèle 1). Cependant le Modèle 1 est conservé dans le système et pourra être ré-exploité pour les apprentissages suivants selon le critère d'âge choisi. Le rectangle bleu en pointillé représente l'historique des modèles et le rectangle vert en pointillé représente les modèles anciens qui ne sont pas utilisés pour la classification. La partie test n'est plus représentée pour des raisons de lisibilité mais elle s'effectue comme sur la figure précédente (Figure 16) en sachant que le modèle utilisé pour le test est le dernier généré (traits verts pleins).

A ce stade, la fin de cette nouvelle étape conduit à la mise à jour de l'âge : l'âge du Modèle 2 est mis à 1, et l'âge du Modèle 1 passe de 1 à 2. Le même mécanisme est appliqué ensuite pour l'étape 3. Lorsque le Lot 3 arrive, seul le Modèle 2 passe le critère de l'âge et donc seuls les vecteurs supports de ce modèle contribuent à la construction du modèle suivant. Les SVs du modèle 1, qui a un âge de 2, ne satisfont pas ce critère d'âge (croix rouge). Le mécanisme se poursuit ainsi sur les étapes suivantes.

### 3.3.1.3. Algorithme d'apprentissage continu du mOC-iSVM.AP

L'algorithme mOC-iSVM.AP est décrit dans l'Algorithme 2. Il diffère de la version générique mOC-iSVM (Algorithme 1) par l'étape de sélection des SVs issus des anciens modèles. La nouveauté se situe dans l'**ÉTAPE 3-1** qui devient:

**//ÉTAPE 3-1 : Sélection des anciens SVs de cette classe issus des modèles qui ont un âge inférieur à  $\alpha$**

Pour chaque  $m$  ;  $0 \leq m < \alpha$

$$SV_{\text{best}C_j} = SV_{\text{best}C_j} \cup SV_{\text{system}}^{i-m} C_j$$

Fin

Algorithme 2 - Modification de l'algorithme d'apprentissage pour le mOC-iSVM.AP

### 3.3.2. Sélection des vecteurs supports selon la performance : version mOC-iSVM.EP

#### 3.3.2.1. Motivation et principe

L'intérêt de l'utilisation de l'historique des modèles pour permettre une adaptation aux variations de contextes (environnements stationnaires et non-stationnaires) a été présenté dans la section précédente à travers le principe du mOC-iSVM.AP. Une autre façon de choisir les meilleurs SVs consiste à tester les nouvelles données entrantes sur tous les modèles précédents issus de l'historique et à ne garder que les meilleurs modèles : ceux qui garantissent les performances les plus élevées. Ce sont les SVs issus de ces modèles qui seront ensuite repris pour construire les nouveaux modèles lors des apprentissages suivants. Le principe de cette version mOC-iSVM, que nous nommerons mOC-iSVM.EP (pour *Multi One-Class Incremental SVM with Error Pruning*), consiste alors à fournir une valeur de seuil de performance, qui servira d'indicateur des « meilleurs » modèles à conserver. Le seuil de performance pourra agir soit sur la précision, le rappel ou le taux moyen de reconnaissance. C'est lors de la définition de la version mOC-iSVM.EP qu'il sera déterminé.

L'approche mOC-iSVM.EP, qui sera expérimentée en section 3.4, doit nous montrer si les performances globales du système sont améliorées par le simple fait de ne retenir que les SVs qui, au fil du temps, ont conduit aux meilleurs résultats. L'intérêt d'utiliser cette version du mOC-iSVM est de ne pas avoir à tenir compte de la nature de l'environnement à la différence de l'approche mOC-iSVM.AP où le choix de l'âge des supports vecteurs est fortement corrélé à la nature des environnements. Par ailleurs, cette proposition de type *Error Pruning* semble être tout à fait appropriée à une bonne adaptation aux environnements réoccurrents (présence répétée d'une même distribution des données au fil du temps), ce qui *a priori* n'est pas le cas avec la version *Age Pruning*.

D'un point de vue purement formel, cette approche semble néanmoins apporter son lot de difficultés. En particulier, si les seuils de performance retenus sont trop grands, on se trouvera dans une situation où aucun modèle ancien ne pourra être réutilisé lors des apprentissages. Le système ne sera donc pas réellement dynamique ni évolutif et ne conduira pas à une amélioration incrémentale de performances, comme cela est attendu. Inversement, si les seuils de performance retenus sont trop petits, des informations inutiles vont continuer à survivre et potentiellement abaisser la performance du système (notamment dans le cas de l'environnement non-stationnaire). Des difficultés d'ordre calculatoire sont enfin à prévoir

lorsque le flux de données est important et les apprentissages nombreux. Les mises à jour des modèles requièrent alors la prise en compte de nombreuses versions antérieures ce qui induit un coût pouvant être relativement important. La recherche automatique du seuil optimal pourrait être intéressante à réaliser d'un point de vue théorique, mais elle conduirait certainement à l'estimation d'un ensemble de valeurs de seuils possibles (et pas un seuil unique) dépendant de la nature de l'environnement et la durée du flux à traiter. Une solution simple pourrait consister à trier les modèles selon leurs performances, et à ne retenir que les  $n$ -meilleurs pour les intégrer dans la construction incrémentale du nouveau. Cette méthode sera présentée dans la section suivante (3.3.3).

### 3.3.2.2. Schéma de principe de l'apprentissage continu selon mOC-iSVM.EP

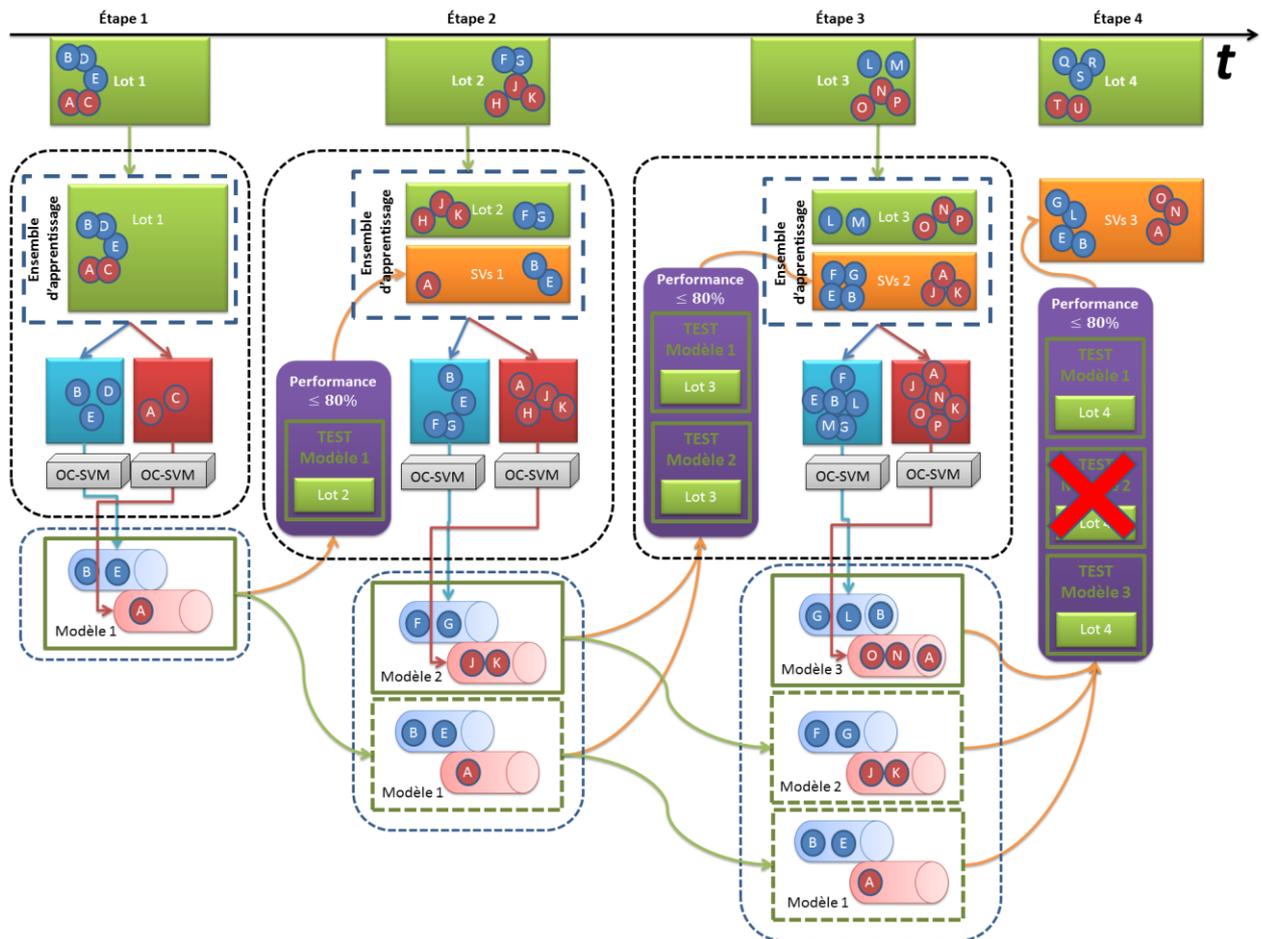


Figure 21 – Principe de la procédure d'apprentissage du mOC-iSVM.AP avec  $f = 80\%$

Le schéma de la Figure 21 illustre le principe du mOC-iSVM.EP toujours en 4 étapes d'apprentissage. La performance prise comme indicateur seuil dans cette illustration est fixée

à 80%. La 1<sup>ère</sup> étape se déroule conformément à ce que nous avons déjà vu dans les cas précédents. A la deuxième étape, lors de l'arrivée du second lot de données et afin de pouvoir réutiliser les vecteurs supports du modèle ancien (Modèle 1) conjointement aux données du lot 2 pour faire l'apprentissage, il faut que le critère de performance (rectangle violet) soit validé. Pour vérifier cela, le Modèle 1 est testé sur le lot 2. Si les performances sont supérieures au seuil défini, les SVs viennent alimenter la nouvelle base d'apprentissage de l'étape 2. Ainsi, au fur et à mesure des itérations, seuls les modèles qui ont une performance supérieure à ce seuil peuvent être réemployés et faire contribuer leurs vecteurs supports à la construction de la base de vecteurs supports de l'étape suivante. Par exemple, dans l'étape 4, seuls les modèles 1 et 3 passent le critère de performance. Par conséquent, seuls les vecteurs supports de ces deux modèles sont intégrés dans l'apprentissage.

### 3.3.2.3. Algorithme d'apprentissage du mOC-iSVM.EP

L'algorithme du mOC-iSVM.EP diffère de la version générique mOC-iSVM (Algorithme 1) par l'étape de sélection des SVs issus des anciens modèles et qui est désormais conditionnée par une contrainte de performances. La modification se situe dans l'*ÉTAPE 3-1* qui devient :

**//ÉTAPE 3-1 : Sélection des anciens SVs de cette classe issus des modèles de l'historique dont la performance est supérieure à f sur le lot courant**

Pour chaque m :  $1 \leq m \leq i$

Perf<sup>m</sup>Cj = Test (M<sub>system</sub><sup>m</sup> Cj, D<sup>i</sup>Cj  $\cup$  D<sup>i</sup>Cj)

Si Perf<sup>m</sup>Cj > f

SV<sub>best</sub>Cj = SV<sub>best</sub>Cj  $\cup$  SV<sub>system</sub><sup>m</sup> Cj

Fin

Fin

Algorithme 3 - Modification de l'algorithme d'apprentissage pour le mOC-iSVM.EP

### 3.3.3. Sélection des vecteurs supports selon les n-meilleurs classificateurs: version mOC-iSVM.nB

#### 3.3.3.1. Motivation et principe

Comme décrit dans la section précédente, le mOC-iSVM.EP possède des avantages non négligeables sur les versions précédentes, notamment en terme d'indépendance relative à

la nature de l'environnement (stationnaire ou non) puisque la sélection des SVs se base sur un critère de performance. Cependant, comme évoqué plus haut, déterminer un seuil de performance minimal permettant de sélectionner les modèles de l'historique, qui serviront à l'établissement de la base d'apprentissage à chaque étape, n'est pas un problème trivial. En effet, le choix d'un seuil absolu sur un critère de performance tel que présenté plus haut n'est pas aisé. Ce seuil reste fortement dépendant du problème (difficulté intrinsèque) mais aussi du flux (variations de la qualité ou de la représentativité des données). Ce seuil dépend donc du contexte général et local. Mal choisi, il peut avoir des effets non négligeables voire extrêmes lorsque qu'aucun modèle passé n'atteint la performance requise sur le lot courant.

Une façon de conserver à la fois les propriétés du système mOC-iSVM.EP tout en s'affranchissant de la définition d'un seuil absolu, consiste à considérer comme critère de sélection les  $n$  meilleurs modèles sur le lot de données courant. Par ailleurs, cette approche permet de limiter le nombre de modèles qui seront pris en compte lors de l'apprentissage et donc de limiter la complexité inhérente à la procédure d'apprentissage. En revanche, comme pour l'approche précédente, il faut tester tous les modèles à chaque étape afin d'évaluer leurs performances sur le lot courant, ce qui reste coûteux. Par ailleurs, pour des valeurs de  $n$  importantes, il se peut qu'une part non négligeable des  $n$ -meilleurs classificateurs soit peu performante, ce qui peut perturber le bon déroulement de la procédure de sélections des SVs finaux par la méthode d'apprentissage classique des SVMs.

### 3.3.3.2. Schéma de principe de l'apprentissage continu du mOC-iSVM.nB

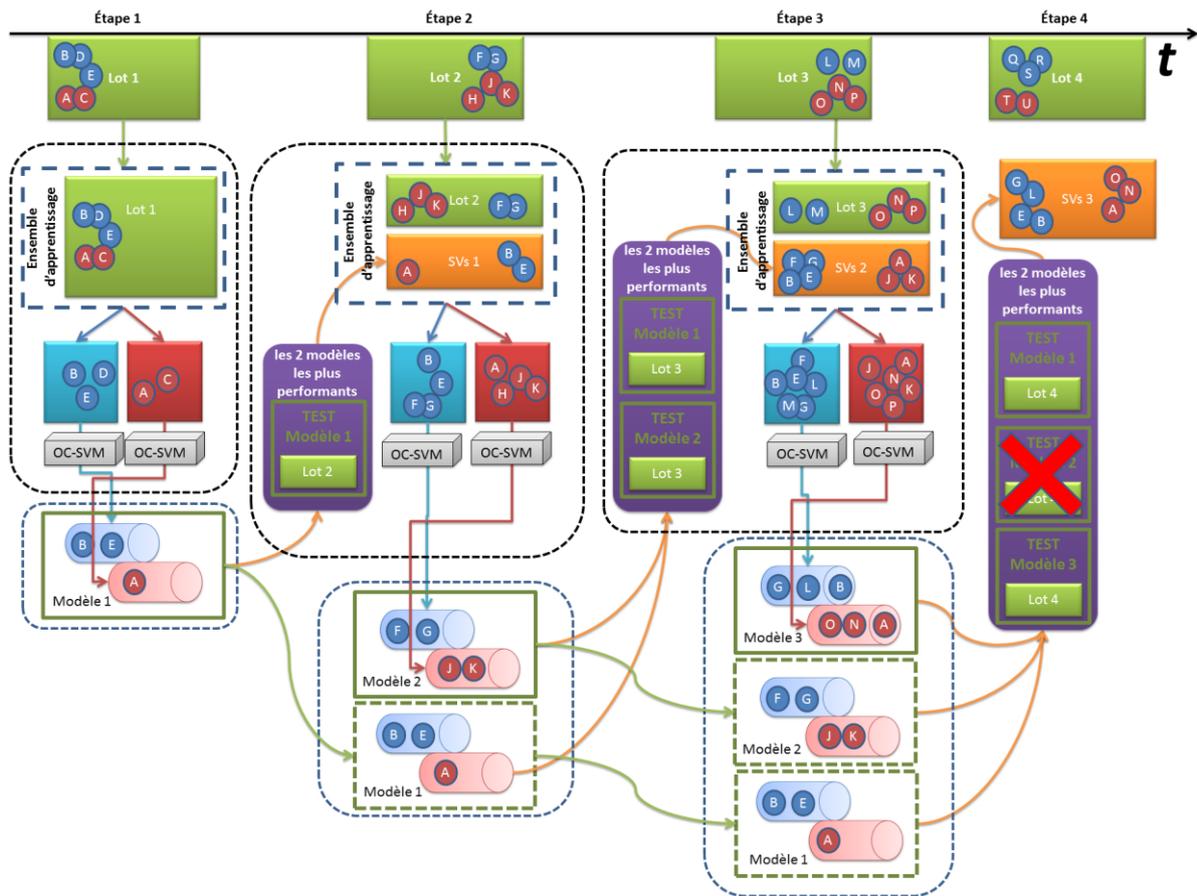


Figure 22 – Principe de la procédure d'apprentissage du mOC-iSVM.nB avec  $n = 2$

Ce schéma est en tout point identique au précédent à l'exception du critère permettant de sélectionner les modèles dont les SVs participeront à l'apprentissage (cadre violet). Le nombre de modèles à conserver dans l'apprentissage dans ce schéma est fixé à 2 ( $n = 2$ ). Pour choisir lesquels, à une étape donnée, on teste tous les modèles de l'historique sur le lot courant et seuls les  $n$  meilleurs seront sélectionnés.

### 3.3.3.3. Algorithme d'apprentissage continu du mOC-iSVM.nB

L'algorithme du mOC-iSVM.nB diffère de la version générique mOC-iSVM par l'étape de sélection des SVs issus des anciens modèles et qui est désormais conditionné par une étude des performances. La nouveauté est indiquée dans l'**ÉTAPE 3-1** qui devient :

**//ÉTAPE 3-1 : Sélection des SVs anciens qui appartiennent aux  $n$  modèles précédents les plus performants**

Pour chaque  $m : 1 \leq m \leq i$

$$\text{Perf}^m = \text{Test} (M_{\text{system}}^m C_j, D^i C_j \cup D^i \bar{C}_j)$$

$$\text{List\_Peft} = \text{TrierSurPerf} (\text{List\_Peft} \cup \{[\text{Perf}^m, m]\})$$

Fin

Pour chaque élément  $\{[Perf^y, y]\}$  dans  $List\_Peft$

$$SV_{best} = SV_{best} \cup SV_{system}^y C_j$$

Fin

Algorithme 4 - Modification de l'algorithme d'apprentissage pour le mOC-iSVM.nB

## 3.4. Expérimentations et étude des performances des quatre versions de mOC-iSVM

### 3.4.1. Objectifs

Nous souhaitons, dans cette section, proposer une validation expérimentale du bien-fondé des quatre modèles (mOC-iSVM, AP, EP, nB) sur différents jeux de données correspondant à des environnements différents relatifs au traitement de données en flux. Les bases exploitées sont issues de benchmarks standards du domaine de l'apprentissage continu. En particulier, on va chercher à évaluer les différentes approches :

- en situation d'ajout de classes en environnement stationnaire (avec la base *UCI Optical Recognition of Handwritten Digits Dataset*<sup>7</sup> - [Kaynak 1995]),
- sur leur capacité d'adaptation aux environnements non-stationnaires avec changement abrupt (sur le jeu de données *SEA dataset*, [Street 2001]),
- sur leur capacité d'adaptation aux environnements non-stationnaires réoccurents avec évolution progressive (sur le jeu de données *CheckerBoard dataset*, [Elwell 2011]).

D'autres expérimentations portant sur un problème réel de traitement d'images de documents (contexte du projet DIGIDOC) seront conduites au chapitre 4. Ces expérimentations mettent en évidence les capacités du système pour créer, fusionner, diviser ou encore étendre des concepts.

### 3.4.2. Plan de l'étude comparative

Pour l'expérimentation en environnement stationnaire, nous allons comparer nos approches avec : *Learn++* [Polikar 2001] ; *Learn++.NC* [Muhlbaier 2009] ; *Bagging* avec

---

<sup>7</sup> La base est la même que précédemment mais le protocole utilisé ici est celui proposé dans l'article de [Muhlbaier 2009].

CART [Breiman 1994] ; *ARC-X4* [Breiman 1998] ; et *DWM* avec classificateur Bayésien (*NB*) [Kolter 2007].

Pour l'expérimentation en environnement non-stationnaire, les comparaisons sont faites avec les méthodes par ensemble suivantes : *Learn++.NSE* [Elwell 2011], *Adaboost* [Freund 1995], *SEA* [Street 2001], *DWM* [Kolter 2007]. Chacune de ces méthodes par ensemble est évaluée avec trois classificateurs de base différents : *NB* (*Naïve Bayes*), *SVM* (*Support Vector Machine*), et *CART* (*Classification And Regression Tree*) sauf *DWM* qui utilise *NB* seulement. Les résultats obtenus avec un classificateur multi-classes de type *NB*, *SVM* (*1-vs-all*) et *CART* sont inclus aussi.

Toutes ces méthodes ont été présentées au chapitre 2, dans les sections 2.2.1.2.3, 2.2.2.1.3 et 2.2.2.2. Nous comparons aussi entre elles les différentes versions de *mOC-iSVM* : *mOC-iSVM* standard, *mOC-iSVM.AP*, *mOC-iSVM.EP*, *mOC-iSVM.nB*. On notera que parmi les approches de l'état de l'art, seuls *Learn++* et *Learn++.NC* sont conçus pour gérer l'apparition de nouvelles classes. Les autres approches permettent de le faire malgré tout (structure suffisamment souple) mais les performances seront nécessairement plus limitées dans ce cas. On notera également que les algorithmes basés sur le *Bagging* et le *Boosting* n'utilisent pas la version *Online* décrite dans l'état de l'art.

En ce qui concerne l'apprentissage des paramètres classiques ( $\nu$  et  $g$ ) du *OC-SVM* à la base de notre approche, nous utilisons le même processus que celui décrit plus haut (noyau *RBF* et méthode *GSES-Grid Search at Each Step*, cf. section 3.2.2.4.) pour l'expérimentation sur la base *UCI Optical Recognition of Handwritten Digits Dataset*. Cependant, pour des raisons calculatoires sur les bases *SEA* et *Checkerboard*, nous avons utilisé une recherche par grille appliquée uniquement lors des premières étapes<sup>8</sup> (méthode *GSFS-Grid Search at First Steps*). Dans ce cas, une fois les paramètres optimaux obtenus, ceux-ci sont conservés dans les étapes suivantes.

Pour les paramètres spécifiques à chacune de nos approches ( $\alpha$  dans *mOC-iSVM.AP*,  $f$  dans *mOC-iSVM.EP*,  $n$  dans *mOC-iSVM.nB*), nous avons testé l'ensemble des valeurs indiquées dans le tableau de synthèse ci-dessous (cf.

Table 2 présentant chacune des expérimentations) afin d'en retenir les meilleures. Seules les performances de ces dernières sont retenues dans les éléments de discussion sur la comparaison entre les différentes approches.

---

<sup>8</sup> Nous fixons les paramètres sur les premières étapes de la base d'apprentissage (sans utiliser le jeu de test). Ce nombre d'étapes dépend de la base et de l'expérimentation : 5 ou 10 étapes pour la base *CheckerBoard*, 50 pour la base *SEA*.

Environnement et type d'adaptation	Version de mOC-iSVM et paramètres spécifiques				Recherche des paramètres ( $v$ et $g$ )	Autres approches	Base de données utilisée
	mOC-iSVM	mOC-iSVM.AP	mOC-iSVM.EP	mOC-iSVM.nB			
Ajout de classe en environnement stationnaire		$a = [1, \dots, 4]$	$f = [75\%, \dots, 95\%]$	$n = [1, \dots, 4]$	GSES	Learn++, Learn++, NC, Bagging (CART), ARC-X4, DWM (NB)	UCI Optical Recognition of Handwritten Digits Dataset (Expérimentation 1)
Environnement non-stationnaire avec changements abrupts		$a = [1, \dots, 9]$	$f = [75\%, \dots, 95\%]$	$n = [1, \dots, 9]$	GSES (5 ou 10 étapes)	L++, NSE (SVM et NB), Adaboost (CART), SEA (SVM), DWM (NB)	SEA Dataset (Expérimentation 2)
Environnement non-stationnaire avec changement progressif							
Environnement non-stationnaire avec réoccurrence des événements		$a = [1, \dots, 9]$	$f = [75\%, \dots, 95\%]$	$n = [1, \dots, 9]$	GSES (5 étapes)	L++, NSE (SVM, NB, CART), Adaboost (SVM, NB, CART), SEA (SVM, NB, CART), DWM (NB)	Cheerboard Dataset (Expérimentation 3)

Table 2 - Tableau de synthèse des expérimentations conduites

### 3.4.3. Évaluation de la capacité d'ajout de concepts en environnement stationnaire

Nous allons tester dans cette section la capacité des différentes versions de notre système mOC-iSVM à intégrer de nouvelles classes représentées par des données portant de nouvelles étiquettes dans les lots qui arrivent au fur et à mesure.

#### 3.4.3.1. Description du jeu de données testé

Le jeu de données utilisé dans ces expérimentations en environnement stationnaire est le même que celui qui a été présenté à la section 3.2.3.2. : *UCI Optical Recognition of Handwritten Digits Dataset* [Kaynak 1995].

#### 3.4.3.2. Description du protocole expérimental

L'expérimentation repose sur un processus en quatre étapes d'apprentissage et de tests respectant le même protocole que celui qui est proposé dans [Muhlbaier 2009] et dont sont issus les résultats des approches qui seront comparées aux nôtres. On considère à chaque étape 10% de la base de données pour les tests (cet ensemble de changera pas), et 90% pour la base d'apprentissage qui sera divisée en quatre lots correspondants aux 4 étapes d'apprentissage successives. Dans la 1ère étape d'apprentissage, on s'assure que les 4 premières classes de digits (classe de 0 à 3) sont présentes et dans les étapes suivantes, on ajoute à chaque fois deux nouvelles classes (4-5, 6-7, 8-9). À l'issue des 4 étapes d'apprentissage, l'ensemble des classes est représenté. On réalise les tests selon le mécanisme de validation croisée en 5 passes pour assurer la fiabilité de l'expérimentation. Dans cette expérimentation, nous utilisons les mesures de performances  $AC_i$  (cf. Équation 1) et  $GA$  (cf. Équation 2).

#### 3.4.3.3. Interprétation des résultats

Les performances ( $GA$ ), lors de la dernière étape, de chacune des versions mOC-iSVM.AP, mOC-iSVM.EP, mOC-iSVM.nB en fonction des paramètres  $\alpha$ ,  $f$ , et  $n$  sont reportées dans les tableaux suivants (**Table 3**, **Table 4**, **Table 5**). Comme expliqué plus haut dans la présentation des variantes autour de mOCi-SVM, la performance du système est normalement relative à la quantité de SVs utiles et inutiles qui sont conservés et réutilisés. Si les seuils sont trop sévères ( $n$ ,  $\alpha$  trop petit ou  $f$  trop grand), les informations utiles ne sont pas suffisantes pour obtenir de bonnes performances. À l'inverse, si le seuil est trop souple, le

système devient plus lourd (plus de SVs à utiliser pour l'apprentissage à chaque étape) sans améliorer la performance réelle. Cependant, dans le cas de cette expérimentation, on notera que les performances en fonction des différentes approches mais aussi en fonction des paramètres ne varient pas beaucoup, sans doute à cause du benchmark.

<i>Seuil <math>f =</math></i>	95%	90%	85%	80%	75%	70%
Performance ( <i>GA</i> ) de la dernière étape (%)	98.24 ± 0.3	98.56 ± 0.2	98.58 ± 0.2	98.56 ± 0.2	98.55 ± 0.2	98.53 ± 0.2

Table 3 - Comparaison des résultats de mOC-iSVM.EP avec différentes valeurs de  $f$

<i>Seuil <math>\alpha =</math></i>	1	2	3	4
Performance ( <i>GA</i> ) de la dernière étape (%)	98.5 ± 0.16	98.51 ± 0.19	98.57 ± 0.15	98.57 ± 0.15

Table 4 - Comparaison des résultats de mOC-iSVM.AP avec différentes valeurs de  $\alpha$

<i>Seuil <math>n =</math></i>	1	2	3	4
Performance ( <i>GA</i> ) de la dernière étape (%)	98.5 ± 0.16	98.52 ± 0.23	98.64 ± 0.16	98.54 ± 0.16

Table 5 - Comparaison des résultats de mOC-iSVM.nB avec différentes valeurs de  $n$

La Figure 23 présente les performances (*GA*) de chaque modèle de mOC-iSVM pour chaque étape d'apprentissage. Initialement, les méthodes débutent avec une performance relativement faible qui s'explique par l'absence de connaissances sur certaines classes (les données de tests sont identiques pour chaque étape et contiennent des données appartenant à toutes les classes. Lors de la première étape, seules les classes 0 à 3 sont apprises, les données appartenant aux autres ne pouvant être identifiées). Pour chaque version du mOC-iSVM, on observe ensuite des gains en performance très similaires. Dans l'étape finale (4<sup>ème</sup> étape où toutes les données ont finalement été traitées et toutes les classes apprises), les performances deviennent similaires entre la version de base et les versions dérivées du mOC-iSVM (entre 95 et 98% de reconnaissance). Ces approches sont par ailleurs bien plus performantes que les méthodes *Bagging*, *ARC-X4*, *DWM-NB* et *Learn++* qui ont des taux de reconnaissance entre 75 et 82%. Comme le montre la Table 6, qui présente le détail des performances ( $AC_i$ ) pour chaque classe (numérotées de 0 à 9), ainsi que la performance globale et son écart type (les

valeurs nulles indiquant que le classificateur n'a pas reconnu d'éléments de cette classe car il n'a pas encore pu en définir le modèle), ces faibles performances sont liées à la difficulté à reconnaître les nouvelles classes. En effet, dans les méthodes de *Bagging* et d'ARC-X4, le nombre fixe de classificateurs ne permet pas de gérer efficacement la prise en compte de nouvelles classes: le nombre de classificateurs les représentant est trop faible et le vote avantage les classes plus anciennes. Comme mentionné dans [Polikar 2001], ces méthodes ne sont pas adaptées à ce type de situation mais bien plus pour les environnements stationnaires ou de dérive de concept classique (sans ajout de classes). Par ailleurs, la plupart des approches permettant de traiter la dérive de concept utilisent une capacité à oublier les anciennes connaissances nécessaires à leur adaptation. Le problème alors est que le système est incapable à la fois d'oublier les anciennes connaissances pour apprendre de nouvelles classes, tout en conservant les performances sur les anciennes classes.

On observe notamment, pour les classes 8 et 9, des performances qui atteignent tout juste les 4% de reconnaissance avec Learn++, 9 % avec le *Bagging*, et à 11% avec ARC-X4. Cependant, au cours du temps, Learn++ corrige cet aspect (*cf.* étape 3 pour les classes 4, 5 et étape 4 pour les classes 6 et 7). L'approche DWM n'a pas les mêmes problèmes car elle possède un mécanisme d'oubli en supprimant les classificateurs trop anciens (critères d'âge) ou trop peu performants. Cependant, les connaissances progressivement oubliées sont définitivement perdues : la performance sur les anciennes classes diminue rapidement (*cf.* classe 1 par exemple). DWM ne peut pas en même temps bien apprendre les nouvelles classes tout en conservant ses connaissances sur les anciennes classes.

Au final, seuls Learn++.NC et nos variantes de mOC-iSVM ont bien pris en compte les nouvelles données relatives aux nouvelles classes dès leurs premières apparitions (performances allant de 80% à 90% pour ces classificateurs). On rappelle ici que Learn++.NC a été conçu spécifiquement pour l'apprentissage de nouvelles classes (NC) et repose sur la méthode de vote dynamique (voir section 2.2.2.1.3). Cette pondération permet de mieux prendre en compte l'apparition de nouvelles classes. En ce qui concerne nos approches à base de mOC-iSVM, l'indépendance des OC-SVM, couplée au fait qu'il n'y a qu'un classificateur par classe, permet d'éviter ces problèmes de représentativité des classificateurs les uns par rapport aux autres que l'on retrouve dans les approches par ensemble.

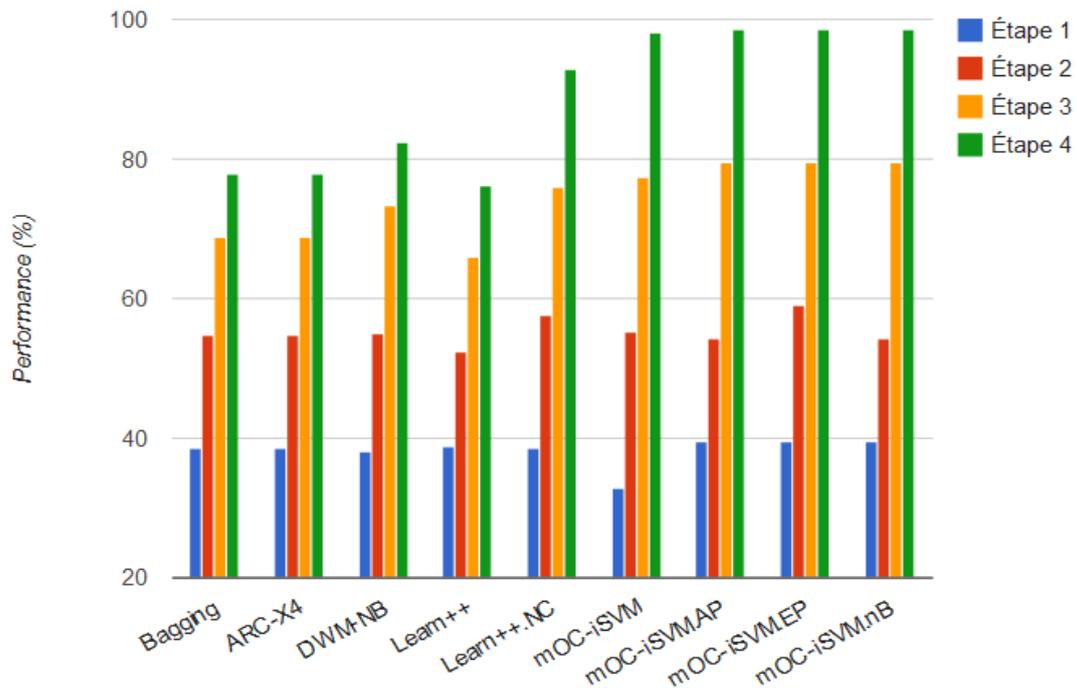


Figure 23 - Évolution de performances ( $GA$ ) de 5 classificateurs et de notre proposition de mOC-iSVM (et de ses variantes) pour 4 étapes d'apprentissage.

Approches	Étape	Performance ( $AC_i$ ) (%)										Performance ( $GA$ )	Écart Type
		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9		
Learn++.NC	1	99	95	95	96	0	0	0	0	0	0	37.9	0.1
	2	98	94	96	95	97	97	0	0	0	0	56.9	0.7
	3	98	92	95	93	90	93	99	99	0	0	75.9	0.8
	4	98	84	93	90	90	90	97	94	96	96	93.4	0.4
Bagging	1	99	95	94	95	0	0	0	0	0	0	37.4	0.1
	2	99	96	95	96	76	84	0	0	0	0	56.9	0.2
	3	99	95	96	96	91	92	55	61	0	0	76	0.3
	4	99	95	97	96	93	95	96	94	9	4	93.3	0.2
ARC-X4	1	99	96	94	95	0	0	0	0	0	0	37.7	0.1
	2	98	95	94	93	77	88	0	0	0	0	57.1	0.3
	3	97	93	93	92	93	95	56	65	0	0	75.4	0.4
	4	97	93	94	92	93	95	97	97	25	11	91.9	0.3
DWM	1	98	94	95	95	0	0	0	0	0	0	38.2	0.1
	2	91	83	94	92	96	92	0	0	0	0	54.9	0.6
	3	90	79	95	93	95	92	96	92	0	0	73.3	0.3
	4	86	63	90	87	90	86	91	85	78	73	82.3	0.3

<b>Learn++</b>	1	99	96	95	96	0	0	0	0	0	0	37.9	0.1
	2	99	96	96	96	62	73	0	0	0	0	56.9	0.1
	3	99	96	97	97	89	92	43	47	0	0	75.9	0.2
	4	99	96	97	96	93	95	89	89	4	2	93.4	0.2
<b>mOC-iSVM</b>	1	100	98	98	98	0	0	0	0	0	0	39.4	0.1
	2	99	99	99	99	99	98	0	0	0	0	59.4	0.1
	3	100	100	100	99	100	99	100	99	0	0	79.6	0.1
	4	99	97	99	98	99	99	100	99	96	97	98.5	0.1
<b>mOC-iSVM.EP (f = 85%)</b>	1	100	98	98	98	0	0	0	0	0	0	39.4	0.1
	2	99	99	98	98	99	98	0	0	0	0	59.1	0.3
	3	100	100	100	100	100	99	100	99	0	0	79.6	0.1
	4	100	98	99	98	99	99	100	99	96	97	98.6	0.2
<b>mOC-iSVM.AP (a = 3)</b>	1	100	98	98	98	0	0	0	0	0	0	39.4	0.1
	2	99	99	99	99	99	98	0	0	0	0	59.4	0.1
	3	100	100	100	99	100	99	100	99	0	0	79.6	0.1
	4	100	98	99	98	99	99	100	99	96	97	98.6	0.1
<b>mOC-iSVM.nB (n = 3)</b>	1	100	98	98	98	0	0	0	0	0	0	39.4	0.1
	2	99	99	99	99	99	98	0	0	0	0	59.4	0.1
	3	100	100	100	99	100	99	100	99	0	0	79.6	0.1
	4	100	98	99	98	99	99	100	99	96	97	98.5	0.2

Table 6 – Performance ( $AC_i$  et  $GA$ ) pour l’ajout de concept en environnement stationnaire.

Si nous comparons les 4 versions du mOC-iSVM, nous constatons que les performances à la dernière étape ( $GA$ ) sont équivalentes (< 1% de différence). Par ailleurs les temps d’apprentissage (cumulés sur toutes les étapes) sont identiques (voir Table 7). En revanche, en termes de stockage de l’historique des SVs, la version standard mOC-iSVM qui ne conserve que les modèles courants, est bien entendu moins gourmande. Enfin, le paramétrage du mOC-iSVM de base étant plus simple (pas de paramètres spécifiques), cette version standard semble mieux adaptée à ce type d’environnement stationnaire.

<b>Approches</b>	<b>Performance (<math>GA</math>) de la dernière étape (%)</b>	<b>Temps d’apprentissage (seconde)</b>
<b>mOC-iSVM</b>	<b>98.5</b>	<b>12.48</b>
<b>mOC-iSVM.AP</b>	<b>98.6</b>	<b>13.68</b>
<b>mOC-iSVM.EP</b>	<b>98.6</b>	<b>11.19</b>
<b>mOC-iSVM.nB</b>	<b>98.5</b>	<b>12.72</b>

Table 7 - Comparaison générale entre les différentes versions de mOC-iSVM.

