$N°$ d'Ordre : D.U. ...
EDSPIC : ...

Université Paris 13

ÉCOLE DOCTORALE GALILÉE

THÈSE

présentée par

Fatma HAMDI

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PARIS 13
Spécialité : INFORMATIQUE

Apprentissage en distributions déséquilibrées

Soutenue publiquement le 06 décembre 2012 devant le jury :

Younès BENNANI	Professeur, Université Paris 13	Directeur de thèse
Pascale Kuntz	Professeur, Polytech Nantes	Rapporteur
Jean-Charles Lamirel	Maître de Conférences, HDR LORIA	Rapporteur
Vladimir Radevski	Professeur, South East European University	Examinateur
Khalid Benabdeslem	Maître de conférences, Université Lyon 1	Examinateur
Christophe Fouqueré	Professeur, Université Paris 13	Examinateur
Stéphane Lallich	Professeur, Université Lyon 2	Examinateur
Vincent Lemaire	Research Scientist, HDR Orange Labs	Examinateur

Remerciements

Résumé

Le travail de recherche présenté dans cette thèse concerne le développement d'approches à base d'apprentissage artificiel pour le traitement des bases de données déséquilibrées. Afin d'apporter des éléments de réponse pour cette problématique, nous proposons différentes contributions. Une première méthode qui agit au niveau des données d'apprentissage $SNCR$, c'est une technique d'échantillonnage structurel adaptatif qui permet de rééquilibrer les données par sous-échantillonnage de la classe majoritaire. La méthode proposée est guidée par la structure topologique des données et leur distribution. La seconde contribution proposée dans cette thèse aborde le problème de l'apprentissage à partir d'une seule classe, c'est un moyen permettant de contourner le problème de classes déséquilibrées à un problème de détection de nouveauté. Le modèle $RS-NDF$ est basée sur un ensemble de filtres adaptatif. Chaque filtre est conçu dans un espace de description dont les composantes et la dimension sont choisies aléatoirement. Nous avons proposé en outre une amélioration de la qualité de $RS-NDF$ par une extension plus économe $SRS-NDF$, permettant de réduire le nombre de modèles participant à la prise de décision. L'objectif est de choisir parmi cet ensemble de filtres, le sous ensemble qui permet d'atteindre les meilleurs performances. Enfin nous avons proposé une adaptation du modèle de base NDF au problème de la détection de la dérive de concept. Les résultats obtenus sur la validation des approches traitées dans cette étude sont encourageants et prometteurs.

Mots clès : apprentissage artificiel, distributions déséquilibrées, apprentissage à partir d'une classe, apprentissage d'ensemble, échantillonnage structurel adaptatif, détection de nouveauté, dérive de concept, détection et identification d'événements rares.

Abstract

The research work exposed in this thesis concerns the development of approaches for processing and modeling unbalanced databases. In order to afford solutions to this problem, we propose different contributions. A first proposition acting at the learning data level $SNCR$, it is a technique of adaptive structural sampling that allowing data rebalancing by sub-sampling of the major class. The proposed method is guided by the topological structure of the data and their distribution. The second proposed approach in this thesis discuss the problem of one class learning, it is a way allowing to bypass the problem of unbalanced classes to a novelty detection problem. The model $RS-NDF$ is based on a set of adaptive filters. Every filter is conceived in a description subspace which the components and the dimension are randomly chosen. Besides, we propose an improvement of the quality of $RS-NDF$ by an extension $SRS-NDF$ allowing to reduce the number of models participating in the decision. The goal is to choose between those filters the sub-set which allows to reach the best performances. Finally, we propose an adaptation of the basic model NDF to the concept drift detection problem. The results obtained using the proposed approaches are encouraging and promising.

Keywords : machine learning, umbalanced distributions, one class learning, ensemble methods, structural and adaptatif sampling, novelty detection, concept drift , detection and identiation of rare events.

Table des matières

Introduction **1**

1 Apprentissage en distributions déséquilibrées et détection de nouveauté **7**

1.1 Introduction 7

1.2 Problèmes liés aux données déséquilibrées 8

1.3 Apprentissage sensible au déséquilibre des classes 10

1.3.1 Stratégies d'échantillonnage 10

1.3.2 Stratégies Algorithmiques 12

1.3.3 Approches ensemble 15

1.4 Conclusion 16

2 Sous Echantillonnage Structurel et Adaptatif **17**

2.1 Le principe de notre approche : SNCR 18

2.1.1 Le modèle de base : Cartes Auto-organisatrices 18

2.1.2 Notre contribution : SNCR 20

2.2 Validation 22

2.2.1 Description des bases de données 22

2.2.2 Protocole expérimental 23

2.2.3 Résultats 26

2.3 Conclusion 29

3 Apprentissage à partir d'une seule classe par filtrage adaptatif **31**

3.1 L'apprentissage à partir d'une seule classe et détection de nouveauté . 32

3.1.1 Analyse en Composantes Principales 33

3.1.2 Réseaux de neurones auto-associatifs de type MLP 35

3.1.3 Les séparateurs à vastes marges : SVM 36

3.1.4 Le principe du filtre détecteur de nouveauté de Kohonen et Oja 36
3.1.5 Le modèle ILoNDF : "Incremental data-driven Learning of Novelty Detector Filter" . . . 41
3.2 Nos contributions . . . 43
3.2.1 Le modèle RS-NDF : " Random Subspace Novelty Detection Filter" 43
3.2.2 Le modèle SRS-NDF : "Selected Random Subspace Novelty Detection Filter" . . . 45
3.3 Vers une description unifiée des différentes approches . . . 47
3.4 Validation . . . 50
3.4.1 Description des bases de données . . . 50
3.4.2 Les mesures de performances et le protocole expérimental . . . 52
3.4.3 Résultats . . . 53
3.5 Conclusion . . . 59

4 Détection de la dérive de concept 63
4.1 Définition et problématique . . . 64
4.1.1 Définition de la dérive de concept . . . 64
4.1.2 Problèmatique . . . 65
4.2 L'apprentissage avec la dérive de concept . . . 67
4.2.1 Fenêtre d'apprentissage . . . 68
4.2.2 Méthodes Ensemblistes . . . 69
4.2.3 Pondération des données . . . 70
4.3 Notre contribution à la dérive de concept . . . 71
4.3.1 Les normes matricielles . . . 71
4.3.2 Fonctionnement de notre approche . . . 73
4.4 Validation . . . 74
4.4.1 Présentation des données . . . 74
4.4.2 Résultats . . . 76

5 Conclusion et perspectives 81

Conclusion 81
5.1 Conclusion . . . 81
5.2 Perspectives . . . 84

Liste des publications 87

Bibliographie **119**

Table des figures

1.1 Manque absolu de données 9

1.2 Manque relatif de données 9

2.1 Quelques étapes de l'apprentissage d'une carte auto-organisatrice. Les données sont dans la zone bleue, les prototypes sont en vert, reliés entre eux par des liens topologiques. A la fin de l'apprentissage les régions de Voronoi déterminent quel neurone sera le plus sensible pour chaque donnée. 20

2.2 Evaluation binaire : faux positif, faux négatif, vrai positif et vrai négatif 24

2.3 Courbe ROC 25

2.4 Comportement des deux algorithmes NCR et SNCR avec la base Post-operative. La classe majoritaire est présentée par des carrés rouges, et la classe majoritaire par des cercles colorés (chaque couleur correspond à une classe). 28

2.5 SNCR le processus d'auto -organisation. Le développement du sous-échantillonnage indiqué par la taille de la base de données à chaque itération. Chaque courbe montre les deux phases : Phase de nettoyage et phase de quantification. . . . 30

3.1 L'architecture neuronale du modèle de filtre détecteur de nouveauté NDF 37

3.2 Projection 38

3.3 L'architecture neuronale du modèle RS-NDF 43

3.4 La fonction de filtrage 48

3.5 Projection ACP de la base Ionosphere 51

3.6 Projection ACP de la base Waveform 51

3.7 Projection ACP de la base Yeast 52

3.8 Radars des bases : Waveform, Oil, Glass et Ionosphère 55

3.9 Radars des bases : Vin, Yeast, WDBC et Spectf 56

3.10 Le signal de sortie de RS-NDF et NDF 57

3.11 La F-measure sur les différentes bases de données 57

3.12 La mesure Acc+ sur les différentes base de données 58

4.1 Exemple d'une dérive de concept progressive, passant du concept 1 au concept 2 [70]. 66

4.2 Dérive lente et dérive brusque de concept. 68

4.3 Exemples de différents types de fenêtrage. 73

4.4 L'évolution de la distance entre les filtres pour Waveform 77

4.5 L'évolution de la distance entre les filtres pour Wine 77

Liste des tableaux

2.1 Bases d'apprentissage. Les colonnes présentent : le nombre d'exemples, le nombre d'attributs (quantitatifs et qualitatifs), la classe minoritaire et majoritaire et leurs distribution 23

2.2 Matrice de confusion pour la classification binaire. 24

2.3 AUC (%), IBA (%) et TS (%) calculés dans le cas des arbres de décision. Les valeurs entre parenthéses correspondent à l'écart type calculé aprés une validation croisée. AUC : Area under curve. IBA : Index of Balanced Accuracy. TS (%) : Taux des données supprimées aprés application de l'algorithme NCR et SNCR. 100% correspond à la taille de la classe majoritaire 27

2.4 TS (%) : Taux des données supprimées après application de l'algorithme NCR et SNCR. 100% correspond à la taille de la classe majoritaire . . 27

3.1 Description des bases 50

3.2 Comparaison des performances sur la base de données Vin 53

3.3 Comparaison des performances sur la base de données Ionosphere . . . 54

3.4 Comparaison des performances sur la base de données Glass 54

3.5 Comparaison des performances 60

4.1 Types et catégories d'attaques 75

4.2 Répartition des données KDD-Cup 1999 75

4.3 Répartition des données d'apprentissage 76

4.4 Matrice de confusion 76

4.5 Matrice de Confusion de WaveForm 77

4.6 Matrice de Confusion de Wine 78

4.7 Matrice de confusion de notre approche 78

4.8 Matrice de l'approche AdaBoostM1 79

Introduction

La fouille de données est devenue ces dernières années une discipline très active dans de nombreux domaines d'applications visant à analyser et explorer des données de plus en plus volumineuses et complexes. D'une manière générale, la fouille de données regroupe l'ensemble des techniques descriptives (ou exploratoires) visant à mettre en évidence des informations présentes mais cachées par le volume de données. Nous parlons ici de techniques non supervisées, ou aussi prédictives cherchant à extrapoler de nouvelles informations à partir des informations présentes dans les données (techniques supervisées). La classification supervisée (ang. classification) est basée sur un ensemble d'observations (appelé ensemble d'apprentissage) de classes connues, le but étant de découvrir la structure des classes à partir de l'ensemble d'apprentissage afin de pouvoir généraliser cette structure sur un ensemble de données plus large. Lorsque nous traitons des données réelles, nous sommes généralement face à une masse importante (un grand nombre d'individus dans des grands espaces descriptifs) de nature variée, et aussi des données manquantes, bruitées et déséquilibrées. En effet plusieurs aspects pourraient influencer les systèmes d'apprentissage existants. Un de ces aspects est lié au déséquilibre des classes dans lequel le nombre d'observations appartenant à une classe, dépasse fortement celui des observations dans les autres classes. Dans ce type de cas qui est assez fréquent, le système d'apprentissage a des difficultés au cours de la phase d'entraînement liées aux déséquilibres inter-classes. Les applications typiques de ce problème sont la détection de fraudes, la maintenance préventive, la détection des intrusions dans le réseau, le diagnostic de maladies rares et de nombreux autres domaines. Pour le cas de la la détection de fraudes, il peut y avoir d'énormes transactions légitimes entre deux transactions frauduleuses.

La question du déséquilibre dans la répartition des classes est devenue plus prononcée avec les applications des algorithmes d'apprentissage sur des applications réelles. Le déséquilibre peut être un artefact de la distribution des classes et/ou des différents coûts d'erreurs ou d'exemples. Il a reçu une attention importante dans la communauté d'apprentissage artificiel et de l'exploration de données sous forme de nombreux ate-

liers ou sessions spéciales dans les plus grandes conférences du domaine. La gamme de papiers dans ces manifestations scientifiques exhibe la nature omniprésente des questions de déséquilibre des classes rencontrées par les communautés de l'apprentissage artificiel et la fouille de données. Les méthodes d'échantillonnage (objet du chapitre 2) continuent de se développer dans le domaine comme remède possible pour cette problématique. Cependant, la recherche continue d'évoluer avec différentes applications, car chaque application fournit un problème spécifique et pertinent dans son propre cas. Un des objectifs de ces ateliers initiaux et sessions spéciales portait principalement sur les critères d'évaluation de l'apprentissage et l'exploration des ensembles de données déséquilibrées. La limitation de la précision comme mesure de la performance a été rapidement établie. Par contre, les courbes ROC sont vite apparues comme un choix populaire. La question impérieuse, compte tenu des différentes distributions est la suivante : Qu'est-ce qu'une distribution correcte pour un algorithme d'apprentissage ?
La distribution des classes a un effet considérable sur l'apprentissage d'un classificateur. En effet, la répartition naturelle des classes n'est souvent pas la meilleure distribution pour l'apprentissage d'un classificateur. En outre, le déséquilibre dans les données peut être plus caractéristique de "rareté" dans l'espace des données que le déséquilibre des classes. Diverses stratégies de rééchantillonnage ont été utilisées comme le sur-échantillonnage aléatoire avec remise, le sous-échantillonnage aléatoire, l'échantillonnage ciblé, le sous-échantillonnage centré, le sur-échantillonnage basé sur la génération synthétique avec de nouveaux échantillons à partir d'informations connues, et plusieurs combinaisons de ces techniques. En plus de la question de la distribution inter-classes, une autre importante question résultant de la faible densité des données est la distribution des données dans chaque classe.

Par ailleurs, une autre école de pensée concerne les approches fondées sur les apprenants à classe unique (objet du chapitre 3). Les approches d'apprentissage à partir d'une seule classe constituent une alternative intéressante à l'approche traditionnelle discriminante, où les classificateurs sont appris sur la classe cible seule.

Dans cette thèse, nous présenterons deux types de contributions pour l'apprentissage en distributions déséquilibrées suivant les deux grandes tendances des approches proposées dans le domaine :

- Une contribution basée sur une technique d'échantillonnage structurel et adaptatif
- et des contributions permettant un apprentissage à partir d'une seule classe, la classe cible.

Organisation du mémoire

Chapitre 2

Ce chapitre présente un récapitulatif des différentes problématiques liées aux bases de données déséquilibrées, Ensuite, un résumé des méthodes proposées dans la littérature pour les traiter. Ces différentes méthodes peuvent être regroupées en trois catégories : méthodes opérant au niveau de données, des méthodes opérant au niveau algorithmique et les méthodes ensemble. Les méthodes opérant au niveau de données proposent des stratégies d'échantillonnage qui permettent d'équilibrer les données, ou de constituer des échantillons de manière à encourager les algorithmes d'apprentissage à converger vers un type de modèle spécifique. Les méthodes opérant au niveau algorithmique proposent des téchniques qui tiennent intrinséquement compte du déséquilibre en compensant les données sans altérer la distribution des classes. Enfin les méthodes ensemble permettant de rendre n'importe quel type d'algorithme sensible à l'asymétrie, notamment par des méthodes de *boosting* ou du *bagging*.

Chapitre 3

Dans ce troisième chapitre, nous nous intéressons aux techniques adaptées pour le traitement de bases de données déséquilibrées et plus précisément les techniques de sélection de données. En effet, nous proposons notre première contribution basée sur la structure sous-jacente et la notion de voisinage en particulier autour des frontières entre classes. C'est une technique d'échantillonnage structurel adaptatif qui permet de rééquilibrer les données par sous-échantillonnage de la classe majoritaire. La méthode proposée est guidée par la structure topologique des données et leur distribution. Notre méthode est évaluée sur différents jeux de données afin de montrer son intérêt et sa validité.

Chapitre 4

Ce chapitre expose d'autres contributions algorithmiques qui concerne l'apprentissage à partir d'une seule classe. D'abord, nous présentons quelques approches classiques fréquemment utilisées pour ce problème. Ensuite, une première contribution dans ce chapitre est illustrée sur un problème de détection d'événements rares (nouveautés). Le nouveau modèle d'apprentissage que nous proposons est une extension d'un modèle

de détection de nouveauté proposé dès les années 1976. Nous proposons l'approche "Random Subspace Novelty Detection Filter" ($RS-NDF$) qui est basée sur un ensemble de filtres adaptatifs, combinant le pouvoir d'apprentissage à partir d'une seule classe et les capacités des approches ensemble. Chaque filtre est conçu dans un espace de description dont les composantes et la dimension sont choisies aléatoirement. Ensuite une deuxième contribution permet d'améliorer l'algorithme $RS-NDF$ pour de meilleurs performances en détection de nouveauté. En effet, la première contribution consiste à construire un ensemble de filtres adaptatifs. Chacun de ces filtres est obligatoirement censé donner une réponse face à l'arrivée d'une nouvelle observation. Nous avons donc opté pour la methode la plus "naturelle" qui consiste à évaluer tous les sous modèles. Ceci peut exiger des temps de calcul très important. Nous proposerons donc une stratégie "Selected Random Subspace Novelty Detection Filter" ($SRS-NDF$), plus économe, permettant de réduire le nombre de modèles participant à la prise de décision. L'objectif est de choisir parmi cet ensemble de modèles (filtres) le sous ensemble qui permet d'atteindre les meilleurs performances. Nos méthodes sont évaluées sur différents jeux de données afin de montrer leur intérêt et leur robustesse.

Chapitre 5

Dans le chapitre 5, nous nous intéressons au problème de la dérive de concept dans les flux de données. La détection de dérive de concept peut être abordée par d'éventuels changements dans la distribution des données. Ce chapitre contient un état de l'art sur les méthodes existantes pour ce genre de problématique, puis nous proposerons des premiers travaux pour répondre à ce problème. En effet nous avons adapté le modèle de filtre détecteur de nouveauté présenté dans le chapitre précédent au problème de dérive de concept. Nos premières expérimentations sont présentées sur des jeux de données synthétiques et réelles.

Chapitre 6

Enfin, dans un dernier chapitre, nous présentons une conclusion générale ainsi que les perspectives envisagées pour la poursuite de ce travail.

Remerciements

Ce travail de thèse à été financé par le projet ANR E-Fraud.

Chapitre 1

Apprentissage en distributions déséquilibrées et détection de nouveauté

1.1 Introduction

La plupart des systèmes d'apprentissage suppose que l'ensemble des jeux de données utilisés pour apprendre soient équilibrés. Cependant, s'agissant des applications réelles, cet équilibre n'est pas toujours vérifié. Le traitement de classes issues de données déséquilibrées est un problème classique mais toujours largement ouvert en fouille de données, surtout qu'elle est très utilisée dans de nombreux domaines. A titre d'exemple, dans le domaine médical pour prédire une maladie rare, en diagnostic pour prédire une panne et dans le domaine bancaire pour détecter les clients insolvables ou les transactions frauduleuses. Le déséquilibre des jeux de données pouvant atteindre 1 pour 100, 1 pour 1000, 1 pour 10 000 et souvent encore plus [58]. En effet, si par exemple, 99% des données appartiennent à une seule classe, il sera difficile de faire mieux que le 1% d'erreur obtenue en classant tous les individus dans cette classe. Il convient donc de trouver d'autres solutions et hypothèses adaptées au problème de déséquilibre sans remettre en cause les fondements des algorithmes. Dans ce chapitre, nous présentons les différents problèmes liés à l'asymétrie des classes ainsi qu'une synthèse d'approches proposées afin de traiter le problème de déséquilibre en apprentissage. Ce chapitre est structuré comme suit.

La première section présente une description des problèmes liés à l'asymétrie. La

section suivante décrit les principales approches permettant de traiter ces problèmes. Nous abordons tout d'abord les stratégies d'échantillonnage, puis les méthodes algorithmiques et enfin les approches ensemble.

1.2 Problèmes liés aux données déséquilibrées

Dans un problème de classification à deux classes, les données d'apprentissage de la classe majoritaire sont largement supérieures en nombre à celles de la classe minoritaire. Tous les algorithmes permettant d'obtenir un taux d'erreur minimal auront toujours tendance à négliger la classe minoritaire, qui est dans la majorité des cas la plus intéressante, à cause de cette disproportion. Ce qui justifie la grande liaison entre les deux formes d'asymétrie en apprentissage supérvisé. En effet l'asymétrie se présente sous deux formes principales : le déséquilibre des classes et l'asymétrie des coûts. Le déséquilibre des classe concerne les problèmes où l'une des modalités de la variable cible est beaucoup moins représentée que les autres, ce qui perturbe les algorithmes d'apprentissage. L'asymétrie des coûts concerne les cas où les coûts des erreurs ne sont pas symétriques. Dans le cas de la détection des fraudes, une classification d'une transaction frauduleuse en transaction légitime s'avère plus coûteuse que le cas contraire.

Weiss [2] propose de distinguer six catégories de problèmes liés aux données déséquilibrées, et à l'apprentissage des classes rares. Ces catégories sont [1] :

- Manque "absolu" de données : ce problème est observé lorsque les données disponibles ne sont pas assez suffisantes pour définir clairement les frontières de la classe. C'est le problème principal du déséquilibre. La figure (1.1) illustre ce problème : en effet les observations en rouge ne sont pas suffisantes pour définir clairement le concept de cette classe.
- Manque "relatif" de données : il s'agit d'un problème similaire au manque absolu, sauf que dans ce cas ce manque est relatif à la taille de la base de données majoritaires. Les observations de la classe minoritaire ne sont pas rares au sens absolu mais beaucoup moins représentées que celles de l'autre classe (classe majoritaire). Comme le montre la figure (1.2), on a les objets de la classe minoritaire (en rouge) qui sont représentés dans les données avec une proportion de 50% par rapport aux objets de la classe majoritaire (en bleue).
- Métriques inappropriées : dans ce cas, les mesures utilisées au cours du processus d'apprentissage ainsi que pour l'évaluation des résultats ne sont pas adaptées aux

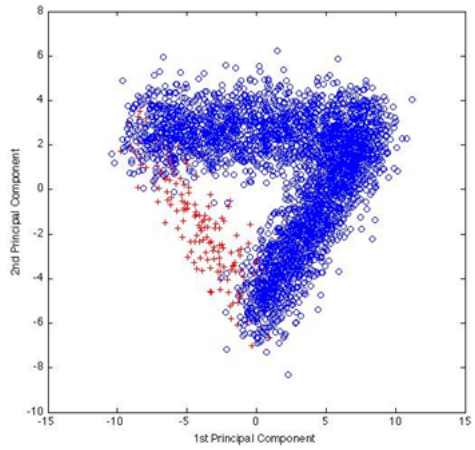


FIGURE 1.1 – Manque absolu de données

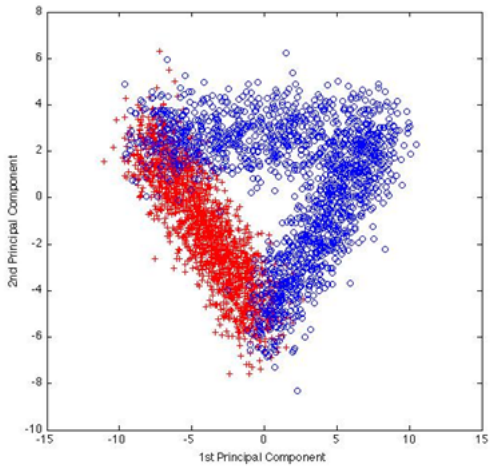


FIGURE 1.2 – Manque relatif de données

problèmes des classes déséquilibrées.

- Fragmentation des données : ce problème est lié aux algorithmes ayant une approche descendante, qui partent de l'espace de tous les individus et le partitionnent récursivement en sous-espaces de plus en plus petits.
- Marge d'induction inappropriée : il s'agit de la marge appliquée à la règle apprise sur les données d'apprentissage pour pouvoir généraliser.
- Données bruitées : le bruit a plus d'impact sur les classes rares que sur les classes fréquentes.

La distribution inégale des classes n'est pas le seul problème responsable de l'échec des algorithmes d'apprentissage. Il existe d'autres sources de difficulté pour l'apprentissage comme : la variabilité intra-classe qui peut perturber l'apprentissage de la classe minoritaire, le chevauchement (ou le recouvrement) des classes, et la duplication des données.

Différentes méthodes ont été proposées pour traiter les problèmes de l'asymétrie des classes. Ces différentes méthodes peuvent être regroupées en trois catégories :

- Méthodes opérant au niveau des données : les stratégies d'échantillonnage qui permettent d'équilibrer les données, ou de constituer des échantillons de manière à encourager les algorithmes d'apprentissage à converger vers un type de modéle spécifique.

- Méthodes opérant au niveau algorithmique : nous trouvons des méthodes qui tiennent intrinséquement compte du déséquilibre en compensant les données sans altérer la distribution des classes.

- Les approches ensemble : le principe de ces techniques permet de rendre n'importe quel type d'algorithme sensible à l'asymétrie, notamment par des méthodes de *boosting* ou du *bagging*.