### 3.4.4. Évaluation du mOC-iSVM et de ses variantes en environnement non-stationnaire avec changements brusques

Cette section est dédiée à l'étude des capacités des différentes versions de notre système mOC-iSVM à gérer les changements brusques de concepts (les données d'une même classe présentent dans le temps des variations brutales).

#### 3.4.4.1. Description du jeu de tests

Le jeu de données testé se nomme *SEA Concept Dataset*. Il a été proposé dans [Street 2001] pour tester l'algorithme SEA (*Streaming Ensemble Algorithm*). Cette base de données est considérée comme une référence pour l'étude des classificateurs en environnement non stationnaire. La base de données comprend deux classes et trois caractéristiques prenant des valeurs dans l'intervalle  $[0, 10]$ . Seules les deux premières caractéristiques sont pertinentes au regard des classes. En effet, une instance est affectée à la classe 1 si la somme des deux premières caractéristiques est en dessous d'un seuil. Sinon, elle est affectée à la classe 2. La base est composée de quatre périodes, chacune constituée de 50 étapes pendant lesquelles la même quantité de données arrive. Au début de chaque période, le seuil définissant les classes est changé de façon brutale, passant successivement de 8 à 9 puis à 7.5 et enfin à 9.5 (voir Figure 24).

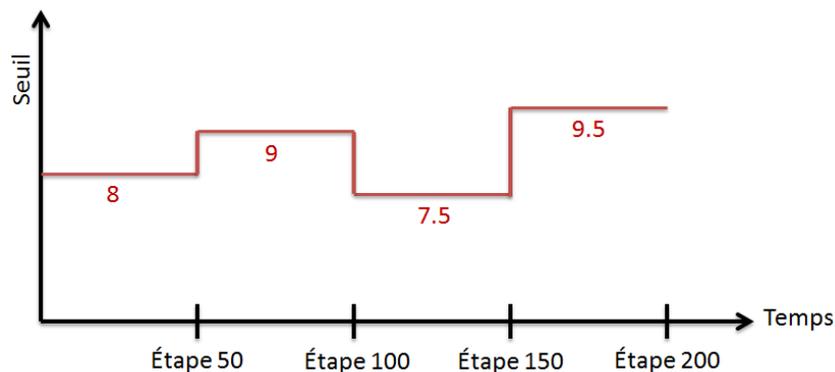


Figure 24 – Évolution du seuil définissant les classes de la base SEA au cours du temps.

#### 3.4.4.2. Description du protocole expérimental

Nous appliquons le même protocole que celui proposé dans [Elwell 2011]. Il existe 50000 instances pour l'ensemble d'apprentissage et 50000 pour l'ensemble de tests. Chaque étape contient donc 250 exemples pour l'apprentissage et la même quantité pour les tests, tirés

aléatoirement, sur un ensemble de 200 étapes. L'évaluation se fait lors de chacune de ces 200 étapes. L'auteur ajoute également 10% de bruit dans l'apprentissage (voir Figure 25).

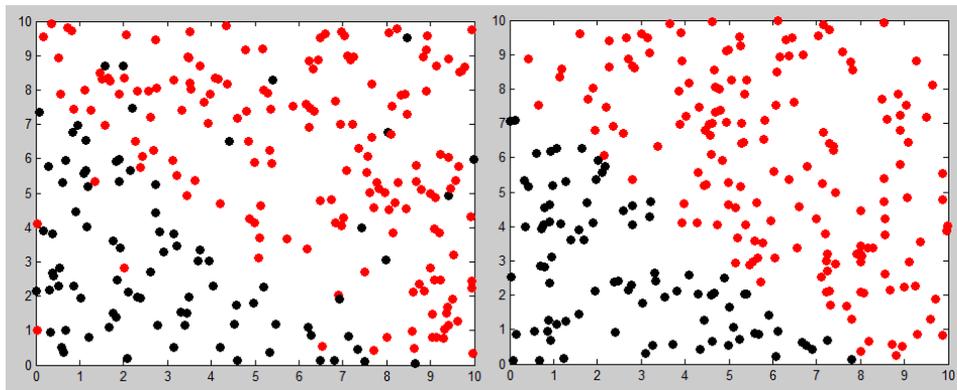


Figure 25 - Illustration des données de l'ensemble d'apprentissage (à gauche) et de l'ensemble de test (à droite) correspondant à l'étape 1 : seuil de 8 (classe 1 en noir, classe 2 en rouge) et 10% de bruit ajouté dans l'ensemble d'apprentissage (à gauche).

Nous souhaitons analyser les performances des trois variantes de mOC-iSVM en les comparant principalement à L++.NSE (*Learn++ for Non-Stationary Environment*), qui est un des meilleurs classificateurs dynamiques répondant aux enjeux de l'adaptation en environnement non-stationnaire. Les résultats sont issus de [Elwell 2011]. Nous inclurons cependant, comme précédemment, les performances des autres classificateurs DWM, SEA et *Adaboost* tirées de [Elwell 2011].

Dans cette expérimentation, nous utilisons la mesure *GA* (cf. Équation 2) et *SAC* (cf. Équation 3) pour comparer les résultats.

#### 3.4.4.3. Interprétation des résultats

##### **Comparatifs avec les classificateurs de l'état de l'art**

La Table 8 présente les résultats obtenus sur la base SEA. Parmi les approches concurrentes, on constate, comme attendu, que Learn++.NSE obtient en général les meilleures performances. Celles-ci sont de 96,8% en utilisant les SVMs comme classificateur de base (plus performants que les NB et CART). Les performances du mOC-iSVM standard sont relativement faibles (proches du *Boosting*). En revanche, les trois autres versions (mOC-iSVM.AP, mOC-iSVM.EP, mOC-iSVM.nB) affichent de bonnes performances. Les versions mOC-iSVM.EP et mOC-iSVM.nB sont même les plus performantes avec un taux de 97.2% de bonne classification, la version mOC-iSVM.AP étant un peu moins performante que les approches concurrentes avec ses 96%. Ces différences de résultats entre la version standard et

les autres montrent bien l'intérêt de conserver l'historique. Par ailleurs, le fait de se baser sur un critère de performance est également préférable à l'usage strict de l'âge (récent). On remarque par ailleurs que l'écart type plus important pour la version mOC-iSVM.AP semble témoigner des difficultés que le système rencontre à s'adapter aux changements brusques.

<b>Approches</b>	<b>Performance (SAC)</b>	<b>Écart type (%)</b>
<b>L++.NSE (NB)</b>	<b>96.6</b>	<b>0.2</b>
<b>Single (NB)</b>	<b>94.7</b>	<b>0.6</b>
<b>DWM (NB)</b>	<b>96.6</b>	<b>0.6</b>
<b>SEA (NB)</b>	<b>95.4</b>	<b>0.4</b>
<b>Adaboost (NB)</b>	<b>93.2</b>	<b>0.4</b>
<b>L++.NSE (SVM)</b>	<b>96.8</b>	<b>0.2</b>
<b>Single (SVM)</b>	<b>95.6</b>	<b>0.4</b>
<b>SEA (SVM)</b>	<b>95.7</b>	<b>0.2</b>
<b>Adaboost (SVM)</b>	<b>93.2</b>	<b>0.2</b>
<b>L++.NSE (CART)</b>	<b>95.8</b>	<b>0.5</b>
<b>Single (CART)</b>	<b>86.7</b>	<b>1</b>
<b>SEA (CART)</b>	<b>95.6</b>	<b>0.3</b>
<b>Adaboost (CART)</b>	<b>93.3</b>	<b>0.5</b>
<b>mOC-iSVM</b>	<b>93</b>	<b>4.3</b>
<b>mOC-iSVM.AP</b>	<b>96</b>	<b>2.9</b>
<b>mOC-iSVM.EP</b>	<b>97.2</b>	<b>1.9</b>
<b>mOC-iSVM.nB</b>	<b>97.2</b>	<b>1.6</b>

Table 8 - Résumé des performances sur la base SEA

Malgré les bons taux de reconnaissance des classificateurs mOC-iSVM, ces derniers présentent des valeurs d'écart type élevées. Cela peut s'expliquer par le fait que le système est fondé sur une re-sélection complète des SVs à chaque étape. Cette sélection affecte directement la frontière des concepts modélisés, rendant le système particulièrement sensible. Les autres méthodes, basées sur des ensembles de classificateurs, sont plus stables du fait de l'inertie inhérente à leur modélisation. Ils mettent en revanche plus de temps à corriger le modèle suite à un changement brusque. Ces phénomènes sont particulièrement visibles sur la Figure 26 présentant l'évolution des performances au cours du temps. Ainsi, on constate que l'ensemble des méthodes mentionnées dans nos tests (L++.NSE, SEA, DWM, *Adaboost*) présente une chute de performance importante lors des étapes 50, 100 et 150 correspondant aux changements de seuils. Par exemple, dans l'étape 150, L++.NSE affiche une baisse de

reconnaissance de 10%, SEA, une baisse de 20% et DWM une baisse de 7%. Ensuite, il faut compter en moyenne plus de 30 étapes ( $30 \times 250 = 75000$  exemples) pour avoir des performances proches de celles d'avant le changement. Pour nos versions de mOC-iSVM, l'adaptation nécessite un maximum de 5 étapes et les chutes de performances sont bien moindres en intensité (même si elles sont plus fréquentes).

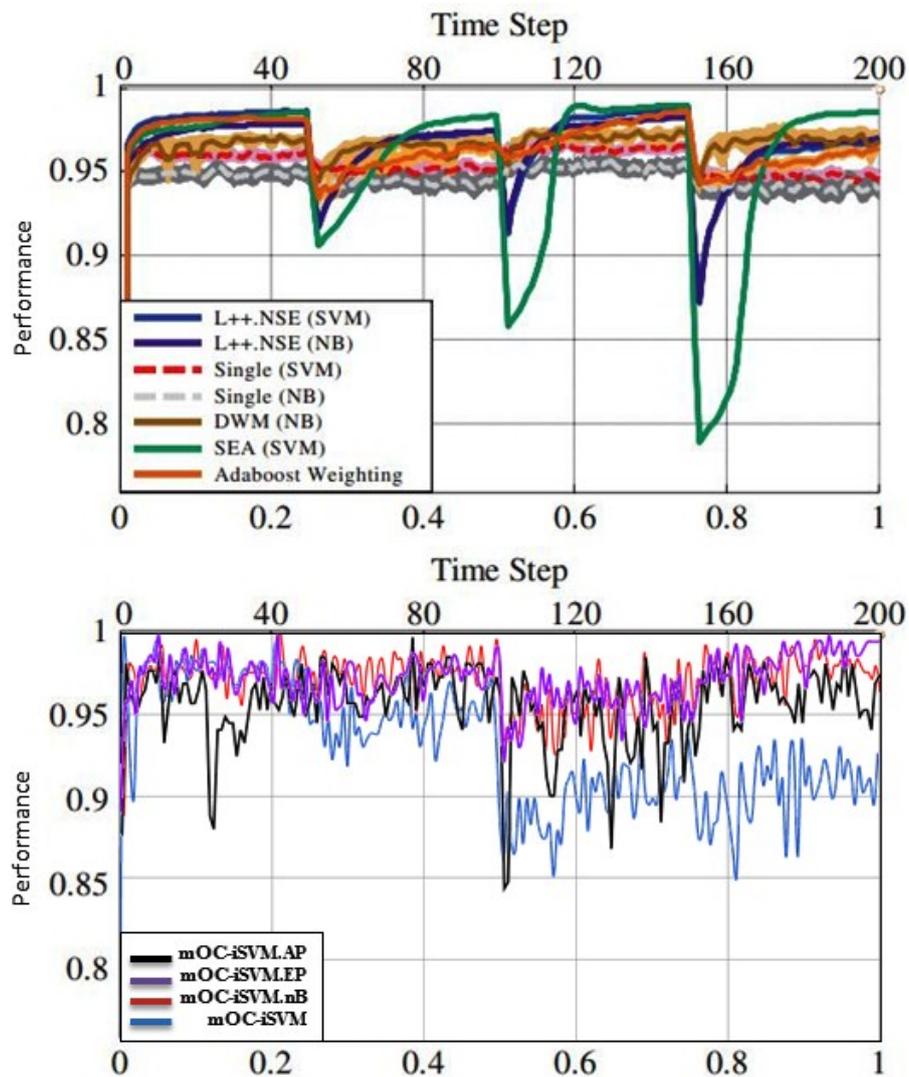


Figure 26 - Evolution des performances ( $GA$ ) au cours du temps sur la base SEA.

### Études des performances en fonction des paramètres spécifiques $\alpha$ , $f$ , et $n$ .

Pour chacune des versions mOC-iSVM.AP, mOC-iSVM.EP, et mOC-iSVM.nB, nous avons testé diverses valeurs pour les paramètres  $\alpha$ ,  $f$ , et  $n$  (voir Table 9, Table 10, Table 11). Seul le meilleur résultat de chaque version a été utilisé comme référentiel pour l'étude comparative ci-dessus. Dans le cas du mOC-iSVM.AP, une valeur trop faible de l'âge ( $\alpha = 1$ ) conduit à des performances moindres (94.87%). Entre un âge de 1 à 6, les

performances ne cessent d'augmenter. Au-delà de 6, les performances stagnent voire décroissent un peu. Au-delà de 9, elles décroissent plus sensiblement. Il faut noter par ailleurs que le changement d'âge n'augmente pas de façon importante les temps de calcul et a très peu d'effet sur l'écart type.

<i>Seuil <math>\alpha</math> =</i>	1	2	3	4	5	6	7	8	9
<b>Performance</b> (SAC)	94.87	95.56	95.63	95.78	95.96	96.01	95.9	96	95.66
<b>Écart Type</b> (%)	2.67	2.4	2.7	2.75	2.62	2.74	2.95	2.78	3.09
<b>Temps de calcul (s)</b>	70.51	77.54	86.47	99.55	100.89	111.3	120.88	127.82	131.18

Table 9 - Comparaison des résultats de mOC-iSVM.AP avec différentes valeurs de  $\alpha$

Dans le cas mOC-iSVM.EP (Table 10), le choix de la performance  $f$  qui détermine les modèles à conserver est plus critique, d'autant plus sur ce problème où les performances sont bonnes et stables (difficile d'évaluer précisément l'impact de ces valeurs). Lorsque  $f$  est élevée ( $> 97\%$ ), les modèles candidats sont moins nombreux et ne permettent d'intégrer qu'un nombre restreint de SVs. Inversement, si le seuil  $f$  est plus faible ( $< 90\%$ ), des SVs inadaptés participent aux apprentissages et conduisent à la détermination de frontières mal ajustées et à une chute de performance globale ( $< 95\%$ ). La répercussion du choix du seuil est donc importante car elle affecte les résultats tout au long du processus (et pas uniquement sur quelques étapes comme dans le cas de mOC-iSVM.AP).

Notons également que les temps de calcul sont très importants par rapport à la version précédente. Ils dépendent beaucoup de  $f$  car en abaissant cette valeur, de nombreux modèles viennent ajouter leurs SVs à la base d'apprentissage à chaque étape, ce qui joue directement sur la complexité et le temps d'apprentissage. Ainsi, les temps de calcul passent de 545s à 2155s pour un décalage de seuil  $f$  de 7.5%. On en conclut que le contrôle du temps de calcul dans mOC-iSVM.EP est plus difficile à obtenir que dans le cas du mOC-iSVM.AP. En revanche, les performances du mOC-iSVM.AP sont un peu plus faibles que dans le cas mOC-iSVM.EP (96% de bonne reconnaissance maximum avec mOC-iSVM.AP contre 97.22% avec mOC-iSVM.EP). Enfin le choix d'un seuil adapté pour  $f$  permet d'obtenir un écart-type plus faible avec mOC-iSVM.EP.

<i>Seuil f =</i>	97.5%	95%	92.5%	90%
<b>Performance (SAC) (%)</b>	96.61	97.22	96.9	95.49
<b>Écart Type (%)</b>	2.86	1.78	1.58	2.45
<b>Temps de calcul (s)</b>	545	1120	1636	2155

Table 10 - Comparaison des résultats de mOC-iSVM.EP avec des différentes valeurs  $f$

Pour la version mOC-iSVM.nB, les écarts-types sont similaires à ceux de mOC-iSVM.AP, mais les performances sont plus élevées (voir Table 11). Pour chaque seuil de 1 à 9, le nombre de SVs qui intègrent l'apprentissage de mOC-iSVM.nB est presque le même que dans le cas mOC-iSVM.AP. Ces deux versions apprennent avec une même quantité de données, mais avec des performances globales différentes. C'est donc bien le fait de ne retenir que les modèles les plus performants qui marque une vraie différence. En évitant ainsi l'intégration de SVs mal adaptés, le mOC-iSVM.nB se concentre sur un historique trié de SVs pour l'apprentissage. Il permet ainsi de ne sélectionner que les SVs les meilleurs en limitant les données éventuellement erronées ou peu efficaces. Cette observation est importante car finalement, avec mOC-iSVM.nB, on peut obtenir la meilleure performance avec un écart-type plus faible (il n'y a pas de forte variance parmi les valeurs d'écart-types de mOC-iSVM.nB, alors qu'il y en a dans la version mOC-iSVM.EP). Cependant, la méthode est très coûteuse en temps de calcul comparativement à mOC-iSVM.AP et même par rapport à la version EP avec une valeur de  $f$  élevée. Cela semble signifier que la méthode EP n'utilise pas toujours l'historique pour construire les nouveaux modèles (pas de classificateur assez performant). D'où l'augmentation de l'écart-type. En revanche le paramètre  $n$  permet de contrôler plus facilement les performances et le temps de calcul par rapport au paramètre  $f$  de mOC-iSVM.EP.

<i>Seuil n =</i>	1	2	3	4	5	6	7	8	9
<b>Performance (SAC) (%)</b>	96.07	96.63	96.96	97.11	97.11	97.06	97.2	97.13	97.17
<b>Écart Type (%)</b>	1.69	1.65	1.51	1.6	1.63	1.67	1.64	1.69	1.59
<b>Temps de calcul (s)</b>	1573.15	1597.89	1632.34	1693.16	1989.96	1773.13	1730.35	1889.1	1734.14

Table 11 - Comparaison des résultats de mOC-iSVM.nB avec des différentes valeurs  $n$ .

Nous pouvons conclure que, dans l'environnement non-stationnaire présentant des changements abrupts, la version mOC-iSVM.nB présente d'excellentes performances, des

écarts-types relativement faibles mais des temps de calcul assez grands. Si le temps de calcul est un critère important, mOC-iSVM.AP peut être une solution à retenir.

### 3.4.5. Évaluation du mOC-iSVM et de ses variantes en environnement non-stationnaire avec dérive progressive et réoccurrence de contexte

#### 3.4.5.1. Description du jeu de tests

Les données du « damier tournant » (CB : *Rotating Checkerboard Dataset*) décrit dans [Elwell 2011] est un jeu de données très intéressant à tester avec nos algorithmes pour illustrer différentes situations associées à la dérive progressive de concept. Le principe est de générer des données appartenant à 2 classes représentées par les cases d'un damier (une couleur représente une classe). Ce damier, et donc la distribution des classes dans l'espace, change au cours du temps en faisant tourner le damier autour de son centre. Quatre vitesses de dérive sont considérées : une vitesse de dérive constante de  $\frac{2\pi}{400} = 0,016\text{rad/étape}$  ; une vitesse de dérive exponentielle (croissante) ; une vitesse de dérive sinusoïdale et une vitesse de dérive impulsionnelle gaussienne (selon la progression : lent-rapide-très rapide-rapide-lent). Cela conduit à former quatre bases appelées : CB Constant ; CB sinusoïdale ; CB exponentielle ; et CB Pulse. Les données sont extraites de la manière suivante. Une fenêtre d'échantillonnage fixe sur le damier permet d'extraire les données des deux classes (points rouges et noirs sur la Figure 27). La fenêtre change d'un angle de rotation qui est défini par une des quatre fonctions de vitesse décrite précédemment. Cette expérimentation est contrôlée par deux paramètres : la dimension de la fenêtre et l'angle de rotation  $\alpha$ . Afin d'augmenter la complexité, 10% du bruit aléatoire sont introduits dans la base d'apprentissage. Les descripteurs des pixels au sein de la fenêtre d'échantillonnage sont leurs coordonnées.

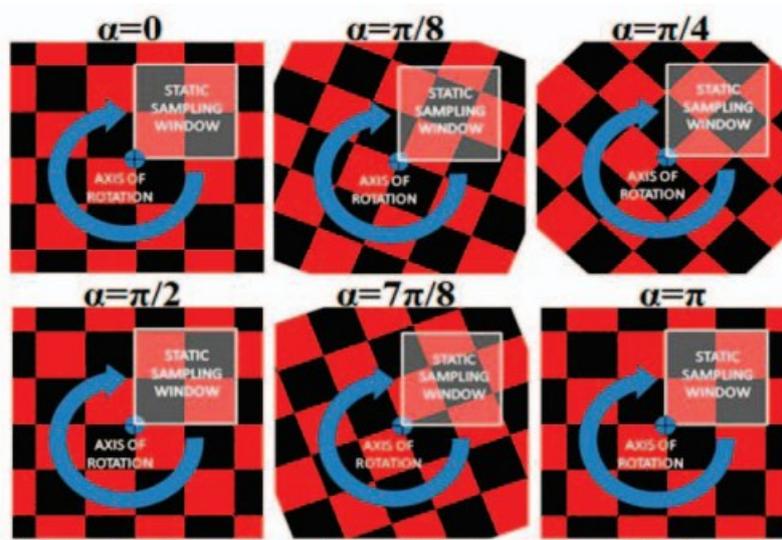


Figure 27 - Extraction des données par fenêtres d'échantillonnage fixe sur un damier tournant (*Rotating CheckerBoard*).

Cette base est également intéressante car elle permet d'obtenir des situations réoccurrentes : la rotation de 0 à  $\pi$  et la rotation de  $\pi$  à  $2\pi$  reproduisent la même situation entre les régions noires et rouges (voir Figure 28). La seule différence est que la vitesse de réapparition des mêmes situations n'est pas la même : elle demeure identique dans le cas de la dérive constante, exponentiellement croissante dans le cas de la dérive exponentielle, inversement rapide à lente (ou lente à rapide) pour la dérive à impulsion gaussienne et sinusoïdale.

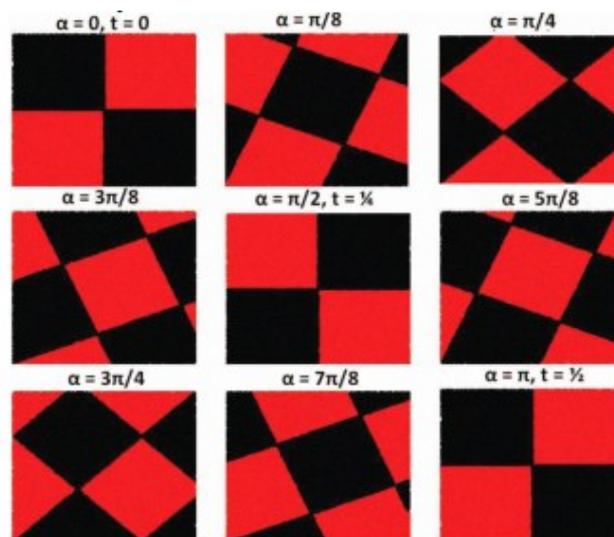


Figure 28 - Quelques résultats de la fenêtre d'échantillonnage (vitesse de dérive constante).

Aux étapes  $\alpha = 0$  et  $\alpha = \pi$ , on observe la même relation entre les régions noires et rouges (situation réoccurrente).

### 3.4.5.2. Description du protocole expérimental

Nous appliquons le même protocole que celui initialement proposé dans [Elwell 2011]. Chaque expérimentation (une par vitesse de dérive) est composée de 400 étapes. Pour chaque étape, un ensemble d'apprentissage est constitué d'un très petit nombre de données (25 données au total pour les deux classes : noire et rouge) à partir de la fenêtre d'échantillonnage statique. Pour l'ensemble de tests, 1024 exemples sont sélectionnés uniformément à partir d'une grille de résolutions 32 par 32 positionnée sur la fenêtre d'échantillonnage (voir Figure 29).

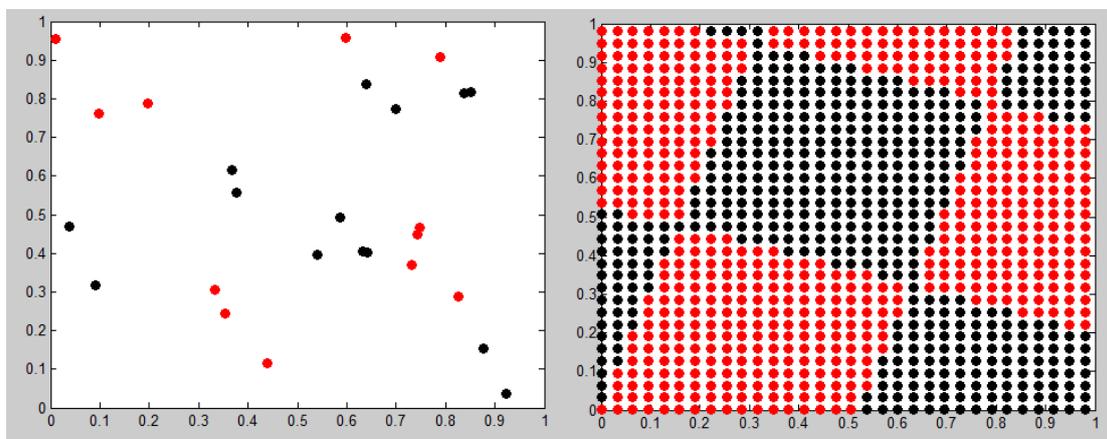


Figure 29 - Ensemble d'apprentissage (25 échantillons) et l'ensemble de tests (1024 échantillons) dans l'étape numéro 20 de la base CB Constant.

Nous cherchons à comparer les performances du mOC-iSVM et de ses trois variantes avec les classificateurs incrémentaux de l'état de l'art. Dans cette expérimentation, nous utilisons la mesure  $GA$  (cf. Équation 2) et  $SAC$  (cf. Équation 3).

### 3.4.5.3. Interprétation des résultats

Il faut tout d'abord faire le constat que la base *Checkerboard* est une base complexe : les méthodes classiques en mode « batch » n'obtiennent que des performances de classification moyennes entre 56% et 76% (voir la Table 12). D'autres approches connues pour leurs performances dans ce type d'environnement non-stationnaire comme DWM ou *Adaboost* n'obtiennent guère de meilleurs résultats. Ce phénomène vient principalement de la complexité introduite dans les classificateurs liés à la présence d'un environnement en changement permanent causé par une vitesse de dérive de concept très instable. Les approches de classification développées pour un environnement non-stationnaire sont souvent attachées

à une configuration spécifique des données, ce qui contraint fortement le système devenu moins adaptable à des configurations variables (comme le changement intempestif de vitesses dans le cas du *Checkerboard*). Cette évaluation est donc particulièrement intéressante dans la mesure où elle permet d'illustrer la capacité de chaque système à s'adapter à la complexité d'un tel cas.

Globalement, la version standard du mOC-iSVM offre des performances comparables aux autres classificateurs dédiés à l'apprentissage continu comme SEA, DWM, L++.NSE (version NB, CART), *Adaboost* (version NB). Par contre, ces performances restent plus faibles que celles obtenues par L++.NSE (version SVM) et *Adaboost* (version SVM et CART). En revanche, nos trois versions mOC-iSVM.EP, mOC-iSVM.AP, et mOC-iSVM.nB présentent des performances beaucoup plus élevées que l'ensemble des classificateurs testés (avec une différence de 4 à 7% avec L++.NSE version SVM), atteignant des taux avoisinant les 90% de reconnaissance, grâce au renforcement de connaissances durant l'apprentissage et la faculté du système à s'adapter à la dynamique de l'évolution de l'environnement.

Un inconvénient (qui peut être jugé rédhibitoire selon le cas) est que le mOC-iSVM et ses variantes présentent des valeurs d'écart-types beaucoup plus élevées que les autres classificateurs. Ce phénomène, déjà expliqué dans la section précédente pour les environnements non-stationnaires (section 3.4.4.3) peut ici être expliqué par la grande instabilité locale des SVs. Celle-ci est compensée par une grande stabilité globale qui se traduit par des performances moyennes très satisfaisantes. On observe également de petites chutes de performance au moment de l'apparition de la dérive de concept, accompagnées d'une adaptation très rapide en quelques étapes seulement, tandis que les autres méthodes nécessitent un plus grand nombre d'étapes pour améliorer leurs résultats (plus d'une dizaine d'étapes) et obtenir leur score optimal. Ce phénomène est très visible dans le cas de CB Pulse et CB Sinus (voir Figure 33, Figure 32).

Performance (SAC) (%) et Écart Type - EctT (%)								
Approach	CB Const	EctT	CB Pulse	EctT	CB Exp	EctT	CB Sin	EctT
L++.NSE (NB)	69.9	1.3	70.5	1.6	69.1	1.4	71.1	1.5
Single (NB)	56.6	1.7	54.3	1.7	56.5	1.7	55.3	1.7
DWM (NB)	59.6	1.6	56.4	1.7	59.6	1.7	57.9	1.7
SEA (NB)	60.1	1.2	64.2	1.6	61	1.3	63.7	1.4
Adaboost (NB)	59.9	1.7	59	1.8	59.9	1.6	59.2	1.7
L++.NSE (SVM)	81.9	0.9	84	0.7	81.6	0.9	83.5	0.9
Single (SVM)	76.6	1.5	79.9	1.5	76.6	1.5	78.6	1.5

<b>SEA (SVM)</b>	<b>71.6</b>	<i>0.7</i>	<b>78.5</b>	<i>0.6</i>	<b>73</b>	<i>0.7</i>	<b>75.4</b>	<i>0.6</i>
<b>Adaboost (SVM)</b>	<b>81</b>	<i>1</i>	<b>82</b>	<i>1.1</i>	<b>80.2</b>	<i>1.1</i>	<b>81.9</b>	<i>1.1</i>
<b>L++.NSE (CART)</b>	<b>77.3</b>	<i>1.1</i>	<b>81.2</b>	<i>1</i>	<b>77</b>	<i>1.1</i>	<b>79.4</b>	<i>1</i>
<b>Single (CART)</b>	<b>67.8</b>	<i>1.9</i>	<b>69.3</b>	<i>2.6</i>	<b>67.7</b>	<i>1.9</i>	<b>68.7</b>	<i>2.3</i>
<b>SEA (CART)</b>	<b>69.3</b>	<i>0.9</i>	<b>77.2</b>	<i>0.8</i>	<b>70.6</b>	<i>0.9</i>	<b>73.2</b>	<i>0.8</i>
<b>Adaboost (CART)</b>	<b>74.7</b>	<i>1.3</i>	<b>80.5</b>	<i>1</i>	<b>74.8</b>	<i>1.3</i>	<b>77.4</b>	<i>1.2</i>
<b>mOCiSVM</b>	<b>73.1</b>	<i>5.4</i>	<b>75.1</b>	<i>8.8</i>	<b>72</b>	<i>6.9</i>	<b>74</b>	<i>5.7</i>
<b>mOC-iSVM.AP</b>	<b>87</b>	<i>2.6</i>	<b>86.56</b>	<i>4.4</i>	<b>86.9</b>	<i>2.7</i>	<b>87.4</b>	<i>3.7</i>
<b>mOC-iSVM.EP</b>	<b>88.2</b>	<i>4.5</i>	<b>87.9</b>	<i>7.8</i>	<b>89</b>	<i>3.7</i>	<b>89.1</b>	<i>5.9</i>
<b>mOC-iSVM.nB</b>	<b>86.9</b>	<i>2.3</i>	<b>87</b>	<i>2.7</i>	<b>89.1</b>	<i>2.4</i>	<b>87.1</b>	<i>3.2</i>

Table 12 - Résumé des résultats sur CB Dataset.

### Détails de l'évaluation sur CB-Const.

Dans le cas de la base CB Const (voir Figure 30), l'environnement non-stationnaire change avec une vitesse constante : l'environnement récursif revient dans la période de  $\pi$  à  $2\pi$ . L++.NSE (SVM) s'adapte très bien à ce type de changement en réutilisant les informations utiles issues des classificateurs précédents (qui ont été définis à la période allant de 0 à  $\pi$ ). Ses performances sont plus élevées en environnement récursif qu'en environnement non-stationnaire avec changement abrupt. Le même phénomène peut être observé dans nos méthodes (et plus particulièrement dans le cas mOC-iSVM.EP). Mais avec une capacité d'adaptation au changement bien meilleure : on n'observe pas les chutes de performances aux premières étapes suivant les points de changements d'environnement (voir Figure 30, les étapes 100<sup>ème</sup>, 200<sup>ème</sup>, 300<sup>ème</sup> et 400<sup>ème</sup>). La version standard du mOC-iSVM maintient des scores moyens supérieurs à L++.NSE, *Adaboost*, SEA, DWM version NB, mais néanmoins inférieurs à L++.NSE version SVM, CART et nos trois versions de mOC-iSVM.

Dans le cas de la base CB Const (voir Figure 30), l'environnement change avec une vitesse constante : la réoccurrence de situations précédentes survient dans la période de  $\pi$  à  $2\pi$ . L++.NSE (SVM) s'adapte très bien à ce type de changement en réutilisant les informations utiles issues des classificateurs précédents (qui ont été définis à la période allant de 0 à  $\pi$ ). Ainsi, ses performances sont plus élevées quand l'environnement est réoccurent (entre  $\pi$  à  $2\pi$ ) par rapport à la première occurrence (entre 0 à  $\pi$ ). Le mOC-iSVM.EP possède également une bonne capacité d'adaptation aux changements sur ce problème. On notera d'ailleurs qu'il n'y a pas de chute de performances lors des étapes suivant les changements d'environnement comme on pouvait les observer sur le problème précédent (voir Figure 30, étapes 100, 200, 300 et 400). C'est également le cas pour les versions mOC-iSVM.AP et

mOC-iSVM.nB qui sont très performantes également quoique légèrement inférieures à celles de mOC-iSVM.EP sur les dernières étapes. La version standard du mOC-iSVM obtient, quant à elle, des performances honorables au regard de nombreuses autres approches (*Adaboost*, SEA, DWM version NB). Elles sont également en moyenne meilleures que L++.NSE, mais ce dernier affiche une tendance générale à la hausse ce qui n'est pas le cas du mOC-iSVM. Par ailleurs, L++.NSE version SVM et CART sont plus performantes que cette version standard de nos approches.

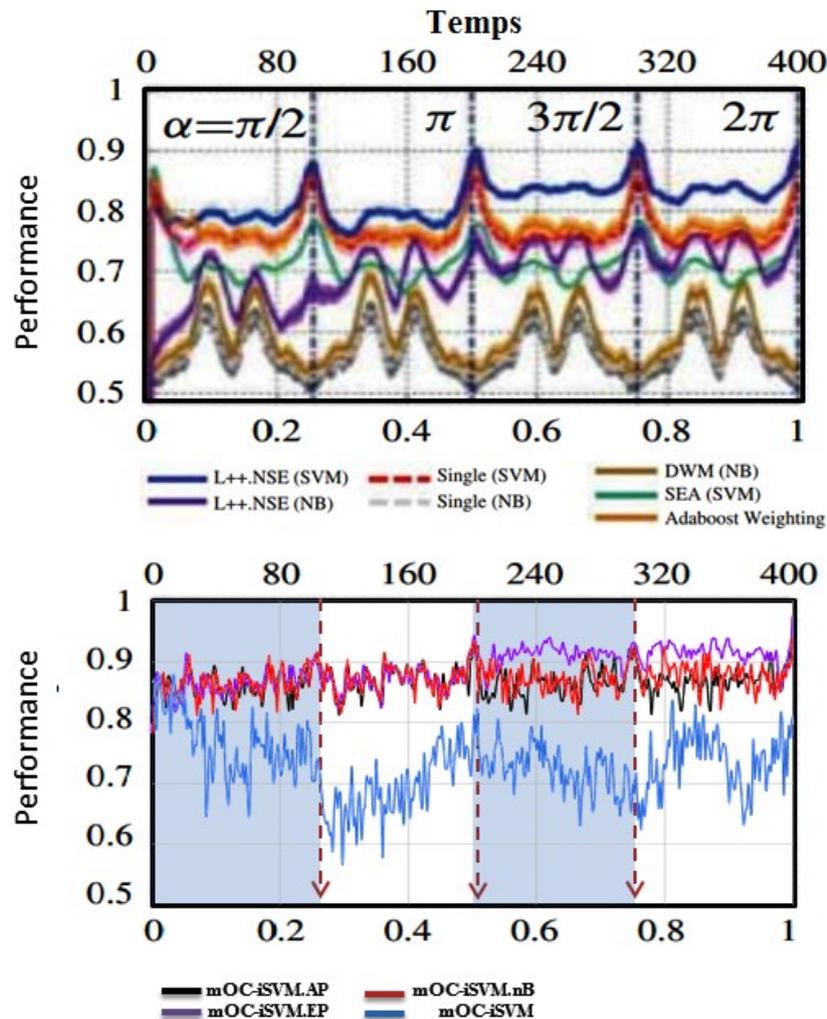


Figure 30 - Courbe d'évolution des performances (*GA*) sur la base CB Constant

#### Détails de l'évaluation sur CB-Exp.

Dans le cas de la base CB-Exp (voir la Figure 32), l'environnement est caractérisé par une évolution constante de la vitesse sur la période allant de 0 à  $\pi$ . Cette évolution se déroule sur plus de 200 étapes. L'environnement durant cette période peut être assimilé à un environnement quasiment stationnaire. Sur les autres périodes, la vitesse d'évolution est

beaucoup plus rapide (augmentation exponentielle), spécialement dans la dernière période. L'évolution des performances de L++.NSE sur cette base ressemble à celle constatée avec le jeu de données « CB Constant ». On retrouve quatre chutes de performances et des résultats meilleurs dans les deux dernières phases qui correspondent aux situations récurrentes. Dans nos trois versions de mOC-iSVM, les performances du mOC-iSVM.nB sont très bonnes au début sur la longue période d'évolution lente. Dans les deux périodes de grands changements (entre  $\pi$  et  $3\pi/2$  puis entre  $3\pi/2$  et  $2\pi$ ), c'est au tour du mOC-iSVM.EP de s'adapter très rapidement et d'acquies de bons résultats. Le mOC-iSVM.AP fonctionne correctement dans cet environnement avec une stabilité globale satisfaisante, mais avec l'incapacité de ré-exploiter les connaissances plus anciennes pourtant nécessaires ici pour gérer la récurrence. C'est pour cette raison que la version AP est globalement moins performante que les deux autres versions (EP et nB). Enfin la version standard mOC-iSVM obtient encore une fois des résultats moyens, comme dans le cas précédent. Cependant, on constate tout de même une chute des performances lors des dernières étapes correspondant à l'accélération des changements.