1.3 Apprentissage sensible au déséquilibre des classes

1.3.1 Stratégies d'échantillonnage

L'échantillonnage est une technique incontournable en statistique et en $DataMining$, notamment lorsque la classe à prédire est rare. Ces techniques permettent de redresser l'échantillon de façon à augmenter la fréquence de cette classe ou diminuer celle des classes majoritaires [43]. Tous les algorithmes permettant d'obtenir un taux d'erreur minimal auront toujours tendance à négliger la classe minoritaire, qui est la plus intéressante, à cause de cette disproportion. La redéfinition de la distribution des données est dans ce cas, primordiale. Les techniques d'échantillonnage permettent d'apporter des éléments de réponse efficaces à ce problème. Parmi les différentes techniques d'échantillonnage existantes on trouve : le sous-échantillonnage et le sur-échantillonnage aléatoire, qui se basent sur la notion de voisinage. Le principal défaut des algorithmes de

sur-échantillonnage est le fait qu'ils peuvent créer des observations dans des régions où il y a présence dbservations de la classe majoritaire, ce qui crée des risques d'erreur important pour tout algorithme de classification. On trouve aussi des méthodes basées sur la combinaison de sur-échantillonnage et sous-échantillonnage. Les événements rares (les observations de la classe minoritaire) par rapport à la masse des événements fréquents, les comportements d'événemens rares extrêmement évolutifs dans le temps, et les volumes de données très importants motivent le développement de techniques d'échantillonnage adaptatives tenant compte de la structure et la nature de ces données.

1.3.1.1 Le sous-échantillonnage

Le sous-échantillonnage est un moyen pour rééquilibrer les jeux de données en supprimant un certain nombre d'individus appartenant à la classe majoritaire. Une méthode très simple, le sous-échantillonnage aléatoire, qui consiste à supprimer aléatoirement du jeu d'apprentissage des individus appartenant à la classe majoritaire, de manière à équilibrer le jeu de données. Il s'agit d'une méthode simple mais présente le risque de supprimer des individus importants pour le concept de la classe majoritaire. Afin d'éviter ce problème, plusieurs techniques ont été proposées pour guider le sous-échantillonnage. Kubat et al. [9] proposent d'utiliser le lien de Tomek comme une méthode de sous-échantillonnage. Considèrons deux individus $\mathbf{x}_1$ et $\mathbf{x}_2$ appartenant respectivement à la classe i et à la classe j, et $d(\mathbf{x}_1, \mathbf{x}_2)$ la distance entre ces deux individus. La paire $(\mathbf{x}_1, \mathbf{x}_2)$ est un lien de Tomek s'il n'existe aucun individu $\mathbf{x}_3$ tel que $d(\mathbf{x}_3, \mathbf{x}_1) < d(\mathbf{x}_1, \mathbf{x}_2)$ ou $d(\mathbf{x}_3, \mathbf{x}_2) < d(\mathbf{x}_1, \mathbf{x}_2)$. Si ces deux individus forment un lien de Tomek, c'est que l'un des deux est du bruit, ou que les deux sont des points frontières. L'idée se base sur le fait que les individus éloignés de la frontière étant plus sûrs pour l'apprentissage, et moins sensibles au bruit. Les mêmes auteurs proposent une méthode pour sous-échantillonner les observations de la classe majoritaire proches de la frontière. La première phase consiste à tirer aléatoirement un individu dans la classe majoritaire et tous les individus de la classe minoritaire dans un échantillon E pour former un nouvel échantillon E'. La deuxième phase consiste à classer tous les individus de E, non sélectionnés au cours de la première phase, avec la classe de leur plus proche voisin (1-PPV) dans E'. Les individus mal classés sont ensuite déplacés vers E'. L'objectif est donc de ne conserver parmi les exemples de la classe majoritaire que ceux qui sont éloignés de la frontière de décision, car ils sont considérés comme plus pertinents pour l'apprentissage. Une autre technique de sous-échantillonnage a été proposée par Wilson [10]. L'idée consiste à utiliser la régle des plus proches voisins pour supprimer des individus de la classe majoritaire "Neighborhood Cleaning Rule"

(NCR). Après la sélection des trois voisins les plus proches pour chaque exemple $\mathbf{x}_i$ l'une des règles suivantes est appliquée :
- si $\mathbf{x}_i$ appartient à la classe majoritaire ("-"), et les trois voisins les plus proches sont classés dans la classe minoritaire ("+"), alors l'observation $\mathbf{x}_i$ est supprimée.
- si $\mathbf{x}_i$ appartient à la classe minoritaire, et les trois voisins les plus proches sont classés dans la classe majoritaire, alors les trois voisins les plus proches sont supprimés.

1.3.1.2 Le sur-échantillonnage

A l'inverse du sous-échantillonnage, le sur-échantillonnage consiste à augmenter le nombre d'individus de la classe minoritaire. Comme première solution, il a été proposé de dupliquer aléatoirement les individus, mais cette solution risque de ralentir les algorithmes en ajoutant des individus, tout en fournissant des modèles incapables de généraliser (risque de sur-apprentissage). Pour éviter ces problèmes plusieurs méthodes ont été proposées : L'approche SMOTE [44] est une technique qui permet de générer des individus artificiels dans la classe minoritaire. Pour chaque individu de la classe minoritaire, ses k plus proches voisins de la même classe sont calculés, puis un certain nombre d'entre eux sont sélectionnés, des individus artificiels sont ensuite disséminés aléatoirement le long de la ligne entre l'individu de la classe minoritaire et ses voisins sélectionnés. Une autre technique de sur-échantillonnage [45] qui traite surtout le problème des cas rares et le déséquilibre intra-classe a été proposée. Les individus d'une classe sont regroupés dans des sous-parties de l'espace. Dans chaque classe les sous regroupements sont détectés par une technique de classification non supervisée, puis les sous-regroupements sont sur-échantillonnés indépendamment pour que chacun, quelque soit sa classe, ait le même nombre d'individus.

1.3.2 Stratégies Algorithmiques

Afin de gérer le problème de l'apprentissage à partir de classes asymétriques une autre famille d'approches a été mise en place et qui consiste à agir au niveau algorithmique et non pas sur les données d'apprentissage. Ces approches ont été proposées afin de rendre les algorithmes d'apprentissage sensibles à l'asymétrie des distributions.

1.3.2.1 Introduction d'un biais dans les algorithmes d'apprentissage

Plusieurs approches proposent d'introduire un biais dans les algorithmes d'apprentissage. Barandela, et al [46] proposent d'introduire une mesure de distance pondérée dans l'algorithme des k plus proches voisins (K-PPV). Le principe de cette idée consiste à assigner des poids aux classes et non pas aux individus référents. Les distances aux prototypes de la classe minoritaire deviennent plus faibles qu'à celles de la classe majoritaire. D'autres travaux consistent à biaiser l'algorithme d'apprentissage SVM, en modifiant la fonction noyau, de manière à ce que l'hyperplan appris soit éloigné de la classe positive [49] et [50].

1.3.2.2 Modification du seuil de décision

Une autre stratégie algorithmique propose de modifier le seuil de décision. Certains algorithmes fournissent une probabilité pour chaque individu d'appartenir à une classe. La décision est donc prise en fixant un seuil sur cette probabilité. C'est le cas du bayésien naif, ou de certains réseaux de neurones. Il est donc possible de tenir compte du déséquilibre des données en diminuant ce seuil de décision pour la classe minoritaire (et à l'inverse en augmentant ce seuil pour les individus de la classe majoritaire). Ceci permet d'améliorer mécaniquement la sensibilité du modèle sur la classe minoritaire.

1.3.2.3 L'apprentissage à partir d'une seule classe

Une troisième catégorie s'intéresse à l' apprentissage centré sur une seule classe (One class learning) : Raskutti et Kowalczyk [47] montrent que l'apprentissage à partir d'une seule classe est particulièrement approprié lorsque les données sont très déséquilibrées. Plusieurs travaux ont été proposés dans cette famille citant le travail de Kubat et al [51] qui proposent l'algorithme SHRINK. Cet algorithme recherche la meilleure région positive dans l'espace et considère qu'il n'existe qu'une seule région positive, ce que les auteurs justifient en remarquant que si la classe est rare, c'est qu'il ne s'agit probablement que d'un seul concept. SHRINK recherche donc la région de l'espace où les individus positifs sont le plus concentrés, qu'ils soient majoritaires dans cette zone ou non. Les frontières de cette zone sont déterminées en fonction de la moyenne géométrique des erreurs sur les individus positifs et négatifs. D'autres méthodes plus populaires ont été proposées pour le problème d'apprentissage à partir d'une seule classe. En effet les différentes techniques qui seront citées ci-dessous ont été utilisées pour le problème

de détection de nouveauté. Une donnée nouvelle est une donnée qui apporte une nouveauté par rapport aux données de référence (données d'apprentissage), ce qui amène au problème de la classification à partir d'une seule classe. Cependant, le problème de détection de nouveauté représente un moyen permettant de contourner le problème de distributions déséquilibrées à un problème d'apprentissage à partir d'une seule classe. Ces différentes techniques sont : L'analyse en composantes principales ACP, les Séparateurs à Vastes Marges mono-classe ($SVM-1C$) ainsi que le Perceptron Multi-Couches (MLP [39]). L'analyse en composantes principales ACP [40] est une transformation linéaire de l'ensemble des données de l'espace original dont les variables sont corrélées, vers un espace orthogonal où tous les axes, qui sont des combinaisons linéaires des variables d'origine, sont non corrélées deux à deux. Cette technique statistique a été utilisée pour détecter les valeurs aberrantes [30]. Le problème de détection de nouveauté est très lié à celui de la détection des valeurs aberrantes. Après l'application de l'ACP, le sous-espace représentant le modèle (qui est généré par les p premières composantes principales) est associé à des variations systématiques dans les données. La projection de $x_i \in \mathbb{R}^d$ sur l'espace orthogonal, induit $x \in \mathbb{R}^p, (p \ll d)$ avec une erreur de reconstruction. D'autres techniques très populaires se sont les Séparateurs à Vastes Marges mono-classe ($SVM-1C$) [41]. Ce sont des adaptations des méthodes à noyau pour ce cas particulier. Par essence, ces techniques sont une solution naturelle aux cas de classes à effectifs déséquilibrés. En terme plus précis, $SVM-1C$ cherche une hypersphère de volume minimal qui englobe la plupart des exemples positifs disponibles pour l'apprentissage. Les réseaux de neurones auto-associatifs sont des réseaux de type MLP entrainés pour produire une approximation de la fonction identité entre les entrées et les sorties du réseau. Le réseau de neurones auto-associatif se compose d'une couche de neurones d'entrée, suivie d'une ou plusieurs couches cachées, et une couche de neurones de sortie ayant la même dimension que la couche de neurones d'entrées.

1.3.2.4 Apprentissage sensible aux coûts

L'apprentissage sensible aux coûts est une autre stratégie algorithmique. Le principe des algorithmes appartenant à cette catégorie consiste à fixer des coûts inégaux sur les différents types d'erreurs de mauvaise classification. *Metacost* une approche proposée par Domingos [52] consiste à estimer pour chaque individu sa classe optimale, c'est à dire, celle qui minimise au final le coût. C'est une approche qui permet de rendre n'importe quel algorithme d'apprentissage sensible aux coûts.

1.3.3 Approches ensemble

Les différentes méthodes présentées ci-avant agissent soit au niveau des données soit au niveau algorithmique pour résoudre le problème de l'asymétrie des classes. Une autre famille de techniques a été proposée dans la littérature pour traiter aussi le problème de déséquilibre des classes. Ce sont les approches ensemble. Le principe de ces techniques permet de rendre n'importe quel type d'algorithme sensible à l'asymétrie, notamment par des méthodes de *boosting* ou du *bagging*.

Le boosting est un algorithme itératif qui consiste à affecter des poids différents aux individus du jeu d'apprentissage. Après chaque itération le poids sur les individus mal classés augmente et celui sur les individus classés correctement diminue. Les erreurs étant souvent concentrées sur les classes rares, on peut penser que le boosting permet d'améliorer l'apprentissage sur les jeux de données déséquilibrés en augmentant les poids des individus appartenant à la classe minoritaire. Différentes variantes du boosting ont été proposées pour résoudre les problèmes de déséquilibre des données. AdaCost [53] assigne des poids plus élevés pour les erreurs sur la classe minoritaire. RareBoost [54], consiste à appliquer la règle suivante : si le nombre de vrais positifs est supérieur au nombre de faux positifs le poids des individus bien classés diminue, et si le nombre de vrais négatif est supérieur à celui des faux négatifs le poids des individus mal classés augmente. Constatant que le boosting risque de souffrir du sur-apprentissage en surpondérant les individus de la classe minoritaire, Chawla propose l'algorithme SMOTEBoost [55] qui ajoute des individus artificiels par la méthode SMOTE au lieu d'augmenter le poids des individus de la classe minoritaire.

Le Bagging (Bootstrap aggregating) est basé sur un processus stochastique de modification de l'échantillon d'apprentissage A pour créer un ensemble de classifieurs diversifié. Cette méthode consiste à construire chaque hypothése à partir d'un rééchantillonnage par bootstrap de A, sachant qu'un bootstrap est le tirage aléatoire avec remise, sur un échantillon de taille n d'un échantillon initial de même taille (c'est à dire n). Les hypothéses ainsi construites sont ensuite combinées par un vote majoritaire. Un bagging d'arbre de décision aléatoire (chaque arbre est construit sur un sous-ensemble des variables, tirées aléatoirement) connu sous le nom de forêt aléatoire à été proposé par Breiman [56]. Chen et al. [57] ont proposé deux méthodes pour utiliser les forêts aléatoires sur les jeux de données déséquilibrées. La premiére, "Balanced Random Forest", consiste à effectuer un bootstrap sur la classe minoritaire, puis à tirer le même nombre d'individus dans la classe majoritaire (avec remise). Ainsi l'échantillon de chaque arbre est équilibré. La deuxiéme approche est intitulée "Weighted Random Forest" et consiste à construire des arbres sensibles aux coûts.

1.4 Conclusion

Dans ce chapitre introductif et bibliographique, nous avons présenté une formulation du problème d'apprentissage en distributions déséquilibrées avec des classes asymétriques. Cette asymétrie se présente généralement à deux principaux niveaux : au niveau des classes ou au niveau des coûts. Nous avons exposé les principaux éléments des différentes problématiques liées aux bases de données déséquilibrées ainsi qu'une synthèse des méthodes proposées pour les traiter. Ces différentes méthodes ont été regroupées en deux grandes catégories : les techniques d'échantillonnage, et les algorithmes sensibles à l'asymétrie. En effet, l'apprentissage à partir de données déséquilibrées revient souvent à effectuer un arbitrage entre la sensibilité du modèle, qui est sa capacité à détecter les individus de la classe minoritaire, et sa précision qui correspond à la proportion d'individus réellement positifs parmi ceux classés comme positifs par le modèle. Enfin, l'apprentissage à partir d'ensembles de données déséquilibrées ouvre un front de problèmes et de directions de recherche intéressantes. Étant donné que le *DataMining* devient omniprésent dans de nombreuses applications, il est important d'étudier les problématiques liées au phénomène du déséquilibre à la fois dans la distribution des classes et des coûts.

Chapitre 2

Sous Echantillonnage Structurel et Adaptatif

Introduction

Dans ce chapitre nous nous plaçons dans un contexe d'apprentissage en distributions déséquilibrées avec une classe majoriaire et une classe minoritaire. Nous présentons une nouvelle approche de sélection de données basée sur la structure sous-jacente et la notion de voisinage en particulier autour des frontières entre classes. Cette méthode est semi supervisée, elle utilise les données de la classe minoritaire pour guider le processus du sous échantillonnage. C'est un moyen pour rééquilibrer les jeux de données en supprimant un certain nombre d'individus appartenant à la classe majoritaire. Notre méthode SNCR (Structural Neighborhood Cleaning Rule) utilise la règle des plus proches voisins NCR de Wilson [10] pour supprimer des individus de la classe majoritaire. Afin de mieux guider le sous-échantillonnage et de le rendre moins "aveugle", l'utilisation des cartes auto-organisatrices SOM [13], nous paraît une solution efficace pour choisir d'une façon intelligente les données à supprimer de la classe majoritaire en tenant compte de leur topologie.

2.1 Le principe de notre approche : SNCR

2.1.1 Le modèle de base : Cartes Auto-organisatrices

Une carte auto-organisatrice ou Self Organizing Map SOM [13] est un algorithme d'apprentissage compétitif non-supervisé à partir d'un réseau de neurones artificiels. C'est une technique non linéaire très populaire pour la réduction de dimensions et la visualisation des données. Lorsqu'une observation est reconnue, l'activation d'un neurone du réseau, sélectionné par une compétition entre les neurones, a pour effet le renforcement de ce neurone et l'inhibition des autres (c'est la régle du "Winner Takes All"). Chaque neurone se spécialise donc au cours de l'apprentissage dans la reconnaissance d'un certain type d'observations. La carte auto-organisatrice est composée d'un ensemble de neurones connectés entre eux par des liens topologiques qui forment une grille bi-dimensionnelle. Chaque neurone est connecté à n entrées (correspondant aux n dimensions de l'espace de représentation) selon n pondérations $w_j = (w_{1j}, ..., w_{nj})$ (qui forment le vecteur prototype du neurone). Les neurones sont aussi connectés à leurs voisins par des liens topologiques. Le jeu de données est utilisé pour organiser la carte selon les contraintes topologiques de l'espace d'entrée. Ainsi, une configuration entre l'espace d'entrée et l'espace du réseau est construite ; deux observations proches dans l'espace d'entrée activent deux unités proches sur la carte. Une organisation spatiale optimale est déterminée par la SOM à partir des données et quand la dimension de l'espace d'entrée est inférieure à trois, aussi bien la position des vecteurs de poids que des relations de voisinage directes entre les neurones peuvent être représentées visuellement. Pour chaque donnée présentée en cours d'apprentissage, le meilleur neurone (le plus sensible à cette donnée) met à jour son vecteur prototype w de façon à améliorer sa sensibilité à ce type de données. Pour assurer la conservation de la topologie de la carte, les autres neurones mettent à jour leurs prototypes de la même manière, mais selon une amplitude qui dépend de leurs distances par rapport au meilleur neurone. Ainsi, les prototypes les plus proches d'une donnée correspondent à des neurones voisins sur la carte.

L'apprentissage connexionniste est souvent présenté comme la minimisation d'une fonction de coût. Dans notre cas, cela correspond à la minimisation de la distance entre les données et les prototypes de la carte, pondérée par une fonction de voisinage $\mathcal{K}^{\mathcal{T}}$. Pour ce faire, nous utilisons un algorithme de gradient. La fonction de coût à minimiser est définie par :

$$\mathcal{R}(\chi, \mathcal{W}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{r \in \mathcal{C}} \mathcal{K}^{\mathcal{T}}(\delta(\chi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{w}_r||^2 \tag{2.1}$$

Avec A l'ensemble de données, $\mathcal{C}$ l'ensemble de neurones de la carte, $\chi(\mathbf{x}_{(i)})$ est le neurone dont le vecteur prototype est le plus proche de la donnée $\mathbf{x}_{(i)}$ (le "best match unit" : BMU). $\mathbf{w}_r$ est le prototype associé au neurone r. $\mathcal{K}(\delta(c, r))$ est une fonction symétrique positive à noyau : la fonction de voisinage. L'importance relative d'un neurone c comparé à un neurone r est pondérée par la valeur de $\mathcal{K}(c, r)$, qui peut être définie ainsi :

$$\mathcal{K}^{\mathcal{T}}(\delta(c, r)) = \frac{1}{T \times \exp^{-\frac{d_1^2(c,r)}{T}}} \tag{2.2}$$

T est une fonction de température qui contrôle l'étendue du voisinage qui diminue avec le temps t de T_i à T_f (par exemple $T_i = 2$ à $T_f = 0,5$) :

$$T(t) = T_i(\frac{T_f}{T_i})^{\frac{t}{t_{max}}} \tag{2.3}$$

t_{max} est le nombre maximum d'itérations autorisé pour l'apprentissage. $d_1(c, r)$ est la distance de Manhattan définie entre deux neurones c (de coordonnée (k, m)) et r (de coordonnée (b, s)) sur la grille de la carte :

$$d_1(c, r) = \parallel b - k \parallel + \parallel s - m \parallel \tag{2.4}$$

L'algorithme SOM batch est le suivant :

1) **Phase d'initialisation** :
   - Définir la topologie de la carte.
   - Initialiser aléatoirement tous les prototypes $\mathbf{w}_j = (w_{1j}, ..., w_{nj})$ pour chaque neurone j.
2) **Phase de compétition** :
   - Présenter une donnée $\mathbf{x}_{(i)}$ choisie aléatoirement.
   - Parmi les $|w|$ neurones, choisir le meilleur, $\chi(\mathbf{x}x^{(i)})$, pour représenter cette donnée :

$$\chi(\mathbf{x}_{(i)}) = \underset{1 \leq j \leq |w|}{Argmin} \parallel \mathbf{x}_i - \mathbf{w}_j \parallel^2$$

3) **Phase d'adaptation** :
   – Mettre à jour les prototypes w_c de chaque neurone c selon la règle :

$$\mathbf{w}_c = \frac{\sum_{r \in C} \mathcal{K}^T(\delta(c,r)) \sum_{\mathbf{x}_i \in \mathcal{A}, \chi(\mathbf{x}_i)=r} \mathbf{x}_i}{\sum_{r \in C} \mathcal{K}^T(\delta(c,r))|w|} \tag{2.5}$$

   avec T le taux d'apprentissage, qui diminue avec le temps.

4) **Répéter les phases 2 et 3** jusqu'à ce que les mises à jours des prototypes soient négligeables.

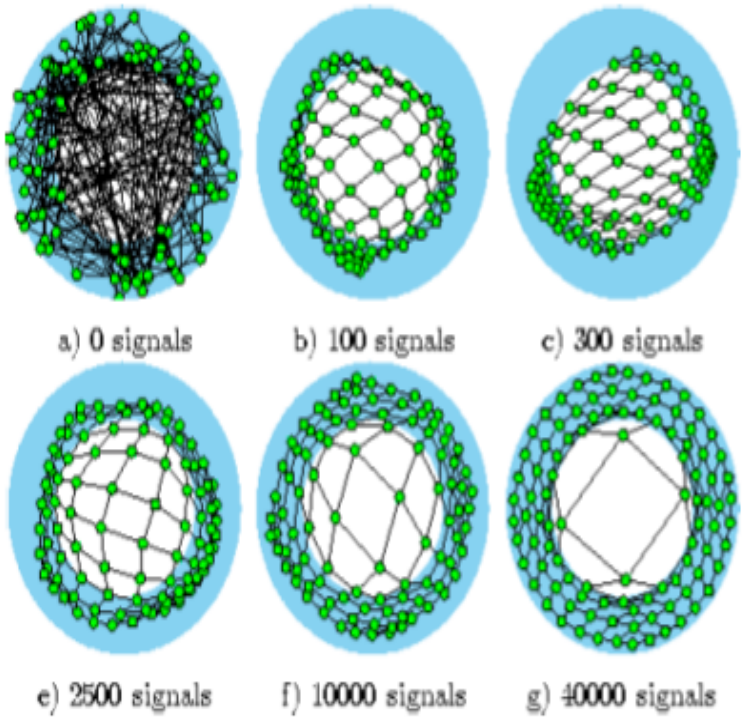


FIGURE 2.1 – Quelques étapes de l'apprentissage d'une carte auto-organisatrice. Les données sont dans la zone bleue, les prototypes sont en vert, reliés entre eux par des liens topologiques. A la fin de l'apprentissage les régions de Voronoi déterminent quel neurone sera le plus sensible pour chaque donnée.

2.1.2 Notre contribution : SNCR

La méthode que nous proposons consiste à modifier l'algorithme d'apprentissage des SOM en proposant de supprimer à chaque itération les observations qui "gênent" les

données minoritaires. L'approche consiste à intégrer les règles de nettoyage de l'algorithme NCR [10] comme une troisième étape dans l'algorithme SOM. Ces règles seront appliquées localement au niveau de chaque cellule de la carte. Par conséquent, l'algorithme SOM sera utilisé dans le cas semi-supervisé, puisque les étiquettes associées à la classe positive ("+", minoritaire) seront utilisées. Ces étiquettes ne sont pas utilisées comme variable de la base, mais uniquement dans la phase de nettoyage. L'élimination au cours de l'apprentissage d'observations implique la modification de la base d'apprentissage $\mathcal{A}$ qui diminue au fur et à mesure des itérations ($\mathcal{A} = \{\mathbf{x}_i \in \mathcal{R}^n, i = 1...N\}$ où l'individu $\mathbf{x}_i = (x_{i1}, x_{i2}..., x_{ij}, ..., x_{in})$). Chaque observation $\mathbf{x}$ dispose d'une étiquette $label(\mathbf{x})$ positive ("+", données minoritaires) ou négative ("-", données majoritaires). L'étiquette positive "+" est utilisée uniquement dans la régle de nettoyage. On notera par la suite $\mathcal{A}^+$ l'ensemble des données positives et par $\mathcal{A}^-$ l'ensemble des données nègatives. L'approche que nous proposons est donc une approche hybride : une action sur les données avec la phase de nettoyage par voisinage et une modification algorithmique de SOM. A l'opposé de NCR qui est dans l'obligation de supprimer des données de la classe majoritaire, notre approche SNCR ne l'est pas puisque le nettoyage par voisinage s'applique d'une manière locale au niveau de chaque cellule.

Nous proposons donc de minimiser la fonction de coût suivante :

$$\mathcal{R}(\mathcal{A}_t, \chi, \mathcal{W}) = \sum_{\mathbf{x}_i \in \mathcal{A}_t} \sum_{r \in \mathcal{C}} \mathcal{K}^{\mathcal{T}}(\delta(\chi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{w}_r||^2 = \sum_{\mathbf{x}_i \in \mathcal{A}_t^-} \sum_{r \in \mathcal{C}} \mathcal{K}^{\mathcal{T}}(\delta(\chi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{w}_r||^2 + \sum_{\mathbf{x}_i \in \mathcal{A}_t^+} \sum_{r \in \mathcal{C}} \mathcal{K}^{\mathcal{T}}(\delta(\chi(\mathbf{x}_i), r)) ||\mathbf{x}_i - \mathbf{w}_r||^2 \quad (2.6)$$

Où χ affecte chaque observation $\mathbf{x}$ à une cellule unique de la carte $\mathcal{C}$.