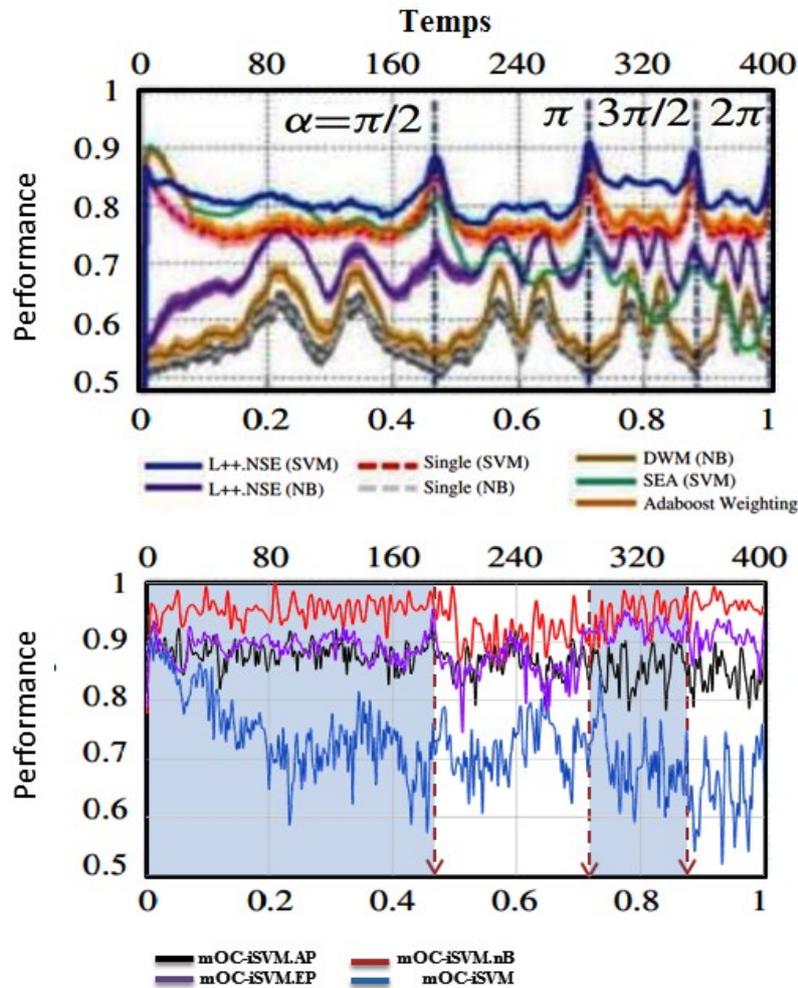


Figure 31 - Courbe d'évolution des performances ( $GA$ ) sur la base CB Exp

### Détails de l'évaluation sur CB-pulse et CB-sinus.

Dans le cas de l'évolution provoquée par la fonction sinus (jeu de données CB-Sinus) et l'impulsion gaussienne (jeu de données CB-Pulse), on observe une alternance entre une longue période (une centaine d'étapes) pendant laquelle la vitesse de changement varie très lentement et des périodes courtes avec une vitesse qui varie très vite (voir Figure 32 et Figure 33). Le point important à remarquer dans ces expérimentations est que les écarts de variations de performances lors des périodes très dynamiques (changements les plus importants) sont réduits pour nos approches (10% d'amplitude maximum hors version standard) par rapport aux écarts observés sur les autres approches de la littérature (plutôt de l'ordre de 20%). Ainsi, L++.NSE et les autres méthodes sont caractérisées par des adaptations lentes accompagnées d'importantes chutes de performances et de très longues plages d'adaptation nécessaires pour retrouver le niveau optimal. Globalement, nous observons que les versions étendues de mOC-iSVM s'adaptent rapidement à l'environnement et que les pertes de performances sont

minimisées en amplitude. En revanche, cela n'est pas le cas pour mOC-iSVM qui présente des caractéristiques semblables à L++.NSE, avec des performances plus faibles.

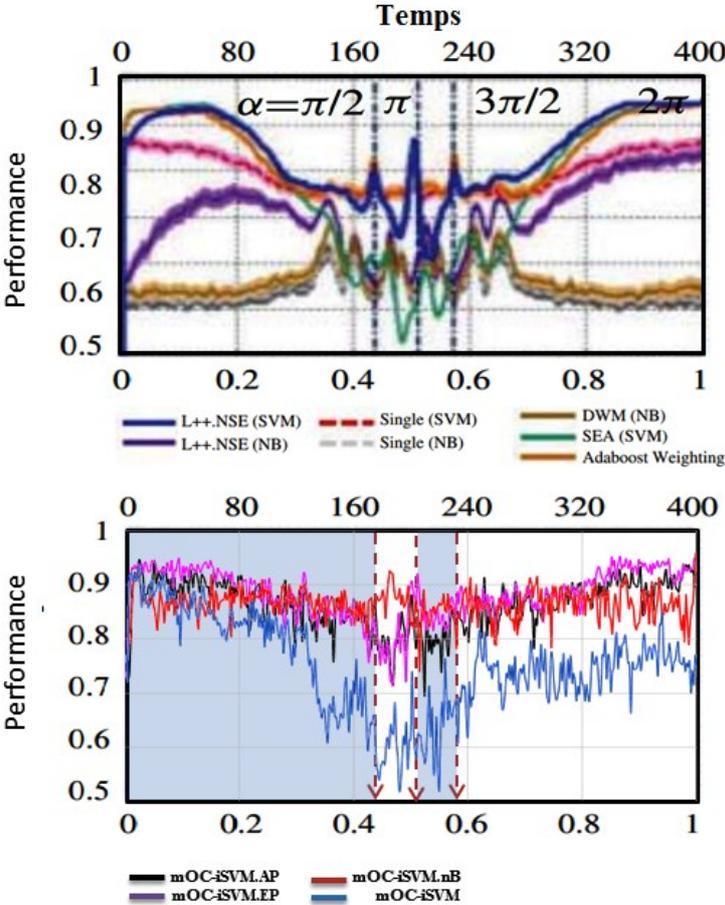


Figure 32 - Courbe d'évolution des performances (*GA*) sur la base CB Pulse

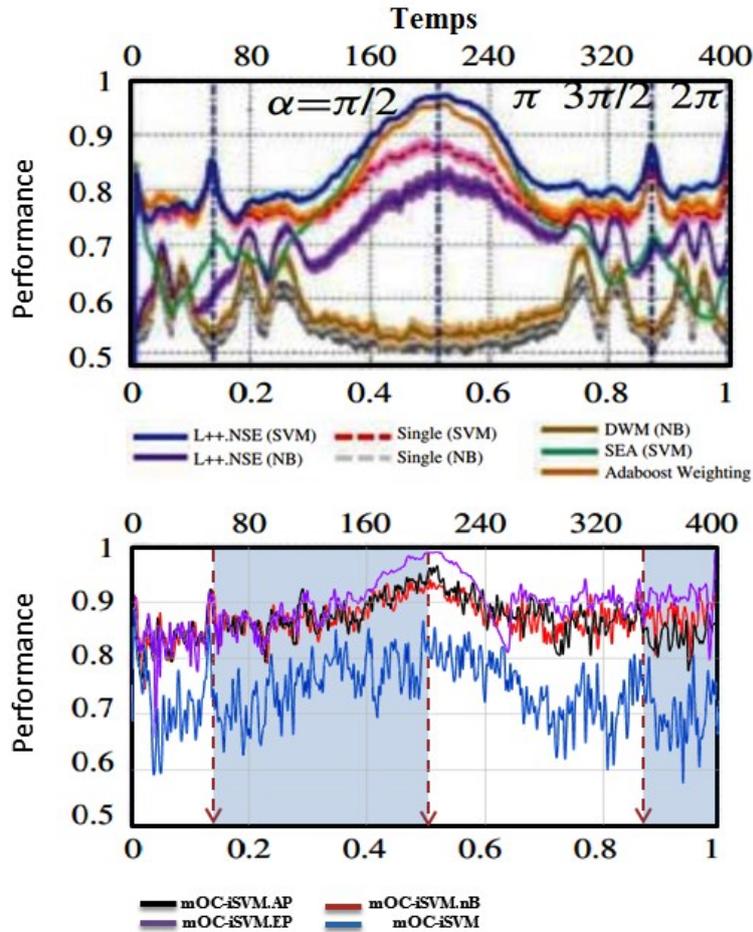


Figure 33 - Courbe d'évolution des performances ( $GA$ ) sur la base CB Sinusoïdal

### Bilan sur les variantes de mOC-iSVM avec des paramètres spécifiques $\alpha$ , $f$ , et $n$ .

Concernant les performances des différentes versions du mOC-iSVM, nous pouvons tout d'abord constater que pour mOC-iSVM.AP, le choix du paramètre  $\alpha$  (voir Table 13, Table 14, Table 15, Table 16) est très dépendant de la « dynamique » de l'environnement. L'écart type est ainsi d'autant plus grand que les variations sont rapides. Par exemple, dans CB Const (l'évolution est très lente), il n'y a pas de grande différence provoquée par les différentes valeurs de  $\alpha$  ; l'écart-type (resp. la performance) ne change pas de plus de 1% (resp. de 3%). Dans CB-Sinus et CB Exp, comme il y a des périodes où les évolutions sont très rapides, les valeurs d'écart-type et de performance deviennent très dépendantes des choix effectués pour les valeurs de  $\alpha$  (plus de 5% pour l'écart-type et plus de 6% pour la performance notamment lorsque  $\alpha$  est trop grand par rapport aux vitesses de changement). Nous pouvons ainsi en conclure que dans un environnement présentant des dérives rapides, il est préférable de choisir des valeurs de  $\alpha$  faibles pour assurer une adaptation optimale au contexte.

CB Const										
Seuil $\alpha$	1	2	3	4	5	6	7	8	9	10
Performance (SAC) (%)	81.8	84.74	86.27	86.82	86.95	86.82	86.47	86.04	85.45	84.77
Écart Type (%)	3.9	2.85	2.62	2.28	2.15	2.16	2.06	2.12	2.14	2.15
Temps de calcul (s)	96.54	105.47	103.11	110.19	114.19	114.68	118.81	125.94	129.46	127.98

Table 13 - Résultats du mOC-iSVM.AP avec différentes valeurs de  $\alpha$  sur CB Const.

CB Pulse										
Seuil $\alpha$	1	2	3	4	5	6	7	8	9	10
Performance (SAC) (%)	82.8	85.5	86.56	86.87	86.96	86.71	86.33	85.93	85.59	85.19
Écart Type (%)	4.93	4.36	4.38	4.99	5.91	6.76	7.65	8.58	9.26	9.8
Temps de calcul (s)	99.27	105.01	109.69	109.47	118.25	117.53	123.13	126.36	131.57	129.64

Table 14 - Résultats du mOC-iSVM.AP avec différentes valeurs de  $\alpha$  sur CB Pulse.

CB Sinus										
Seuil $\alpha$	1	2	3	4	5	6	7	8	9	10
Performance (SAC) (%)	82.74	85.86	87.01	87.4	87.43	87.23	86.78	86.27	85.6	84.97
Écart Type (%)	5.1	3.82	3.58	3.72	3.98	4.49	5.04	5.75	6.45	7.05
Temps de calcul (s)	104.16	100.69	106.69	113.78	112.74	112.4	119.83	155.59	129.89	122.53

Table 15 - Résultats du mOC-iSVM.AP avec différentes valeurs de  $\alpha$  sur CB Sinus.

CB Exp										
Seuil $\alpha$	1	2	3	4	5	6	7	8	9	10
Performance (SAC) (%)	81.51	84.56	86.17	86.91	87.11	87.07	86.77	86.23	85.53	85.01
Écart Type (%)	3.99	3.18	2.65	2.7	2.95	3.27	3.71	4.23	4.77	5.39
Temps de calcul (s)	100.27	105.47	108.41	111.4	113.1	117.4	123.3	126.4	129.6	176.68
				7	3	9		6	4	

Table 16 - Résultats du mOC-iSVM.AP avec différentes valeurs de  $\alpha$  sur CB Exp.

Pour la version mOC-iSVM.EP, les tableaux suivants (voir Table 17, Table 18, Table 19, et Table 20) illustrent la difficulté à établir des valeurs de seuils de performances  $f$  efficaces en environnement non-stationnaire. Une variation de seulement 2% du seuil peut induire des variations très importantes de performances (voir Table 17 par exemple). La connaissance des performances globales attendues du système peuvent aider à choisir le bon seuil à retenir. Cette version est moins aisée à exploiter lorsqu'on ne dispose pas de cette information. Nous pouvons enfin noter les temps de calcul importants qui sont requis dans cette version (plus de 20 fois ceux de la version mOC-iSVM.AP).

CB Pulse							
Seuil $f$ (%)	75	77.5	80	82.5	85	87.5	90
Performance (SAC) (%)	88.79	88.41	87.87	88.23	82.76	82.46	76.51
Écart Type (%)	6.62	6.73	7.75	4.58	11.5	11.52	7.75
Temps de calcul (s)	3129.47	2700.57	2472.49	1750.89	2248.29	2117.61	1369.92

Table 17 - Résultats du mOC-iSVM.EP avec différentes valeurs de  $f$  sur CB Pulse.

CB Sinus							
Seuil $f$ (%)	75	77.5	80	82.5	85	87.5	90
Performance (SAC) (%)	89.24	89.8	87.22	83.19	79.88	78.34	77.61
Écart Type (%)	5.33	4.73	7.71	10.17	10.53	10.04	9.65
Temps de calcul (s)	2267.87	2209.2	1981.34	1830.45	1797.71	1675.38	1476.42

Table 18 - Résultats du mOC-iSVM.EP avec différentes valeurs de  $f$  sur CB Sinus.

CB Exp									
Seuil $f$ (%)	70	72.5	75	77.5	80	82.5	85	87.5	90
Performance (SAC) (%)	87.86	88.54	88.91	89.15	89	87.02	78.27	74.91	74.82
Écart Type (%)	3.29	3.09	3.07	3.05	3.65	6.43	8.27	5.73	5.59
Temps de calcul (s)	2075.13	1878.77	1811.05	1704.78	1625.98	1571.15	1450.51	1365.91	1615.01

Table 19 - Résultats du mOC-iSVM.EP avec différentes valeurs de  $f$  sur CB Exp.

CB Const									
Seuil $f$ (%)	70	72.5	75	77.5	80	82.5	85	87.5	90
Performance (SAC) (%)	88.26	88.81	89.2	89.16	88.15	84.48	76.84	74.84	74.61
Écart Type (%)	3.11	3.04	2.92	3.19	4.47	7.45	7.42	6.2	6.03
Temps de calcul (s)	2253.33	1897.89	2057.85	1804.12	1629.34	1466.56	1391.96	1361.64	1379.53

Table 20 - Résultats du mOC-iSVM.EP avec différentes valeurs de  $f$  sur CB Const.

Pour la version mOC-iSVM.nB, les résultats sont décrits dans les tableaux suivants : Table 21, Table 22, Table 23, et Table 24. On peut y observer de nombreux avantages à exploiter cette version du mOC-iSVM. Tout d'abord, on peut constater que les valeurs d'écart-type sont dans l'ensemble très stables (moins 2% de variations sur l'ensemble des expérimentations réalisées) comme cela est également le cas pour la version mOC-iSVM.AP. Ces performances absolues restent cependant très légèrement inférieures à celles du mOC-iSVM.EP. Par conséquent, si les connaissances *a priori* sur la nature de l'environnement et les performances atteignables ne sont pas disponibles, il est préférable d'utiliser la version mOC-iSVM.nB.

CB Const									
Seuil $n$	1	2	3	4	5	6	7	8	9
Performance (SAC) (%)	81.09	84.84	85.97	86.87	86.93	86.88	86.54	87.9	87.55
Écart Type (%)	3.93	2.8	2.86	2.31	2.36	2.36	2.38	2.71	2.79
Temps de calcul (s)	1445.36	1489.66	1505.34	1626.42	1628.32	1798	1681.87	1790.72	1883.9

Table 21 - Résultats du mOC-iSVM.nB avec différentes valeurs de  $n$  sur CB Constant.

CB Exp									
<i>Seuil n</i>	1	2	3	4	5	6	7	8	9
Performance (SAC) (%)	82.32	85.47	86.97	87.79	88.48	88.96	89.13	88.51	88.57
Écart Type (%)	3.85	2.71	2.65	2.41	2.31	2.37	2.38	2.41	2.56
Temps de calcul (s)	1573.15	1597.89	1632.34	1693.16	1989.96	1773.13	1730.35	1889.1	1734.14

Table 22 - Résultats du mOC-iSVM.nB avec différentes valeurs de  $n$  sur CB Exp.

CB Sinus									
<i>Seuil n</i>	1	2	3	4	5	6	7	8	9
Performance (SAC) (%)	81.21	85.6	86.68	87.09	87.11	87.15	86.75	88.1	87.84
Écart Type (%)	4.87	4.01	3.53	3.24	3.43	4.12	4.68	5.25	5.64

Table 23 - Résultats du mOC-iSVM.nB avec différentes valeurs de  $n$  sur CB Sinus.

CB Pulse									
<i>Seuil n</i>	1	2	3	4	5	6	7	8	9
Performance (SAC) (%)	82.32	85.47	86.97	86.8	87.51	87.17	87.33	88.41	88.27
Écart Type (%)	2.85	2.71	2.65	4.49	4.64	5.3	4.75	6.31	6.27
Temps de calcul (s)	1366.8	1459.44	1416.24	1448.61	1496.8	1093.59	1896.49	1823.37	1966.05

Table 24 - Résultats du mOC-iSVM.nB avec différentes valeurs de  $n$  sur CB Pulse.

### 3.5. Conclusion

À travers ces expérimentations, nous avons cherché à montrer la faisabilité de l'usage de classificateurs *one-class* pour l'élaboration d'un système de reconnaissance dynamique : mOC-iSVM. Ce modèle de base et ses trois variantes basées sur un filtrage différent des vecteurs supports selon leur âge ou leur performance, ont été implémentés et évalués. Nous avons comparé leurs performances avec les meilleurs classificateurs dynamiques du moment : SEA, DWM, L++.NC, L++.NSE. Nous avons obtenu des résultats très prometteurs surpassant, dans un grand nombre de cas, les scores des classificateurs de la littérature. Nous

avons également mis en évidence les forces et faiblesses de chacune de ces versions selon le temps de calcul, le critère de stockage et l'importance du paramétrage.

Le point très positif que nous avons souligné est que toutes les versions de mOC-iSVM sont utilisables quel que soit le type d'environnement considéré : stationnaire et non-stationnaire, à changement brusque ou progressif, avec ou sans réoccurrence. La plupart des autres classificateurs de la littérature sont spécialisés pour l'apprentissage dans des environnements soit stationnaires, soit non-stationnaires, mais rarement les deux (sauf peut-être Learn++.NSE).

L'étude qualitative des performances de chacun des classificateurs étudiés dans ce chapitre est finalement résumée à la Table 25. Elle permet de prendre connaissance des avantages et inconvénients dans le choix d'un classificateur selon les critères mentionnés jusqu'ici.

	<b>Performance</b>					
	<b>Performance en environnement stationnaire</b>	<b>Performance en environnement non-stationnaire</b>	<b>Écart-type</b>	<b>Temps de calcul</b>	<b>Critère de stockage</b>	<b>Contrôle des paramètres</b>
<b>mOC-iSVM Standard</b>	Bon	Moyen	Faible	Bon	Sans stockage	Sans paramètres
<b>mOC-iSVM.AP</b>	Bon	Bon	Moyen	Bon	Grand Stockage	Facile
<b>mOC-iSVM.EP</b>	Très bon	Très bon	Moyen	Faible	Grand Stockage	Difficile
<b>mOC-iSVM.nB</b>	Bon	Très Bon	Moyen	Faible	Grand Stockage	Facile
<b>L++.NC</b>	Moyen	impossible	Très Bon	Très Bon	Sans information	Sans information
<b>L++.NSE</b>	Impossible	Moyen	Très Bon	Très Bon	Sans information	Sans information
<b>Les autres...</b>	Faible	Moyen/Faible	Très bon/ Bon	Très Bon	Sans information	Sans information

Table 25 - Synthèse qualitative des classificateurs étudiés dans ce chapitre et de nos quatre approches de classification Multi one-class.

Troisième partie

Application à la numérisation cognitive

## IV. Application à la numérisation cognitive

**D**ans le contexte du projet DIGIDOC, les classificateurs proposés dans cette thèse doivent être repris pour définir un module de « numérisation cognitive » qui sera, à terme, intégré aux scanners pour aider au paramétrage en fonction des documents numérisés et de l'usage que l'on souhaite en faire par la suite.

*L'objectif applicatif de cette thèse est donc de mettre en place des mécanismes permettant au processus de numérisation de s'auto-adapter en fonction de la nature du document et de l'image à traiter ainsi qu'en fonction des objectifs de numérisation définis par l'opérateur. Ces derniers sont directement liés aux traitements futurs auxquels les images seront soumises et aux applications qui les manipuleront : OCR, archivage, recherche par le contenu... Ainsi, à la différence des processus de numérisation habituels qui procèdent en une capture unique à haute résolution, notre ambition est de permettre des acquisitions multiples et paramétrables en fonction des objectifs d'utilisation et du contenu même du document. C'est pourquoi nous parlons de numérisation cognitive ou de scanner intelligent.*

*Ce chapitre commence par un rapide passage en revue des caractéristiques des scanners du marché afin de comparer leurs capacités actuelles avec les objectifs visés dans le cadre du projet DIGIDOC. Les travaux scientifiques en lien avec la problématique de la classification d'images de documents, sous-jacente à la mise au point de scanner intelligent, sont également abordés dans ce chapitre.*

*Les objectifs initiaux du projet DIGIDOC sont ensuite rappelés avant de décrire en détail comment il est possible d'exploiter les modèles de classificateurs proposés dans les chapitres précédents afin de produire un prototype de « scanner intelligent ».*

*Pour finir, plusieurs scénarios permettant de valider l'intérêt des classificateurs mOC-iSVM, dans le cadre de la classification d'images de documents, sont proposés en fin de chapitre.*

## 4.1. Des scanners intelligents à la numérisation cognitive

La plupart des scanners actuels, qu'ils soient destinés à un large public ou bien à visée professionnelle, peuvent parfois être qualifiés de « scanners intelligents ». Cette appellation, à fort impact commercial, n'a pas de définition précise et regroupe un ensemble de fonctionnalités relatives à la fois aux avancées matérielles des scanners mais aussi et surtout aux suites logicielles qui leur sont associées. Le tableau ci-dessous donne quelques exemples de ces scanners aux fonctionnalités avancées. L'objectif n'est pas ici d'être exhaustif mais plutôt de fournir un échantillon représentatif de ce qui se fait à l'heure actuelle autour de l'aptitude du scanner à s'adapter au document.

Scanner	Technologies matérielles	Technologies logicielles
Panasonic KV- S5055C (2011)	<i>Toughfeed</i> : repère automatiquement les agrafes sur les documents et détecte le passage simultané de deux feuilles pour stopper la numérisation en cours  Peut numériser simultanément des documents de différentes tailles et de différentes épaisseurs depuis le chargeur de documents	Image Capture Plus : pour l'édition des pages scannées. Permet de : supprimer les éventuelles pages blanches ; de modifier l'ordre des pages ; de recadrer les images ; d'effacer automatiquement les traces laissées par d'éventuels trous sur le document original
i1320 de Kodak (2006)	scanne les deux côtés d'une feuille en un seul passage (2 CCD)	Numérisation de page; iThresholding; traitement de seuil adaptatif ; redressement ; découpage automatique ; recadrage relatif ; recadrage agressif ; suppression automatique de couleur ; numérisation à deux volets ; couleur interactive, le réglage du contraste et

		la luminosité ; orientation automatique ; détection automatique des couleurs ; lissage de la couleur de fond ; remplissage des bords d'image ; fusion de l'image ; traitement des contenus basés sur la détection des pages blanches ; filtrage de série ; remplissage du trou de l'image ; filtre de netteté.
IRISPen Express		Inclut un OCR (128 langues) : lecture d'images floues et déformées ; technologie de suivi de ligne unique ; gestion de la couleur du texte et du fond ; pratiquement n'importe quelle police, dans un large éventail de styles et de tailles (de 8 à 20 points), ainsi que des caractères spéciaux, des marques de lecture, des symboles spéciaux et des éléments de mise en forme, tels que les barres verticales séparant les cellules d'un tableau) ; alphabets mixtes ; traduction, synthèse vocale, reconnaissance de codes barre
I2S Digibooks	prévisualisation en temps réel, détection de format, mise au point automatique, temps d'exposition, correction d'éclairage,	redressement, correction de courbure, effacement des doigts, découpage des pages, accentuation des détails, accentuation du contraste, étalement des niveaux, courbe en S.

Table 26 – Scanners sur le marché.

Pour résumer, de nombreuses avancées techniques ont été faites sur le plan matériel et notamment sur la capacité du scanner à traiter une quantité importante de documents tout en gérant différents formats. Ces fonctionnalités sont souvent couplées à des systèmes permettant la détection de problèmes mécaniques au plus tôt pour éviter les blocages. Bien que ces avancées soient nécessaires, elles ne concernent pas directement le projet DIGIDOC qui est centré sur la numérisation elle-même.

Sur cet aspect, certains scanners avancés permettent de régler automatiquement plusieurs paramètres en fonction du document. Ils s'appuient pour cela sur des technologies que l'on retrouve fréquemment dans le domaine de l'optique ou de la vision pour le calibrage de capteurs. Plus le scanner est avancé et plus cette mise au point peut se faire sur un nombre de paramètres importants (*cf.* Digibooks). Ce paramétrage automatique, tout à fait intéressant dans notre contexte, n'en reste pas moins limité, pour deux raisons. La première est qu'il nécessite souvent une intervention de l'opérateur pour vérifier le calibrage, notamment en fonction du résultat souhaité. Cela est particulièrement vrai lorsqu'il s'agit de numériser des documents pour lesquels on souhaite une grande fidélité (patrimoine en particulier). La seconde raison, qui résulte en partie de la première, est que le temps nécessaire au calibrage est souvent conséquent. Il ne peut donc se faire que ponctuellement.

Bien que cela n'apparaisse pas explicitement dans le tableau ci-dessus, les scanners possèdent aussi souvent des profils standards de numérisation qui peuvent être choisis par l'utilisateur, voire automatiquement par un pré-scan. Ces profils permettent d'adapter notamment la résolution et la couleur en fonction du contenu du document (image, image et texte, texte seul) afin de trouver un compromis entre la taille des images acquises et l'information qu'elles contiennent. Là encore, bien que cette notion de profil corresponde à ce que l'on souhaiterait avoir dans un scanner intelligent, leur nombre reste très limité et ceux-ci ne dépendent absolument pas de l'usage que l'on souhaite faire du document numérique.

Une grande majorité (pour ne pas dire la quasi-totalité) de ces scanners embarque également des outils logiciels qui permettent « d'améliorer la numérisation » en corrigeant les défauts principaux (recadrage, suppression de pages blanches, amélioration du contraste, suppression du « bruit » ou de défauts (trous, doigts, *etc.*), *etc.*). Il faut bien comprendre ici que cette « amélioration de la numérisation » constitue un abus de langage dans le sens où la numérisation n'est en rien changée. Seul le résultat de la numérisation, c'est-à-dire l'image, est amélioré, dans le but de satisfaire au mieux à des critères visuels subjectifs de l'utilisateur standard. Ces améliorations sont le plus souvent standards et ne correspondent bien souvent qu'à un usage de consultation du document pour un utilisateur quelconque. La seule exception concerne l'ensemble des traitements, là encore standards, qui serviront à améliorer, outre le rendu de la numérisation, les performances d'un OCR.

Dans ce travail de thèse, l'appellation « scanner cognitif » (ou « scanner intelligent ») souligne l'intégration, au sein du scanner, d'un dispositif logiciel (semi-automatique supervisé par un utilisateur expert) capable de diagnostiquer la nature d'un contenu, ou sa qualité au fur

et à mesure des acquisitions. L'objectif ici est naturellement de contribuer à une numérisation de qualité éventuellement par plusieurs captures selon l'objectif. Des mécanismes de classification des contenus, permettant d'aider la prise de décision pour la validation des acquisitions faites au fil de la chaîne de numérisation, sont donc à mettre en place. Nous parcourons donc ici les principales méthodes actuellement exploitées dans le contexte de la classification de documents.

## 4.2. Panorama des méthodes de classification de documents

La mise en place, dans les scanners, d'une capacité de détection du type de documents, de la qualité ou des défauts de numérisation, nécessite l'utilisation de techniques de classification d'images de documents. Cette section synthétise les principaux travaux menés dans ce domaine ces dernières années.

Dans cette section, le document sera vu comme une organisation d'objets (de symboles textuels et graphiques de toutes sortes) ayant une disposition aléatoire ou structurée de ses éléments de contenu. La classification des documents sera envisagée ici comme le processus capable de regrouper divers documents en sous-ensembles homogènes sur la base de représentation et de description de leurs contenus (en lien avec les structures physiques ou logiques) de telle sorte que les observations dans les mêmes classes de documents soient les plus similaires possible, et que les observations entre les classes soient les plus dissemblables.

### 4.2.1. Architecture de la chaîne de traitement

Une tâche classique de classification d'images comporte généralement les étapes suivantes :

- i. La préparation des données : pré-traitement et amélioration de la qualité des images.
- ii. L'extraction et la sélection de caractéristiques discriminantes.
- iii. La comparaison des données entre elles (non supervisé) ou via un modèle obtenu à l'aide de données d'apprentissage (supervisé) et à l'aide d'une mesure de proximité/similarité.
- iv. Eventuellement des post-traitements de vérification de la décision.

L'état de l'art dressé par Chen [Chen 2007], dont nous conseillons la lecture, fournit un excellent panorama des techniques usuelles de classification d'images de documents. La Figure 34 décrivant les points fondamentaux de ces systèmes est tirée de ce document.

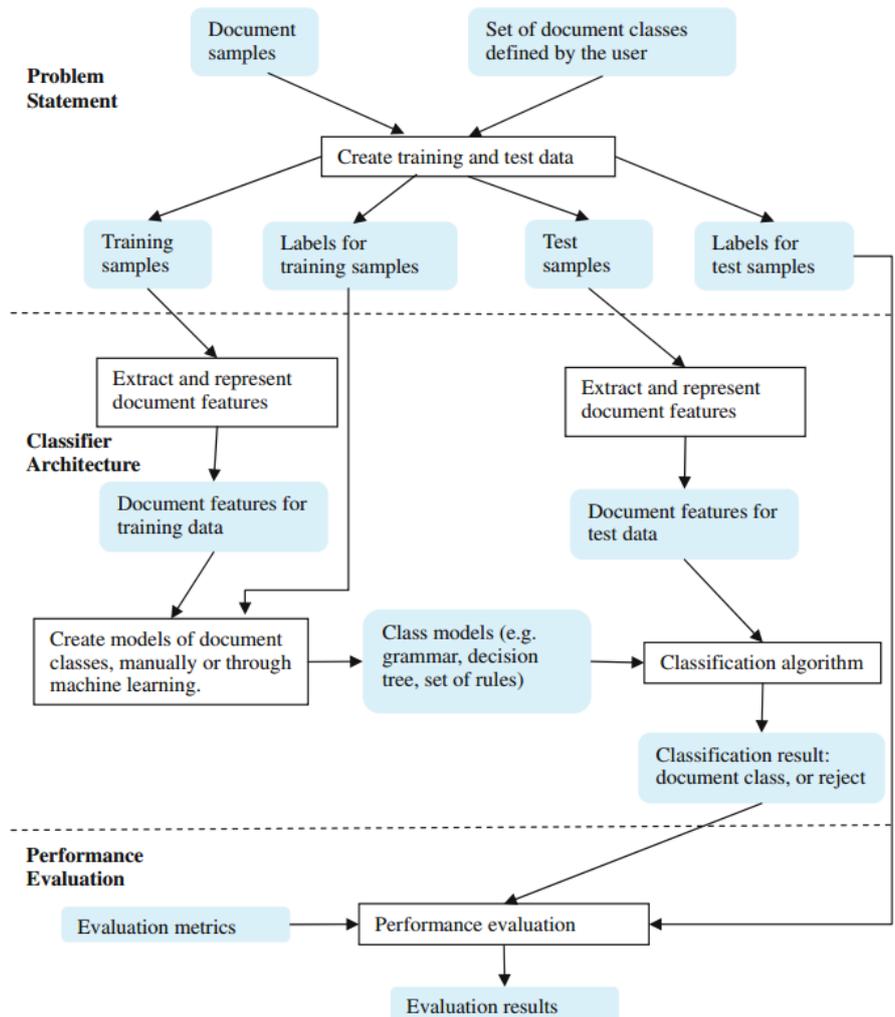


Figure 34 - Architecture extrait de [Chen 2007]

Depuis cet état de l'art, les techniques de classification ont considérablement évolué et, il est aujourd'hui devenu plus difficile de faire la distinction entre les différentes étapes issues du processus classique de classification. Nous proposons donc, dans la suite, une remise à jour de ce panorama de l'état de l'art selon un découpage différent mettant plus en valeur les évolutions les plus récentes. Dans les sections suivantes, nous instancierons les étapes précédentes au cas spécifiques des données numérisées, avec un impact notamment pour les étapes de prétraitement.

#### 4.2.2. Préparation des images

L'étape de préparation des données est obligatoire lorsque les outils utilisés pour extraire les descripteurs ne sont pas robustes au bruit ou aux transformations géométriques. Son objectif n'est pas de procurer de l'information supplémentaire sur le contenu de l'image mais de simplifier la suite de l'analyse en normalisant ou nettoyant les données.

Les principales sources de dégradation introduites dans les images lors d'une numérisation sont liées essentiellement au positionnement de la feuille à numériser, à la présence de défauts dans l'image (dus à un mauvais paramétrage du scanner ou inhérents à l'image elle-même, comme la présence de bruit sur la forme ou le fond) et plus généralement à la présence d'une bordure très visible (bords noirs, présence d'autres feuillets sur l'image...)

### ***Redressement des pages***

Lors de la numérisation d'un document (surtout lors d'acquisition de flux), il est fréquent que la feuille ne soit pas positionnée parfaitement (horizontalement) sur le scanner. De nombreuses techniques ont été mises au point pour corriger ce défaut. On peut citer celles se basant sur les profils de projection, la détection de lignes par transformée de Hough [Louloudis 2006] ou plus récemment par différence de gradients et R-signature [Felhi 2011]. Au-delà du redressement de l'inclinaison globale de la page, il peut également être nécessaire de corriger les défauts d'inclinaison impactant les lignes de texte, essentiellement dues à la présence de la reliure au centre d'un ouvrage et qui peut dénaturer le texte et compromettre sa reconnaissance, [Trinh 2003].

### ***Traitement du bruit***

Le bruit est très souvent présent sur les images de documents, provenant soit d'impressions successives ou de dégradations subies avant numérisation, soit de la numérisation elle-même. Des traitements classiques tels que les filtres (filtres médians [Arias-Catro 2009], [Arce 2005] ; filtres linéaires de Wiener [Lim 1990]) sont utilisés pour filtrer des pixels isolés sur l'image (bruit impulsionnel). Les opérateurs de morphologie mathématique tels que la dilatation ou l'érosion et leurs compositions (ouverture et fermeture) sont également très souvent utilisés pour corriger de légers défauts sur les images ou recoller des morceaux de caractères cassés [Bishnoi 2014]. On peut également trouver des traitements plus avancés de dé-bruitage d'images procédant par régularisation de la diffusion champs de gradients apparentés à des filtres anisotropes rehaussant les traits tout en lissant l'arrière-plan ([Drira 2006]).

### ***Suppression des bords***

Il arrive également souvent que la numérisation introduise l'apparition de régions noires sur les bords des images de document (ou la présence d'autres éléments parasites, comme des doigts, d'autres feuillets...). Un recentrage de l'image autour de l'information d'intérêt doit alors être réalisé, afin que ces composantes ne perturbent pas la suite de l'analyse [Shafait 2009].

### 4.2.3. De la représentation à la décision

#### 4.2.3.1. Approches perceptuelles

En classification d'images naturelles, il est courant d'utiliser des descripteurs tels que la couleur, la texture et la forme pour caractériser des images ou parties d'images. Bien que souvent binaires ou en niveaux de gris, ces descripteurs ont également été exploités pour caractériser les images de documents.

En dehors de ceux en lien avec la forme, parmi les descripteurs pouvant être exploités dans un contexte appliqué aux images de documents (qui peuvent s'apparenter selon le cas à des images de textures, des images de traits géométriques, des images de plages de couleurs), on peut citer:

- Descripteurs globaux de couleur :
  - Histogramme de couleurs (RGB Color Histogram) [Sergyan 2008]
  - *Scalable Color Descriptor* (SCD),
- Descripteurs globaux de texture :
  - Texture de Tamura [Keysers 2007]
  - Texture de Haralick [Chen 2006]
- Descripteurs construits par analyse des plages blanches/noires :
  - Histogramme de « *Run-Length* » [Gordo 2013], [Keysers 2007]
  - Distribution des « *Run-Length* » [Bulacu 2007]
- Descripteurs de contraste et de flou :
  - Différence du contraste gaussien (D.O.G, *Difference of gaussian contrast*) [Tadmor 2000]
  - Contraste entre des 8-voisinages locaux (*8-Neighborhood local contrast*) [Rizzi 2004]
  - Différence du contraste gaussien appliquant aux 8-voisinages locaux (D.O.G *Neighborhood contrast*) [Rizzi 2008]

- Descripteurs des orientations et des courbures :
  - Cooccurrence horizontale des angles de courbures et distribution des « edge-hinge » (*Horizontal co-occurrence of edge angles & Edge-hinge distribution*) [Bulacu 2007]
  - Distribution des transitions des composants dans la ligne [Beusekom 2009]
  - Plages verticales (Vertical Component Run, VCR) [Lu 2006]

Afin d'illustrer, de manière non exhaustive, les descripteurs visuels pouvant être exploités pour classifier des images de documents, les travaux de [Keysers 2007] constituent un bon exemple. Ils proposent la combinaison assez complète de descripteurs suivante:

- TTFH : Histogramme de texture de Tamura (*Tamura Texture Features Histogram*)
- RIFH : Histogramme des descripteurs invariants (*Relational Invariant Feature Histograms*)
- DSI : Imagettes sous-échantillonnées redimensionnées (*Down-scaled images of size 32×32*)
- FR : Ratio de remplissage (*Fill Ratio*)
- RLH : Histogramme de plages noires (*Run-Length*)
- RL {B,W} de plages noires/blanches
- RL {X,Y,M,S}V : Un vecteur contient des valeurs du nombre total de pixels noirs, la moyenne de pixels noirs trouvés sur la ligne médiane, la variance des plages noires et des plages blanches dans les directions horizontales, verticales, diagonales ([Wang 2006])
- CCXH, CCYH : Histogramme de la hauteur et de la largeur des composants connexes.
- CCXYH : Distribution de la hauteur et de la largeur des composants connexes.
- CCNNH : Histogramme des distances entre des composants connexes et des composants voisins.

Dans le cas du document, il faut noter qu'il existe une grande diversité de descripteurs de forme (binaires ou en niveaux de gris) permettant d'apprécier ou de quantifier leur niveau de rondeur, de sinuosité, d'étalement par exemple ou encore de renseigner sur leur complexité et leur densité, de façon globale. Cependant il faut bien constater que l'utilisation de caractéristiques décrivant l'image dans sa globalité est souvent insuffisante pour caractériser un contenu et doit être complétée par des descripteurs spatiaux locaux. Dans ce cadre,

l'approche la plus simple consiste à découper l'image à l'aide d'une grille de taille fixe ou encore à exploiter la notion de fenêtre glissante. L'extraction des descripteurs globaux précédemment décrits a alors lieu à l'intérieur de chacune des zones (caractéristiques locales) plutôt que globalement. À titre d'exemple, les travaux dans [Chen 2006] peuvent être cités :

- Découpage de l'image en tuiles (5x5)
- Extraction caractéristique
  - Densité
  - Projection de H et V
  - Corrélogramme (corrélation spatiale des couleurs)
  - Lignes
  - Taille des CC

De nombreuses variantes dans le mode de découpage (grilles, tuiles recouvrantes, fenêtre glissante, multi-résolution, ...) ont été proposées [Rizzi 2004], [Lemaitre 2007].

La plupart des méthodes procédant à une analyse globale sont très sensibles à la présence de graphiques. Sakhi souligne ce phénomène dans [Sakhi 2012]. En effet, les graphiques sont souvent très complexes et de formes variables. Leur description peut manquer de robustesse si elle n'intègre pas une bonne prise en charge de la diversité (invariance à l'échelle, aux transformations, aux dégradations, aux occlusions partielles...) Cela est particulièrement vrai pour les images composées d'une grande mixité d'éléments graphiques et textuels, où l'homogénéité de la page n'est plus garantie. Sakhi cite quelques éléments visuels qui peuvent être la cause de ces difficultés :

- Présence de tableaux et de bordures épaisses sur les images de publicité.
- Présence d'interlignes ou d'espaces inter-mots insuffisants
- Présence de défauts d'alignement du fait du trop grand proximité des mots et d'espaces interlignes insuffisants.
- Présence de signes de ponctuations, diacritiques ou de caractères brisés mal alignés avec les mots auxquels ils sont associés.

Pour pallier ce problème, certains auteurs proposent de procéder à une analyse de structure de la page à classifier. C'est alors plus la manière dont est organisée la page que le type des éléments contenus qui est utilisé pour discriminer les documents. Comme nous allons le voir dans la section suivante, ce type d'approches est intéressant mais nécessite l'introduction de plus de connaissances *a priori* pour pouvoir bien traiter les images.

#### 4.2.3.2. Approches orientées structure et éléments de contenus

Pour caractériser la structure d'un document, la principale problématique est de segmenter le document en différentes parties (également appelées blocs) puis de représenter cette structure sous forme d'arbre ou de graphe. Comparer des images de documents reviendra alors à comparer leur contenu (plus sémantique) ou leur mise en page. Si l'on souhaite effectuer une tâche de classification, ceci implique que des documents avec des mises en page similaires seront classés dans la même catégorie.

La première étude sur ce type de méthodes a été menée par Watanabe dans [Watanabe 1999]. Le tableau suivant présente une synthèse de son analyse mettant en relation la nature des dispositions des éléments de contenus au sein des documents, et le type de caractéristiques et de modes de représentations utilisé pour les décrire.

<b>Disposition des éléments de contenus</b>	<b>Exemples</b>	<b>Présentation des descripteurs</b>
<b>Position fixe,</b>	Formulaire, chèques	Listes
<b>Position variée, existence d'une structure <i>a priori</i></b>	Cartes, lettres	Arbres de représentation
<b>Structure rigide sous forme de lignes</b>	Tableaux	Arbre binaire (global et local)
<b>Existence d'une charte graphique mais positions des éléments flexibles</b>	Articles, Journaux	Base des règles
<b>Existence d'éléments de représentation précise mais de mise en page variable</b>	Graphes de présentation	Graphes ou réseaux

Table 27 - Résultat de l'étude de Watanabe dans [Watanabe 1999]

Dans de nombreux cas, une analyse basique mais souvent suffisante de la structure et du contenu peut être réalisée par un procédé de séparation fond/forme obtenu à l'aide d'une binarisation de l'image. Les composantes connexes correspondent alors aux formes (éléments de contenu) tandis que les régions blanches sont supposées traduire l'arrière-plan du document. Une analyse de la taille, de la position, des relations entre les composantes des

formes, produit une information très riche sur la classe du document. Cette information peut être complétée par une analyse de l'arrière-plan, comme proposé dans [Breuel 2004] par extraction des « *Maximal Whitespace Rectangles* ».

Pour classer des images de documents, Shin dans [Shin 2001] propose une approche basée sur les arbres de décision. Les auteurs calculent un grand nombre de descripteurs tels que des statistiques sur la taille, la densité, le périmètre des composantes connexes, le nombre de lignes horizontales, verticales, la présence d'espaces blancs, etc. Un arbre de décision est construit à partir de la vérité terrain extraite d'un échantillon représentatif de la base d'images. Cet arbre est ensuite utilisé pour classer le reste de la base.

Dans [Cesarini 2001], la structure est extraite sous forme d'arbre M-XY qui segmente récursivement le document horizontalement et verticalement. La segmentation horizontale et verticale se fait, soit le long d'espaces blancs, soit le long de lignes séparatrices. Les documents sont alors représentés par des graphes. La valeur d'un nœud non terminal représente la manière dont ses fils sont segmentés tandis que les feuilles peuvent prendre 4 valeurs: HL (*horizontal line*), VL (*vertical line*), T (texte) et I (image). Ces valeurs représentent le contenu des blocs. La structure de chaque document est ensuite encodée dans un vecteur de caractéristiques et les auteurs proposent de classer les images de documents en utilisant des réseaux de neurones.

La littérature sur les méthodes d'analyse de structures et de segmentation d'images de documents abonde mais est hors propos pour ce travail; nous ne développerons donc pas plus ce point ici.

Une fois la mise en page ou les principaux blocs de contenus extraits, leur organisation doit généralement être représentée à l'aide de structures plus évoluées que de simples vecteurs numériques de caractéristiques. Les modélisations sous forme d'arbres ou de graphes de contenus sont alors utilisées et renvoient vers l'utilisation de méthodes structurelles de reconnaissance des formes pour ce qui concerne l'étape de prise de décision.

Gordo propose dans [Gordo 2009], de faire une comparaison de mise en page des documents à l'aide d'un descripteur invariant en rotation exploitant une représentation sous forme de graphe. Ces mêmes auteurs proposent, dans [Gordo 2012], une extension originale de leur travaux en définissant une méthode de classification multi-vues permettant d'exploiter plusieurs vues (descriptions de l'image) durant la phase d'apprentissage et une seule vue (calculer toutes les vues étant trop coûteux) pendant la phase de reconnaissance. Ceci est

possible grâce à un mécanisme de projection des vues dans un espace commun et d'analyse des corrélations canoniques (CCA).

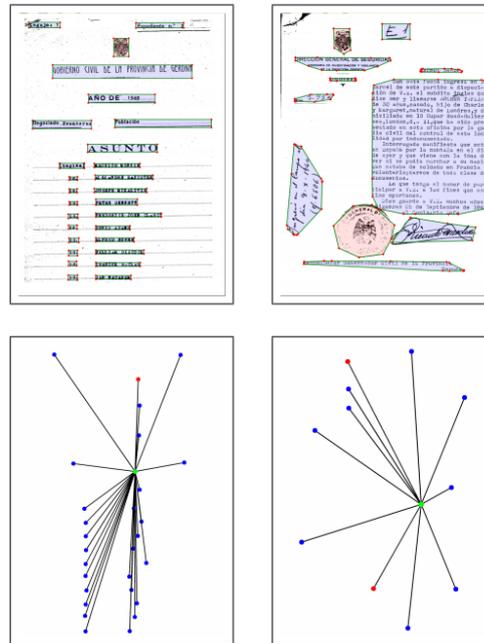


Figure 35 - Descripteur structurel de Gordo dans [Gordo 2009]

Une autre approche pour décrire la structure d'une image du document est basée sur la coloration de graphes [Gaceb 2008]. L'idée est d'utiliser la puissance de la méthode de coloration de graphes pour distinguer les éléments pertinents et les regrouper en ensembles homogènes tout en rejetant les éléments parasites. Le principe consiste donc à distinguer, par coloration des composantes connexes, toutes les zones textuelles et des zones non textuelles, puis de regrouper les composantes connexes des zones textuelles en lignes. Grâce à cela, on peut construire un graphe avec une structure hiérarchique. La structure physique d'un document est présentée par les caractéristiques globales (relatives au document dans son ensemble) et locales (propres à une ligne de texte). Pour comparer la similarité entre deux documents, l'auteur utilise la combinaison de deux distances basées sur la distance euclidienne, (voir Figure 36).

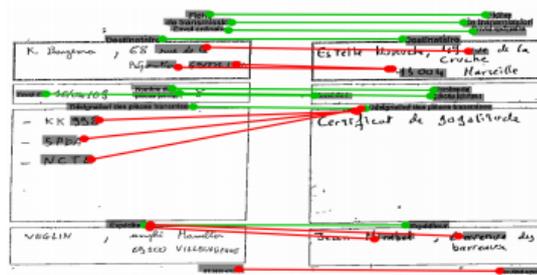


Figure 36 – Comparaison dynamique des documents avec la coloration de graphes. [Gaceb 2008].

Certains auteurs, comme [Hamza 2008], extraient la structure physique et logique des documents en se basant uniquement sur les mots extraits par OCR. Les structures obtenues sont représentées à l'aide de graphes ensuite mis en correspondance par *graph probing*.

Méthode	Résultats	Commentaires
<b>Descripteurs statistiques globaux + Arbre de décision</b>	Base de formulaires de 5590 images répartis en 20 classes  Précision = 99.7% sur les 2000 images de la base de test.	[Shin 2001]
<b>Descripteurs (graphe) de structure invariants en rotation + « Cyclic Dynamic Time Warping »</b>	Base de 823 documents répartis en 8 classes  <i>Leave-one-out cross validation</i>  Précision = 65,9 %	[Gordo 2009]
<b>Graphes de structures construits à partir des sorties OCR + « graph probing »</b>	Base de 324 documents industriels répartis en 8 classes pour l'apprentissage  Meilleur taux de reconnaissance = 99.40 % sur 169 éléments de test	[Hamza 2008]

Table 28 - Quelques indications de performances obtenues par les méthodes citées dans cette section.

#### 4.2.3.3. Approches orientées indexation

##### *Indexation textuelle : Exploitation du texte (sortie d'OCR)*

Inspirés par les techniques de fouille de texte, de nombreux travaux se basent sur l'utilisation du texte reconnu par OCR pour classer des documents. Ceci s'explique principalement par les progrès réalisés ces dernières années par les logiciels d'OCR. Ces travaux sont principalement basés sur l'utilisation de mots-clés permettant l'identification de classes spécifiques (RIB, facture, ...) ou l'utilisation des fréquences d'apparition de mots ou de groupes de lettres (*n-gram*) tels que « *tf-idf* » [Sebastiani 2002], [Paradis 2007].

Les techniques de sacs de mots (*Bags of Words* - BoW) supposent que le type d'un document peut être défini à partir des mots qu'il contient. Pour mettre en place cette technique, il est tout d'abord nécessaire de constituer un dictionnaire qui représente le vocabulaire à prendre en compte pour la caractérisation. Ce dictionnaire peut être constitué à partir de tous les mots présents dans l'ensemble des documents à traiter en excluant tout de même certains mots n'apportant pas de sens (mots vides ou *stop words*). L'exclusion des mots non pertinents permet également de s'affranchir des problèmes liés aux mots mal reconnus par l'OCR. Une autre étape permettant de réduire le dictionnaire est la lemmatisation dont l'objectif est de remplacer des mots d'une même famille par une racine commune.

Une fois le dictionnaire créé (souvent par *K-means clustering* [Leung 2001]), chaque document est décrit par un histogramme représentant le nombre d'occurrences de chacun des mots du dictionnaire présents dans le document. La comparaison entre deux documents se fait par la comparaison de leur histogramme respectif. Les vecteurs sont généralement de très grandes dimensions selon le nombre de mots dans le dictionnaire et sont comparés à l'aide d'une mesure de similarité construite à partir du cosinus [Sebastiani 2002].

Des mécanismes de pondération permettant de prendre en compte les caractéristiques de chaque classe, ou encore l'usage de tables de hachage permet d'optimiser ce genre de procédés.

### ***Couplage d'informations textuelles et visuelles par approche BoW***

La technique *BoW* a été adaptée pour être applicable à la recherche d'images par l'exemple. Cette déclinaison, appelée « *Bag of Features* » (*BoF*) ou « *Bag of Visual Words* » (*BoVW*), est directement inspirée du principe des sacs de mots. Le principe repose sur l'utilisation de « mots visuels » correspondant en fait à des parties (morceaux) d'images généralement sélectionnées et décrites à l'aide des techniques de sélection de points d'intérêt comme SIFT (*Scale-Invariant Feature Transform*) [Lowe 1999] ou SURF (*Speeded Up Robust Features*) [Bay 2008].

L'algorithme des *BoF* peut être résumé en 4 principales étapes :

- Extraction de points d'intérêts sur l'ensemble des images que l'on souhaite classifier.
- Partitionnement en  $k$  groupes de ces points d'intérêt. Ces  $k$  groupes définissent le vocabulaire c.-à-d. les  $k$  dimensions de l'histogramme qui servira à décrire chaque image.

- Pour caractériser une image, chacun de ses points d'intérêt est classé/affecté dans l'un des groupes précédemment définis afin d'incrémenter l'un des pics de l'histogramme représentant l'image.
- Les images sont comparées en comparant leur histogramme respectif.

Habituellement, le partitionnement des descripteurs (points d'intérêt) est effectué avec l'algorithme *k-means* [Csurka 2004], [Leung 2001], [Winn 2005], [Weber 2000] et l'affectation d'un point d'intérêt à un groupe se fait par SVM [Csurka 2004], [Zhang 2007], [Marszalek 2006].

On peut utiliser aussi *Naive Bayes* [Csurka 2004], *pLSA (Probabilistic latent semantic analysis)* [Russell 2005], ou *Adaboost* [Serre 2005].

Dans [Laroum 2009], les auteurs classent des documents en fonction de leur contenu textuel. Les techniques de sacs de mots sont testées sur une base de documents provenant de la société ITESOFT. La base est constituée de 2000 fichiers répartis dans 24 catégories. Le texte provient du résultat de logiciels d'OCR. Les documents présents dans la base sont de type : bulletins de salaires, certificats, avis d'impôt, etc. La base est caractérisée par la variété des documents, leur faible contenu textuel (130 mots en moyenne par document) et la répartition très hétérogène des différentes classes. L'utilisation des sacs de mots avec pondération *tf-idf* et apprentissage par SVM permet d'obtenir de bonne performance (précision moyenne de 95,8 %).

Bien que certains travaux récents dans [Rusinol 2012], [Augereau 2014] relativisent l'intérêt du couplage *BoW + BoVW* pour la recherche d'images de documents, l'usage des *BoVW* générant une chute des performances comme montré dans Table 29, certaines expérimentations, proposant l'exploitation de méthodes de fusion plus poussées, démontrent tout de même qu'un gain peut être obtenu lorsque les résultats OCR sont de moins bonne qualité [Augereau 2014].

	Rappel (%)	Précision (%)
<b>BoVW</b>	0.871	0.869
<b>BoW</b>	0.982	0.977
<b><i>Borda Fusion</i></b>	0.899	0.902

Table 29 - Méthodes combinant BoW et BoVW par fusion simple fournie dans [Augereau 2014]. La base de tests utilisée, fournie par la société GestForm, est composée de 1985 documents répartis en 12 classes. 5 documents servent à l'apprentissage, le reste au test en validation croisée.

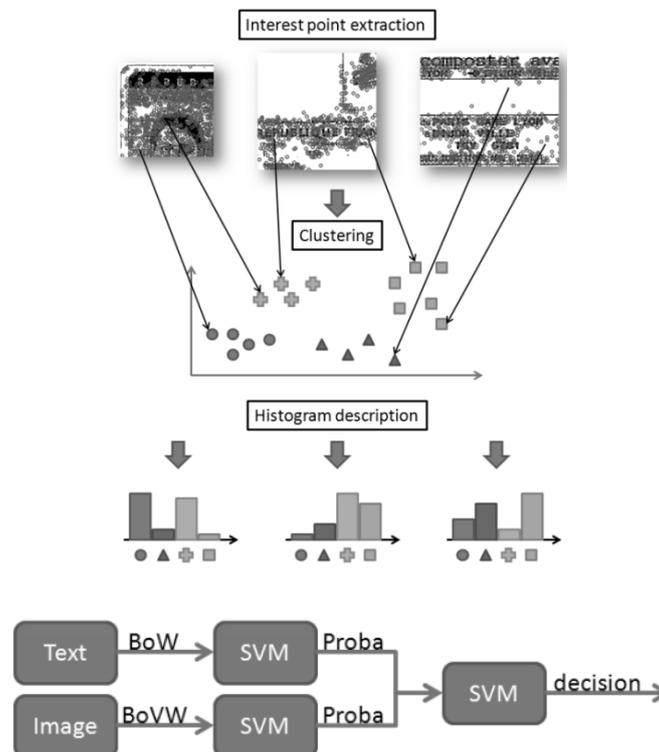


Figure 37 - Méthode combinant BoW et BoVW proposée par Augereau [Augereau 2014]

Les travaux de recherche d'Augereau [Augereau 2013] ont permis de fournir différentes solutions de classification d'images de documents en fonction des connaissances relatives aux images à traiter. Le couplage des techniques *BoW* et *BoVW* a permis la proposition de différentes contributions telles que :

- la mise en place d'une méthode permettant à un utilisateur de classer, de manière interactive, des images de documents dont la nature est inconnue,
- la proposition d'une technique de recherche d'images de documents par l'exemple, basée sur l'extraction et la mise en correspondance de points d'intérêts,
- l'élaboration d'une méthode de classification non supervisée d'images de documents.

Dans ces travaux, la méthode SURF et l'algorithme *k-means* (avec  $k = 1000$ ) sont utilisés pour l'extraction des points d'intérêt. Un SVM multi-classes (RBF) est utilisé pour la classification (voir Figure 37).

Plutôt que d'utiliser des points d'intérêt SIFT ou SURF, [Sarkar 2006] propose de rechercher des motifs de texture particuliers (voir Figure 38) pour caractériser les images de documents en s'inspirant de la méthode proposée par Viola dans [Viola 2001] pour classifier ou rechercher des objets dans différents types d'images naturelles.

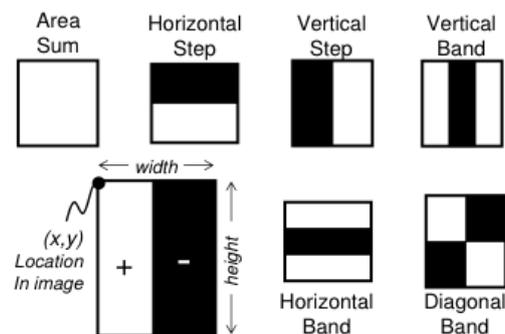


Figure 38 - Motifs de texture exploités dans [Sarkar 2006]

#### 4.2.4. Traitement de flux de documents et limites actuelles

Les documents qui ont une forte variabilité intra-classe ou une faible variabilité interclasse rendent parfois difficile la différenciation de certains documents. Le couplage d'informations de différentes natures (perceptuelle, structurelle, textuelle, ...) est alors important. Les temps de calcul peuvent cependant devenir prohibitifs, nécessitant de travailler à la mise au point de nouvelles approches multi-vues (telles que celles proposées par [Gordo12]) ou des mécanismes d'adaptation dynamique de l'espace de représentation pour chaque classe. L'utilisation de techniques d'apprentissage incrémental permet ainsi de prendre en compte ces diversités.

Afin d'obtenir un bon taux de reconnaissance, les méthodes usuelles de classification de documents doivent être entraînées en utilisant de nombreux documents étiquetés.

Beaucoup d'entre elles sont donc entièrement supervisées et coûteuses étant donné qu'elles nécessitent l'étiquetage manuel d'un grand nombre de documents.

Les méthodes existantes de classification de documents ont besoin de connaître à l'avance l'ensemble des documents utilisés pour l'apprentissage. Ceci n'est pas possible dans le cas d'un flux où les documents arrivent en continu et deviennent disponibles au fil du temps (flux infini). Dans ce cas, l'apprentissage doit se faire au fur et à mesure que les documents arrivent : chaque nouveau document est susceptible d'être utilisé immédiatement et une seule fois, pour mettre à jour le modèle de façon incrémentale. La plupart des méthodes de classification incrémentale pour les flux de données supposent généralement que le nombre de classes de documents dans le flux est connu. Cependant, dans les applications réelles, cette supposition est souvent incorrecte. En effet, il est difficile d'obtenir des échantillons de documents étiquetés à partir de toutes les classes de documents possibles qui apparaîtront dans le futur. De plus, la nature évolutive du flux fait que de nouvelles classes de documents peuvent apparaître à tout moment. Si une nouvelle classe de document n'est pas automatiquement détectée, toutes les instances de documents de cette classe seront inévitablement classées par erreur dans des classes connues, ou rejetées.

L'identification de nouvelles classes est donc importante pour éviter de telles erreurs de classification. L'usage de mécanismes interactifs nous semble alors être indispensable.

#### 4.2.5. Introduction de l'apprentissage continu pour la classification de document

Cette partie intervient à l'intersection entre, d'une part, le domaine de l'*apprentissage continu bien décrit dans la littérature* (et rapporté dans le chapitre 2 de cette thèse) et, d'autre part, *la classification de document en flux*, qui correspond à la problématique soulevée par ces travaux. Toutes les méthodes de la classification statique de documents ne sont pas mentionnées dans cette section.

Au sein des méthodes de classification de document en flux, la plupart des articles font référence à des classes de documents bien identifiées (avec la présence de connaissances *a priori* en nombre suffisant). Ces approches sont fondées sur des mécanismes de classification (en ligne ou hors ligne) non-incrémentaux, c'est-à-dire sans avoir besoin de créer une adaptation à la volée. On peut citer ici quelques méthodes dynamiques appliquées à la classification de documents: Bianchi dans [Bianchi 2006] utilise un classificateur basé sur un modèle d'arbre de décision pour la classification du texte et Song dans [Song 2008] utilise un

classificateur SVM pour réaliser une classification incrémentale avec un concept stable et **sans** ajout de classe (les notions citées ici sont présentées au chapitre 2 aux sections 2.2.1.1.1 et 2.2.1.1.2). On peut également citer les travaux de Chen dans [Chen 2007] qui utilise les réseaux de neurones incrémentaux capables d'ajouter au fil du temps de nouveaux types de documents dont les modèles restent stables dans le temps (concepts stables). La structure de ce classificateur est similaire à celle d'ARTMAP présenté dans la section 2.2.2.1.2. Ces méthodes sont essentiellement dédiées à la classification en *environnement stationnaire*. Seul Natee dans [Natee 2006] utilise la méthode Winnow (présenté dans la section 2.2.1.2.3) permettant de traiter des données en environnement non-stationnaire mais **sans** avoir la capacité d'ajout de concepts. Bouguelia dans [Bouguelia 2013] propose un mécanisme capable de détecter une nouvelle classe automatiquement en utilisant un simple classificateur K-NN et une distance de rejet (distance Euclidienne) pour détecter une nouvelle classe de documents. Cette méthode n'est cependant pas une approche incrémentale car elle doit utiliser l'ensemble des anciennes données pour calculer la distance d'un exemple aux autres données déjà classifiées.

Pour les autres catégories d'évolution en matière de classification de document, il n'existe aucune solution à ce jour capable de réaliser un *ajout de concept en environnement non-stationnaire*, une *fusion* ou une *division de concepts* alors que ces facultés sont de notre point de vue *les fonctions de base requises par le scanner cognitif* que nous cherchons à concevoir.

### 4.3. Rappel des objectifs du projet DIGIDOC

Le scanner cognitif devra permettre la définition de protocoles de numérisation adaptés au mieux à la numérisation d'un type de contenu pour un usage donné. Il peut donc y avoir plusieurs protocoles pour un même usage, en fonction du contenu du document. Par la suite, étant donné ces protocoles et les usages correspondants, le système devra être capable de sélectionner automatiquement, pour un nouveau document et en fonction d'un usage choisi, le protocole de numérisation adéquat. Pour cela, il faudra être capable de caractériser le contenu d'un document par un ensemble de descripteurs plus ou moins bas niveau et de lier ces descripteurs aux protocoles de numérisation existants en fonction de l'usage. Pour ce faire, l'utilisation de processus d'apprentissage et de classification incrémentale a été envisagée. La difficulté vient alors du fait que ces mécanismes devront être capables de supporter une mise à jour incrémentale et interactive des connaissances sur lesquelles ils se

fondent. La mise en place de cette acquisition incrémentale de connaissances (par exemple mise à jour continue des modèles et de la ou des bases d'apprentissage des classificateurs) est un véritable verrou scientifique. En effet, elle nécessite d'être capable, au fil de l'eau, d'évaluer la pertinence d'une décision ou d'un exemple d'apprentissage, de décider ou non de créer une nouvelle classe (lien descripteur-protocole de numérisation) et, le cas échéant, de produire les données à associer à cette nouvelle classe à partir d'une quantité très faible d'informations apportées par les algorithmes d'analyse d'images bas niveau. Une solution envisageable et pertinente pour contrecarrer ce problème nous semble être d'accepter de faire appel à l'utilisateur lorsque l'incertitude est trop importante. C'est pourquoi nous parlons également de classification interactive. Un système de classification avec rejet répondant aux objectifs ci-dessus doit donc être proposé : le classificateur doit être capable de déterminer un seuil de rejet de manière automatique pour déterminer si une page ou un fragment de page scanné correspond à l'une des classes déjà connues ou bien s'il est nécessaire de définir une nouvelle classe et de construire le protocole de numérisation associé. Dans ce dernier cas, la définition du nouveau protocole se fera de façon interactive avec l'aide de l'opérateur.

#### 4.3.1. Modes de fonctionnement prévus du système

La problématique concerne l'introduction d'une « intelligence » dans les dispositifs de numérisation. Sur la base de descripteurs d'images, l'objectif est de concevoir des méthodes permettant l'auto-adaptation du processus de numérisation (par définition d'un protocole de numérisation) en fonction du contenu de chaque page et des objectifs de l'opérateur de numérisation (usager). Le schéma général de fonctionnement prévu pour ce système est le suivant :

- Demande des objectifs à l'opérateur (se ramenant à la sélection d'un cas d'usage prédéfini (profil/cas d'usage))
  1. Numérisation de l'image en fonction du protocole choisi
  2. Extraction des descripteurs de contenu d'images nécessaires (selon le profil choisi)
  3. Soumission des descripteurs au module cognitif (module de décision)
    - a. Prise de décision (via des mécanismes d'apprentissage et classification)
    - b. Si les descripteurs correspondent à une classe existante et si le protocole est déjà défini, alors la sélection du protocole adéquat se fait automatiquement

- c. Si les descripteurs ne correspondent pas à une classe existante, il est nécessaire de faire intervenir l'utilisateur (*Interaction Homme-Machine*) pour définir le nouveau protocole de numérisation (situation de *rejet*)
  - d. La fin du processus est atteinte lorsque l'image obtenue est satisfaisante. Dans ce cas, il y a une mémorisation ou une mise à jour éventuelle du protocole (si celui-ci vient d'être créé ou juste modifié)) utilisé dans la base de connaissances.
4. Rebouclage en 3) sauf cas c)

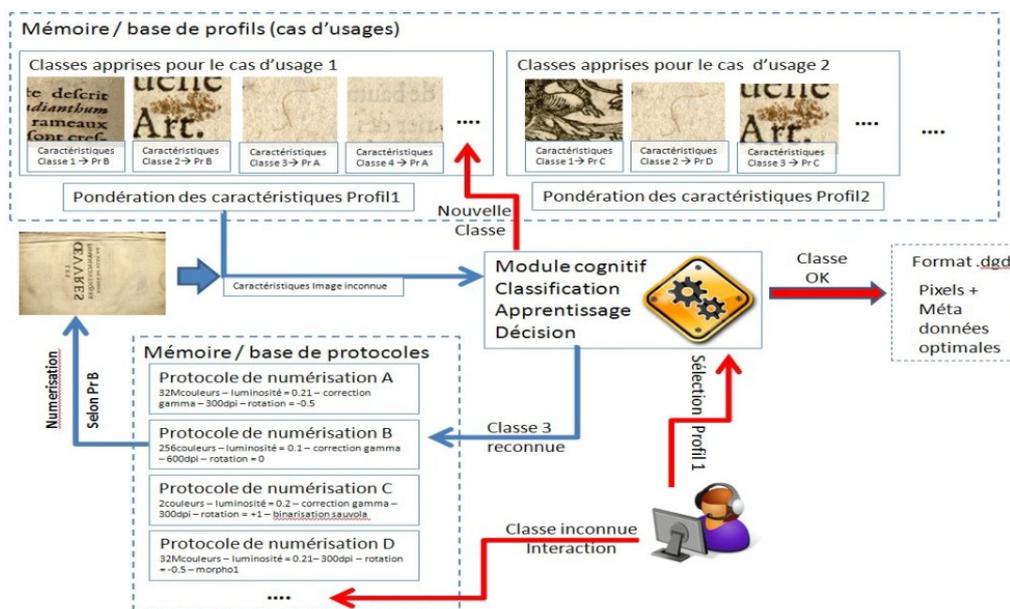


Figure 39 - Schéma de principe du système cognitif

Pour les images rejetées par la classification, un opérateur aura donc la charge d'interagir avec le système pour définir un protocole de numérisation dédié à ce nouveau type d'images. L'originalité de notre proposition réside aussi dans la mise au point de nouveaux modes d'interaction. Il devrait être possible de fournir à l'opérateur de nouveaux indices pour l'aider à juger de la qualité de la numérisation en fonction des valeurs prises par les descripteurs extraits sur les images et des usages envisagés. Si, après plusieurs boucles entre l'utilisateur et le système, le protocole échoue, l'image sera alors classifiée comme étant une image ne pouvant pas être corrigée par le scanner. L'image sera alors automatiquement identifiée comme potentiellement à retoucher.

#### 4.3.2. Spécifications et cas d'usages retenus

Dans une étude préalable, nous nous sommes attachés à mettre en évidence les différentes catégories d'images que le scanner pourra être capable d'identifier. À chacune de ces classes pourra correspondre un ou plusieurs protocoles de numérisation (jeu de paramètres) en fonction de l'usage choisi. Nous les avons recensés de façon assez large dans un premier temps, sachant que toutes ces catégories ne pourront pas toutes être intégrées dès le départ dans le scanner. Les connaissances embarquées seront amenées à évoluer au cours de l'apprentissage incrémental. Nous présentons également les classes sur lesquelles nous insisterons en priorité afin de pouvoir constituer les bases d'images qui serviront à l'apprentissage et à l'évaluation de la méthode proposée (mOC-iSVM). Ces choix sont donc dépendants des images à disposition.

#### 4.3.2.1. Cas d'usage et fonctionnalités d'un scanner cognitif

Dans une campagne de numérisation où la qualité est prépondérante, la majeure partie des coûts de production sont des coûts de main-d'œuvre, principalement liés au temps de réglage ou de re-réglage du scanner. Il est même parfois nécessaire de recommencer la phase d'acquisition lorsque les images produites ne peuvent être utilisées par les autres maillons de la chaîne. C'est le cas par exemple lorsque le taux de reconnaissance de l'OCR est trop faible. Il est donc primordial d'essayer d'offrir un processus de numérisation permettant aux scanners de s'auto-adapter aux données, de détecter et éventuellement corriger les défauts de prises de vues.

Les collections de documents à numériser se caractérisent par une forte hétérogénéité tant au niveau de leurs contenus (langue, thème,...), que des techniques d'imprimerie utilisées (insertion d'illustrations et de texte, technique de mise en page, qualité des techniques d'imprimerie...), que de leur qualité physique après plusieurs siècles d'archivage (apparition de taches d'encre, détérioration du papier).

Il nous a donc paru judicieux de faire apparaître deux types de cas d'usages des scanners :

- Les cas d'usage utilisateurs
- Les cas d'usage fonctionnels

Les deux sections suivantes listent, pour chacun de ces deux cas, les principaux usages qui ont été prévus pour ces scanners cognitifs.

##### 4.3.2.1.1. Cas d'usage Utilisateurs

Il s'agit ici de lister les objectifs que peuvent avoir les utilisateurs lorsqu'ils exploitent un scanner pour numériser des documents, l'objectif étant de permettre à terme le choix des paramètres de numérisation optimaux vis-à-vis de cet objectif.

***Usage 1 : Préservation, archivage***

Il peut s'agir simplement de garder une trace numérique d'un document permettant plus tard d'accéder au message qu'il véhiculait ou au contraire d'en faire une copie fidèle dans tous ses détails.

***Usage 2 : Reproduction, diffusion, impression, web, DVD***

Dans ce cas, il s'agit bien souvent de trouver le meilleur compromis possible entre fidélité à l'original et mémoire de stockage ou débit nécessaire.

***Usage 3 : Enrichissement du document : Transcription, OCR sur texte + indexation des images, GED***

L'objectif est ici différent puisque la fidélité à l'original n'est plus forcément requise. L'objectif est plutôt d'améliorer la qualité de l'image, d'éliminer certains défauts afin que l'image produite soit la plus adaptée possible aux traitements qui vont suivre.

***Usage 4 : Recherche : épigraphie, paléographe, philologie, codicologie, historien, ...***

Il s'agit là de faire des acquisitions très précises ou voire même avec des paramétrages très spécifiques nécessaires aux traitements et analyses très spécifiques qui vont suivre. L'analyse de ces cas d'usages amène à la conclusion qu'il s'agit bien souvent de trouver le meilleur compromis entre fidélité à l'original et espace mémoire ou temps de calcul associé.

Il est aussi possible que l'utilisateur veuille à la fois archiver et enrichir les documents (combinaison de cas d'usage). Il semble donc inopportun de baser la conception et le fonctionnement futur des scanners sur le choix *a priori* d'un de ces cas d'usage lors du lancement d'une session de numérisation.

4.3.2.1.2. Cas d'usage fonctionnels

Pour concevoir le scanner cognitif, il nous semble préférable de se baser sur la liste des cas d'usage fonctionnels suivants :

***Usage 1 : usage de base***

Bien évidemment le principal usage d'un scanner concerne la production d'images (pixels, TIFF) de qualité adaptée aux besoins. Les scanners cognitifs produiront bien évidemment, comme leurs prédécesseurs, des fichiers images.

Cependant, en plus des images, les scanners cognitifs produiront un fichier au format DIGIDOC correspondant à une session complète de numérisation d'un document ou ouvrage (et pouvant donc correspondre à plusieurs pages)

***Usage 2 : Auto-calibration ou calibration assistée (réglages des paramètres)***

Dans certains cas, avant la phase de numérisation, il est nécessaire de régler le scanner selon les spécifications fournies par le client. Le scanner cognitif devra fournir une aide à ces réglages :

- Respect des niveaux de gris, balance des blancs, précision colorimétrie et résolution, profondeur de champ, focale (flou), défaut d'éclairage, ...
- Sélection des prétraitements (correctifs) à appliquer.

***Usage 3 : Détection des défauts (classes de défauts)***

À l'aide de son module d'apprentissage, le scanner cognitif sera capable d'apprendre et donc de détecter certains défauts de numérisation, et ce, à 3 niveaux différents :

- Niveau document : Page manquante, pliée, dérive progressive...
- Niveau image : Prise de vue incorrecte : orientation, structure, cadrage, bords, ...
- Niveau Élément de Contenu (EoC) : résolution, contraste, couleur, luminosité, bruits, flou,...

***Usage 4 : Détection de types de contenus (documents, pages, régions : classes d'EoC)***

Afin d'utiliser les bons paramètres d'acquisition, le scanner sera capable d'apprendre et ensuite de reconnaître certains types de contenus spécifiques. Le protocole correspondant au type reconnu sera alors appliqué pour obtenir une image de qualité optimale. Les types de contenus (EoC) qui pourront être appris seront par exemple :

- Texte imprimé
- Texte manuscrit
- Dessins, plans, ...
- Tableaux, lignes, figures
- Photos, illustrations
- Pages vides
- ...

L'opérateur pourra définir lui-même les types de contenus à reconnaître.

#### 4.3.2.2. Classes d'images retenues pour la réalisation du prototype

Dans la **classification de document**, un document peut être vu comme une **organisation d'objets** (de symboles textuels et graphiques de toutes sortes) ayant une disposition aléatoire ou structurée de symboles textuels et graphiques de toutes sortes. La **classification des documents** consiste, en général, à regrouper divers documents **en sous-ensembles homogènes** sur la base des représentations descriptives de leurs structures physiques ou de leurs contenus textuels de telle sorte que les observations dans les mêmes classes de documents soient le plus similaires possible, et que les observations dans des classes différentes soient le plus dissemblables possible. Bagdanov dans [Bagdanov 2001] propose de distinguer les classes selon deux types de granularités : séparation grossière (coarse-grained) comme elle peut s'observer entre une carte et un journal par exemple et granularité fine, comme celle que l'on rencontre entre deux documents imprimés selon des indicateurs qui relèvent d'une analyse plus fine des contenus. Nagy dans [Nagy 2000] divise les types de documents dans 2 catégories : catégorie décrivant les contenus essentiellement textuels (manuscrits ou imprimés) et la catégorie décrivant les documents essentiellement graphiques (*Engineering drawings, diagrams, sheet music, maps...*). Dans la catégorie des documents imprimés, par exemple, Nagy propose une division entre des contenus très structurés (journaux, magazines..) et des documents de structure pauvre, texte brut. Doermann dans [Doermann 1998] divise les documents selon leur structure fonctionnelle reposant sur la présence de titres, d'entêtes et pieds de pages, de listes, de tableaux, et de graphiques. Afin de proposer de travailler sur des classes d'images en cohérence avec nos objectifs, nous pouvons partir d'une distinction entre **trois points de vue** autour de la classification de documents : **la classification de contenus** au niveau de la page vue comme une unité de contenu, **la classification de textes** portant sur les données textuelles (et non visuelles et graphiques) et enfin **la classification de structures** rendant compte des variétés de mise en page et s'intéressant à l'organisation de l'information (répartition des éléments de contenus sur la surface du document).