Les phases principales de l'algorithme d'apprentissage SNCR sont :

- **Entrèes** : Base d'apprentissage, $\mathcal{A}_0 = \mathcal{A}_0^- \cup \mathcal{A}_0^+$; le paramétre k : nombre des plus proches voisins.
- **Sortie** :
  1) Référents de la carte $\mathcal{C}$.
  2) Base d'apprentissage sous échantillonnée : $\mathcal{A}_{final}$ ($|\mathcal{A}_{final}| \leq |\mathcal{A}_0|$), telle que $|\mathcal{A}_{final}^+| = |\mathcal{A}_0^+|$.
- **Phase d'affectation** : chaque observation $\mathbf{x}_i$ est affectée au referent $\mathbf{w}_c$, dont elle est la plus proche au sens de la distance euclidienne : $\chi(\mathbf{x}_i) = \arg\min_c \left(\|\mathbf{x}_i - \mathbf{w}_c\|^2\right)$

– **Phase d'adaptation** : les vecteurs référents sont mis à jour avec l'expression suivante :

$$\mathbf{w}_c = \frac{\sum_{r \in \mathcal{C}} \mathcal{K}^T(\delta(c,r)) \sum_{\mathbf{x}_i \in \mathcal{A}_t, \chi(\mathbf{x}_i)=r} \mathbf{x}_i}{\sum_{r \in \mathcal{C}} \mathcal{K}^T(\delta(c,r))|w|} \tag{2.7}$$

– **Phase de nettoyage** : pour chaque cellule $c \in \mathcal{C}$ à l'itération t
  – si $\mathbf{x}_i \in \mathcal{A}_t^- \wedge k\text{-}ppv(\mathbf{x}_i) \subset P_{\chi(\mathbf{x}_i)} \subset \mathcal{A}_t^+$ alors $\mathcal{A}_{t+1} \leftarrow \mathcal{A}_t - \{\mathbf{x}_i\}$;
  – si $\mathbf{x}_i \in \mathcal{A}_t^+ \wedge k\text{-}ppv(\mathbf{x}_i) \subset P_{\chi(\mathbf{x}_i)} \subset \mathcal{A}_t^-$ alors $\mathcal{A}_{t+1} \leftarrow \mathcal{A}_t - k\text{-}ppv(\mathbf{x}_i)$;

Les trois phases permettent de minimiser la fonction de coût (eq. 2.6). Les deux premières phases sont similaires à l'algorithme de type nuées dynamiques classique. La troisième phase permet de minimiser la fonction de coût par rapport à $\mathcal{A}$. A la fin de l'apprentissage, la carte auto-organisatrice détermine une partition des données en $|\mathcal{W}|$ groupes associés à chaque référent $\mathbf{w}_c \in \mathcal{R}^n$ de la carte. Il est important de noter qu'en plus de la carte topologique, nous obtenons aussi une nouvelle base d'apprentissage de taille inférieure ou égale à la base initiale $\mathcal{A}$ ($|\mathcal{A}_{final}| \leq |\mathcal{A}|$).

2.2 Validation

2.2.1 Description des bases de données

Nous avons utilisé différents types de bases de données provenant du répertoire UCI, [14], qui sont utilisés de telle maniére à avoir des degrés de déséquilibre variables pour évaluer notre approche (tableau 2.1). Le tableau 2.1 présente pour chaque base, le nombre d'individus, le nombre d'attributs (quantitatifs et qualitatifs), la classe minoritaire et majoritaire et la distribution des classes minoritaires et majoritaires. Pour les ensembles de données ayant plus de deux classes, nous avons choisi la classe minoritaire comme classe positive, et le reste comme classe négative (classe majoritaire).

– La base Post-operative [59] : c'est une base qui contient 90 observations (patients) décrites par 8 variables. La tâche de classification pour cette base consiste à déterminer la phase post-opérative de chaque patient (la classe ou variable cible) et les différents attributs présentent des mesures de température prises pour chaque patient.
– La base Ecoli [60] : c'est une base pour la prédiction des sites de localisation des protéines des cellules eucaryotes. La base contient 336 observations décrites par 7 variables et groupées dans 9 classes de distribution de protéines.

Bases	Taille	Dim. (quanti, quali)	Taille min/maj	Classe % (min, maj)
Post-operative	90	8(1,7)	24,66	26.67, 73.33
Thyroid	215	5 (5,0)	30,185	13.95, 86.04
Ecoli	336	7 (7,0)	35,301	10.42, 89.58
Satimage	6435	36 (36,0)	626, 5809	9.72, 90.27
Glass	214	9 (9,0)	17, 197	7.94, 92.06
Flag	194	28(10,18)	17,177	8.76, 91.24

TABLE 2.1 – Bases d'apprentissage. Les colonnes présentent : le nombre d'exemples, le nombre d'attributs (quantitatifs et qualitatifs), la classe minoritaire et majoritaire et leurs distribution

- La base Satimage [61] : C'est une base qui décrit des images satellite, elle contient 6435 données et 36 variables.
- La base Glass [62] : elle est composée de 214 exemples décrits sur 9 variables. Les données sont regroupées sur 9 classes.
- La base Flag [63] : c'est une base qui contient des informations concernant différentes nations et leurs drapeaux. La base est composée par 194 observations et 28 attributs.
- La base Thyroid [64] : il s'agit d'une base contenant 215 observations décrites sur 5 variables quantitatives.

2.2.2 Protocole expérimental

Plusieurs indices d'évaluation existent en littérature, mais pour nos expérimentations nous avons choisi de calculer deux indices synthétiques. Le premier est l'indice classique AUC "Area under curve" [16] et un nouvel indice appelé IBA "Index of Balanced Accuracy", présenté par [15]. Les mesures de qualité utilisées sont calculées à partir d'une matrice de confusion qui représente les exemples correctement et incorrectement reconnus pour chaque classe. Le tableau suivant présente une matrice de confusion pour la classification binaire, où VP sont les vrais positifs, FP les faux positifs, FN les faux négatifs, et VN les vrais négatifs (figure 2.2).

Deux indices sont souvent observés conjointement. Le taux de sensibilité qui est la

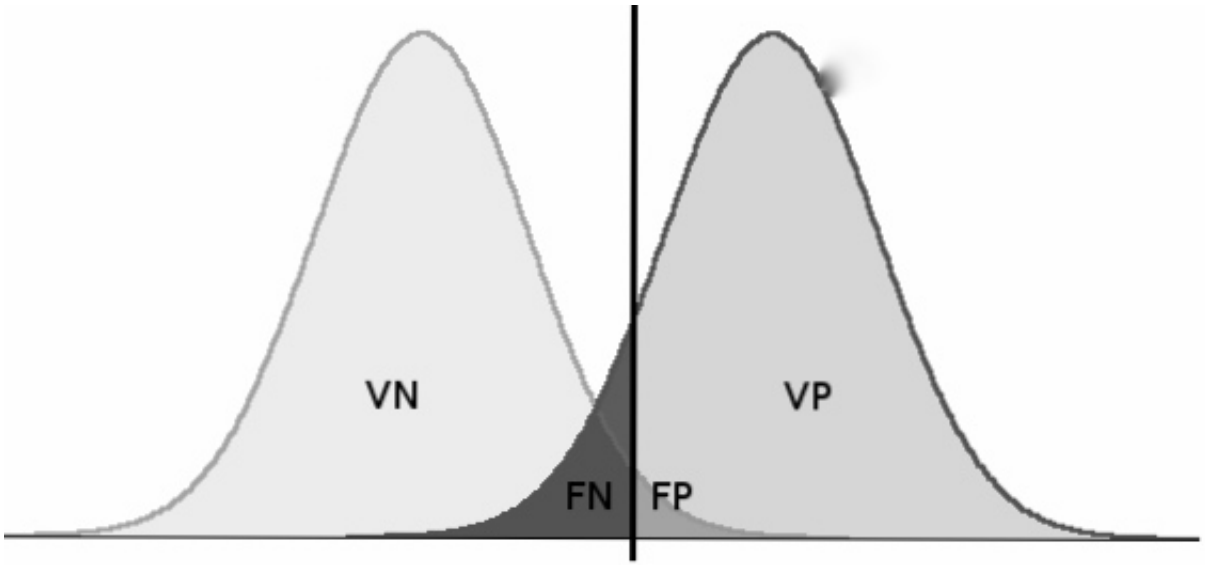


FIGURE 2.2 – Evaluation binaire : faux positif, faux négatif, vrai positif et vrai négatif

	Classe Positive Prédite	Classe Negative Prédite
Classe Positive Réelle	VP	FN
Class Negative Réelle	FP	VN

TABLE 2.2 – Matrice de confusion pour la classification binaire.

probabilité qu'un individu positif soit classé positif par le modèle :

$$TauxdeVraispositif(Acc+)(Rappel) = \frac{VP}{VP + FN} \tag{2.8}$$

Le taux de spécificité qui est la probabilité qu'un individu négatif soit classé négatif :

$$TauxdeVraisNgatif(Acc-) = \frac{VN}{VN + FP} \tag{2.9}$$

"L'espace ROC"

L'aire sous la courbe ROC ("Area under curve", AUC) est un indicateur synthétique de la courbe ROC (figure 2.3). Il existe plusieurs méthodes pour estimer l'aire sous la courbe ROC. Dans le cas du classement binaire, la courbe ROC est donnée par le point de coordonnées $Acc-$; $Acc+$, (1,0) et l'AUC est simplement l'indice égale à [16] : $AUC = [(\text{Acc}+) + (\text{Acc-})]/2$.

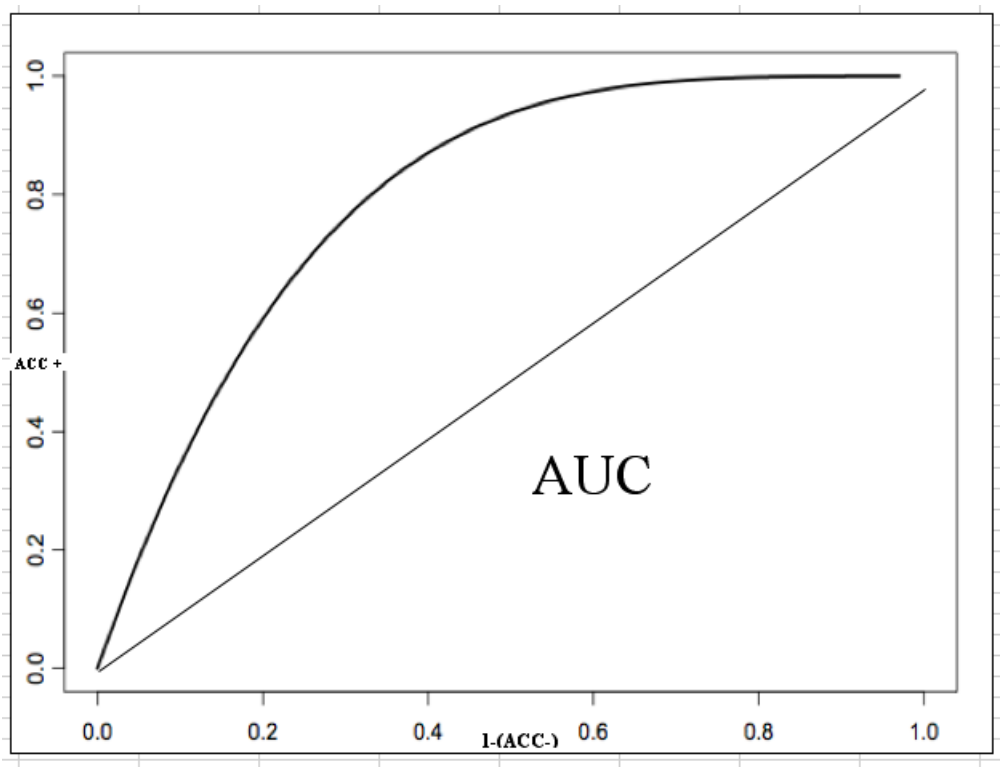


FIGURE 2.3 – Courbe ROC

"Index of Balanced Accuracy", IBA

L'AUC minimise l'influence négative de la distribution inégale des classes, mais ne distingue pas la contribution de chaque classe pour la performance globale. Cela signifie que différentes combinaisons de $Acc+$ et $Acc-$ produisent la même valeur AUC. Pour remédier à cette contrainte en donnant de l'importance à la classe positive, [15] proposent une nouvelle mesure, appelée (IBA), dont l'expression calcule la superficie d'une région rectangulaire dans un espace à deux dimensions appelé "Balanced Accuracy Graph" : $IBA = (1 + Dominance) \times Gmean^2$
Cet espace est défini par le produit de deux termes. Le premier terme est un simple indice qui introduit un degré de prévalence qui est le taux de la classe dominante par rapport à l'autre, défini par l'expression suivante : $Dominance = (Acc+) - (Acc+)$.
Le deuxième terme est la précision de chaque classe ($Gmean^2$), qui est la moyenne géométrique ([17]), et qui est une mesure appropriée de la précision globale pour les classes déséquilibrées : $Gmean = \sqrt{(Acc+) \times (Acc-)}$.
Une étude théorique et expérimentale de cet indice est présentée dans le papier [15].