Pour la classification au niveau page, on désigne généralement le type de document en fonction de son contenu principal (page de texte imprimée, graphique, partition musicale...). Les descripteurs de très bas niveau sont généralement suffisants pour permettre de telles séparations. La classification des données textuelles porte sur des descripteurs de textes en lien avec des informations linguistiques et des dictionnaires permettant d'introduire des considérations en lien avec le sens dégagé des contenus. La classification de structures repose

sur la l'agencement de l'information sur l'espace de la page. On parle de *physical layout*. L'état de l'art du domaine foisonne de propositions permettant de produire une description structurée des contenus. Afin de déterminer les fonctionnalités du scanner intelligent, nous avons essentiellement concentré nos efforts dans les premières expérimentations sur la prise en compte de données facilitant la **classification de contenus au niveau page**.

Pour la mise en place du scanner cognitif, nous allons nous intéresser à la fois à des documents issus de classes de **contenus** différentes et des documents de classes de **qualité** variable. À partir de catégories existantes, en adaptant avec nos objectifs, le scanner va donc devoir être capable de classer les documents selon ces deux modalités ;

- Domaine A : Les classes relatives à la nature du document. Ces classes peuvent servir pour détecter le type de contenus d'un document ou encore la nature du document en lui-même et donc faire correspondre à un type un jeu de paramètre.

- Domaine B : Les classes relatives à la qualité de document. Ces classes peuvent servir directement pour l'auto-calibration ou calibration assistée ainsi que pour la détection des défauts que le module cognitif doit être capable de faire.

À partir de cette structure des classes d'images, nous avons choisi différentes classes principales que le scanner devrait pouvoir identifier. Différents critères entrent en jeu : ils sont, d'une part, en lien avec le niveau de détails de contenu sur lequel l'analyse portera, mais portent également sur les besoins du projet et les corpus accessibles.

Dans le cadre d'une scénarisation du scanner cognitif, nous avons, pour chacun des domaines A et B, choisi de proposer la classification suivante en 7 classes de contenus (domaine A) et en 5 classes de qualité (domaine B).

- Classes de contenus basiques (7 types):
  - **Texte manuscrit** : images de traits contenant du texte manuscrit exclusivement.
  - **Texte imprimé** : images de traits contenant du texte imprimé (de typographie variable selon les périodes d'impression).
  - **Graphique** : images de traits contenant des symboles graphiques
  - **Photos** : images naturelles en variation de niveaux de gris ou en couleurs.
  - **BD** : images combinées de textes et de dessins.
  - **Partitions musicales** : images spécifiques à la musique.
  - **Complexe** : images composites aux propriétés multiples.

- Classes de qualité basique (5 types) :
- **Couleur** : document couleur
- **Niveau de gris** : document en niveau de gris.
- **Noir/blanc** : document noir/blanc.
- **Mauvaise qualité** : pliure, bruit,...
- **Mauvaise acquisition** : flou, homogénéité de lumière, mauvais cadrage...

Ces classes ne sont pas exclusives : une même image de document peut correspondre à plusieurs des classes ci-dessus. C'est en fonction de celles-ci que des premiers jeux de paramètres seront déterminés (et en fonction de l'usage).

Lors de l'utilisation du scanner, de nouvelles catégories pourront être proposées (automatiquement par le scanner ou bien par l'utilisateur) avec le jeu de paramètres de numérisation correspondant (déterminé également automatiquement ou non).

#### 4.3.3. Interfaces et implémentation du prototype

Afin d'offrir les fonctionnalités listées dans les sections précédentes, les IHM suivantes ont été définies pour le pilotage du scanner :

- IHM principale (Figure 40) : elle permet de visualiser l'état du scanner, le document en cours de numérisation et les résultats de classification proposés par le module cognitif. L'utilisateur peut à tout moment intervenir pour corriger les propositions de classification émises par le scanner. Cette IHM donne également accès à d'autres IHM permettant notamment la définition de nouveaux protocoles de numérisation (paramétrages du scanner) et l'association de ces paramétrages à différentes classes de documents pouvant également être définies et gérées par l'opérateur.

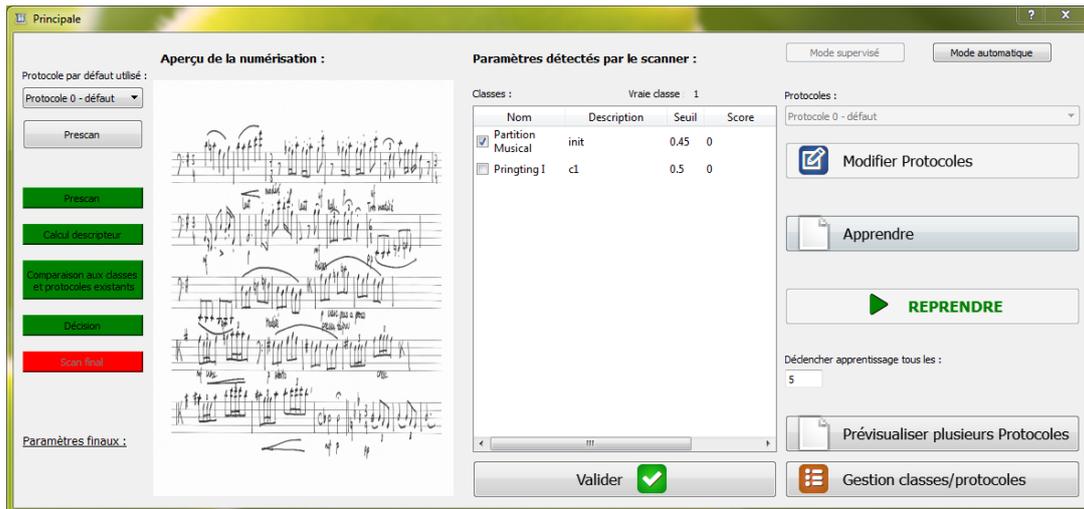


Figure 40 - IHM principale du scanner

- IHM de gestion des protocoles de numérisation (Figure 41) : Elle permet à l'opérateur de définir les réglages du scanner pour une classe de documents donnée (résolution, corrections, pré-traitement, luminosité, éclairage, ...). Cette IHM dépend du type de scanner et du SDK de communication avec le dispositif matériel qui sera fourni par I2S.
- IHM de gestion des classes de documents (Figure 42) : Cette interface permet de définir ou de supprimer des classes de documents qui seront reconnues automatiquement par le classificateur : il s'agit donc d'ajouter ou de supprimer des exemples d'apprentissage dans les bases de connaissances associées à chacune des classes à reconnaître par la suite.

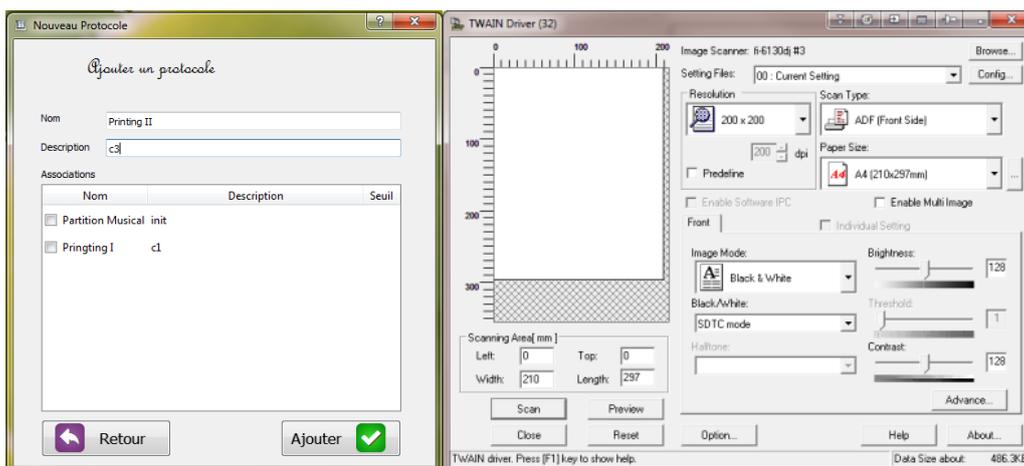


Figure 41 - IHM de définition de protocoles



Figure 42 - IHM de définition de classes

## 4.4. Exploitation des mOC-ISVM pour la numérisation cognitive

### 4.4.1. Sélection des descripteurs d'images

Dans cette partie, notre objectif est de déterminer un jeu de descripteurs qui permettra de caractériser les classes d'images que nous avons sélectionnées pour alimenter et évaluer notre scanner cognitif. Ce choix s'effectue selon plusieurs critères. Le premier est d'avoir des descripteurs simples à calculer pour des raisons d'embarquement dans le scanner. Ensuite, ils doivent être d'un assez bas niveau pour ne pas être trop dépendants du type de documents et représenter la variabilité permettant de distinguer les classes sélectionnées et à venir.

Ces descripteurs pourront évoluer au cours du projet ou dans le futur. Nous avons choisi de partir d'une démarche de caractérisation sans segmentation, et sans *a priori* sur le contenu de l'image afin de permettre au système de déterminer automatiquement la présence d'informations relatives aux classes de contenus proposées dans la section précédente (classes graphiques, photo, BD, complexe, texte imprimé et manuscrit...). La notion de caractérisation portera sur une description multi échelle qui nous semble être une voie intéressante en procédant à une exploration par zones ne présumant d'aucune information relative à la mise en page et à la présence de telle ou telle information de contenu spécifique (graphique, texte ou image).

D'une façon générale, pour caractériser l'image de façon globale et locale, on a besoin de disposer d'une liste de descripteurs de contenus qui peuvent avoir été choisis en fonction de l'objectif de la classification et de certains présupposés même très grossiers sur la nature des images (présence d'informations de couleur, de texture, de contours, de traits simples

binaires, et/ou encore de points d'intérêt...) ainsi que du type de région de l'image sur laquelle porteront les calculs (simple pavage multi échelle de l'image, zones d'intérêt extraites automatiquement, régions prédéfinies...). En fonction de ces données de départ, il est possible de produire une description qui pourra revêtir diverses formes : une description scalaire contenant une valeur unique pour une zone ou l'image entière (il s'agit généralement de valeurs moyennes, ou de valeurs d'écart à la moyenne qui permettent de rendre compte d'une information non uniformément distribuée : comme une densité, ou une entropie...), une description multi-valorée de type vecteur de caractéristiques contenant une description plus complète des contenus, calculée sur des représentations de l'image différenciées: image de texture, couleurs ou niveaux de gris, image binarisée, image de contours, image améliorée...), ou enfin une représentation cumulée de valeurs de type « histogramme de distributions » (couleurs, niveaux de gris, ou valeurs cumulées calculées sur une région) ou une signature de l'image produite comme une matrice d'accumulation (comme les descripteurs de formes de type histogrammes log-polaires *Shape context* de Belongie, ou encore les signatures cumulant les valeurs calculées en tous les points de la région comme avec les descripteurs d'Haralick, les coefficients maximaux de Curvelets...).

Notons aussi que nous avons exclu l'usage de mots-clés obtenus par OCR ou de structures complexes (graphes) comme descripteurs puisque le module de classification a vocation à être embarqué dans un scanner et à fonctionner en temps réel. Seuls des descripteurs d'assez bas niveau peuvent donc être utilisés.

Nous avons donc choisi de définir plusieurs stratégies de caractérisation complémentaires : portant soit sur l'image dans sa globalité, soit sur des régions d'extraction simple (type partitionnement régulier ou pavage multi échelle) et enfin sur plusieurs niveaux de résolutions possibles. Cette façon de considérer l'image selon plusieurs niveaux d'échelles, et plusieurs zones de focus, nous semble bien convenir à une caractérisation visant à définir des classes de contenus différenciées: textes, dessins, photos, traits ainsi que des catégories plus hétérogènes et complexes en liant plusieurs. Sans chercher à produire une segmentation des images, nous visons à produire une description permettant de rendre compte de la présence d'un élément de contenu sans conduire à sa localisation fine.

Les étapes d'extraction des caractéristiques retenues sont les suivantes :

### ***Étape 1 : Ré-échantillonnage et découpage de l'image en zones***

Pour l'extraction de régions dans l'image, nous avons choisi de traiter massivement l'information sans la segmenter et d'y appliquer des transformations globales, soit en

procédant à une série d'opérations de divisions contrôlées par des règles de décomposition régulière, soit en procédant à une sélection de régions à plusieurs niveaux d'échelles permettant de traiter itérativement l'information de contenus :

- En sous-régions : il s'agit de diviser l'image en régions séparées avant d'appliquer une méthode pour chacune d'elle.
- En régions multi-échelles : il s'agit de sélectionner différentes versions d'une image avec des niveaux d'échelles variables (niveaux caractères, niveaux blocs, niveaux fragments/paragraphes...).

***Étape 2 : Une fois les régions déterminées (globales, partielles, multi échelles), il est alors possible de procéder à différentes transformations de l'image facilitant les calculs de descripteurs.***

- Les transformations d'espaces couleur : RVB, TSL, HSL, HSV,  $L^*u^*v$  Les transformations des niveaux de gris : conduisant à une modification de l'intensité des niveaux de gris
- Les opérations de binarisation conduisant à une segmentation bitonale de l'image. Les changements d'espaces de représentations : transformées de Hough, fonction de Haar, domaines fréquentiels (et les transformées usuelles : ondelettes, multi-échelles, anisotropes...)
- Les détecteurs de contours (Roberts, Sobel, Canny-Deriche, Laplacien de Gaussiens...) ou les squelettisations (discrètes et fondées sur la morphologie mathématique, ou plus continues fondées sur des opérations de régularisation du gradient par diffusion ...)
- La segmentation en blocs : analyse des connexités, extractions de régions d'intérêts multi échelles (par Différences de Gaussiennes), découpage en quad-tree...

***Étape 3 : Calcul des descripteurs de forme***

- Par histogramme : représenter l'information par une distribution des classes de valeurs consécutives
- Par valeur scalaire : représenter l'information par une valeur scalaire unique pour ce descripteur et pour chacune des zones considérées

- Liste de représentation : représenter par une dizaine de valeurs pour chaque zone ou pour chaque descripteur.

Les descripteurs retenus doivent pouvoir être extraits sur une image de document ou sur une partie de celui-ci. Ils doivent également être capables de caractériser grossièrement la complexité et l'homogénéité (structure) de l'image.

Les sections suivantes décrivent les descripteurs finalement retenus.

#### 4.4.1.1. Liste exhaustive des descripteurs retenus

Les descripteurs retenus doivent pouvoir être extraits sur une image de document ou sur une partie de celui-ci. Ils doivent également être capables de caractériser grossièrement la complexité et l'homogénéité (structure) de l'image.

- ***Couleurs utilisées dans l'encre***

Ce descripteur de caractérisation colorimétrique a été choisi pour décrire principalement les couleurs de l'encre utilisée dans l'image. D'abord, nous séparons des pixels de fond et pixels d'élément. Après, nous divisons les valeurs de R, G, B des pixels d'élément en 8 mixte intervalles de couleurs et calculons le pourcentage de pixels dans chaque mixte intervalle. Contribution du descripteur à la caractérisation du contenu : ce descripteur est un indicateur intéressant des couleurs utilisées dans l'encre du document sans compter les couleurs naturelles du fond. À ce titre, il offre par ailleurs une capacité à discriminer plus finement les variations colorimétriques dans les différents types de documents.

- ***Variance des plages verticales et horizontales de noir et de blanc (RLE)***

Ce descripteur permet de mesurer la distribution des éléments de contenu binaires selon les orientations privilégiées du texte (horizontale et verticale). Il pourra également permettre d'estimer une éventuelle inclinaison de la page numérisée en fonction des pics maximaux relevés. Le principe consiste à diviser la hauteur et la largeur d'image en 10 intervalles. On calcule le pourcentage des longueurs/largeurs des plages qui appartient à chaque intervalle. Enfin, on calcule la variance de ces 10 valeurs. Ce descripteur nous aide à distinguer les types de contenus comme un graphique, un texte ou une photo par exemple; il peut présenter indirectement le niveau de qualité comme le bruit, des plis ... Contribution du descripteur à la caractérisation du contenu : ce descripteur sera, de façon privilégiée, considéré comme un indicateur de la nature des données présentes sous la forme de traits

(contenus graphiques, images vs textuels). Ce descripteur peut présenter indirectement une information de qualité des traits dans sa capacité à en livrer des informations d'épaisseur (épaisseurs importantes du fait d'un flou exagéré par exemple) et de longueur (longueurs de plages très faibles liées à la présence de traits trop fins) ...

- ***Descripteurs de densités de niveaux de gris***

Ce descripteur est indicateur de quantité d'informations qui est choisi pour informer du niveau de remplissage dans le document. On calcule le nombre de pixels noirs (après la binarisation) sur le nombre total de pixels. Contribution du descripteur à la caractérisation du contenu : ce descripteur fournit une indication de densité locale des traits permettant notamment d'écarter la présence d'images ou de photos sur certaines pages. Il nous aide en ce sens à bien distinguer les types de contenus comme les graphiques, les textes ou les photos.

- ***Descripteurs portant sur les variations de luminance***

- *Variance régionale de luminance*

Cette méthode est choisie pour obtenir la largeur de la distribution des valeurs de luminance entre des zones qui peut annoncer indirectement la structure générale de l'image. Nous divisons l'image en 9 zones et calculons la moyenne de luminance de chaque zone. Puis, nous calculons la variance entre ces 9 zones. Ce descripteur nous aide à annoncer la présence d'informations de contenus dans un document à la structure dense ou complexe.

- *Moyenne régionale de luminance sur 9 zones*

Ce descripteur est choisi pour présenter les changements de la luminance dans la structure générale de l'image. Cette caractéristique est choisie pour décrire la densité locale d'informations et de distributions de luminance dans le document par une division régulière en régions : nous divisons pour cela l'image en 25 zones et nous calculons la moyenne de niveaux de gris pour chacune d'elle.

- *Gradient directionnel*

Ce descripteur est choisi pour identifier l'intensité du contraste dans 3 directions principales par l'information du gradient. Nous calculons la différence de niveaux de gris entre un pixel et les pixels voisins. On additionne les différences et on divise par le nombre de pixels total. On peut avoir un gradient horizontal, vertical, diagonal et un ensemble des trois. Ce descripteur nous aide à distinguer une qualité de flou et les densités locales.

- *Variance globale de luminance*

Ce descripteur est choisi pour présenter les informations du contraste grâce à l'information de la variance. D'abord, nous calculons le niveau de gris moyen de toute l'image. Après, nous calculons la différence entre cette valeur et la valeur propre de chaque pixel, et nous faisons la somme de toutes les différences. Enfin, nous divisons cette somme avec le nombre des pixels total. Ce descripteur informe sur la qualité du contraste.

- *Contraste global de l'image*

Ce descripteur est choisi pour présenter les informations du contraste entre l'encre et le fond de l'image. D'abord, nous extrayons les pixels des composantes connexes (pixels de l'encre) et les pixels qui ne sont pas des composantes connexes (pixels du fond) en binarisant l'image en noir/blanc. Puis, nous calculons le quotient entre le niveau de gris moyen des pixels de l'encre et des pixels de fond. Ce descripteur informe sur la qualité du contraste.

- *Entropie des transitions de pixels*

Un document peut être vu comme une organisation d'objets (de symboles textuels et graphiques de toutes sortes) ayant une disposition aléatoire ou structurée de symboles textuels et graphiques de toutes sortes. La notion d'ordre et de désordre dans l'organisation des données est importante et elle permet d'apprécier la façon dont les informations sont effectivement distribuées. Pour cela nous avons introduit un descripteur, l'entropie. Il a été choisi pour quantifier la proportion entre le nombre de transitions entre pixels noirs (objet) et blancs (arrière-plan) selon des droites orientées horizontalement et verticalement. L'entropie possède de grandes valeurs dans le cas de textes présentant de fortes variations et permet de marquer des différences significatives entre les images de traits (images de graphiques vs images de textes écrits) qui risqueraient d'être confondus avec d'autres descripteurs. Le principe repose sur le calcul des transitions de chaque ligne horizontale et verticale, suivi du calcul de l'entropie de ces valeurs. Ce descripteur nous aide à faire la distinction entre des types de contenus tels que graphique, texte ou photo.

- *Descripteurs portant sur les orientations principales*

L'orientation est une caractéristique très discriminante pour les images de traits. Nous avons choisi de proposer 4 descripteurs fondés sur cette dimension. Au départ nous avons estimé, à partir de l'autocorrélation de l'image, le diagramme polaire des orientations rendant compte des orientations principales sur l'ensemble de l'image traitée. Nous avons retenu

comme descripteurs : l'orientation principale globale, l'orientation principale par quadrant directionnel (selon des tranches fixes de 45 degrés), la variance des orientations (rendant compte de la diversité interne et de la prégnance d'une orientation particulière).

- ***Descripteurs basés sur les objets connexes***

Identifier des éléments permettant de discriminer géométriquement les données textuelles, graphiques et images. Les méthodes suivantes ont été choisies pour caractériser des éléments de contenu dans les documents. Elles sont utilisées pour fournir des indicateurs sur la présence de connexités, leur nombre afin de distinguer les éléments graphiques, des éléments textuels avec encore les images et les photographies.

- *Nombre de composantes connexes selon 3 tranches de surface*

Cette méthode consiste à déterminer dans un premier temps l'ensemble des composantes connexes dans l'image puis à exploiter leurs dimensions : hauteur, largeur et surface. Une discrétisation en 3 intervalles des valeurs de surface est ensuite réalisée pour classifier les occurrences rencontrées. Elles sont utilisées pour distinguer les éléments graphiques, des éléments textuels avec encore les images et les photographies.

- *Nombre des composantes connexes*

Cette méthode consiste à calculer le nombre de composantes connexes de l'image. Ce descripteur est exploité à des fins de discrimination entre des documents en fonction de leur nombre de connexités (jugé pertinent pour distinguer les documents de contenu textuel, des documents graphiques ou des images).

- *Nombre de composantes connexes selon 3 tranches de ratio hauteur/largeur*

Cette méthode exploite la quantité des différentes formes des composantes connexes existantes dans l'image. On peut ainsi distinguer différents types de lettres ou de typographies dans l'image. Une discrétisation en 3 intervalles des valeurs est ensuite réalisée pour classifier les occurrences rencontrées. Elles sont utilisées pour fournir des indicateurs sur les formes des éléments textuels souvent réguliers ou des graphiques et photographies souvent irrégulières.

## **Synthèse des descripteurs**

On dispose finalement de 7 caractéristiques pour un total de 41 valeurs scalaires. Le tableau ci-dessous synthétise ces descripteurs.

No	Valeurs (41 valeurs)	Descripteurs
1	Nombre des composantes connexes	<b>Descripteurs basés sur les objets connexes</b>
2	Nombre de composantes connexes selon 3 tranches de surface (< 10%)	
3	Nombre de composantes connexes selon 3 tranches de surface (10%< ... < 80%)	
4	Nombre de composantes connexes selon 3 tranches de surface (> 80%)	
5	Nombre de composantes connexes selon 3 tranches de ratio hauteur/largeur (< 10%)	
6	Nombre de composantes connexes selon 3 tranches de ratio hauteur/largeur (10%< .. < 80%)	
7	Nombre de composantes connexes selon 3 tranches de ratio hauteur/largeur (> 80%)	
8	Gradient directionnel horizontal	<b>Descripteurs portant sur les variations de luminance</b>
9	Gradient directionnel vertical	
10	Gradient directionnel diagonal	
11	Variance global de luminance	
12	Variance moyenne régionale de luminance (zone 1)	
13	Variance moyenne régionale de luminance (zone 2)	
14	Variance moyenne régionale de luminance (zone 3)	
15	Variance moyenne régionale de luminance (zone 4)	
16	Variance moyenne régionale de luminance (zone 5)	
17	Variance moyenne régionale de luminance	

	(zone 6)	
18	Variance moyenne régionale de luminance (zone 7)	
19	Variance moyenne régionale de luminance (zone 8)	
20	Variance moyenne régionale de luminance (zone 9)	
21	Variance régionale de luminance sur 9 zones	
22	Contraste global de l'image	
23	Densités de niveaux de gris	<b>Descripteurs de densités de niveaux de gris</b>
24	Variance des plages horizontales	<b>Variance des plages verticales et horizontales de noir et de blanc (RLE)</b>
25	Variance des plages verticales	
26	Entropie horizontal des transitions de pixels	<b>Entropie des transitions de pixels</b>
27	Entropie vertical des transitions de pixels	
28	Orientation principale par quadrant directionnel 0° - 45°	<b>Descripteurs portant sur les orientations principales</b>
29	Orientation principale par quadrant directionnel 45° - 90°	
30	Orientation principale par quadrant directionnel 90° - 145°	
31	Orientation principale par quadrant directionnel 145° - 180°	
32	Orientation principale globale	
33	Variance des orientations	
34	Couleur 1 - Noir	<b>Couleurs utilisés</b>

35	Couleur 2	<b>dans l'encre</b>
36	Couleur 3	
37	Couleur 4	
38	Couleur 5	
39	Couleur 6	
40	Couleur 7	
41	Couleur 8 - Blanche	

Table 30 – Synthèse des descripteurs retenus

#### 4.4.1.2. Sélection de caractéristiques pertinentes sur une base réduite

##### **Présentation de la base réduite pour l'étude des caractéristiques**

La base de données qui est utilisée pour ces expériences contient six classes de documents, chacune avec un contenu différent (voir Figure 43): « partitions musicales » (490 instances), « textes imprimés I » (230 instances), « textes imprimés II » (99 instances), « textes manuscrits I » (379 instances), « textes manuscrits II » (103 instances), « cartes » (279 instances). Les classes de « Texte imprimé I et II » sont composées de documents dont l'origine est différente (documents historiques et documents modernes) Ces sous catégories diffèrent de par leur fond, le bruit et les défauts, les polices,... présents dans les images. Ce découpage a été réalisé afin de permettre de les considérer comme deux sous-classes de la même classe plus générale: « Textes imprimés ». La même chose a été faite pour la classe « Textes manuscrits ». Pour l'ensemble des six catégories, le contenu intra-classe est relativement homogène. Pour les classes imprimés et manuscrits, les images avec des éléments graphiques ont été supprimées afin de permettre une meilleure compréhension des résultats de classification obtenus.

Partitions musicales	Textes imprimés		Textes manuscrits		Cartes
	I	II	I	II	
					

Figure 43 - Six classes de documents orientées « Types de contenus »

La Figure 44 représente la distribution des valeurs des 41 descripteurs retenus entre l'intervalle de [0,1] sur cette base d'étude. On peut constater que les descripteurs basés sur les objets connexes permettent de bien discriminer les différentes catégories de documents de notre base sans ambiguïté (spécialement les descripteurs 1, 2, 5). Les descripteurs portant sur les variations de luminosité (de 12 à 21) sont discriminants mais ils n'empêchent pas la confusion entre la catégorie «textes manuscrits I» et la catégorie «cartes» ou encore la catégorie «textes manuscrits II» et «partitions musicales». Le contraste global de l'image, la densité de niveaux de gris et les entropies des transitions de pixels sont également des descripteurs de contenus très intéressants (les 22, 23, 26 et 27). Pour les quatre caractéristiques associées à la description de l'orientation principale par quadrant directionnel, seul les deux quadrants 45° - 90° et 90° - 145° séparent très clairement les classes entre elles (les 29 et 30). L'orientation principale globale fonctionne bien mais n'évite pas la confusion entre les classes de «textes manuscrits I» et de «textes imprimés II» (descripteur 32). Les couleurs de l'encre n'étant pas significatives dans les images de la base traitée, seuls les niveaux *blanc* et *noir* (34 et 41) sont exploitables.

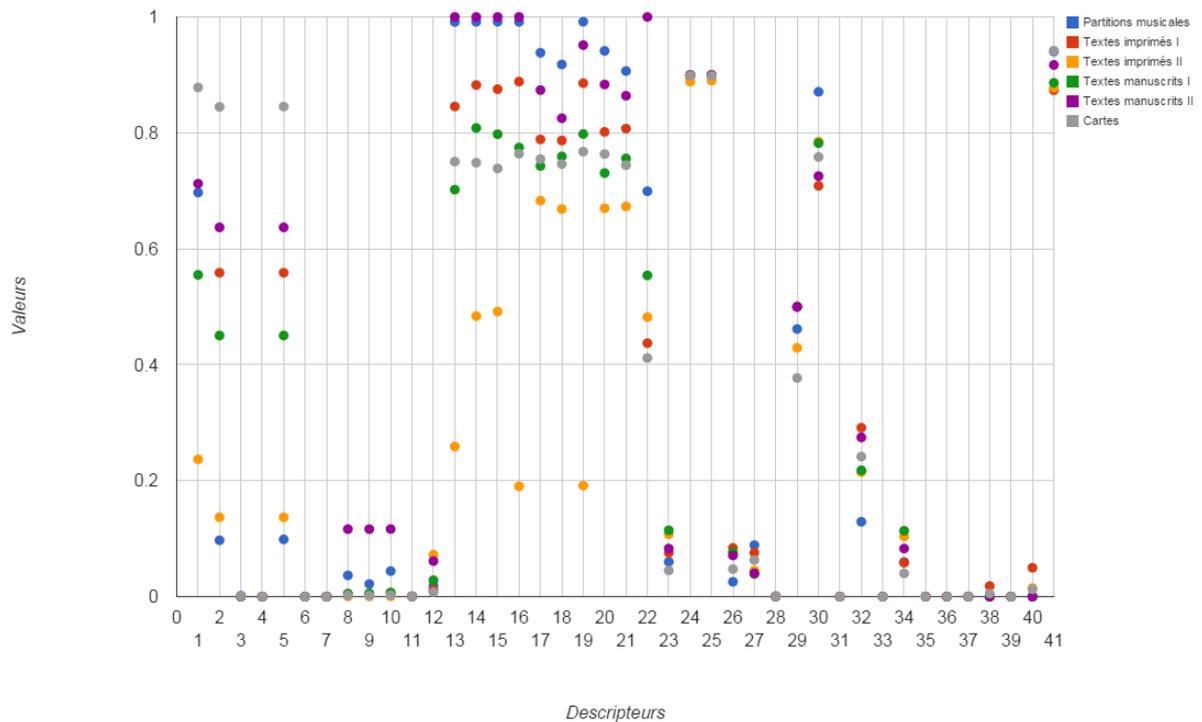


Figure 44 – Distribution des valeurs des descripteurs retenus

#### 4.4.1.3. Sélection des descripteurs pertinents par SVM-RFE

Le but de la sélection des caractéristiques est d'améliorer l'interprétabilité du modèle, de réduire les temps d'entraînement, et de renforcer la généralisation en réduisant le sur-apprentissage (*overfitting problem*). Nous avons exploité les méthodes génétiques comme *baseline genes* dans [Golub 1999], ou la méthode de *Stepwise Regression* dans [Draper 1998]. Nous pensons qu'il faut choisir une méthode de sélection de caractéristiques qui doit être cohérente avec la méthode mOC-iSVM, basée sur l'utilisation des SV. C'est pourquoi nous avons choisi d'utiliser le principe de la sélection de caractéristiques reposant sur le SVM-RFE (*Support Vector Machine with Recursive Feature Elimination*) défini dans [Guyon 2002] qui a été validé par comparaison à deux méthodes de sélection : *baseline genes* définie dans [Golub 1999], et l'analyse linéaire discriminante (LDA) et *Pseudo-inverse* définies dans [Duda 1973].

Les documents sont d'abord représentés par un ensemble de 41 caractéristiques qui ont été utilisées pour leur capacité à caractériser la variété de contenus. Considérant que les documents sont composés principalement par des traits contrastés et le contenu assez binaire (des objets noirs sur des fonds blancs), la plupart des descripteurs sont des valeurs scalaires qui expriment la distribution des pixels d'objets, leur entropie, leur densité, leur compacité. Ensuite, nous avons sélectionné 26 parmi les 41, grâce à la méthode SVM-RFE. Cela conduit

à la prise en compte des caractéristiques purement géométriques suivantes : blob (6 attributs), variation de luminance (10 attributs), densité de l'information (1 attribut), complexité (1 attribut), caractéristiques de texture (1 attribut), direction basée caractéristiques (5 attributs), et caractéristiques de couleur (2 attributs). Les documents analysés présentent des différences évidentes dans la quantité d'éléments correspondant à leur contenu (texte imprimé ou manuscrit, dessins...). Parce que les images sélectionnées de notre base de données contiennent principalement des images en niveaux de gris, les caractéristiques de couleur sont utilisées comme caractéristiques de luminance complémentaires.

#### 4.4.2. Réalisation des scénarios « orientés contenus » et résultats obtenus

A l'issue de l'étude des bonnes propriétés dont le scanner cognitif va devoir faire preuve (stockage limité, temps de calcul faible...) et faisant suite aux conclusions issues de l'analyse des performances des quatre versions de mOC-iSVM (présenté à la section 3.5), nous proposons d'utiliser la version standard mOC-iSVM pour résoudre le problème de classification de documents par classes de contenus.

##### 4.4.2.1. Description du protocole expérimental

Pour assurer la fiabilité du modèle et des résultats, nous utilisons pour l'ensemble des tests une validation croisée de 5 passes. Afin de simuler l'arrivée des données en flux successifs (incrémentalité du processus), l'ensemble d'apprentissage provenant de la validation croisée est divisé en 12 sous-ensembles qui correspondent à une succession de 12 étapes d'apprentissage. Chaque ensemble est mutuellement exclusif et contient respectivement 2%, 2%, 4%, 4%, 6%, 6%, 8%, 8%, 10%, 10%, 20% et 20% des données d'entraînement, tirées au hasard.

Par conséquent, de petits nombres de données par classe viennent d'abord pour évaluer la capacité de travailler avec très peu d'exemples (de 2 à 8 données dans une première étape selon les classes). L'ensemble de tests utilisé à chaque étape est la partie restante de l'ensemble de données séparées par validation croisée. Bien sûr, selon le scénario retenu, les étiquettes de chaque document sont adaptées pour répondre aux exigences des expériences. Les différents scénarios qui sont proposés dans les sections suivantes permettent de simuler des données de flux en entrée du système de classification. Ils ont été définis pour analyser le

comportement du système vis à vis de situations différentes d'apprentissage incrémental correspondant à une tâche de classification d'images de documents.

Les paramètres  $v$  et  $g$  sont sélectionnés par une recherche par grille appliquée à chaque étape (méthode GSES-*Grid Search at Each Step*). Pour l'évaluation, nous utilisons  $BA_i$  (Équation 6) de chaque classe prise séparément et  $GBA$  (Équation 7) pour comparer les résultats.

#### 4.4.2.2. Apprentissage incrémental simple

L'objectif de ce premier scénario est simplement d'évaluer la capacité d'apprentissage incrémental du scanner cognitif au cours du temps. Il s'agit donc d'étudier les intérêts et/ou limites de l'approche incrémentale par rapport à une approche plus classique dans laquelle le scanner conserverait toutes les données d'apprentissage et ré-apprendrait un nouveau modèle à chaque demande de l'opérateur. Dans ces scénarios, à chaque pas de temps, de nouvelles images de chaque classe sont mises à disposition du module d'apprentissage. Toutes les classes sont connues par le scanner dès le départ du scénario et de nouveaux exemples supplémentaires peuvent être exploités pour compléter l'apprentissage à chaque pas de temps. Les classes sont apprises interactivement par présentation de quelques exemples (2%, 2%, 4%...) de chaque classe et les capacités du module de numérisation cognitive à reconnaître les classes apprises sont testées pour chaque pas de temps.

À la fin du scénario, lorsque toutes les données d'apprentissage sont disponibles, les performances du système doivent être semblables à celles d'un système statique exploitant directement l'ensemble des données d'apprentissage. Dans ce scénario, nous comparons notre classificateur à :

- un classificateur SVM Multi-classes non-incrémental exploitant la stratégie 1 contre tous (avec la règle de décision du max) en utilisant toute la base d'apprentissage.
- un classificateur SVM Multi-classes construit à partir d'un ensemble d'one-class SVM en utilisant toute la base d'apprentissage (non-incrémental).

Ces deux classificateurs sont appris directement avec l'ensemble des données d'apprentissage et avec une approche *grid-search* pour trouver le meilleur paramétrage. Pour le système one-class classique, comme pour notre système, des données négatives sont utilisées lors de l'apprentissage pour l'optimisation des paramètres.

Performance (%)							
	Partitions musicales	Textes imprimés I	Textes imprimés II	Textes manuscrits I	Textes manuscrits II	Cartes	<i>GBA</i>
<b>SVM Binaire</b>	99.95	99.53	98.88	99.52	100	99.96	99.64
<b>one-class SVM</b>	97.94	72.9	93.78	88.73	99.42	82.71	89.25
<b>mOC-iSVM</b>	98.33	72.31	93.78	88.51	99.46	83	89.23

Table 31 – Résultat comparatif des approches.

Les résultats présentés Table 31 montrent que les SVM traditionnels fonctionnent bien sur ce type de problème, mieux que les classificateurs *one-class* ce qui n'est pas surprenant [Manevitz 2001]. Cependant le taux *GBA* est intéressant à analyser. Les performances du mOC-iSVM à l'étape finale sont similaires à ceux du système statique démontrant que l'apprentissage incrémental ne dégrade, en aucun cas, les performances. Ce constat avait déjà été fait dans la section 3.2.3 lors des expérimentations sur des benchmarks plus classiques. La Figure 45, montre l'évolution des taux de reconnaissance pour chaque classe au cours du temps. On peut noter que le *GBA* augmente progressivement, même si, pour certaines classes, les taux de reconnaissance diminuent à certains pas de temps. Ceci s'explique par l'apparition de données qui peuvent localement différer significativement des données vues préalablement.

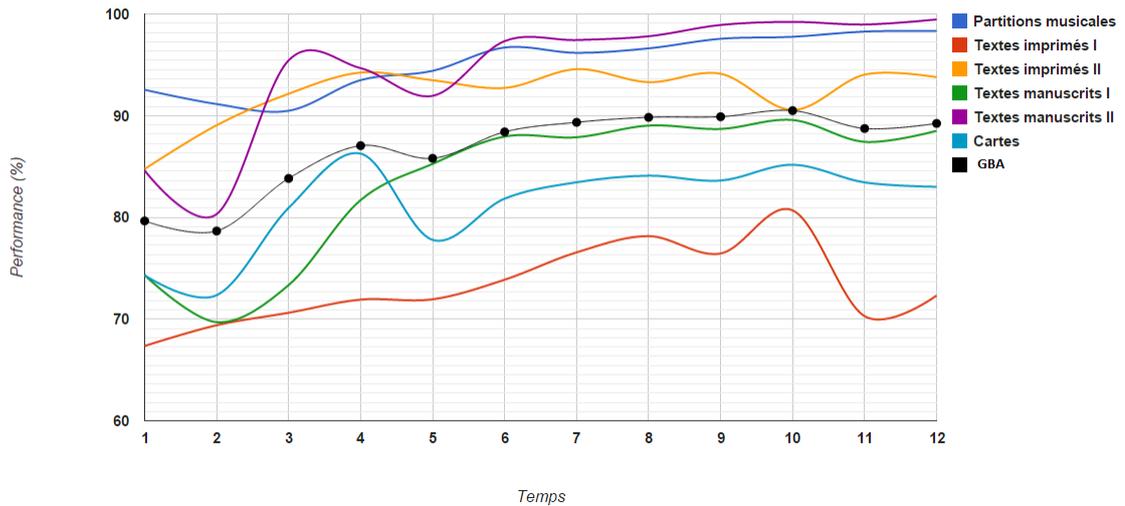


Figure 45 - Apprentissage incrémental du mOC-iSVM ( $BA_i$  pour chaque classe et  $GBA$ ).

#### 4.4.2.3. Apparition de nouveaux concepts au cours du temps

Ce scénario a pour objectif d'évaluer le comportement du scanner cognitif lors de l'apparition de classes inconnues au cours d'une campagne de numérisation. Il s'agit de voir si l'apparition d'un nouveau concept perturbe la reconnaissance de ceux préalablement existants ainsi que le temps (ou le nombre de données) nécessaire pour que ce nouveau concept soit bien assimilé par le module cognitif.

C'est ici que l'opérateur aura la charge de fournir, pas après pas, les exemples d'images de la nouvelle classe afin que celle-ci soit ensuite reconnue par le module cognitif. Il est important de noter que le fait d'avoir utilisé un modèle composé d'un ensemble de classificateurs *one-class* procure un large avantage pour ce scénario puisque l'apparition d'une nouvelle classe (image inconnue) devra se traduire par un rejet de cette image par l'ensemble des classificateurs *one-class* existants. Une alerte pourra ainsi être levée et l'opérateur pourra intervenir pour définir un nouveau protocole de numérisation associé à cette nouvelle classe de type de contenus (et donc également un nouveau classificateur *one-class*).

Le flux mis en place pour ce scénario est le suivant :

- A l'étape 1 : la classe de contenus « Partitions musicales » est connue.
- La classe « Textes manuscrits I » apparaît à l'étape 2
- La classe « Textes manuscrits II » apparaît à l'étape 3
- La classe « Textes imprimés I » apparaît à l'étape 4
- La classe « Textes imprimés II » apparaît à l'étape 5

- Finalement, la classe « Cartes » apparaît à l'étape 6
- Durant les pas suivants (de 6 à 12), de nouvelles images de chacune de ces classes sont soumises au classificateur.

Notons que, durant les étapes 1 à 6, seules les images correspondant aux classes connues sont soumises au classificateur lors des phases de tests successives.

Les résultats sont présentés en Figure 46. On observe que les premières classes apprises sont bien reconnues et ne souffrent pas trop de l'apparition de nouvelles classes. Bien sûr, de légères baisses sont visibles mais les taux restent comparables à ceux obtenus lors du scénario 1 démontrant que chaque *one-class* n'est pas trop perturbée par la création d'autres *one-class* supplémentaires. Par exemple, au pas 2, le  $BA_i$  de la classe « Partitions musicales » est de 3% inférieur par rapport au scénario 1 et ré-augmente ensuite. De même, « Textes imprimés I » commence avec un taux assez élevé et diminue jusqu'au pas 6 correspondant à l'apparition de l'ensemble des classes. Ensuite le comportement est similaire à celui du scénario 1. Toutes les autres classes voient leur  $BA_i$  augmenter régulièrement jusqu'au pas 12. Le  $GBA$  obtenu est le même que celui du scénario 1 bien que, globalement, moins d'exemples aient été utilisés pour l'apprentissage.

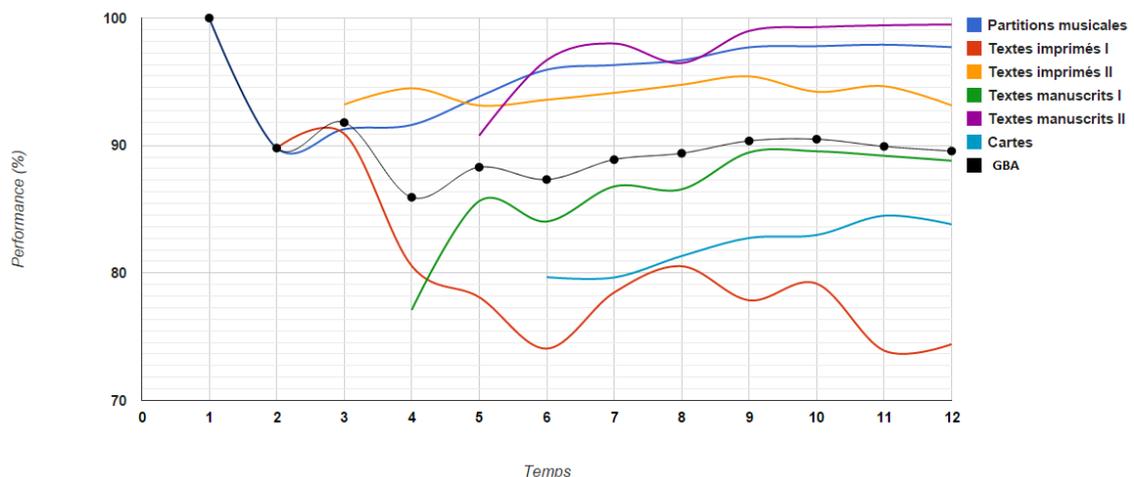


Figure 46 - Apparition de nouveaux concepts au cours du temps

#### 4.4.2.4. Extension de Concepts

Ce scénario a pour objectif d'étudier le comportement de notre module cognitif en cas d'extension de concept. L'extension de concept correspond au fait que les représentants d'une même classe peuvent se diversifier significativement au cours du temps.

Par exemple, dans le scénario prévu, nous avons à disposition deux sous-catégories d'images dans les classes « Textes imprimés » et « Textes manuscrits ». Si lors des étapes 1 à 5, seuls des exemples de la sous-catégorie « Textes imprimés I » (resp. « Textes manuscrits I ») sont utilisés, le classificateur *one-class* risque de se spécialiser pour cette sous-catégorie. Lors des étapes suivantes (6 à 12), on lui présente alors des exemples de la deuxième sous-catégorie et on analyse le temps qui est nécessaire aux classificateurs pour s'adapter à cette extension de concept.

Comme dans les scénarios précédents, seuls des exemples des classes et sous-catégories connues à chaque pas sont insérés dans les ensembles de tests. Un second scénario similaire est conduit en inversant l'apparition des sous-catégories : « Textes imprimés II » (resp. « Textes manuscrits II ») apparaît avant « Textes imprimés I » (resp. « Textes manuscrits I ») au lieu de l'inverse. Comme nous pouvons le voir Figure 47, le  $BA_i$  est assez élevé avant l'étape 6 et la chute du  $BA_i$  est significative lors de l'extension du concept. On note que même la classe « Cartes » est impactée par cette chute alors que la classe « Partitions musicales » l'est moins.

Néanmoins, les  $BA_i$  remontent ensuite régulièrement durant les étapes suivantes (6 à 12) pour atteindre des taux similaires à ceux atteints lors des scénarios précédents lors de l'étape finale. Cette expérience semble donc démontrer que les capacités d'adaptation de notre module de numérisation cognitive seront suffisantes pour supporter des extensions de concepts et réagir (s'adapter) assez rapidement.

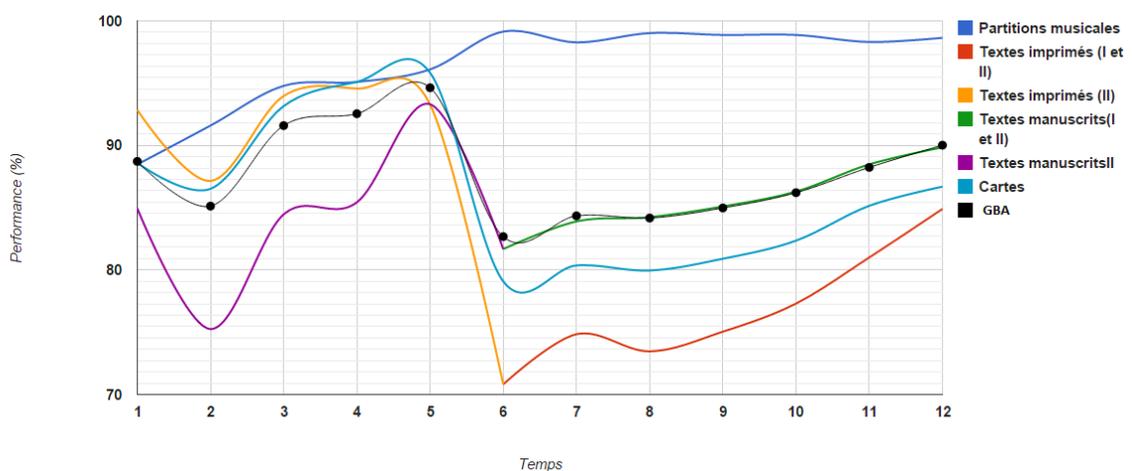


Figure 47 - Extension de Concept

#### 4.4.2.5. Évolution de concept (Concept drift)

L'évolution de concept est une configuration typique pouvant apparaître lors de la classification de flux de données. Il correspond aussi à un cas pouvant typiquement apparaître lors d'une campagne de numérisation de documents.

Ce phénomène se produit lorsque la fonction de densité de probabilité d'une classe change au cours du temps. Le scénario précédent peut d'ailleurs être vu comme un cas particulier de l'évolution de concept. Cependant, dans l'évolution de concept, on considère généralement que la fonction densité de probabilité initiale doit être oubliée et remplacée par la nouvelle ; ce qui n'est pas le cas pour l'extension de concept. L'évolution de concept correspond donc au fait que les représentants d'une même classe peuvent changer significativement au cours du temps (plutôt que simplement se diversifier).

Pour simuler ce phénomène, nous allons ré-exploiter nos 2 classes comportant des sous-catégories (« Textes imprimés » et « Textes manuscrits ») mais cette fois ci, après l'étape 5, « Textes imprimés I » (*resp.* « Textes manuscrits I ») viendra remplacer complètement « Textes imprimés II » (*resp.* « Textes manuscrits II ») plutôt que de venir en complément. Comme précédemment, il est possible de générer 2 scénarios différents en inversant l'ordre d'apparition des sous-catégories.

Notre système s'adapte assez rapidement aux évolutions démontrant une bonne plasticité (ce qui peut parfois constituer une limite plutôt qu'un avantage). Entre l'étape 6 et l'étape 7, le  $BA_i$  maximum atteignable pour les classes cibles est presque atteint. On note que la classe « Partitions musicales » n'est pas impactée par les changements, contrairement à la classe « Cartes » qui obtient une large augmentation de son taux de bonne classification probablement due à la suppression de confusion entre la classe « Cartes » et les classes « Textes manuscrits I »/« Textes imprimés I ».

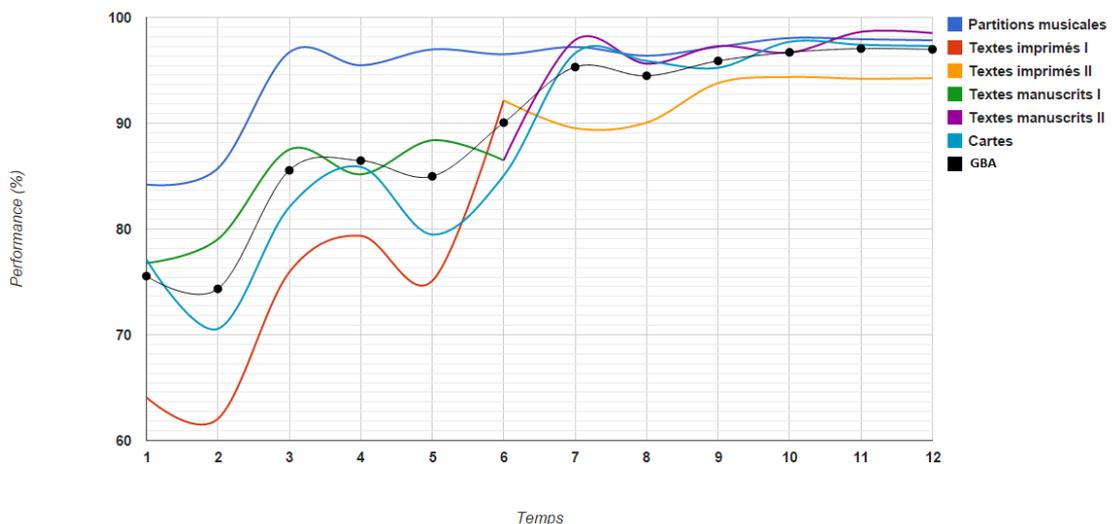


Figure 48 - Concept drift de « Textes manuscrits I » (resp. « Textes imprimés I ») vers « Textes manuscrits II » (resp. « Textes imprimés II »)

#### 4.4.2.6. Fusion et division de concepts

Pour finir, le dernier phénomène pouvant apparaître lors d'une campagne de numérisation menée par un opérateur est la fusion ou la division de concepts. Cela correspond à la volonté qu'aurait un opérateur de transformer :

- une classe unique de type de documents (« Textes imprimés ») en 2 sous-classes (« Textes imprimés I » et « Textes imprimés II ») parce qu'il considère, par exemple, qu'il serait préférable d'avoir deux protocoles de numérisation différents pour chacune des sous classes.
- deux classes différentes (« Textes manuscrits I » et « Textes manuscrits II ») en une seule classe parce qu'il constate que ces deux classes sont similaires et nécessitent le même protocole de numérisation.

À notre connaissance, ce type de phénomène n'a jamais été réellement traité par les chercheurs travaillant sur la classification incrémentale supervisée. Nous présentons les résultats obtenus dans différents scénarios de fusion et division de concepts en lien avec l'identification des classes de contenus que nous avons défini pour notre cadre applicatif (DIGIDOC). Sur ces figures, nous observons logiquement une diminution des taux de reconnaissance lorsque la division ou la fusion a lieu. Mais comme pour les scénarios précédents, on constate ensuite que les capacités d'adaptation de nos classificateurs sont suffisamment bonnes pour permettre d'atteindre rapidement des performances similaires à

celles qui auraient été obtenues sans l'occurrence du phénomène perturbateur (fusion ou division).

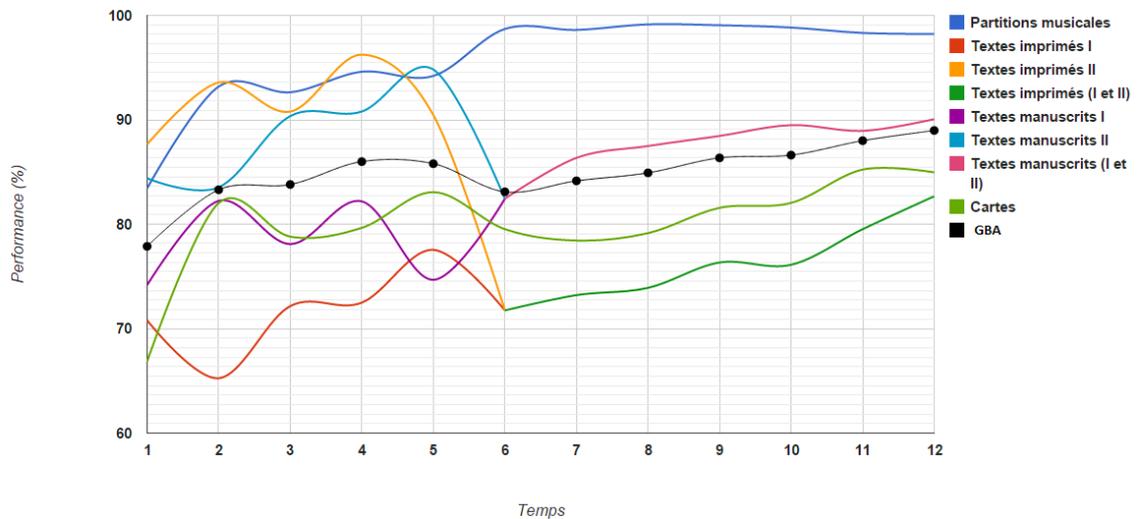


Figure 49 - Fusion de « Textes imprimés I », « Textes manuscrits I », « Textes imprimés II », « Textes manuscrits II » vers « Textes imprimés » (I and II), « Textes manuscrits » (I et II).

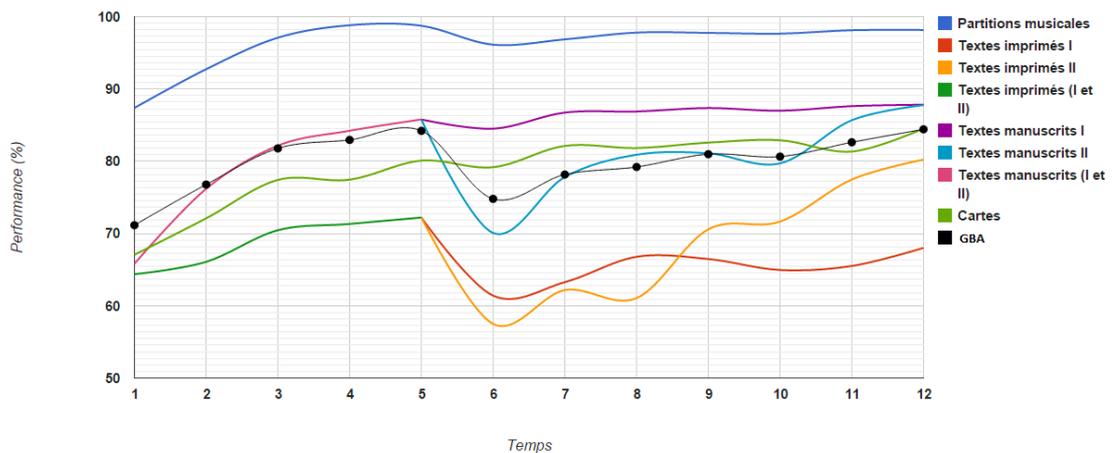


Figure 50 - Division de « Textes imprimés » (I and II), « Textes manuscrits » (I and II) vers « Textes imprimés I », « Textes manuscrits I », « Textes imprimés II », « Textes manuscrits II ».

#### 4.4.3. Procédure en ligne du scanner cognitif

Afin de proposer un scénario simulant le fonctionnement du scanner cognitif en situation d'acquisition continue d'images de documents, nous avons choisi de procéder en deux temps : tout d'abord, il faut que nous montrions la capacité de notre système à pouvoir traiter des données arrivant en flux (et pas en lot) et pour cela faire varier les quantités de

données d'apprentissage et observer le comportement de notre modèle. Dans un second temps, nous proposons une mise en situation réelle reposant sur la numérisation « cognitive » de 130 puis de 450 images, impliquant un apprentissage qui se fait au fur et à mesure de l'arrivée des données. L'objectif est ici de montrer les forces et les limites d'un tel système, notamment à travers l'étude des performances pas toujours égales selon la nature des contenus et l'étude des variations de performances en lien avec les quantités d'apprentissage (qui traduit la capacité d'adaptation des classificateurs one-class à la variété des données).

#### 4.4.3.1. Simulation du scanner cognitif par lot de données

Nous exploitons ici un scénario du scanner cognitif simulant l'arrivée d'un flux d'images numérisé au fil du temps. Le but est d'illustrer la capacité du classificateur mOC-iSVM à bien catégoriser les données entrantes dans une situation où les apprentissages sont fréquents et fonctionnent sur des quantités de données réduites et où l'utilisateur a une place importante qui lui permet de contrôler (valider/modifier) la catégorie proposée par le classificateur pour chaque nouvelle donnée traitée. L'utilisateur a aussi un rôle à jouer dans la détermination de la fréquence des apprentissages à effectuer sur des données valides.

**Modalités d'apprentissage.** Le principe du classificateur embarqué dans le scanner est qu'il dispose de deux modes possibles d'apprentissage : un mode *continu* supervisé où les données sont traitées *une par une* et où l'apprentissage s'effectue sur des données validées par l'utilisateur au fur et à mesure des acquisitions et un mode *par lot* supervisé où cette fois les apprentissages sont réalisés à une certaine fréquence seulement.

**Démarrage de la procédure.** Le démarrage du scanner est dans tous les cas engagé sur un mode supervisé pour assurer le «pré»-apprentissage des classes (sauf si on relance le scanner sur une numérisation interrompue : dans ce cas on « charge » un modèle existant) et la définition des protocoles de numérisation. En mode supervisé, le classificateur propose des scores de reconnaissance pour l'ensemble des classes associées au scanner et c'est l'utilisateur qui valide la classe et l'association classe – protocole (il crée le cas échéant la nouvelle classe, le nouveau protocole, les deux ou aucun). Au démarrage, il n'y a, *a priori*, aucun modèle. Quand une image est numérisée, le classificateur embarqué dans le scanner est chargé de reconnaître le contenu associé à la page et de proposer un protocole associé à cet exemple selon les associations « classe-protocole » connues du système. A ce stade, c'est un protocole par défaut qui est utilisé pour numériser l'image afin de disposer d'une description « normalisée » et comparable entre images. Au démarrage, ne disposant d'aucune connaissance, le système n'est pas en mesure de lancer de classification. L'utilisateur doit

donc manuellement étiqueter l'image et la conserver pour un apprentissage ultérieur (la fréquence de l'apprentissage est adaptable, l'utilisateur peut choisir d'attendre de disposer de plusieurs images pour améliorer l'apprentissage ou de lancer l'apprentissage image par image).

**Fonctionnement au fil de l'eau** Une fois le premier apprentissage réalisé et le (ou les) premier(s) modèle(s) de classe(s) établi(s), le classificateur peut être testé sur les nouvelles images juste numérisées. Si le classificateur renvoie un résultat incorrect, l'utilisateur peut proposer de modifier la classe désignée ou de créer une nouvelle classe de contenus associée à la nouvelle image (la description de cette image nouvellement étiquetée sera conservée pour les apprentissages suivants). Au fur et à mesure des apprentissages, de nouveaux modèles peuvent ainsi être créés, ou enrichis (mis à jour).

Voyons désormais sur des jeux de données de tests et des ensembles d'apprentissages comment se comporte le scanner.

#### 4.4.3.2. Expérimentations du mOC-iSVM et de ses variantes

Dans le scénario que nous proposons, nous utilisons une série de 130 images (qui alimenteront la base d'apprentissage) qui sont traitées au fur à mesure. Cette série est composée de 6 classes de contenus. La classe 1 est la classe des « Partitions musicales », la classe 2, celle des « textes imprimés I », la classe 3, celle des « textes imprimés II », la classe 4, celle des « textes manuscrits I », la classe 5, celle des « textes manuscrits II », et la classe 6 enfin, celle des « Cartes ».

Les images sont ordonnées dans la chaîne simulant une situation d'acquisition réelle créant des alternances dans l'arrivée des images à numériser selon une fréquence prédéfinie.

Partitions musicales	Textes imprimés		Textes manuscrits		Cartes
Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
					

Figure 51 - Six classes de contenus

Un ensemble de tests contenant 1450 images est utilisé pour observer l'évolution de la capacité de chaque classificateur à bien identifier les données au fil du temps. Le scénario de classification incrémentale est réalisé 5 fois à partir d'ensembles d'apprentissages de 5 tailles différentes : 1 donnée, 2 données, 5 données, 10 données et 25 données considérées à chaque étape. La table suivante illustre la répartition des données pour les tests pour chacune des classes.

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Total
<b>Ensemble d'apprentissage</b>	1, 2, 5, 10	1, 2, 5, 10	1, 2, 5, 10	1, 2, 5, 10	1, 2, 5, 10	1, 2, 5, 10	1, 2, 5, 10
<b>Ensemble de test</b>	465	205	74	354	88	264	1450

Table 32 – Distribution des données dans la base DIGIDOC pour les tests de performance du mOC-iSVM.

En premier lieu, nous allons nous intéresser au comportement du mOC-iSVM lors d'un apprentissage continu (apprentissage d'une donnée à chaque fois). La Figure 52 illustre le résultat en termes de précision globale (*GPR* - Équation 10), sensibilité globale (ou rappel global; *GSN* - Équation 8) et spécificité globale (*GSP* - Équation 9) à partir de l'évaluation sur les ensembles de tests dont la répartition est indiquée à la Table 32.

On observe sur ces courbes deux phénomènes essentiellement. D'une part les taux moyens de sensibilité (rappel) et de précision augmentent au fur et à mesure de l'introduction des nouveaux modèles permettant de faire diminuer le rejet; nous constatons que l'apprentissage sur une seule donnée pour une classe donnée conduit à un pic important de reconnaissance à l'étape de test sur cette classe en question. D'autre part, les performances de chaque classificateur sont également améliorées à chaque nouvel apprentissage, par le simple fait qu'une donnée mal étiquetée par un classificateur *one-class* (par exemple le classificateur *one-class* de la classe 1 (OC1) peut classer une donnée relevant normalement de la classe 2 dans sa propre classe. Un apprentissage réalisé pour modéliser la classe 2 peut alors permettre de faire ressortir cette donnée et la reclasser correctement dans sa catégorie d'appartenance. La progression observée tout au long du processus incrémental illustre donc ces deux phénomènes.

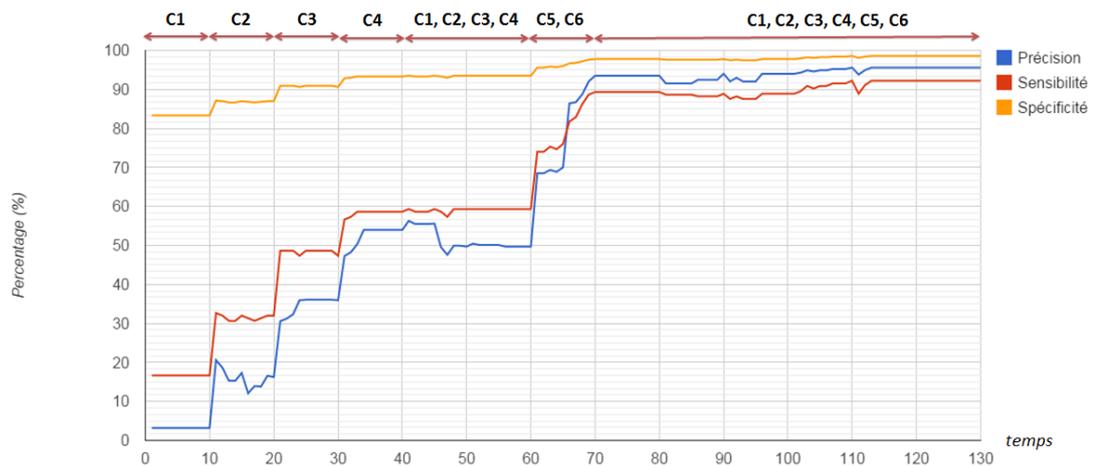


Figure 52 – Précision globale (*GPR*), sensibilité globale (*GSN*) et spécificité globale (*GSP*) mesurée pour un apprentissage un à un. Il est indiqué sur cette figure les périodes où les données arrivent et leur classe d'appartenance réelle.

Nous observons qu'à partir du moment où tous les modèles sont connus (étape 70 notée en abscisse), les taux moyens de rappel et de précision n'augmentent plus que très doucement au fil du temps. Le classificateur mOC-iSVM progresse en oscillant jusqu'à se stabiliser une fois l'ensemble des modèles requis connus.

La Figure 53 illustre la performance du mOC-iSVM configuré cette fois avec des ensembles d'apprentissages de tailles variables. Les valeurs indiquées correspondent aux taux *GBA* (moyenne calculée entre la spécificité et la sensibilité). Cette expérience indique que la variation de performances calculée entre ces différentes configurations ne dépasse pas 5% : on peut globalement considérer qu'elle n'influence pas directement le résultat des classificateurs. Cela provient de la combinaison qui est réalisée à chaque nouvel apprentissage avec les anciens vecteurs supports qui représentent l'historique des modèles issus des étapes précédentes et qui permettent donc de les renforcer.

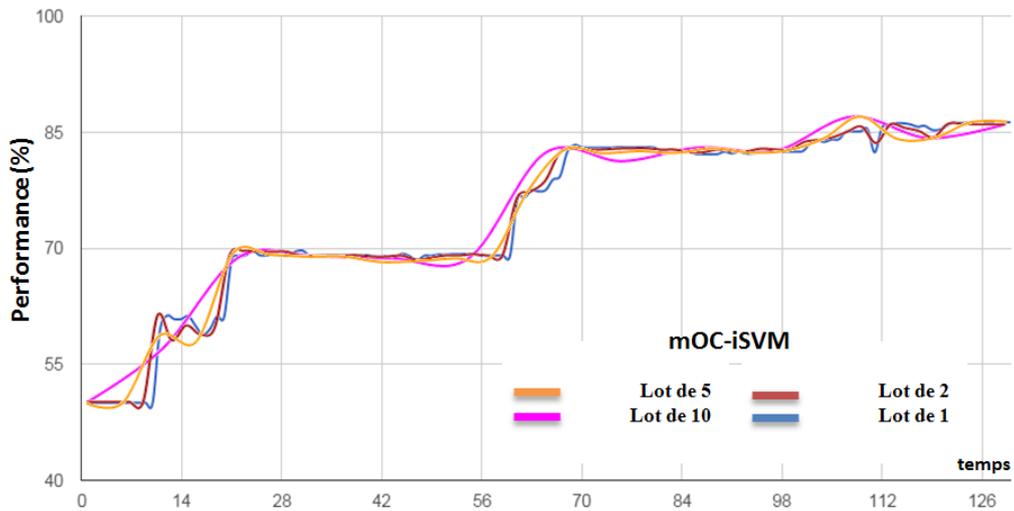


Figure 53 - Courbe d'évolution de la performance (*GBA*) avec différentes dimensions de lots d'apprentissage.

La Figure 54 présente l'évolution du nombre de vecteurs supports au fil du temps dans le mode en ligne. En faisant la référence avec l'évolution de la performance (Figure 53), on peut observer que le nombre de SVs se stabilise à l'étape 70, étape à partir de laquelle l'ensemble des classes a été balayé au moins une fois avec un minimum de 5 données. Après la 70<sup>ème</sup> étape, le nombre des vecteurs supports continue à augmenter mais sans augmentation majeure de performance (moins de 1% de performance *GBA*). C'est à ce moment qu'il est possible d'arrêter l'apprentissage (en mode « supervisé » du scanner) pour passer à la procédure de la reconnaissance automatique sans apprentissage (mode « automatique » du scanner).

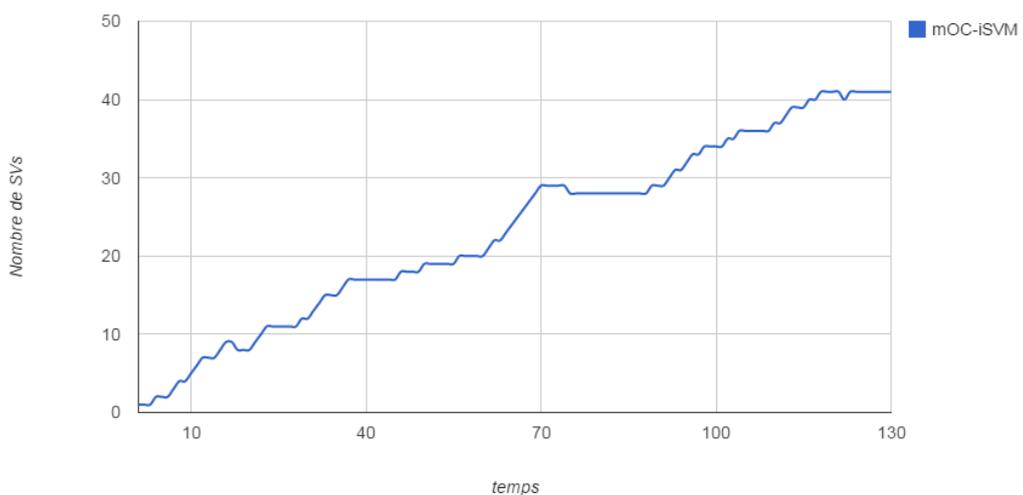


Figure 54 - Courbe d'évolution du nombre de SVs dans le mode « en ligne »

Le scénario, tel que nous l'avons choisi, décrit une situation typique d'un environnement stationnaire. Afin de comparer les performances de notre classificateur mOC-iSVM, nous avons choisi de considérer également ici ses différentes variantes. Dans la situation en lien avec le fonctionnement d'un scanner, nous sélectionnons la configuration la plus économique (au sens du coût calculatoire). Nous avons donc choisi de paramétrer les variantes du mOC-iSVM de la façon suivante : mOC-iSVM.AP avec  $\alpha = 1$ , mOC-iSVM.EP avec  $f = 80\%$  et mOC-iSVM.nB avec  $n = 1$ . La Figure 55 présente les résultats superposés des trois évolutions de performances (*GBA*) issues du mOC-iSVM, du mOC-iSVM.AP, du mOC-iSVM.EP et du mOC-iSVM.nB. Les courbes associées au mOC-iSVM et au mOC-iSVM.EP sont très similaires et correspondent aux meilleurs résultats. Nous pouvons constater sur cette figure que la configuration idéale demeure la version standard du mOC-iSVM, comme nous l'avons observé lors des tests sur des données synthétiques de benchmarks. Par ailleurs, on peut s'étonner des résultats faibles du mOC-iSVM.nB qui jusqu'ici se révélait très performant en environnement stationnaire. Cependant ici avec une valeur de  $n$  limitant les quantités de données à manipuler, ses performances décroissent sensiblement. Il se trouve que pour simuler le fonctionnement du scanner, il est recommandé de limiter au maximum les calculs pour produire des résultats de classification dans des temps raisonnables.

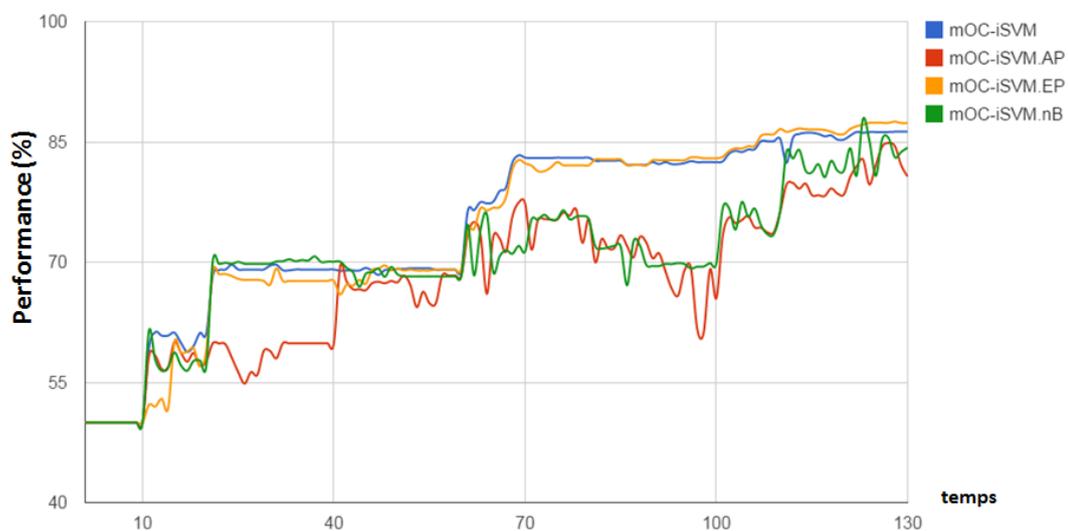


Figure 55 - Courbe d'évolution de la performance (*GBA*) des versions de mOC-iSVM dans la base de scanner

Le Table 33 illustre les résultats de temps de calcul, de performances, les conditions de mise en œuvre de chacune des versions. On peut conclure que le choix de mOC-iSVM

standard pour ce type de problème est certainement le plus approprié en mettant en vis-à-vis les besoins associés au fonctionnement du scanner cognitif (temps de calcul faible, bonne performance, sans stockage et sans paramètre à choisir) et les contraintes calculatoires des différentes versions de mOC-iSVM.

Approches	Temps de Calcul (s)	Performance ( <i>GBA</i> ) de la dernière étape (%)	Note
<b>mOC-iSVM</b>	35	86.26	Sans stockage, sans paramètre à choisir
<b>mOC-iSVM.AP</b>	27.6	77.19	Petite stockage, un paramètre à choisir
<b>mOC-iSVM.EP</b>	329.85	87.01	Grand stockage, un paramètre à choisir
<b>mOC-iSVM.nB</b>	364.42	84.28	Grand stockage, un paramètre à choisir

Table 33 – Temps de calcul et performance des versions de mOC-iSVM dans la base de scanner

## 4.5. Conclusion

Dans ce chapitre, nous avons choisi d'appliquer notre classificateur mOC-iSVM à une situation de traitement de documents en flux. Ce chapitre aborde tout naturellement la problématique soulevée par le projet ANR DIGIDOC autour de la conception d'un scanner cognitif doté d'un module de reconnaissance de contenus incrémental. Pour cela nous avons tout d'abord rappelé les études de l'état de l'art concernant la classification de documents et plus spécifiquement la classification de documents traités en flux. Le constat actuel est qu'il existe un vide méthodologique important en matière de classification de documents en flux et que les solutions proposées ne sont que des réponses partielles aux problèmes soulevés par la classification dynamique : adaptation à la nature de l'environnement (stationnaire ou non stationnaire), capacité à gérer l'intégration de nouveaux modèles et à assurer leur mise à jour. Notre proposition mOC-iSVM a été formulée pour permettre de répondre à ces différents enjeux. Notre motivation première étant d'intégrer un classificateur dynamique au sein d'un scanner, nous avons ensuite balayé le spectre large des scanners existant sur le marché. Nous avons constaté, une fois encore, qu'il n'existait pas de dispositif de numérisation capable d'analyser les données en flux avec la capacité d'intégrer incrémentalement au système de

nouvelles connaissances associées aux nouveaux contenus rencontrés. Or cette capacité de diagnostiquer un nouveau contenu (comme de déterminer automatiquement si une qualité de numérisation est satisfaisante) fait partie des propriétés qu'un scanner doit pouvoir assurer. Dans un contexte supervisé, le mOC-iSVM (et ses variantes) offre la possibilité de traiter, de façon incrémentale, des données de contenus variables au fil du temps : en créant dynamiquement de nouveaux modèles de contenus, en gérant l'évolution de concepts, leur extension et leur éventuelle dérive et en assurant une bonne capacité de fusion ou de division de concepts dès que nécessaire). Les résultats obtenus sur la base de données d'images de DIGIDOC sont prometteurs et comblent le vide existant dans le domaine de l'analyse et la reconnaissance de documents en matière de classification incrémentale.