2.2.3 Résultats

Tous les résultats présentés ci-dessous sont obtenus avec le paramétre de voisinage local $k = 3$. En ce qui concerne les paramètres de la carte auto-organisatrice, nous avons pris les paramétres fournis par la "SOM Toolbox" de Kohonen :
($http://www.cis.hut.fi/projects/somtoolbox/$).

Le tableau 2.3 présente les mesures AUC et IBA qui sont calculées dans le cas des arbres de décision. Les valeurs entre parenthèses indiquent l'écart type calculé sur 100 expériences correspondant à une validation croisée, en divisant la base en 10 sous ensembles et répétant ce processus 10 fois. L'analyse des résultats permet en premier lieu de confirmer que le sous-échantillonnage structurel et adaptatif permet d'obtenir de meilleurs performances en terme de l'indice AUC ou IBA, sur la plupart des bases de données. Nous avons constaté une légère baisse pour la base Flag et la base thyroid en observant uniquement l'indice AUC. Cette baisse est due uniquement à la faible valeur du $Acc-$ sur les deux bases. Par exemple pour la base Flag, $Acc-$ passe de 93.59% avec NCR à 93.33% avec SNCR, par contre concernant la classe positive, $Acc+$ passe de 20.82% (NCR) à 22.82% (SNCR). Ceci se traduit par une augmentation de l'indice IBA qui donne un avantage à la classe positive.

Pour mieux comprendre le comportement des deux méthodes SNCR et NCR, nous avons calculé le taux de suppression (TS) obtenu à la fin de l'apprentissage semi-supervisé (table 2.3). Nous observons clairement que la méthode SNCR fournit majoritairement un taux de suppression élevé en le comparant à celui de NCR. Nous avons constaté que nous atteignons des performances meilleures ou similaires avec moins de données que la méthode NCR. Sur la plupart des bases nous avons constaté une progression positive sur les trois indices (AUC, IBA et TS) à l'exception de la base "Post-operative" où nous obtenons une baisse du taux de suppression.

En effet, la base Post-operative a une distribution de 66 (+) sur 24 (-). L'application de SNCR a permis d'atteindre un AUC de 47.89% et un IBA de 12.08% avec un TS de seulement de 46.97% inférieur à 75.76% (NCR). L'action de suppression par le voisinage de la méthode SNCR, permet de ne détecter aucune donnée à éliminer dans le voisinage local, à l'inverse de l'algorithme NCR qui est une approche globale dans l'obligation de supprimer des données. Les figures (2.4(a,b,c)) montrent une projection des données à l'aide d'une ACP (Analyse en composante Principale), des données originales et celles obtenues après échantillonnage avec NCR et SNCR. La classe minoritaire est présentée par des carrés rouges, et la classe majoritaire par des cercles colorés (chaque couleur correspond à une classe). Nous observons clairement que NCR supprime des données appartenant à un voisinage relativement loin, qui sont considérées hors du voisinage local par la méthode SNCR.

Bases	*AUC* (%)		*IBA* (%)	
	NCR	SNCR	NCR	SNCR
Post-operative	45.62 (16.49)	**47.89** (18.46)	9.89 (10.92)	**12.08** (18.56)
Thyroid	**95.84** (6.51)	95.12 (8.58)	80.64 (27.95)	**81.85** (27.71)
Ecoli	83.70 (12.24)	**84.46** (12.33)	25.42 (19.37)	**26.69** (18.78)
Satimage	82.30 (2.42)	**83.42** (2.21)	24.05 (5.21)	**24.92** (4.87)
Flag	**61.61** (19.32)	61.18 (18.66)	12.10 (23.10)	**14.20** (25.84)
Glass	97.37 (7.58)	**97.62** (7.61)	85.92 (28.44)	**91.26** (24.84)

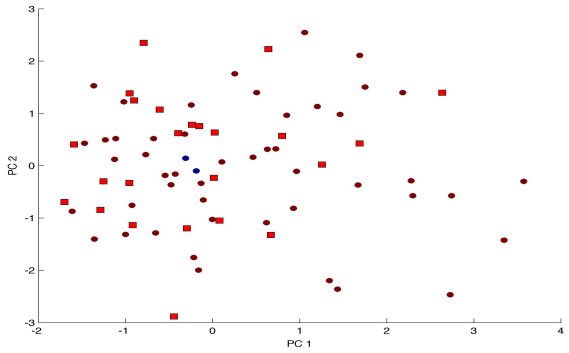
TABLE 2.3 – *AUC* (%), *IBA* (%) et TS (%) calculés dans le cas des arbres de décision. Les valeurs entre parenthéses correspondent à l'écart type calculé aprés une validation croisée. *AUC* : Area under curve. *IBA* : Index of Balanced Accuracy. TS (%) : Taux des données supprimées aprés application de l'algorithme NCR et SNCR. 100% correspond à la taille de la classe majoritaire

	TS (%)	
Bases	NCR	SNCR
Post-operative	75.76	46.97
Thyroid	4.87	7.03
Ecoli	14.29	19.61
Satimage	10.61	13.74
Flag	25.43	38.42
Glass	7.11	12.7

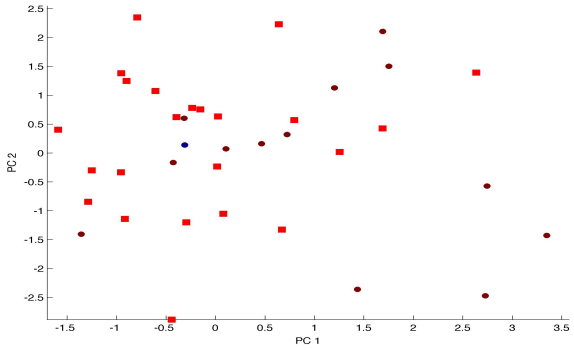
TABLE 2.4 – TS (%) : Taux des données supprimées après application de l'algorithme NCR et SNCR. 100% correspond à la taille de la classe majoritaire

.

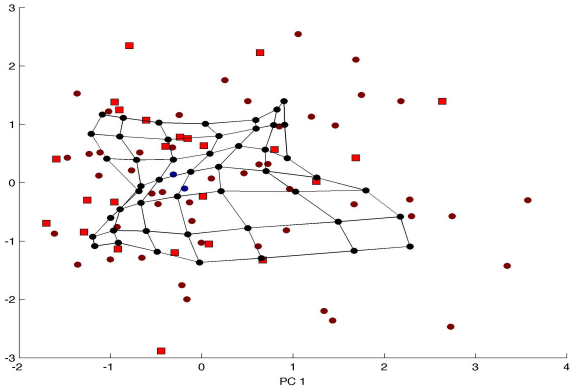
Comme pour chaque algorithme de classification topologique, une étude de l'influence du voisinage est indispensable. En effet nous avons procédé à la diminution progressive de la température T entre la valeur de t_{max} et la valeur de t_{min}. Pour chaque valeur de T, le comportement de SNCR se présente sous deux phases comme l'indique la figure



(a) Données avant échantillonnage



(b) Après application de NCR



(c) Après application de SNCR

FIGURE 2.4 – Comportement des deux algorithmes NCR et SNCR avec la base Post-operative. La classe majoritaire est présentée par des carrés rouges, et la classe majoritaire par des cercles colorés (chaque couleur correspond à une classe).

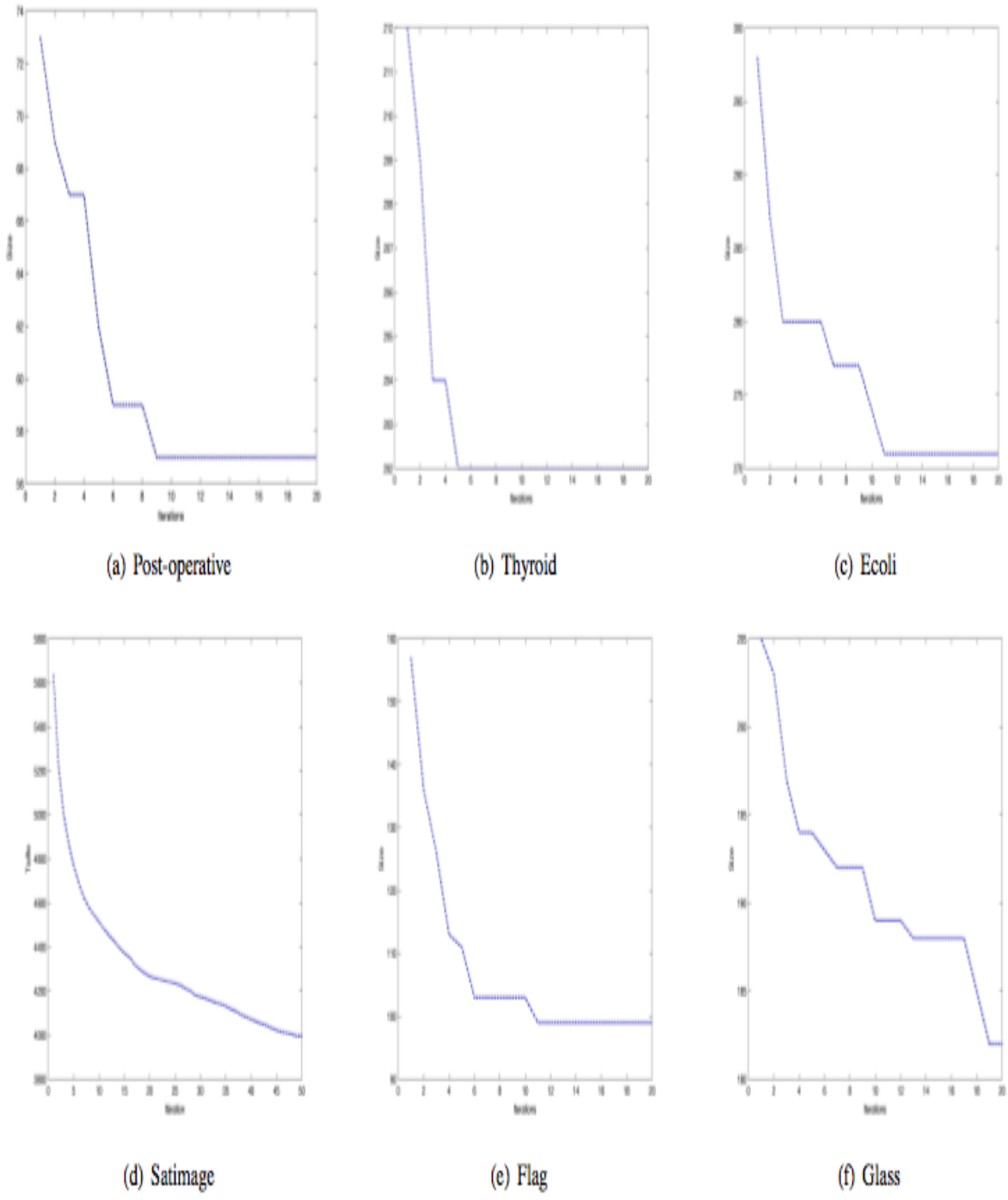
(2.5) :

– Phase de nettoyage : Cette étape correspond à des grandes valeurs de T. Pendant

cette phase l'algorithme SNCR diminue le nombre d'exemples négatifs et converge vers une taille optimale de la base de données.
– Phase de Quantification : Elle correspond à des petites valeurs de T ce qui provoque une adaptation très locale. Les paramètres sont calculées à partir d'une densité locale de données. Nous observons donc pendant cette phase une stabilité de la taille de données.

2.3 Conclusion

L'apprentissage à partir d'ensembles de données déséquilibrées est en effet un problème très important à la fois du point de vue perspective algorithmique et performance. Ne pas choisir la bonne distribution ou la bonne fonction objective tout en développant un modèle de classification peut introduire un biais vers la classe majoritaire (potentiellement sans intérêt). En outre, les mesures de performance dans ce contexte peuvent avoir un impact sur le choix de la méthode. Les méthodes d'échantillonnage sont très populaires en rééquilibrant la répartition des classes avant l'apprentissage d'un classificateur, utilisant une fonction objective basée sur une erreur de recherche dans l'espace des hypothèses. Dans ce chapitre, nous avons présenté une approche de sous-échantillonnage qui se base sur les cartes topologiques. Cette solution guide le choix des données à supprimer dans un voisinage local, en prenant en considération la distribution et la topologie des données. Une série d'expériences a été réalisée pour valider la méthode proposée. Les résultats obtenus ont été comparés avec une méthode de sous échantillonnage connue qui nous a permis de mieux évaluer notre approche qui s'est avérée prometteuse comme solution au problème de déséquilibre des classes.