Nous avons ensuite testé le fonctionnement du mOC-iSVM et de ses variantes à l'occasion d'une mise en situation réelle de numérisation de documents en flux et montré la capacité du classificateur à reconnaître, avec des performances satisfaisantes, des contenus à partir d'un entraînement effectué uniquement sur des jeux de données de très faibles dimensions (1, 2, 5, 10 données d'apprentissage). Les configurations retenues pour le mOC-iSVM et ses versions pour fonctionner en temps raisonnable sur une chaîne de numérisation offrent à ce jour des scores perfectibles. On observe cependant que le système présente une capacité d'adaptation aux nouvelles données très rapide sans nécessiter d'apprentissage lourd. L'indépendance des SVM *one-class* est un atout important pour la classification de données traitées en flux. Le simulateur fonctionne de façon supervisée liée à une intervention humaine importante (phase d'initialisation des classes de contenus, analyse du rejet, création des nouvelles classes dès que nécessaire, validation des décisions prises par le classificateur). Il semble désormais important également de réfléchir à une implantation qui permette d'augmenter l'autonomie du système afin d'alléger l'intervention humaine.

## Conclusion et perspectives

# V. Conclusion et perspectives

## 5.1. Conclusion

Avec comme but applicatif initial, la conception d'un moteur de scanner cognitif, nous nous sommes intéressés, dans cette thèse, à la problématique de l'apprentissage et de la classification dynamique en environnements stationnaires et non stationnaires. L'objectif est d'obtenir un système de classification pouvant supporter de faibles quantités de données d'apprentissage au départ et capable d'ajuster, ensuite, au fil du temps, ses modèles selon la variabilité des données entrantes pour faire progresser ses taux de performances lorsque plus de données sont disponibles et au travers d'interactions avec l'utilisateur.

Le moteur de scanner cognitif dont il est question ici constitue l'un des objectifs majeurs du projet ANR DIGIDOC visant à améliorer et à simplifier les paramétrages lors de la numérisation de gros volumes de données. Ce type de scanner doit donc pouvoir décider seul des paramètres de numérisation en fonction des propriétés des documents à numériser (en particulier en fonction de leur contenu et de leur qualité) et de l'usage que souhaite en faire l'utilisateur. Cela sous-tend deux choses :

- d'une part que le système soit capable, au fil du temps, d'améliorer sa capacité à distinguer les différentes catégories de documents prédéfinis par l'opérateur ;
- d'autre part, que le système ait la capacité d'identifier de nouvelles situations et de s'y adapter, que celles-ci soient intrinsèquement liées aux nouvelles données traitées ou bien qu'elles correspondent à un nouveau besoin de l'utilisateur.

Ces ambitions correspondent aux avantages pouvant être procurés par des mécanismes d'apprentissage et de classification incrémentaux. Ainsi, au travers d'un état de l'art des principales méthodes d'apprentissage et de classification dynamique, le chapitre 2 offre une nouvelle lecture des différents mécanismes d'apprentissage existants afin de les mettre en regard et d'identifier ceux qui permettront d'élaborer le moteur du scanner cognitif. Cette analyse met en exergue la difficulté de formaliser ou comparer les travaux existants du fait de la variété et de l'inconstance dans le vocabulaire utilisé pour spécifier les différentes problématiques, situations, mécanismes proposés : renforcement de concepts, fusion de concepts, dérive de concepts, apprentissage en ligne, apprentissage incrémental, apprentissage continu, plasticité ...

Nous essayons, dans cette partie, de montrer les spécificités des approches selon deux considérations principales : l'évolution des concepts et l'évolution du nombre de concepts. Ces axes de lecture nous permettent de dépasser les difficultés liées à la variété et à la « polysémie » des termes et qualificatifs utilisés dans la littérature afin de pouvoir ensuite mieux positionner nos contributions.

Dans le chapitre 3, sur la base de cet état de l'art, nous proposons une méthode unifiée capable de résoudre plusieurs des problèmes de classification incrémentale dans des situations où les concepts peuvent évoluer à la fois dans leur description (au fil du temps) et dans leur nombre. Cette approche repose sur une classification par combinaison de classificateurs indépendants SVM. Le modèle proposé est un classificateur conçu à partir de plusieurs classificateurs *one-class* qui ont chacun la capacité d'évoluer indépendamment des autres (évolution de leur description) et qui autorise l'ajout de nouveaux classificateurs *one-class* sans impact sur les performances des autres. Ce modèle générique (qui est nommé mOC-iSVM, pour classificateur incrémental « *Multi One-Class Incremental SVM* ») est conçu pour permettre la génération de plusieurs versions spécialisées considérant différemment les vecteurs supports qui seront exploités dans les étapes successives d'apprentissage.

Trois versions spécialisées ont été développées pour répondre efficacement à des problèmes de classification relevant des environnements à la fois stationnaires et non-stationnaires

- mOC-iSVM.AP, sélectionnant les vecteurs supports selon leur ancienneté ;
- mOC-iSVM.EP, sélectionnant les vecteurs supports selon les performances associées à leurs modèles ;
- mOC-iSVM.nB, sélectionnant les vecteurs supports selon les performances associées à leurs modèles mais se limitant aux n dernières étapes d'apprentissage.

Comme le montrent les nombreuses expérimentations réalisées sur les benchmarks classiquement utilisés dans le domaine de l'apprentissage continu, chacune des variantes est applicable aussi bien en environnement stationnaire que non-stationnaire et propose des capacités d'ajout, de modification (extension, dérive), de fusion/division de concepts. Cependant, chacune apporte des avantages différents. La première version mOC-iSVM standard est idéale pour les environnements stationnaires de par son faible coût de calcul. La deuxième variante, mOC-iSVM.AP basée sur la sélection des SVs selon leur âge, est efficace sur des environnements non-stationnaires avec toujours un faible coût de calcul. Les deux dernières variantes, mOC-iSVM.EP et mOC-iSVM.nB sont plus gourmandes en temps de

calcul mais s'adaptent à tous les types d'environnements (stationnaires et non-stationnaires). Nous avons comparé leurs performances avec les meilleurs classificateurs dynamiques du moment : SEA, DWM, L++.NC, L++.NSE. Nous avons obtenu des résultats très prometteurs surpassant dans un grand nombre de cas les scores des classificateurs de la littérature. Nous avons également mis en évidence les forces et faiblesses de chacune de ces versions selon divers critères (temps de calcul, espace mémoire, robustesse du paramétrage, ...).

Alors que la plupart des autres classificateurs de la littérature sont spécialisés pour l'apprentissage dans des environnements soit stationnaires, soit non-stationnaires (sauf peut-être Learn++.NSE), notre proposition est capable d'appréhender différentes situations : stationnaire et non-stationnaire, à changement brusque ou progressif, avec ou sans réoccurrence.

Le chapitre 4 concerne l'application du modèle proposé dans un scénario d'usage lié à la numérisation en masse de documents par un scanner intelligent. L'intelligence du scanner est liée à la performance du modèle de classification embarqué qui, dans le cas présenté, doit être capable de reconnaître des catégories de documents au fur et à mesure de l'acquisition des images par le scanner. Différentes capacités du système (ajout de classes et évolution de concepts pour des environnements stationnaires et non stationnaires) sont testées sur une base d'images réelles utilisées dans le cadre du projet ANR DIGIDOC. Les interactions avec l'utilisateur sont fortement mises en avant dans cette partie car ce sont elles qui sont responsables de l'introduction des nouvelles connaissances et de la gestion du rejet. Ce dernier chapitre nous donne l'occasion de passer en revue les caractéristiques des scanners du marché afin de comparer leurs capacités actuelles avec les objectifs visés dans le cadre du projet DIGIDOC. Les travaux scientifiques en lien avec la problématique de la classification d'images de documents, sous-jacente à la mise au point de scanner intelligent, sont également abordés dans ce chapitre.

Les objectifs initiaux du projet DIGIDOC sont ensuite rappelés avant que l'on décrive en détail comment il est possible d'exploiter les modèles de classificateurs proposés dans les chapitres précédents afin de produire un prototype de « scanner intelligent ». Plusieurs scénarios permettant de valider l'intérêt des différentes versions de classificateurs mOC-iSVM, dans le cadre de la classification d'images de documents, sont proposés. De nombreuses illustrations graphiques permettent de visualiser les résultats obtenus, de les comparer avec des méthodes de classifications plus classiques (statiques) montrant que les approches dynamiques obtiennent la plupart du temps des résultats aussi bons que les

méthodes classiques sans pour autant nécessiter, à chaque étape, un réapprentissage complet des modèles.

Ces expérimentations prouvent que notre approche d'apprentissage continu peut répondre à des situations complexes apparaissant fréquemment en classification de documents arrivant par lots ou en flux. Cette classification peut être guidée par la nature propre du document mais aussi par les décisions de l'utilisateur. Ces situations complexes correspondent à des situations d'apprentissages continus très différentes : ajout, modification, extension, dérive, division, fusion de concepts, ... qui semblent pouvoir être appréhendées de manière correcte par l'approche proposée dans cette thèse.

Ce travail nous semble donc jeter les bases d'une méthodologie nouvelle pour la classification des documents et ouvre la voie à de nouvelles applications très importantes pour la gestion des grandes masses de données traitées en flux, par lot ou à l'unité.

## 5.2. Perspectives

Le travail présenté dans cette thèse aborde des problématiques scientifiques variées et largement étudiées depuis de nombreuses années par les chercheurs de la communauté du « Machine Learning ». Nous avons essayé d'apporter notre pierre à l'édifice en proposant une approche unifiée susceptible d'être utilisée dans diverses situations où l'apprentissage continu apporte un intérêt supplémentaire.

Il reste néanmoins de nombreux points à approfondir et des études complémentaires à mener afin de pouvoir valider définitivement nos propositions :

- Une des faiblesses de nos approches est liée aux grandes valeurs d'écart-type local remarquées lors des expérimentations en environnement non-stationnaire. Une voie possible d'amélioration pour limiter ce phénomène pourrait être la mise au point d'un mécanisme de renforcement des SVs et des nouveaux exemples pertinents à l'aide de poids. Ce type de solutions a été exploité par Rüping dans [Rüping 2001] qui propose une approche basée sur le contrôle des poids des échantillons participant à la procédure d'apprentissage (SVM binaire) pour améliorer la stabilité du système. Cette solution semble envisageable également pour améliorer la stabilité de notre iSVM *one-class* et mériterait donc d'être explorée.
- Grâce à l'indépendance des classificateurs dans la structure mOC-iSVM, on peut envisager de résoudre des problèmes de classification multi-étiquettes. Ce

problème est très souvent rencontré dans la réalité : un document peut être classé comme « document manuscrit » et aussi comme « document couleur ». Dans ce cas, l'espace des concepts peut comprendre des superpositions. On peut imaginer que chaque classificateur de notre mOC-iSVM focalise sur un aspect spécifique (par exemple, un classificateur pour « document couleur » et un autre classificateur pour « document manuscrit »). La décision peut ne pas être unique mais correspondre à l'acceptation d'une donnée par plusieurs classificateurs simultanément. Plus généralement, nous n'avons pas beaucoup étudié, durant ce travail, les stratégies de fusion de classificateurs ou de résultats de classification par exemple, en étudiant plus profondément les valeurs de décision renvoyées par les classificateurs. Cette étude mériterait également d'être menée

- Le choix d'utiliser plusieurs classificateurs *one-class* plutôt qu'un classificateur multi-classes unique est aussi lourd de conséquences et ouvre plusieurs perspectives :
  - Du fait de l'usage de classificateur *one-class*, le pouvoir discriminant de chacune des caractéristiques disponibles n'est pas facilement évaluable et exploitable dans le système proposé. Actuellement, il n'est même pas du tout utilisé et une réflexion pourrait donc avoir lieu sur la manière d'incorporer ce type d'information dans le système actuel et serait tout à fait pertinente.
  - A l'inverse, l'utilisation de *one-class* indépendant pour chacune des classes offre la possibilité à chaque classificateur d'exploiter un espace de caractéristiques différent des autres et adapté à la classe à identifier. Certaines caractéristiques peuvent être fondamentales pour identifier certaines classes et inintéressantes voire perturbantes pour d'autres : la couleur est par exemple très importante pour distinguer différentes classes de magazines tandis qu'elle est inutile pour distinguer l'imprimé du manuscrit. Une structure dans laquelle chaque classificateur travaille séparément sur un espace de caractéristiques différent est alors intéressant. Par contre, il devient nécessaire d'appliquer des méthodes de sélection des caractéristiques et d'avoir des moyens de stockage de méta-informations sur chacun des modèles de classificateurs.

- Le système proposé actuellement comporte également quelques lacunes venant du fait que certains de ces paramètres n'ont pas toujours pu être optimisés. Par exemple, notre mOC-iSVM est encore très sensible vis-à-vis du contenu de sa base d'apprentissage initiale. Ce problème est lié à la nature même des classificateurs SVM *one-class* (sensibles aux petites quantités de données d'apprentissage). Même si, grâce à la procédure incrémentale, il améliore sa performance au cours du temps, on aimerait qu'il soit performant au plus tôt. Une solution pourrait être d'utiliser des exemplaires négatifs (appartenant à d'autres classes) pour améliorer au plus vite les résultats. Nous n'avons, pour l'instant, pas eu le temps d'étudier l'effet de l'introduction d'exemples négatifs dans le mode « en ligne » avec très peu de données d'apprentissage.

Concernant les aspects plus applicatifs, nous aimerions également mener des expérimentations sur la capacité de notre système à déterminer la qualité (flou, bruité, incliné,...) des documents plutôt que leur type (manuscrit, imprimé, graphique, ...). Il s'agit ici d'une fonctionnalité importante pour la mise en place d'un scanner cognitif que nous n'avons pas eu le temps d'explorer, principalement à cause de la non-adéquation entre les caractéristiques disponibles actuellement et la capacité à détecter ce type de défauts (concepts).

Enfin, nous aimerions pouvoir appliquer et évaluer notre approche sur d'autres types d'applications nécessitant de classifier des flux de données et pour lesquelles l'utilisateur intervient au cours du temps. Un exemple typique concerne la reconnaissance de lettres manuscrites saisies en ligne sur dispositif mobile et pouvant nécessiter que le système apprenne, au fil du temps, le style d'écriture du scripteur.

# Bibliographie

[Aha 1991] D.W.Aha, D.Kibler, M.K.Albert ; Instance-Based Learning Algorithms ; *Machine Learning*, Pp.37-66 (1991).

[Alippi 2007] C.Alippi, M.Roveri ; Just-In-Time Adaptive Classifiers In Non-Stationary Conditions ; *In: Ijcn*, Pp.1014–1019, *Ieee, New York* (2007).

[Alippi 2008] C.Alippi, M.Roveri ; Just-In-Time Adaptive Classifierspart Ii: Designing The Classifier ; *Tnn19(12)*, Pp.2053–2064 (2008).

[Alippi 2009] C.Alippi, G.Boracchi, M.Roveri ; Just In Time Classifiers: Managing The Slowdrift Case ; *In: Ijcn*, Pp. 114–120. *Ieee, Newyork* (2009).

[Almaksour 2008] E.Anquetil, A.Almaksour, H.Mouchère ; Fast Online Incremental Learning With Few Examples For Online Handwritten Character Recognition ; *Proceedings Of The Eleventh International Conference On Frontiers In Handwriting Recognition (Icfhr'08)*, *Canada* (2008).

[Almaksour 2009] A.Almaksour, E.Anquetil ; Fast Incremental Learning Strategy Driven By Confusion Reject For Online Handwriting Recognition ; *Tenth International Conference On Document Analysis And Recognition (Icdar2009)*, *Spain* (2009).

[Almaksour 2010] A.Almaksour, E.Anquetil, S.Quiniou, M.Chretien ; Evolving Fuzzy Classifiers Application To Incremental Learning Of Handwritten Gesture Recognition ; *International Conference On Pattern Recognition (Icpr)*, *Istanbul : Turkey* (2010).

[Almaksour 2011] A.Almaksour, E.Anquetil ; Improving Premise Structure In Evolving Takagi-Sugeno Neuro-Fuzzy Classifiers ; *Evolving Systems*, *Vol. 2*, Pp. 25–33, (2011).

[Anagnostopoulos 2001] G.C.Anagnostopoulos, M.Georgiopoulos ; Ellipsoid Art And Artmap For Incremental Clustering And Classification ; *In Ijcn'01, International Joint Conference On Neural Networks*, *Volume: 2* (2001).

[Angelov 2004] P.Angelov, D.Filev; An Approach To Online Identification Of Takagi-Sugeno Fuzzy Models ; *Ieee Transactions On Systems, Man And Cybernetics - Part B: Cybernetics*, *34 (1)*. Pp. 484-498, (2004).

[Appiani 2001] E.Appiani, F.Cesarini, A.M.Colla, M.Diligenti, M.Gori, S.Marinai, G.Soda ; Automatic Document Classification And Indexing In High-Volume Applications ; *Int. J. Doc. Anal. Rec- Ognit.* *4(2)*, 69–83 (2001).

[Arce 2005] G.R.Arce ; Nonlinear Signal Processing: A Statistical Approach ; *Wiley:New Jersey, Usa* (2005).

[Arias-Castro 2009] E.Arias-Castro, D.L.Donoho ; Does Median Filtering Truly Preserve Edges Better Than Linear Filtering ? ; *Annals Of Statistics*, *Vol. 37, No. 3*, Pp. 1172 (2009).

[Augereau 2011] O.Augereau, N.Journet, J-P.Domenger ; Document Images Indexing With Relevance Feedback: An Application To Industrial Context ; *Icdar 2011, Pp.1190-1194, (2011)*

[Augereau 2013] O.Augereau ; Reconnaissance Et Classification D'images De Documents ; *Thesis, University Of Bordeaux I (2013)*.

[Augereau 2013b] O.Augereau, N.Journet, J-P.Domenger ; Semi-Structured Document Image Matching And Recognition ; *In DRR'13 (2013)*.

[Augereau 2014] O.Augereau, N.Journet, A.Vialard, J-P.Domenger ; Improving Classification Of An Industrial Document Image Database By Combining Visual And Textual Features ; *Document Analysis Systems 2014: Pp.314-318 (2014)*.

[Bagdanov 2001] A.D.Bagdanov, M.Worring ; Fine-Grained Document Genreclassification Using First Order Random Graphs ; *In: Proceed- Ings Of The 6th International Conference On Document Anal- Ysis And Recognition, Seattle, Usa, 10-13 September 2001, Pp. 79-90 (2001)*.

[Bagdanov 2003] A.D.Bagdanov, M.Worring ; First Order Gaussian Graphs For Efficient Structure Classification ; *Pattern Recognition 36(6),1311-1324 (2003)*.

[Baldi 2003] S.Baldi, S.Marinai, G.Soda ; Using Tree-Grammars For Training Set Expansion In Page Classification ; *In: Proceedings Of The 7th International Conference On Document Analysis And Rec- Ognition, Edinburgh, Scotland, 3-6 August 2003, Pp. 829-833 (2003)*.

[Baumann 1997] S.Baumann, M. Ali, A. Dengel, T.Jäger, M.Malburg, A.Weigel, C.Wenzel ; Message Extraction From Printed Documents – A Complete Solution ; *In: Proceedings Of The 4th International Conference On Document Analysis And Recognition, Ulm, Germany, 18-20 August 1997, Pp. 1055-1059 (1997)*.

[Bay 2008] H.Bay, A.Ess, T.Tuytelaars, L.V.Gool ; Surf: Speeded Up Robust Features ; *Computer Vision And Image Understanding (Cviu), Vol. 110, No. 3, Pp. 346-359 (2008)*.

[Bengio 1995] Y.Bengio, P.Frasconi ; An Input Output Hmm Architecture ; *In: Advances In Neural Information Processing Systems, Vol. 7, Pp. 427-434. Mit, Cambridge (1995)*.

[Bifet 2007] A.Bifet, R. Gavaldà ; Learning From Time-Changing Data With Adaptive Windowing ; *In: Sdm, Pp. 443-448 (2007)*.

[Bifet 2009] A.Bifet ; Adaptive Learning And Mining For Data Streams And Frequent Patterns ; *Phd Thesis, Universitat Politcnica De Catalunya (2009)*.

[Bifet 2010] A.Bifet ; Adaptive Stream Mining: Pattern Learning And Mining From Evolving Data Streams ; *Ios Press Inc (2010)*.

[Bishnoi 2014] A.Bishnoi ; Noise Removal With Morphological Operations Opening Andclosing Using Erosion And Dilation ; *International Journal Of Modern Engineering Research (Ijmer), Vol. 4, Iss. 5 (2014)*.

[Bouchachia 2011] A.Bouchachia ; Incremental Learning With Multi-Level Adaptation ; *Neurocomputing, Volume 74, Issue 11, May 2011, Pages 1785–1799 (2011)*.

[Bouguelia 2013] M.R.Bouguelia, Y.Belaid, A.Belaid ; Document Image And Zone Classification Through Incremental Learning ; *20th Ieee International Conference On Image Processing (Icip), 4230-4234 (2013)*.

[Bouguelia 2013b] M.R.Bouguelia, Y.Belaid, A.Belaïd ; A Stream-Based Semi-Supervised Active Learning Approach For Document Classification ; *Document Analysis And Recognition (Icdar), (2013)*.

[Bouillon 2012] M.Bouillon, Apprentissage incrémental et décremental Classification avec un système d'inférence floue évolutif appliquée à la reconnaissance de gestes manuscrits ; *Report of Master of INSA Rennes (2012)*.

[Bouillon 2013] M.Bouillon, E.Anquetil, A.Almaksour ; Decremental Learning of Evolving Fuzzy Inference Systems: Application to Handwritten Gesture Recognition ; *Machine Learning and Data Mining in Pattern Recognition Lecture Notes in Computer Science Volume 7988, 2013, pp 115-129 (2013)*.

[Boukharouba 2009] K.Boukharouba, L.Bako, S.Lecoeuche ; Incremental And Decremental Multi-Category Classification By Support Vector Machines ; *In Machine Learning And Applications, 2009. Icmla '09. International Conference On, Pages 294 –300 (2009)*.

[Breiman 1994] L.Breiman ; *Bagging Predictors ; Machine Learning 24 (2): 123–140 (1994)*.

[Breiman 1998] L.Breiman ; Arcing Classifiers ; *The Annals Of Statistics, 26(3):801–849, (1998)*.

[Breiman 1999] L.Breiman ; Pasting Small Votes For Classification In Large Databases And On-Line ; *Machine Learning, 36:85 103 (1999)*.

[Breuel 2003] T.M.Breuel ; An Algorithm For Finding Maximal Whitespace Rectangles At Arbitrary Orientations ; *International Conference On Document Analysis And Recognition, (2003)*.

[Brodersen 2010] K.H.Brodersen, C.S.Ong, K.E.Stephan, J.M.Buhmann ; The Balanced Accuracy And Its Posterior Distribution ; *Icpr'10, (2010)*.

[Brükner 1996] T.Brükner, P.Suda, H.Block, G.Maderlechner ; In-Housemail Distribution By Automatic Address And Content Interpretation ; *In: Proceedings Of The 5th Annual Symposium On Document Analysis And Information Retrieval, Las Vegas, Usa, April 1996, Pp. 67–75 (1996)*

[Bulacu 2007] M.L.Bulacu ; Statistical Pattern Recognition For Automatic Writer Identification And Verification ; *Phd Thesis, Artificial Intelligence Institute, University Of Groningen, The Netherlands (2007)*.

[Bunke 2000] H.Bunke ; Recent Developments In Graph Matching ; *In: Pro-Ceedings Of The 15th International Conference On Patter Recognition, Barcelona, Spain, 3–8 September 2000, Vol. 2, Pp. 2117–2124 (2000).*

[Byun 2000] Y.Byun, Y.Lee ; Form Classification Using Dp Matching ; *In:Proceedings Of The 2000 Acm Symposium On Applied Com- Puting, Como, Italy, 19–21 March 2000, Pp. 1–4 (2000).*

[Carpenter 1991] G.A.Carpenter, S.Grossberg, J.H.Reynolds ; Artmap: Super-Vised Real-Time Learning And Classification Of Nonstationary Data By A Self Organizing Neural Network ; *Neural Network, Vol. 4, No. 5, Pp. 565–588 (1991).*

[Carpenter 1992] G.A.Carpenter, S.Grossberg, N.Markuzon, J.H.Reynolds, D.B.Rosen ; Fuzzy Artmap: A Neural Network Architecture For Incremental Supervised Learning Of Analog Multidimensional Maps ; *Ieee Trans. Neural Netw., Vol. 3, Pp. 698–713, (1992).*

[Cauwenberghs 2001] G.Cauwenberghs, T.Poggio ; Incremental And Decremental Support Vector Machine Learning ; *In Nips 2000, Vol. 13 (2001).*

[Cavnar 1994] W.Cavnar, J.Trenkle ; N-Gram-Based Text Categorization ; *In:Proceedings Of The 3rd Annual Symposium On Document Analysis And Information Retrieval, Las Vegas, Usa, 1994, Pp. 161–175 (1994).*

[Cesarini 2001] F.Cesarini, M.Lastri, S.Marinai, G.Soda ; Encoding Of Modified X–Y Trees For Document Classification ; *In: Proceedings Of The 6th International Conference On Document Analysis And Recognition, Seattle, Usa, 10–13 September 2001, Pp. 1131–1136 (2001).*

[Chakraborty 2010] R.C.Chakraborty ; Adaptive Resonance Theory ; *Soft Computing Course Lecture, (2010).*

[Chang 2011] C-C Chang, C-J Lin ; Libsvm : A Library For Support Vector Machines ; *Acm Transactions On Intelligent Systems And Technology (2011).*

[Chen 2001] Y.Chen, X.S.Zhou, T.Huang ; One-Class SVM for Learning in Image Retrieval ; *Presentation for IEEE International Conference on Image Processing, ICIP'2001 (2001).*

[Chen 2002] C-Y Chen, S-C Hwang, Y-J Oyang ; An Incremental Hierarchical Data Clustering Algorithm Based On Gravity Theory ; *Proceedings Of The 6th Pacific-Asia Conference On Advances In Knowledge Discovery And Data Mining, Pp 237-250, London, Uk (2002)*

[Chen 2006] N.Chen, H.Shatkay, D.Blostein ; Exploring A New Space Of Features For Document Classification ; *In: Cascon'06 (2006).*

[Chen 2007] N.Chen, D. Blostein ; A Survey Of Document Image Classification: Problem Statement, Classifier Architecture And Performance Evaluation ; *International Journal On Document Analysis And Recognition, Vol. 10, No. 1, Pp. 1–16, (2007).*

[Cohen 2008] L.Cohen, G.Avrahami, M.Last, A.Kandel ; Info-Fuzzy Algorithms For Mining Dynamic Data Streams ; *Applied Soft Computing Vol 8, Pp.1283-1294 (2008).*

[Connolly 2008] J-F.Connolly, E.Granger, R.Sabourin ; Supervised Incremental Learning with the Fuzzy ARTMAP ; *Neural Network Artificial Neural Networks in Pattern Recognition Lecture Notes in Computer Science Volume 5064, 2008, pp 66-77.*

[Crawford 1999] S.L.Crawford ; Extensions To The Cart Algorithm ; *International Journal Of Man-Machine Studies 31, Pp.197-217 (1989).*

[Csurka 2004] G.Csurka, C.Dance, L.X.Fan, J.Willamowski, C.Bray ; Visual Categorization With Bags Of Keypoints ; *Proc. Of Eccv International Workshop On Statistical Learning In Computer Vision (2004).*

[Dengel 1994] A.Dengel, R.Bleisinger, F.Fein, R.Hoch, F.Hönes, M.Mal-Burg ; Officemaids – A System For Office Mail Analysis, Interpretation And Delivery ; *In: Proceedings Of International Association For Pattern Recognition Workshop On Document Analysis Systems, Kaiserslautern, Germany, Pp. 253–275 (1994).*

[Dengel 1995] A.Dengel, F.Dubiel ; Clustering And Classification Of Document Structure – A Machine Learning Approach ; *In: Proceedings Of The 3rd International Conference On Document Analysis And Recognition, Montreal, Canada, 14–15 August 1995, Pp. 587–591 (1995).*

[Dengel 1997] A.Dengel ; Bridging The Media Gap From The Gutenberg's world To Electronic Document Management Systems. *In: Proceedings Of 1997 Ieee International Conference On Systems, Man, And Cybernetics, Orlando, Florida, Usa, October 1997, Pp. 3540–3554 (1997).*

[Diligenti 2003] M.Diligenti, P.Frasconi, M.Gori ; Hidden Tree Markovmodels For Document Image Classification ; *Ieee Trans. Pat- Tern Anal. Mach. Intell. 25(4), 519–523 (2003).*

[Ditzler 2010] G.Ditzler, M. Muhlbaier, R. Polikar ; Incremental Learning Of New Classes In Unbalanced Data: Learn++.Udnc ; *In International Workshop On Multiple Classifier Systems, 2010, Lecture Notes In Computer Science, N. El. Gayer Et Al, Vol. 5997, Cairo, Egypt, Pp. 33–42. (2010).*

[Ditzler 2010b] G.Ditzler, R.Polikar, N.Chawla ; An Incremental Learning Algorithm For Nonstationary Environments And Imbalanced Data ; *International Conference On Pattern Recognition, 2010, Istanbul, Turkey, Pp. 2997–3000 (2010).*

[Ditzler 2010c] G.Ditzler, R. Polikar ; An Incremental Learning Framework For Concept Drift And Class Imbalance ; *In International Joint Conference On Neural Networks, 2010, Barcelona, Spain, Pp. 736-743 (2010).*

[Doermann 1998] D.Doermann, E.Rivlin, A.Rosenfeld ; The Function Of Documents. *Int. J. Comput ; Vision 16(11), 799–814 (1998).*

[Domingos 2000] P.Domingos, G.Hulten ; Mining High-Speed Data Streams ; *Proceedings Kdd 2000, Acm Press, New York, Ny, Usa, Pp. 71–80 (2000).*

[Draper 1998] N.R.Draper, H.Smith ; Applied Regression Analysis ; *Hoboken, Nj: Wiley-Interscience, (1998).*

[Drira 2006] F.Drira, F.Lebourgeois, H..Emptoz ; Restoring Ink Bleed-Through Degraded Document Images Using a Recursive Unsupervised Classification Technique ; *7th IAPR Workshop on Document Analysis Systems (DAS2006), Nelson, New Zealand (2006)*.

[Duda 1973] R.O.Duda, P.E.Hart ; Pattern Classification And Scene Analysis ; *New York: Wiley (1973)*.

[Duda 2001] R.O.Duda, P.E.Hart, D.Stork ; Pattern Classification ; *2nd Edn.Wiley, New York (2001)*.

[Eglin 2003] V.Eglin, S.Bres ; Document Page Similarity Based On Layoutvisual Saliency: Application To Query By Example And Document Classification ; *In: Proceedings Of The 7th International Conference On Document Analysis And Recognition, Edin- Burgh, Scotland, 3–6 August 2003, Pp. 1208–1212 (2003)*.

[Eglin 2004] V.Eglin, S.Bres ; Analysis And Interpretation Of Visualsaliency For Document Functional Labeling ; *Int. J. Doc. Anal. Recognit. 7(1), 28–43 (2004)*.

[Elwell 2009] R.Elwell, R.Polikar ; Incremental Learning In Nonstationary Environments With Controlled Forgetting ; *Int. Joint Conf. On Neural Networks (Ijcnnc 2009), Pp. 771–778, Atlanta, Ga (2009)*.

[Elwell 2011] R.Elwell, R.Polikar ; Incremental Learning Of Concept Drift In Non-Stationary Environments ; *Ieee Transactions On Neural Networks, 22(10):1517–1531. Issn 1045-9227 (2011)*.

[Erdem 2005] Z.Erdem, R.Polikar, F.Gurgen, N.Yumusak ; Ensemble Of Svms For Incremental Learning ; *Simulation, Vol. 3541, Springer, Pp. 246–256 Issn: 03029743 (2005)*.

[Esposito 2000] F.Esposito, D.Malerba, F.A.Lisi ; Machine Learning Forintelligent Processing Of Printed Documents ; *J. Intell. Inf. Syst.14(2–3), 175–198 (2000)*.

[Ester 1998] M.Ester, H.P.Kriegel, J.Sander, M.Wimmer, X.Xu ; Incremental Clustering For Mining In A Data Warehousing Environment ; *Proceedings Of 24th Vldb Conference, New York: Acm Sigmod, Pp. 323 – 333 (1998)*.

[Felhi 2011] M.Felhi, N.Bonnier, S.Tabbone ; A Robust Skew Detection Method Based On Maximum Gradient Difference And R-Signature ; *Pp.2617–2620, Icip 2011, Brussels, Belgium (2011)*.

[Freund 1995] Y.Freund, R.Shapire ; A Decision-Theoretic Generalization Of On-Line Learning And An Application To Boosting ; *Proceedings Of The Second European Conference On Computational Learning Theory, Pp. 23 (1995)*.

[Furao 2010] S.Furao, A.Sudo, O.Hasegawa ; An Online Incremental Learning Pattern-Based Reasoning System ; *Neural Networks Volume 23, Issue 1, January 2010, Pages 135–143 (2010)*.

[Gaceb 2008] D.Gaceb, V.Eglin, F.Lebourgeois, H.Emptoz ; Implication De La B-Coloration De Graphes Pour La Reconnaissance Automatique Du Type De Document ; *Colloque International Francophone Sur L'ecrit Et Le Document, France (2008)*.

[Gama 2006] J.Gama, R.Fernandes, R.Rocha ; Decision Trees For Mining Data Streams ; *Intelligent Data Analysis 10* 23-45(2006).

[Golub 1999] T.R.Golub, D.K.Slonim, P.Tamayo, C.Huard, M.Gaasenbeek, J.P.Mesirov, H.Coller, M.L.Loh, J.R.Downing, M.A.Caligiuri, C.D.Bloomfield, E.S.Lander ; Molecular Classification Of Cancer: Class Discovery And Class Prediction By Gene Expression Monitoring ; *Science*, 286, 531–537 (1999).

[Gordo 2009] A.Gordo, E.Valveny ; A Rotation Invariant Page Layout Descriptor For Document Classification And Retrieval ; *Icdar 2009: Pp.481-485 (2009)*.

[Gordo 2010] A.Gordo, F.Perronnin ; A Bag-Of-Pages Approach To Unordered Multi-Page Document Classification ; *Icpr 2010: Pp.1920-1923 (2010)*.

[Gordo 2012] A.Gordo, F.Perronnin, E.Valveny ; Document Classification Using Multiple Views ; *Document Analysis Systems 2012: Pp.33-37 (2012)*.

[Gordo 2013] A.Gordo, F.Perronnin, E.Valveny ; Large-Scale Document Image Retrieval And Classification With Runlength Histograms And Binary Embeddings ; *Pattern Recognition 46(7): 1898-1905 (2013)*.

[Gordo 2013b] A.Gordo, M.Rusiñol, D.Karatzas, A.D.Bagdanov ; Document Classification And Page Stream Segmentation For Digital Mailroom Applications ; *Icdar 2013: Pp.621-625 (2013)*.

[Guyon 2002] I.Guyon, J.Weston, S.Barnhill, V.Vapnik ; Gene Selection For Cancer Classification Using Support Vector Machines ; *Machine Learning 2002, Volume 46, Issue 1-3, Pp 389-422 (2002)*.

[Hamker 2000] F.H.Hamker ; Life-Long Learning Cell Structures - Continuously Learning Without Catastrophic Interference ; *Neural Netw., Vol. 14, No. 4, Pp. 551–573, (2000)*.

[Hamza 2008] H.Hamza, Y.Belaïd, A.Belaïd, B.B.Chaudhuri ; An End-To-End Administrative Document Analysis System ; *In Document Analysis Systems (Das), Pages 175–182 (2008)*.

[Haralick1994] R.Haralick ; Document Image Understanding: Geometric Andlogical Layout ; *In: Proceedings Of The Conference On Computer Vision And Pattern Recognition, Seattle, 20–24 June 1994, Pp. 385–390 (1994) ;*

[Hasab 2006] M.Hasan, F.Boris ; Svm: Machines À Vecteurs supports Ou Séparateurs À Vastes Marges ; *Bd Web, Isty 3, Survey (2006)*.

[Hasan 2006] M.Hasan, F.Boris ; SVM: Machines à Vecteurs de Support ou Séparateurs à Vastes Marges ; *Technical Report (2006)*.

[Heinke 1998] D.Heinke, F.H.Hamker ; Comparing Neural Networks: A Benchmark On Growing Neural Gas, Growing Cell Structures, And Fuzzy Artmap ; *Ieee Trans. Neural Netw., Vol. 9, No. 6, Pp. 1279–1291, (1998)*.