(a) Post-operative (b) Thyroid (c) Ecoli

(d) Satimage (e) Flag (f) Glass

FIGURE 2.5 – SNCR le processus d'auto -organisation. Le développement du sous-échantillonnage indiqué par la taille de la base de données à chaque itération. Chaque courbe montre les deux phases : Phase de nettoyage et phase de quantification.

Chapitre 3

Apprentissage à partir d'une seule classe par filtrage adaptatif

Introduction

Dans ce chapitre nous présentons des contributions concernant l'apprentissage à partir d'une seule classe. C'est un moyen permettant de contourner le problème des distributions déséquilibrées à un problème d'apprentissage à partir d'une seule classe. Pour ce faire, nous illustrons ce problème d'apprentissage à partir d'une seule classe par un problème de détection de nouveauté. En effet, le but essentiel de la détection de nouveauté est de repérer la nouveauté apportée par des données encore inconnues, en exploitant la connaissance extraite à partir d'un ensemble de données de référence (données d'apprentissage). Les données de référence se limitent à des données normales ou familières, du fait de la difficulté, voire l'impossibilité dans certains cas, d'identifier a priori ce qui constituerait une nouveauté par rapport aux données déjà connues (ce qui amène au problème de la classification à partir d'une seule classe). Les deux nouveaux modèles d'apprentissage que nous proposons sont des extensions d'un modèle de détection de nouveauté NDF (acronyme de "Novelty Detector Filter") proposé dès les années 1976 [22]. Le modèle NDF est un système d'apprentissage qui fonctionne en ligne et doté d'une capacité intéressante de classification avec des classes disjointes. Afin d'adapter le modèle NDF au problème de classification avec des distributions chevauchées, une extension a été proposée par Kassab et al, [26], [27] : ILoNDF pour "Incremental data-driven Learning of Novelty Detector Filter". De nouvelles méthodes d'initialisation, de nouvelles règles d'apprentissage et de nouvelles stratégies pour l'exploitation du modèle pour des fins de classifications ont donc été proposées. Par ailleurs, nous proposons

deux nouvelles extensions de ce modèle de détection de nouveauté, le modèle RS-NDF (Random Subspace Novelty Detection Filter) ainsi que le modèle SRS-NDF (Selected Random Subspace Novelty Detection Filter).

L'approche $RS - NDF$, que nous proposons, est fondée sur les opérateurs de projection orthogonale et l'idée de bootstrap. Notre approche, combine une technique de rééchantillonnage et l'idée d'apprentissage d'ensemble. RS-NDF est un ensemble de filtres ILoNDF, induits à partir d'échantillons bootstrap des données d'apprentissage, en utilisant une sélection aléatoire des variables pour l'apprentissage des filtres. RS-NDF utilise un rééchantillonnage avec remise sur les observations et un rééchantillonnage sans remise sur les variables. Chaque filtre est donc conçu dans un espace de description dont les composantes et la dimension sont choisies aléatoirement. La prédiction est faite par l'agrégation des prédictions de l'ensemble des filtres. Grâce à son algorithme d'apprentissage en ligne, l'approche RS-NDF est également en mesure de suivre les changements dans les données au fil du temps. La deuxième approche que nous proposons est une extension concernant notre modèle RS-NDF. En effet SRS-NDF présente l'avantage de sélectionner l'ensemble de filtres les plus performants permettant de donner de meilleurs résultats.

3.1 L'apprentissage à partir d'une seule classe et détection de nouveauté

Plusieurs travaux de recherche ont été proposés pour le problème de la détection de nouveauté [65], [66] et [67] avec une grande variété d'applications et méthodes. De façon générale, la problématique de la détection de nouveauté vise à identifier, dans un ensemble de données, celles qui différent significativement des autres, ne se conforment pas à un "comportement attendu" (qui est à définir ou à apprendre automatiquement), et qui indiquent par là un processus de génération différent. De nombreux termes sont utilisés pour désigner cette tâche, qui a été étudiée dans la communauté statistique dès le XIXème siècle, ou des problématiques très proches : on peut citer par exemple la détection des points aberrants, le traitement d'anomalies, la détection de bruit, ou encore le traitement d'exceptions. Les applications typiques de ce problème sont la détection de fraudes, la maintenance préventive, la détection des intrusions dans les réseaux, le diagnostic de maladies rares et de nombreux autres domaines. La détection de nouveauté est particulièment utile quand une classe importante est sous représentée dans les données. L'exemple typique de ce problème est la détection de fraudes où il peut y avoir un intervalle de plusieurs heures entre deux transactions frauduleuses. Le terme

"nouveauté" peut donc désigner une observation qui s'écarte des autres observations au point d'éveiller des soupçons. Cette observation peut être un comportement frauduleux, une intrusion dans un réseau, une panne imprévue dans le système, etc.

On distingue généralement trois grandes familles d'approches de détection de nouveauté :

- Les méthodes qui déterminent la nouveauté sans aucune connaissance préalable sur les données. Il s'agit essentiellement d'approches d'apprentissage analogues à la classification non supervisée. Ces approches traitent les données comme une distribution statique, et cherchent à identifier les points les plus éloignés. Ces points sont concidérés comme des valeurs potentiellement aberrantes ;
- Un deuxième type de méthodes consiste à modéliser à la fois les données normales et la nouveauté. Ces approches sont analogues à la classification supervisée et nécessitent des données pré-étiquetées.
- Et finalement des méthodes de détection de nouveauté analogues à une tâche de classification semi-supervisée où une petite partie des données étiquetées est disponible.

Dans le cadre de nos travaux, nous nous sommes situés au niveau de la troisième famille d'approches. Les approches auquelles nous nous sommes intéressées consistent à apprendre un modèle ou un ensemble de modèles sur des données normales disponibles et de l'employer après pour identifier la nouveauté apportée par les données entrantes. Différentes approches ont été proposées afin de résoudre le problème de l'apprentissage à partir d'une seule classe. Il s'agit des méthodes basées sur l'analyse en composantes principales (ACP), les réseaux de neurones auto-associatifs de type MLP (Multi Layer Perceptron), les SVM (Séparateur à Vaste Marge) monoclasses et les filtres détecteurs de nouveauté NDF. Dans notre travail, ces méthodes interprètent un problème à deux classes, généralement une classe majoritaire contenant les données de référence (données apprentissage) et une classe minoritaire contenant les données représentant la nouveauté.

3.1.1 Analyse en Composantes Principales

L'analyse en composantes principales (ACP) [40] fait partie du groupe des méthodes descriptives multidimensionnelles appelées méthodes factorielles. L'ACP permet, à partir d'un ensemble de n individus observés sur p variables, des représentations géométriques de ces individus et ces variables. L'ACP permet d'explorer les liaisons entre

variables et les ressemblances entre individus. L'ACP présente aussi une technique populaire et élémentaire pour la réduction de la dimensionnalité de données numériques. En effet, l'ACP est une transformation linéaire de l'ensemble des données de l'espace original dont les variables sont corrélées, vers un espace orthogonal où tous les axes, qui sont des combinaisons linéaires des variables d'origine, sont non corrélés deux à deux. Autrement dit, nous cherchons à définir k nouvelles variables combinaisons linéaires des p variables initiales qui feront perdre le moins d'informations possible. Ces variables seront appelées Composantes Principales, les axes qu'elles déterminent les Axes Principaux et les formes linéaires associées sont les Facteurs Principaux. Dans ce nouvel espace, les axes sont construits de telle sorte que le premier axe explique la variance maximale des données, et le deuxième axe, orthogonal au premier, explique la partie la plus grande de la variance résiduelle, et ainsi de suite. Pour l'analyse en composantes principales, le travail est effectué sur une approximation de la matrice de données, notée P. Donc soit X la matrice des données centrées contenant L observations et d variables. U les vecteurs propres de XX^T correspondant au k premières valeurs propres, P se présente comme suit :

$$P = U^T X \tag{3.1}$$

Après l'application de l'ACP, le sous-espace représentant le modèle (qui est généré par les k premières composantes principales) est associé à des variations systématiques dans les données. Ce qui nous amène à dire que l'ACP peut être vu comme une décomposition de l'espace des données positives en deux sous-espaces orthogonaux : le sous-espace factoriel (le modèle de l'ACP) et le sous-espace résiduel (les résidus de l'ACP). Ce dernier est associé aux variations aléatoires dues aux erreurs ou au bruit dans les données. Donc, toute donnée x_i peut être décomposée sous la forme suivante :

$$x = \hat{x} + \tilde{x} \tag{3.2}$$

$\hat{x} = (P_k P_k^T)x$ et $\tilde{x} = (I - P_k P_k^T)x$ représentent respectivement les projections de x_i sur le sous-espace modèle et le sous-espace résiduel. La correspondance des données à la classe positive (la classe utilisée pour l'apprentissage) peut alors être évaluée par l'erreur de prévision du résiduel définie comme suit :

$$||\tilde{x}||^2 = x_i^T (I - P_k P_k^T) x_i \tag{3.3}$$

3.1.2 Réseaux de neurones auto-associatifs de type MLP

Le Perceptron Multi-Couches PMC ou MLP (Multi-Layer Perceptron) [39] est un réseau de neurones organisé en couches. Une couche est un groupe de neurones uniformes sans connexion les uns avec les autres. Le MLP est constitué d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie et dispose d'une structure sans cycle (feed-forward), c'est à dire une couche ne peut utiliser que les sorties des couches précédentes et seuls les neurone de deux couches consécutives sont connectés. Chaque cellule possède une fonction de transition $f(x)$. Les réseaux de neurones auto-associatifs de type MLP, également connus sous le nom d'auto-encodeurs, sont un type spécial de réseaux multicouches de type $feedforward$. Ces réseaux sont entraînés pour produire une approximation de la fonction identité entre les entrées et les sorties du réseau. La dimension de la couche de sortie dans un réseau de neurones auto-associatif est égale à celle de la couche d'entrée. Une autre restriction est imposée au niveau des couches cachées, en effet la dimension d'une des couches cachées (la couche centrale) devrait être plus petite que celles des couches d'entrée ou de sortie (ou encore des autres couches cachées si le réseau en comporte plusieurs). Cette contrainte du nombre de neurones est dictée en vue de capturer les variables signiatives (similaires aux composantes principales) dans la représentation des données d'apprentissage en comprimant leurs redondances sans chercher à mémoriser les données. La transformation (entrée-sortie) peut être linéaire ou non linéaire suivant l'architecture du réseau et le type de fonctions de transfert de la couche de sortie. Lorsque la fonction de transfert est linéaire, le réseau réalise une analyse en composantes principales. Lorsque des fonctions non linéaires sont utilisées, le réseau pourrait résoudre des problèmes que l'analyse en composantes principales ne parvient pas à résoudre. Afin de résoudre le problème de la classification à partir d'une seule classe, l'utilisation d'un réseau de ce type se base sur l'apprentissage des poids des connexions du réseau w. Durant cette phase d'apprentissage, on présente les exemples familiers en entrée, et on adapte les poids des connexions du réseau de telle sorte que les sorties répliquent les entrées. Ces derniers sont donc les sorties désirées. L'adaptation des poids des connexions se fait le plus souvent à l'aide de l'algorithme de la rétropropagation, telle que la sortie est calculée ainsi :

$$\hat{x} = f\left[w_0^{(2)} + \sum_j w_j^{(2)} f(w_0^{(1)} + \sum_i w_i^{(1)}) x_i\right] \tag{3.4}$$

où :
$w^{(1)}$: matrice des poids de la première couche.

$w^{(2)}$: matrice des poids de la deuxième couche.

Le principe consiste à minimiser l'erreur quadratique (MSE) entre les sorties calculées et les sorties observées $\hat{x}$ en utilisant une descente de gradient :

$$MSE = 1/L \sum_{i=1}^{L} ||x_i - \hat{x}_i||^2 \tag{3.5}$$

où L représente le nombre d'exemples positifs utilisés pour l'apprentissage du réseau.

3.1.3 Les séparateurs à vastes marges : SVM

Une version des séparateurs à vastes marges SVM : le one-class (SVM) [41] est une méthode à une classe qui apprend la délimitation d'une zone de l'espace de description qui correspond aux données normales (exemple familiers). Les données situées à l'extérieur de cette zone sont prédites comme nouveautés. Cette méthode ne nécessite comme exemples d'apprentissage que des données de la classe normale. En termes plus précis, one-class SVM cherche une hypersphère de volume minimal qui englobe la plupart des exemples familiers ou normaux disponibles pour l'apprentissage. Plus précisément, soit $x_1, x_2, ..., x_L$ un ensemble de données d'apprentissage dans $\mathbb{R}^d$, le but de one-class SVM est d'estimer à partir de ces données une fonction $\phi : \mathbb{R}^d \to \mathbb{R}$ vérifiant que la plupart des éléments d'apprentissage appartiennent à un ensemble R $= x \in \mathbb{R}^d; \phi \geq 0$ de volume minimal. La relation d'appartenance d'une nouvelle données x à R indique si cet élément présente ou non une similarité avec les éléments de la classe d'apprentissage.