[Héroux 1998] P.Héroux, S.Diana, A.Ribert, E.Trupin ; Classificationmethod Study For Automatic Form Class Identification ; *In: Pro- Ceedings Of The 14th International Conference On Pattern Rec- Ognition, Brisbane, Australia, 16–20 August 1998, Pp. 926–929 (1998).*

[Ho 2002] T.K.Ho, A.Kandel, H.Bunke ; Multiple Classifier Combination: Lessons And Next Steps ; *In: Hybrid Methods In Patter Recognition. World Scientific, Singapore, Pp. 171–198 (2002).*

[Hoch 1994] R.Hoch ; Using Ir Techniques For Text Classification Indocument Analysis ; *In: Proceedings Of The 17th International Acm-Sigir Conference On Research And Development In Information Retrieval, Dublin, Ireland, July 1994, Pp. 31–40 (1994).*

[Hoens 2012] T.R.Hoens, R.Polikar, N.V.Chawla ; Learning from streaming data with concept drift and imbalance: an overview ; *Progress in Artificial Intelligence, April 2012, Volume 1, Issue 1, pp 89-101 (2012).*

[Hsu 2010] C-W.Hsu, C-C.Chang, C-J.Lin ; A Practical Guide To Support Vector Classification ; *Department Of Computer Science. National Taiwan University (2010).*

[Hu 1999] J.Hu, R.Kashi, G.Wilfong ; Document Classification Using Layout Analysis ; *In: Proceedings Of The 1st International Workshop On Document Analysis And Understanding For Document Databases, Florence, Italy, Pp. 556–560 (1999).*

[Huang 1990] X.D.Huang, Y.Ariki, M.A.Jack ; Hidden Markov Models For Speech Recognition ; *Edinburgh University Press, Edinburgh (1990).*

[Hulten 2001] G.Hulten, L.Spencer, P.Domingos ; Mining Time-Changing Data Streams ; *In: Kdd, Pp. 97–106. Acm, New York (2001).*

[Hunt 1996] K.J.Hunt, R.Haas, R.Murray-Smith ; Extending the functional equivalence of radial basis function networks and fuzzy inference systems ; *IEEE Trans Neural Netw. 1996;7(3):776-81.(1996).*

[Ittner 1995] D.J.Ittner, D.D.Lewis, D.D.Ahn ; Text Categorization Of Low Quality Images ; *In: Proceedings Of The 4th Annual Symposium On Document Analysis And Information Retrieval, Las Vegas, Usa, 1995, Pp. 301–315 (1995).*

[Jain 200] A.K.Jain, P.W.Duin, J.Mao ; Statistical Pattern Recognition: A Review ; *Ieee Trans. Pattern Anal. Mach. Intell. 22(1), 4–37 (2000).*

[Jang 1993] J-S.R.Jang ; Anfis : Adaptive-Network-Based Fuzzy Inference System ; *Systems, Man And Cybernetics, Ieee Transactions (1993).*

[Junker 1997] M.Junker, R.Hoch ; Evaluating Ocr And Non-Ocr Text Representation For Learning Document Classifiers ; *In: Proceed- Ings Of The 4th International Conference On Document Anal- Ysis And Recognition, Ulm, Germany, 18–20 August 1997, Pp. 1060–1066 (1997).*

[Kapp 2010] M.N.Kapp, R.Sabourin, P.Maupin ; Adaptive Incremental Learning With An Ensemble Of Support Vector Machines ; *Pattern Recognition (Icpr), 2010 20th International Conference, Page(S): 4048- 4051, (2010).*

[Karasuyama 2010] M.Karasuyama, I.Takeuchi ; Multiple Incremental Decremental Learning Of Support Vector Machines ; *Ieee Trans. On Neural Networks, Vol. 21 Issue 7, Pp. 1048-1059, (2010).*

[Karnick 2008] M.Karnick, M.Muhlbaier, R.Polikar ; Incremental Learning In Non-Stationary Environments With Concept Drift Using A Multiple Classifier Based Approach ; *International Conference On Pattern Recognition (Icpr2008), Tampa, Fl, (2008).*

[Karnick 2008b] M.Karnick, Metinahiskali, M.D.Muhlbaier, R.Polikar ; Learning Concept Drift In Nonstationary Environments Using An Ensemble Of Classifiers Based Approach ; *In Ieee International Joint Conference On Neural Networks (Ieee World Congress On Computational Intelligence), Ijcn '08, Pages 3455-3462, (2008).*

[Kasabov 2001] N.Kasabov ; Evolving fuzzy neural networks for supervised/unsupervised on-line knowledge-based learning. (2001).

[Kasabov 2002] N.K.Kasabov, Q.Song ; Denfis: Dynamic Evolving Neural-Fuzzy Inference System And Its Application For Time-Series Prediction ; *Ieee T. Fuzzy Systems 10(2): 144-154 (2002).*

[Kaynak 1995] C.Kaynak ; 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).*

[Keysers 2007] D.Keysers, F.Shafait, T.M. Breuel ; Document Image Zone Classification - A Simple High-Performance Approach ; *Visapp (2) 2007: 44-51(2007).*

[Klinkenberg 2000] R.Klinkenberg, T.Joachims ; Detecting Concept Drift with Support Vector Machines ; *In Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00), (2000)*

[Klinkenberg 2004] R.Klinkenberg ; Learning Drifting Concepts: Example Selection Vs. Example Weighting ; *Intelligent Data Analysis, Special Issue On Incremental Learning Systems Capable Of Dealing With Concept Drift, 8 (3), (2004).*

[Kochi 1999] T.Kochi, T.Saitoh ; User-Defined Template For Identifying Document Type And Extracting Information From Documents ; *In: Proceedings Of The 5th International Conference On Document Analysis And Recognition, Bangalore, India, 20-22 September 1999, Pp. 127-130 (1999).*

[Kolter 2007] J.Kolter, M.Maloof ; Dynamic weighted majority: an ensemble method for drifting concepts ; *JMLR8, 2755-2790 (2007).*

[Kopec 1994] G.E.Kopec, P.A.Chou ; Document Image Decoding Using Markov Source Models ; *Ieee Trans. Pattern Anal. Mach. In- Tell. 16(6), 602-617 (1994).*

[Kubat 1989] M.Kubat ; Floating Approximation In Time-Varying Knowledge Bases ; *Pr110(4), 223-227 (1989).*

[Kubat 1994] M.Kubat, G.Widmer ; Adapting To Drift In Continuous Domains ; *Tech. Report Öfai-Tr-94-27, Austrian Research Institute For Artificial Intelligence, Vienna, (1994).*

[Kuncheva 2004] L.Kuncheva ; Classifier Ensembles for Changing Environments ; *Proc. 5th Int. Workshop on Multiple Classifier Systems, Cagliari, Italy, Springer-Verlag, LNCS, 3077, (2004).*

[Lam 1994] S.Lam ; An Adaptive Approach To Document Classification And Understanding ; *In: Proceedings Of International Association For Pattern Recognition Workshop On Document Analysis Systems, Kaiserslautern, Germany, October 1994, Pp. 231–251 (1994).*

[Laroum 2009] S.Laroum, N.Béchet, H.Hamza, M.Roche ; Classification Automatique De Documents Bruités À Faible Contenu Textuel ; *Rnti : Revue Des Nouvelles Technologies De L'information, 1:25, (2009).*

[Laskov 2006] P.Laskov, C.Gehl, S.Kruger, K.-R.Muller ; Incremental Support Vector Learning: Analysis, Implementation And Applications ; *Journal Of Machine Learning Research, Vol. 7, Pp. 1909–1936, (2006).*

[Last 2002] M.Last ; Online Classification Of Nonstationary Data Streams ; *Intell. Data Anal. 6(2) 129–147(2002).*

[Laviola 2007] Jr.Laviola, J.Joseph, R.C.Zeleznik ; A Practical Approach For Writer-Dependent Symbol Recognition Using A Writer-Independent Symbol Recognizer ; *Journal Ieee Transactions On Pattern Analysis And Machine Intelligence Archive Volume 29 Issue 11, Pages 1917-1926 (2007).*

[Lazarescu 2004] M.Lazarescu, S.Venkatesh, H.Bui ; Using Multiple Windows To Track Concept Drift ; *Ida8(1), Pp.29–59 (2003).*

[Lemaitre 2007] A.Lemaitre, J.Camillerapp, B.Couasnon ; Contribution Of Multiresolution Description For Archive Document Structure Recognition ; *In: Proceedings Of International Conference On Document Analysis And Recognition (Icdar'07), Pp. 247–251 (2007).*

[Leung 2001] T.Leung, J.Malik ; Representing And Recognizing The Visual Appearance Of Materials Using Three-Dimensional Textons ; *Ijcv43(2001).*

[Liang 2002] J.Liang, D.Doermann, M.Ma, J.K.Guo ; Page Classification Through Logical Labelling ; *In: Proceedings Of The 16th International Conference On Pattern Recognition, Quebec, Canada, 11–15 August 2002, Pp. 477–480 (2002).*

[Liang 2010] C.Liang, Y.Zhang, Q.Song ; Decision Tree For Dynamic And Uncertain Data Streams ; *Journal Of Machine Learning Research - Proceedings Track 13: 209-224 (2010).*

[Lim 1990] J.S.Lim ; Two-Dimensional Signal And Image Processing ; *Englewood Cliffs, Nj, Prentice Hall, P. 548 (1990).*

[Lim 1997] C.P.Lim, R.F.Harrison ; An Incremental Adaptive Network For On-Line Supervised Learning And Probability Estimation ; *Neural Netw., Vol. 10, No. 5, Pp. 925–939, Jul. (1997)*

[Lin 2008] M.Lin, Hefeiketang, X.Yao; Selective Negative Correlation Learning Algorithm For Incremental Learning ; *Neural Networks, Ijcn 2008, Ieee World Congress On Computational Intelligence (2008)*.

[Littlestone 1988] N.Littlestone ; Learning Quickly When Irrelevant Attributes Abound: A New Linear Threshold Algorithm ; *Machine Learning*, 2:285-318, (1988).

[Littlestone 1994] N.Littlestone, M.K.Warmuth ; The Weighted Majority Algorithm ; *Inform. Computation*, 108:212-261 (1994).

[Liu 1998] Y.Liu ; Negative Correlation Learning And Evolutionary Neural Network Ensembles ; *Phd Thesis, University College, The University Ofnew South Wales, Australian Defence Force Academy, Canberra, Australia, (1998)*.

[Liu 2004] Y.G.Liu, Q.M.He, Q.Chen ; Incremental Batch Learning With Support Vector Machines ; *In: Proceedings Of The 5th World Congress On Intelligence Control And Automation, Hangzhou, P.R. China, Pp. 1857-1861 (2004)*.

[Louloudis 2006] G.Louloudis, B.Gatos, I.Pratikakis, K.Halatsis ; A Block-Based Hough Transform Map-Ping For Text Line Detection In Handwritten Documents ; *Tenth International Workshop On Frontiers In Handwriting Recognition, La Baule, France (2006)*.

[Lowe 1999] D.G.Lowe ; Object Recognition From Local Scale-Invariant Features ; *Proceedings Of The International Conference On Computer Vision 2. Pp. 1150–1157 (1999)*.

[Lu 2006] S.Lu, C.L.Tan ; Script And Language Identification In Degraded And Distorted Document Images ; *Proceedings Of The 21th National Conference On Artificial Intelligence (Aaai), (2006)*.

[Lughofer 2008] E.Lughofer ; Flexfis : A robust incremental learning approach for evolving takagi-sugeno fuzzy models ; *Fuzzy Systems, IEEE Transactions on*, 16(6) :1393 –1410 (2008).

[Lughofer 2011] E.Lughofer, P.Angelov ; Handling drifts and shifts in on-line data streams with evolving fuzzy system ; *Appl. Soft Comput.*, 11(2) :2057–2068 (2011).

[Maderlechner 1997] G.Maderlechner, P.Suda, T.Brückner ; Classification Of Documents By Form And Content ; *Pattern Recognit. Lett.* 18(11–13), 1225–1231 (1997).

[Manevitz 2001] L.M.Manevitz, M.Yousef ; One-Class SVMs for Document Classification ; *Journal of Machine Learning. Res.* 2: 139-154 (2001).

[Mao 2003] S.Mao, A.Rosenfeld, T.Kanungo ; Document Structure Analysis Algorithms: A Literature Survey ; *In: Proceedings Of Document Recognition And Retrieval X (Is&T/Spie Elec- Tronic Imaging), Santa Clara, California, Usa, Pp 197–207 (2003)*.

[Marszaek 2006] M.Marszaek, C.Schmid ; Spatial Weighting For Bag-Of-Features ; *Computer Vision And Pattern Recognition (Vol 2), Pp2118 - 2125, (2006)*.

[Michalski 1973] R.S.Michalski ; Discovering Classification Rules Using Variable-Valued Logic System V11 ; *Proceedings Of The Third International Joint Conference On Artificial Intelligence (Pp. 162-172) Stanford, Ca: Morgan Kaufmann (1973)*.

[Minku 2009] L.Minku, F.Li, H.Inoue, X.Yao ; Negative Correlation In Incremental Learning ; *Journal Natural Computing: An International Journal Archive* volume 8 Issue 2, June 2009 Pages 289 – 320 Kluwer Academic Publishers Hingham, Ma, Usa (2009).

[Mouchère 2005] H.Mouchère, E.Anquetil, A Unified Strategy to Deal with Different Natures of Reject ; Proceeding ICPR '06 Proceedings of the 18th International Conference on Pattern Recognition - Volume 02, Pages 792-795 IEEE Computer Society Washington, DC, USA ©2006

[Mouchère 2005] H.Mouchère, E.Anquetil, N.Ragot ; On-Line Writer Adaptation For Handwriting Recognition Using Fuzzy Inference Systems ; *8th International Conference On Document Analysis And Recognition - Icdar'05, (2005).*

[Muhlbaier 2007] M.Muhlbaier, R.Polikar ; An Ensemble Approach For Incremental Learning In Nonstationary Environments ; *In Multiple Classifier Systems, Volume 4472 Of Lecture Notes In Computer Science, Pages 490-500 (2007).*

[Muhlbaier 2007b] Muhlbaier M., Polikar R ; Multiple Classifiers Based Incremental Learning Algorithm For Learning Nonstationary Environments ; *Ieee International Conference On Machine Learning And Cybernetics (Icmlc 2007), Pp. 3618-3623, Hong Kong, China, August 2007.*

[Muhlbaier 2009] M.Muhlbaier, A.Topalis, R.Polikar ; Learn++.Nc: Combining Ensemble Of Classifiers Combined With Dynamically Weighted Consult-And-Vote For Efficient Incremental Learning Of New Classes ; *Ieee Trans On Neural Networks, Vol. 20, No. 1, Pp. 152 – 168, (2009).*

[Nagy 1984] G.Nagy, S.Seth ; Hierarchical Representation Of Optically Scanned Documents ; *In: Proceedings Of The 7th International Conference On Pattern Recognition, Los Alamitos, California, Usa, 1984, Pp. 347–349 (1984).*

[Nagy 2000] G.Nagy ; Twenty Years Of Document Image Analysis In Pami ; *Ieee Tran. Pattern Anal. Mach. Intell. 22(1), 38–62 (2000).*

[Nattee 2001] C.Nattee, M.Numao ; Geometric Method For Document Understanding And Classification Using On-Line Machine Learning ; *In: Proceedings Of The 6th International Conference On Document Analysis And Recognition, Seattle, Usa, Pp. 602–606 (2001).*

[Ogata 2003] H.Ogata, S.Watanabe, A.Imaizumi, T.Yasue, N.Furukawa, H.Sako, H.Fujisawa ; Form Type Identification For Banking Applications And Its Implementation Issues ; *Proc. Spie 5010, Document Recognition And Retrieval X, (2003).*

[Okun 1999] O.Okun, D.Doermann, M.Pietikäinen ; Page Segmentation And Zone Classification: The State Of The Art ; *Technical Report, Lamp-Tr-036, University Of Maryland, College Park (1999).*

[Oza 2001] N.C.Oza ; Online Ensemble Learning ; *Phd Thesis, University Of California, Berke-Ley (2001).*

[Ozawa 2004] S.Ozawa, S.Pang, N.Kasabov ; A Modified Incremental Principal Component Analysis For On-Line Learning Of Feature Space And Classifier ; *In Pricai 2004: The Pacific Rim International Conferences On Artificial Intelligence (Pricai2004-Auckland, Nz) (2004)*.

[Ozawa 2008] S.Ozawa, S.Pang, N.Kasabov ; Incremental Learning Of Chunk Data For Online Pattern Classification Systems ; *Ieee Trans. Neural Netw., Vol. 19, No. 6, Pp. 1061–1074, Jun. 2008*.

[Pang 2004] S.Pang, S.Ozawa, N.Kasabov ; One-Pass Incremental Membership Authentication By Face Classification ; *In Biometric Authentication, Lecture Notes In Computer Science, Springer-Verlag, Pp. 155-161 (2004)*.

[Pang 2009] S.Pang, S.Ozawa, N.Kasabov ; Curiosity Driven Incremental Lda Agent Active Learning ; *Proc. Int. Joint Conf. On Neural Networks 2009 (Ijcnnc2009-Atlanta, Ga), Pp. 2401-2408 (2009)*.

[Pang 2010] S.Pang, T.Ban, Y.Kadobayashi, N.Kasabov ; Incremental And Decremental Lda Learning With Applications ; *Ijcnnc 2010:Pp.1-8(2010)*.

[Pang 2012] S.Pang, T.Ban, Y.Kadobayashi, N.Kasabov ; Lda Merging And Splitting With Applications To Multiagent Cooperative Learning And System Alteration ; *Ieee Transactions On Systems, Man, And Cybernetics, Part B 42(2): 552-564 (2012)*.

[Paradis 2007] F.Paradis, J.Y.Nie ; Contextual Feature Selection For Text Classification ; *Information Processing & Management, 43(2):344–352, (2007)*.

[Pavlidis 1980] T.Pavlidis ; Structural Pattern Recognition ; *2nd Edn. Springer, Berlin Heidelberg New York (1980)*.

[Phillips 1993] I.T.Phillips, S.Chen, R.Haralick ; Cd-Rom Document Database Standard ; *In: Proceedings Of The 2nd International Conference On Document Analysis And Recognition, Tsukuba, Japan, 20–22 October 1993, Pp. 478–483 (1993)*.

[Platt 2000] J.Platt ; Probabilistic Outputs For Support Vector Machines And Comparison To Regularized Likelihood Methods ; *Advances In Large Margin Classifiers. Cambridge: Mit Press. Pages 61–74. (2000)*.

[Polikar 2001] R.Polikar, L.Udpa, S.Udpa, V.Honavar ; An Incremental Learning Algorithm For Supervised Neural Networks ; *Ieee Transactions On System, Man And Cybernetics (C), Special Issue On Knowledge Management. (2001)*.

[Polikar 2010] R.Polikar, J.DePasquale, H.Syed-Mohammed, G.Brown, L.I.Kuncheva ; Learn++.MF: A Random Subspace Approach for the Missing Feature Problem ; *Pattern Recognition, vol. 43, no. 11, pp. 3817-3832, 2010*.

[Prudent 2005] Y.Prudent, A.Ennaji ; A New Learning Algorithm For Incremental Self-Organizing Maps ; *Esann 2005, 13th European Symposium On Artificial Neural Networks, Bruges, Belgium, April 27-29, 2005, Proceedings (2005)*.

[Quinlan 1993] R.Quinlan ; C4.5: Programs For Machine Learning ; *Morgankaufmann Publishers, San Mateo, Ca (1993)*.

[Ralaivola 2001] L.Ralaivola, F.D'alché-Buc ; Incremental Support Vector Machine Learning : A Local Approach ; *In Proc. Of Iccan'01 (2001)*.

[Ribert 1999] A.Ribert, A.Ennaji, Y.Lecourtier ; An Incremental Hierarchical Clustering ; *Vision Interface '99, Trios- Rivieres, Canada, 19-21 May (1999)*.

[Rizzi 2004] A. Rizzi, T. Algeri, G. Medeghini, D. Marini ; A Proposal For Contrast Measure In Digital Images ; *In Cgiv 2004 – Second European Conference On Color In Graphics, Imaging And Vision, (2004)*.

[Rizzi 2008] A.Rizzi, G.Simone, R.Cordone ; A Modified Algorithm For Perceived Contrast Measure In Digital Images ; *In: Cgiv 2008 And Mcs '08 (2008)*.

[Rüping 2001] S.Rüping ; Incremental Learning With Support Vector Machines ; *Icdm '01, Pp.641, (2001)*.

[Rusinol 2012] M.Rusinol, D.Karatzas, A.D.Bagdanov, J.Lladós ; Multipage Document Retrieval By Textual And Visual Representations ; *Icpr 2012 (2012)*.

[Sadri 2006] J.Sadri, C.Y.Suen, T.D.Bui ; A New Clustering Method For Improving Plasticity And Stability In Handwritten Character Recognition Systems ; *Icpr '06 Proceedings Of The 18th International Conference On Pattern Recognition - Volume 02 Pages 1130 – 1133b(2006)*.

[Sahoo 2006] N.Sahoo, G.Duncan, Incremental Hierarchical Clustering Of Text Documents ; *Cikm '06 Proceedings Of The 15th Acm International Conference On Information And Knowledge Management, Pages 357-366, Acm New York, Ny, Usa (2006)*.

[Sakhi 2012] O.B.Sakhi ; Segmentation of heterogeneous document images : an approach based on machine learning, connected components analysis, and texture analysis ; *PhD Thesis, University of Paris-Est (2012)*.

[Sako 2003] H.Sako, M.Seki, N.Furukawa, H.Ikeda, A.Imaizumi ; Form Reading Based On Form-Type Identification And Form-Data Recognition ; *In: Proceedings Of The 7th International Conference On Document Analysis And Recognition, Edinburgh, Scotland, 3–6 August 2003, Pp. 926–930 (2003)*.

[Salganicoff 1997] M.Salganicoff ; Tolerating Concept And Sampling Shift In Lazy Learning Using Prediction Error Context Switching ; *Ai Review, Special Issue On Lazy Learning, 11 (1-5), 1997, 133-155 (1997)*.

[Sarkar 2006] P.Sarkar ; Image Classification: Classifying Distributions Of Visual Features ; *Icpr '06, 18th International Conference On Pattern Recognition, Volume:2 (2006)*.

[Sato 2012] J.R.Sato, R.Jane, M-M Janaina ; Measuring Abnormal Brains: Building Normative Rules In Neuroimaging Using one-class Support Vector Machines ; *Frontiers In Neuroscience, Volume 6 (2012)*.

[Saund 2011] E.Saund ; Scientific Challenges Underlying Production Document Processing ; *Proceedings Of Document Recognition And Retrieval Xviii, Vol. 7874, P. 787402, (2011)*.

[Sauvola 1999] J.Sauvola, H.Kauniskangas ; Mediateam Document Database ; ([Http://Www.Mediateam.Oulu.Fi/Mtdb/](http://www.Mediateam.Oulu.Fi/Mtdb/)), Oulu Univer- Sity, Finland (1999).

[Schenker 2003] A.Schenker, M.Last, H.Bunke, A.Kandel ; Classification Of Web Documents Using A Graph Model ; *In: Proceedings Of The 7th International Conference On Document Analysis And Recognition, Edinburgh, Scotland, 3–6 August 2003, Pp. 240–244 (2003).*

[Schlimmer 1986] J.C.Schlimmer, D.Fisher ; A Case Study Of Incremental Concept Induction ; *Proceedings Of The Fifth National Conference On Artificial Intelligence (Pp. 496-501). Philadelphia, Pa: Morgan Kaufmann (1986).*

[Schlimmer 1986] J.C.Schlimmer, R.H.Jr.Granger ; Incremental Learning From Noisy Data ; *Machine Learning 1, 317-354 (1986).*

[Sebastiani 2002] F.Sebastiani ; Machine Learning In Automated Text Categorization ; *Acm Computing Surveys (Csur), 34(1):1–4 (2002).*

[Sergyan 2008] S.Sergyan ; A Survey Of Document Image Classification: Problem Statement, Classifier Architecture And Performance Evaluation ; *Applied Machine Intelligence And Informatics, Sami'08, (2008).*

[Serre 2005] T.Serre, L.Wolf, T.Poggio ; Object Recognition With Features Inspired By Visual Cortex ; *Ieee Computer Society Conference On Computer Vision And Pattern Recognition (Cvpr'05) 2. P. 994 (2005).*

[Servedio 2002] R.Servedio ; Pac Analogues Of Perceptron And Winnow Via Boosting The Margin ; *Mach. Learn., Vol. 47, No. 2–3, Pp. 133–151, (2002).*

[Shafait 2009] F.Shafait, D.Keysers, T.M.Breuel ; *Response To Projection Methods Require Black Border Removal ; Ieee Trans. Pattern Anal. Mach. Intell. 31(4): 763-764 (2009).*

[Shilton 2005] A.Shilton, M.Palaniswami, D.Ralph, A.C.Tsoi ; Incremental Training Of Support Vector Machines ; *Ieee Trans. On Neural Networks, Vol. 16 Issue 1, Pp. 114-131, (2005).*

[Shimotsuji 1996] S.Shimotsuji, M.Asano ; Form Identification Based On Cell Structure ; *In: Proceedings Of The 13th International Conference On Pattern Recognition, Vienna, Austria, August 1996, Vol. C, Pp. 793–797 (1996).*

[Shin 2001] C.Shin, D.Doermann, A.Rosenfeld ; Classification Of Document Pages Using Structure-Based Features ; *International Journal On Document Analysis And Recognition, Vol. 3, No. 4, Pp. 232–247, (2001).*

[Shölkopf 1999] B.Scolkopf, J.C.Platt, J.Shawe-Taylor, A.J.Smola, R.C.Williamson ; Estimating the support of a high-dimensional distribution ; *Technical report, Microsoft Research, MSR-TR-99-87, 1999.*

[Sivic 2005] J.Sivic, B.C.Russell, A.A.Efros, A. Zisserman, W.T.Freeman ; Discovering Objects And Their Location In Images ; *Tenth Ieee International Conference On Computer Vision (Iccv'05) Volume 1. P. 370 (2005).*

[Smith 2004] E.B.Smith, D.Monn, H.Veeramachaneni, K.Kise, A.Malizia, L.Todoran, A.El-Nasan, R. Ingold ; Reports Of The Das02 Working Group ; *Int. J. Doc. Anal. Recognit.* 6(3),211–217 (2004).

[Sollich 1999] P.Sollich, Probabilistic Interpretation And Bayesian Methods For Support Vector Machines ; *Machine Learning 20020102, Volume 46, Issue 1-3, Pp 21-52 (1999).*

[Song 2001] Q.Song, N.Kasabov ; A Novel On-Line, Evolving Clustering Method And Its Applications ; *Artificial Neural Networks And Expert Systems - Annes, (2001).*

[Spitz 2000] A.L.Spitz, A.Maghbouleh ; Text Categorization Using Character Shape Codes ; *In: Proceedings Of Document Recognition And Retrieval Vii (Is&T/Spie Electronic Imaging), San Jose, California, Spie Proceedings Series 3967, 174–181 (2000).*

[Street 2001] W.N.Street, Y.S.Kim A Streaming Ensemble Algorithm (Sea) For Large-Scale Classification ; *In Proceedings Of The 7th Acm Sigkdd International Conference On Knowle Dge Discovery And Data Mining, Pages 377-382. Acm Press, (2001).*

[Su 2006] M.C.Su, J.Lee, K.L.Hsieh ; A New Artmap-Based Neural Net-Work For Incremental Learning ; *Neurocomputing, Vol. 69, No. 16–18, Pp. 2284–2300, (2006).*

[Syed 1999] N.Syed, H.Liu, K.Sung ; Incremental Learning With Support Vector Machines ; *In Proc. Of The Workshop On Support Vector Machines Ijcai-99, Stockholm, Sweden, (1999).*

[Tadmor 2000] Y.Tadmor, D.J.Tolhurst ; Calculating The Contrasts That Retinal Ganglion Cells And Lgn Neurons Encounter In Natural Scenes ; *Vision Research, 40:3145–3157, (2000).*

[Tang 1998] Y.Y.Tang, M.Chriet, J.Liu, J.N.Said, C.Y.Suen ; Document Analysis And Recognition By Computers ; *In: Hand-Book Of Pattern Recognition And Computer Vision, 2nd Edn. World Scientific, Singapore, Pp. 579–612 (1998).*

[Taylor 1995] S.Taylor, M.Lipshutz, R.Nilson ; Classification And Functional Decomposition Of Business Documents ; *In: Proceedings Of The 3rd International Conference On Document Analisis And Recognition, Montreal, Canada, 14–15 August 1995, Pp. 563–566 (1995).*

[Ting 1996] A.Ting, M.Leung ; Business Form Classification Using Strings ; *In: Proceedings Of The 13th International Conference On Pattern Recognition, Vienna, Austria, August 1996, Vol. B, Pp. 690–694 (1996).*

[Tontini 1998] G.Tontini ; Robust Learning And Identification Of Patterns In Statistical Process Control Charts Using A Hybrid Rbf Fuzzy Artmap Neural Network ; *Inproc. Int. Joint Conf. Neural Netw., Vol. 3, Pp. 1694–1699 (1998).*

[Trier 1996] D.Trier, A.K.Jain, T.Taxt ; Feature Extraction Methods For Character Recognition – A Survey ; *Pattern Recognit.* 29(4), 641–662 (1996).

[Trinh 2003] E.Trinh ; De la numérisation à la consultation de documents anciens ; *Thèse de doctorat en Informatique, Insa de Lyon. (2003).*

[Tsymbal 2004] A.Tsymbal Technical Report: Theproblem Of Concept Drift: Definitions And Related Work ; *Trinitycollege, Dublin, Ireland, Tcd-Cs-2004-15, (2004).*

[Utgoff 1988] P.Utgoff ; Id5: An Incremental Id3 ; *Fifth International Conference On Machine Learning, Pp. 107-120. Morgan Kaufmann Publishers (1988).*

[Utgoff 1997] P.Utgoff, N.C.Berkman, J.A.Clouse ; Decision Tree Induction Based On Efficient Tree Restructuring ; *Machine Learning 29, 5-44 (1997).*

[Vapnik 1995] V.Vapnik, C.Cortes; Support-vector networks; *Machine Learning 20 (3): 273. doi:10.1007/BF00994018 (1995).*

[Vapnik 1998] V.N.Vapnik ; Statistical Learning Theory ; *Wiley (1998).*

[Verzi 2001] S.J.Verzi, G.L.Heileman, M.Georgiopoulos, M.J.Healy ; Rademacher Penalization Applied To Fuzzy Artmap And Boosted Artmap ; *Inproc. Int. Joint Conf. Neural Netw., Vol. 2, 2001, Pp. 1191–1196 (2001).*

[Viola 2001] P.A.Viola And M.J.Jones ; Rapid Object Detection Using A Boosted Cascade Of Simple Features ; *In Proc. Cvpr, Pages 511–518 (2001).*

[Vlachos 2004] A.Vlachos ; Active Learning With Support Vector Machines ; *Thesis Of University Of Edinburgh, (2004).*

[Vuori 2001] V.Vuori, J.Laaksonen, E.Oja, J.Kangas ; Experiments With Adaptation Strategies For A Prototype-Based Recognition System For Isolated Handwritten Characters ; *International Journal On Document Analysis And Recognition, Vol. 3, Pp. 150–159, (2001).*

[Wang 2002] Y.Wang, I.T.Phillips, R.Haralick ; A Study On The Docu- Ment Zone Content Classification Problem ; *In: Proceedings Of The 5th International Workshop On Document Analysis Sys- Tems, Princeton, Nj, Usa, 19–21 August 2002, Pp. 212–223 (2002).*

[Wang 2003] H.Wang, W.Fan, P.S.Yu, J.Han ; Mining Concept-Drifting Data Streams Classifiers ; *In Proceedings Of The Ninth Acm Sigkdd International Conference Discovery And Data Mining, Pages 226–235. Acm Press, New York, Ny, (2003).*

[Warmuth 2002] M.K.Warmuth, D.P.Helmbold, S.Panizza ; Direct And Indirect Algorithms For Online Learning Of Disjunctions ; *Theor. Comput. Sci., Vol. 284, No. 1, Pp. 109–142, (2002).*

[Watanabe 1995] T.Watanabe, Q.Luo, N.Sugie ; Layout Recognition Of Multi-Kinds Of Table-Form Documents ; *Ieee Trans. Pattern Anal. Mach. Intell. 17(4), 432–445 (1995).*

[Watanabe 1999] T.Watanabe ; A Guideline For Specifying Layout Knowledge ; *In: Proceedings Of Document Recognition And Retrieval Vi (Is&T/Spie Electronic Imaging), San Jose, Ca, Spie Proceedings Series 3651, 162–172 (1999).*

[Weber 2000] M.Weber, M.Welling, P.Perona ; Unsupervised Learning Of Models For Recogni-Tion ; *In: Eccv. (2000).*

[Wen 2007] Y.-M.Wen, B.-L.Lu ; Incremental Learning Of Support Vector Machines By Classifier Combining ; *Advances In Knowledge Discovery And Data Mining Lecture Notes In Computer Science Volume 4426, 2007, Pp 904-911, (2007).*

[Wenzel 1996] C.Wenzel, S.Baumann, T.Jäger ; Advances In Document Classification By Voting Of Competitive Approaches ; *In: Pro- Ceedings Of International Association For Pattern Recognition Workshop On Document Analysis Systems, Malvern, Pennsylv- Vania, Pp. 352-372 (1996).*

[Widmer 1992] G.Widmer, M.Kubat ; Learning Flexible Concepts From Streams Of Examples: Flora2 ; *In: Ecai, P. 467. Wiley, New York (1992).*

[Widmer 1993] G.Widmer, M.Kubat ; Effective Learning In Dynamic Environments By Explicit Context Tracking ; *In: Ecml, Pp. 227-243. Springer, Berlin (1993).*

[Widmer 1996] G.Widmer, M.Kubat ; Learning In The Presence Of Concept Drift And Hidden Contexts ; *Machinelearning, Vol. 23, No. 1, Pp. 69-101, (1996).*

[Widyantoro 2002] D.H.Widyantoro, T.R.Ioerger, J.Yen ; An Incremental Approach To Building A Cluster Hierarchy ; *Proceeding Icdm '02 Proceedings Of The 2002 Ieee International Conference On Data Mining Page 705 Ieee Computer Society Washington, Dc, Usa (2002).*

[Williamson 1996] J.R.Williamson, Gaussian Artmap: A Neural Network For Fast Incremental Learning Of Noisy Multidimensional Maps ; *Neural Netw., Vol. 9, No. 5, Pp. 881-897, (1996).*

[Winn 2005] J.Winn, A.Criminisi, T.Minka ; Object Categorization By Learned Universal Visual Dictionary ; *In: Iccv (2005).*

[Wnek 1999] J.Wnek ; Learning To Identify Hundreds Of Flex Form Documents ; *In: Proceedings Of Document Recognition And Retrieval Vi (Is&T/Spie Electronic Imaging), San Jose, Ca, Spie Proceedings Series 3651, 173-182 (1999).*

[Wong 1990] A.K.C.Wong, J.Constant, M.L.You ; Random Graphs ; *In: Syntactic And Structural Pattern Recognition: Theory And Applications. World Scientific, Singapore. Pp. 197-236 (1990).*

[Ye 2002] N.Ye, X.Li ; A Scalable, Incremental Learning Algorithm For Classification Problems ; *Computers & Industrial Engineering, Vol. 43, No. 4. (September 2002), Pp. 677-692 (2002).*

[Zhang 1997] T.Zhang ; Data Clustering For Very Large Datasets Plus Applications ; *Phd Thesis. University Of Wisconsin, Madison, Wi (1997).*

[Zhang 2007] J.Zhang, M.Marszalek, S.Lazebnik, C.Schmid ; Local Features And Kernels For Classification Of Texture And Object Categories: A Comprehensive Study ; *Ijcv'07 (2007).*

[Zhang 2011] P.Zhang, B.J.Gao, X.Zhu, L.Guo ; Enabling Fast Lazy Learning For Data Streams ; *In Icdm, 2011, Pp. 932-941 (2011).*

[Zhou 2002] Z-H.Zhou, Z-Q.Chen ; Hybrid Decision Tree ; *Knowledge-Based Systems, Vol.15, No.8, Pp.515-528 (2002)*.

[Zhu 2012] L.Zhu, S.Pang, G.Chen, A.Sarrafzadeh; Class Imbalance Robust Incremental Lpsvm For Data Streams Learning; *Ijcnns 2012: 1-8 (2012)*.

# Méthodes de classifications dynamiques et incrémentales

## Résumé

Cette thèse s'intéresse à la problématique de la classification dynamique en environnements stationnaires et non stationnaires, tolérante aux variations de quantités des données d'apprentissage et capable d'ajuster ses modèles selon la variabilité des données entrantes. Pour cela, nous proposons une solution faisant cohabiter des classificateurs one-class SVM indépendants ayant chacun leur propre procédure d'apprentissage incrémentale et par conséquent, ne subissant pas d'influences croisées pouvant émaner de la configuration des modèles des autres classificateurs. L'originalité de notre proposition repose sur l'exploitation des anciennes connaissances conservées dans les modèles de SVM (historique propre à chaque SVM représenté par l'ensemble des vecteurs supports trouvés) et leur combinaison avec les connaissances apportées par les nouvelles données au moment de leur arrivée. Le modèle de classification proposé (mOC-iSVM) sera exploité à travers trois variations exploitant chacune différemment l'historique des modèles. Notre contribution s'inscrit dans un état de l'art ne proposant pas à ce jour de solutions permettant de traiter à la fois la dérive de concepts, l'ajout ou la suppression de concepts, la fusion ou division de concepts, tout en offrant un cadre privilégié d'interactions avec l'utilisateur. Dans le cadre du projet ANR DIGIDOC, notre approche a été appliquée sur plusieurs scénarios de classification de flux d'images pouvant survenir dans des cas réels lors de campagnes de numérisation. Ces scénarios ont permis de valider une exploitation interactive de notre solution de classification incrémentale pour classifier des images arrivant en flux afin d'améliorer la qualité des images numérisées.

**Mots clés** : classification dynamique, incrémentalité, SVM, environnement stationnaire et non-stationnaire, images de documents, numérisation.

## Résumé en anglais

This research contributes to the field of dynamic learning and classification in case of stationary and non-stationary environments. The goal of this PhD is to define a new classification framework able to deal with very small learning dataset at the beginning of the process and with abilities to adjust itself according to the variability of the incoming data inside a stream. For that purpose, we propose a solution based on a combination of independent one-class SVM classifiers having each one their own incremental learning procedure. Consequently, each classifier is not sensitive to crossed influences which can emanate from the configuration of the models of the other classifiers. The originality of our proposal comes from the use of the former knowledge kept in the SVM models (represented by all the found support vectors) and its combination with the new data coming incrementally from the stream. The proposed classification model (mOC-iSVM) is exploited through three variations in the way of using the existing models at each step of time. Our contribution states in a state of the art where no solution is proposed today to handle at the same time, the concept drift, the addition or the deletion of concepts, the fusion or division of concepts while offering a privileged solution for interaction with the user. Inside the DIGIDOC project, our approach was applied to several scenarios of classification of images streams which can correspond to real cases in digitalization projects. These different scenarios allow validating an interactive exploitation of our solution of incremental classification to classify images coming in a stream in order to improve the quality of the digitized images.

**Keywords** : document classification, incremental learning, one-class SVM classifier, stationary and non-stationary environments, selection of support vectors.