3.1.4 Le principe du filtre détecteur de nouveauté de Kohonen et Oja

En 1976, Kohonen et Oja [22] ont proposé le filtre détecteur de nouveauté (NDF). C'est un système linéaire adaptatif qui agit, après son apprentissage sur des données de référence, comme un opérateur de projection dans un espace vectoriel orthogonal à l'espace vectoriel engendré par les données de référence. Le modèle NDF laisse passer seulement les propriétés nouvelles d'une donnée par rapport à l'ensemble de données de référence déjà apprises ; les propriétés sont dites nouvelles si elles ne sont pas représentées dans les données de référence. Une autre description du filtre est donnée

dans [25]. Le filtre détecteur de nouveauté peut être implémenté par un réseau récurrent de neurones élémentaires, étroitement connectés en forme de boucles rétroactives, ou feedback, entre les neurones (figure 3.1).

Théoriquement, le modèle de filtre détecteur de nouveauté se fonde sur les propriétés

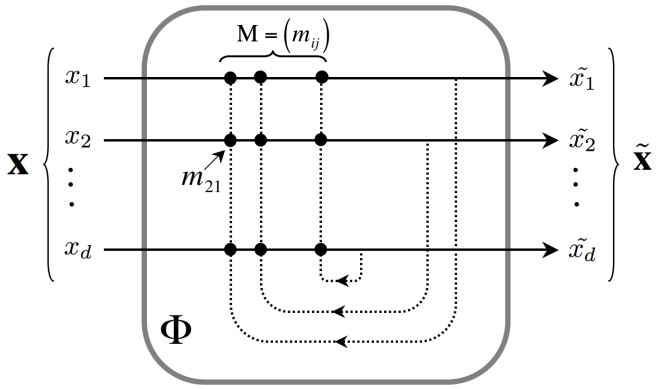


FIGURE 3.1 – L'architecture neuronale du modèle de filtre détecteur de nouveauté NDF

des opérateurs de projection orthogonale dans l'espace vectoriel $\mathbb{R}^d$: soit $x_1, x_2, ..., x_m$, m vecteurs distincts de $\mathbb{R}^d$ engendrant un sous-espace $\zeta \subset \mathbb{R}^d$. Le sous éspace complémentaire de ζ, noté $\zeta^{\perp}$, est engendré par l'ensemble des vecteurs de $\mathbb{R}^d$ orthogonaux à ζ. Alors tout vecteur $x \in \mathbb{R}^d$ peut être décomposé de façon unique sous forme d'une somme de deux vecteurs $\hat{x} \in \zeta$ et $\tilde{x} \perp \zeta$: $x = \hat{x} + \tilde{x}$. Une propriété particulière de la projection orthogonale est importante en approximation : pour toute décomposition de la forme $x = \dot{x} + \ddot{x}/\dot{x} \in \zeta$ dans le cas orthogonal nous avons la propriété suivante : $\| \ddot{x} \|$ est minimum et équivalente à la distance de x au sous-espace ζ (figure 3.2). Ceci est appelé le théorème de la projection.

La représentation matricielle des opérateurs de projection orthogonale donne une indication pour le calcul des vecteurs $\hat{x}$ et $\tilde{x}$ comme suit :
Soit X $\in \mathrm{R}^{d*m}$ dont les colonnes sont les vecteurs x_i et soit X^+ la pseudo-inverse de X alors XX^+ est un opérateur orthogonal qui permet de représenter la projection d'un vecteur x sur ζ sous la forme :

$$\hat{x} = (XX^+)x \tag{3.6}$$

de manière analogue, $I - XX^+$ est l'opérateur qui représente la projection de x sur $\zeta^{\perp}$:

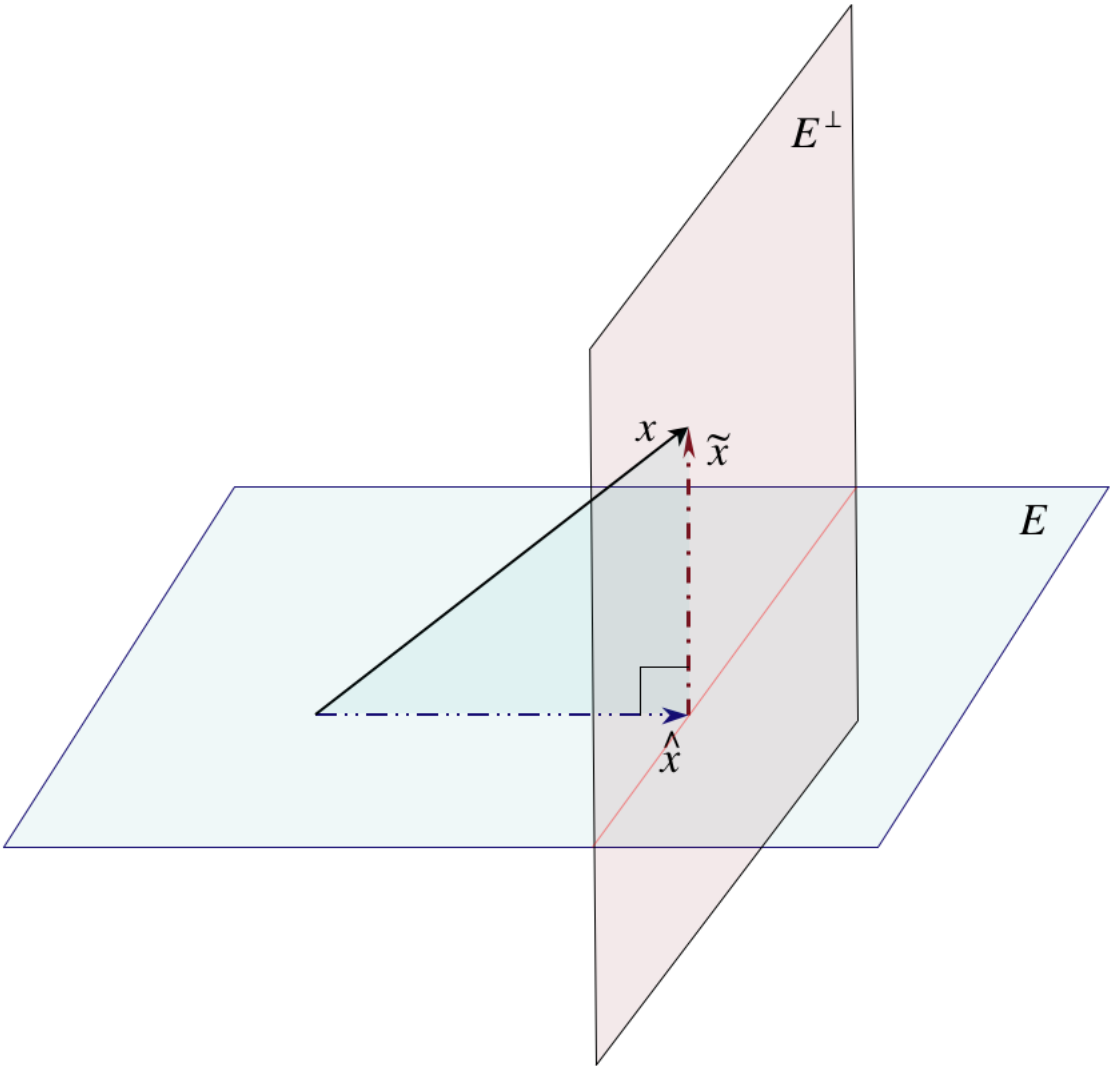


FIGURE 3.2 – Projection

$$\tilde{x} = (I - XX^{+})x \tag{3.7}$$

où I désigne la matrice identité d'ordre m. $\tilde{x}$ peut être considérée comme la contribution résiduelle de x qui subsiste après l'application d'une forme de traitement à x. Si un filtre est conçu dont la fonction de transfert est $(I - XX^{+})$ alors la sortie $\tilde{x}$ du filtre correspond à la composante de x qui est maximalement nouvelle.
Le comportement du modèle NDF s'avère équivalent, sous certaines conditions à celui des opérateurs de projection orthogonale. La sortie de chaque neurone est calculée comme une combinaison linéaire de l'entrée x_i et du feedback qu'il reçoit de la sortie.

L'opérateur de transfert est décrit par l'équation suivante :

$$\tilde{x_i} = x_i + \sum_j m_{ij}\tilde{x_j} \tag{3.8}$$

L'equation précedente peut s'écrire sous forme matricielle comme suit :

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{M}\tilde{\mathbf{x}} \tag{3.9}$$

Les poids des connexions rétroactives m_{ij} caractérisent l'état interne du réseau. Ils sont initialisés à zéro et, ensuite, mis à jour après la présentation de chaque donnée en entrée du réseau selon une règle d'apprentissage de type anti-Hebbian :

$$\frac{d\mathbf{M}}{dt} = -\alpha\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T \tag{3.10}$$

où α est un coefficient positif qui peut être modifié de manière adaptative au cours de l'apprentissage.

La fonction du transfert du réseau $\Phi \in \mathbb{R}^{d\times d}$ est définie par :

$$\tilde{\mathbf{x}} = (\mathbf{I} - \mathbf{M})^{-1}\mathbf{x} = \Phi\mathbf{x} \tag{3.11}$$

où I représente la matrice identité.

L'équation différentielle de Φ est obtenue de la manière suivante :

$$\begin{aligned} \frac{d\Phi^{-1}}{dt} &= -\Phi^{-1}\frac{d\Phi}{dt}\Phi^{-1} = -\frac{d\mathbf{M}}{dt} \\ \frac{d\Phi}{dt} &= -\alpha\Phi^2\mathbf{x}\mathbf{x}^T\Phi^T\Phi \end{aligned} \tag{3.12}$$

Cette équation est identifiable à une équation de Bernouilli du 4-ième degré dont la résolution est difficile. Kohonen et Oja [22] montrent que sous certaines hypothèses, des solutions asymptotiques stables existent

$$\Phi_c = I - XX^+ \tag{3.13}$$

ce qui représente l'équation caractéristique de l'opérateur de projection. Pour réduire la compléxité de calcul, une solution consiste donc à calculer directement l'état stable du filtre Φ en utilisant un algorithme d'estimation de la pseudo-inverse. Dans ce travail nous nous sommes intêressés au théorème de Greville [42]. Soit $X_k = [X_{k-1}, x_k]$ une matrice de k colonnes composée d'une sous matrice de $k-1$ colonnes et une colonne x_k, alors la pseudo-inverse de X_k se calcule comme suit :

$$X_k^+ = \begin{bmatrix} X_{k-1}^+(I - x_k {P_k}^T) \\ {P_k}^T \end{bmatrix}$$

avec

$$P_k = \begin{cases} \frac{(I - X_{k-1}X_{k-1}^+)x_k}{\|(I - X_{k-1}X_{k-1}^+)x_k\|^2} & \text{si} \quad \text{numerateur} \quad \neq 0 \\ \frac{(X_{k-1}^+)^T X_{k-1}^+ x_k}{1 + \|X_{k-1}^+ x_k\|^2} & \text{sinon} \end{cases}$$

La valeur initiale de X_1 est égale à la première colonne de X et :

$$X_1^+ = \begin{cases} {x_1}^T({x_1}^T x_1)^{-1} & \text{si} \quad x_1 \neq 0 \\ 0^T & \text{si} \quad x_1 = 0 \end{cases}$$

L'expression récursive pour estimer la fonction de transfert du réseau peut s'écrire de la manière suivante :

$$\Phi_k = (I - X_k X_k^+) = I - X_{k-1}X_{k-1}^+ - \frac{(I - X_{k-1}X_{k-1}^+)x_k {x_k}^T (I - X_{k-1}X_{k-1}^+)}{\| (I - X_{k-1}X_{k-1}^+)x_k \|^2} \quad (3.14)$$

Si l'on considère que $\Phi_{k-1} = (I - X_{k-1}X_{k-1}^+)$ représente la fonction de transfert de NDF après la présentation des $k-1$ premières données, il est possible de réécrire l'équation précedente sous la forme simplifiée suivante :

$$\Phi_k = \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T}{\| \tilde{\mathbf{x}}_k \|^2} \quad (3.15)$$

où $\mathbf{x}_k = [x_1, x_2, ..., x_d]^T$ est un vecteur de la matrice des données de référence.

Pendant la phase d'apprentissage, le modèle NDF s'habitue aux données présentées en entrée. Une fois l'apprentissage terminé, si une des donnée de référence ou une de

leurs combinaisons linéaires est présentée à l'entrée du modèle, la sortie correspondante sera nulle. D'autre part, si une donnée n'appartenant pas à l'espace formé par les données de référence est choisie comme entrée, la sortie correspondante ne sera pas nulle et elle peut être vue comme représentative des variables nouvelles extraites à partir de la donnée d'entrée vis-à-vis des données de référence qui ont été déjà apprises.

3.1.5 Le modèle ILoNDF : "Incremental data-driven Learning of Novelty Detector Filter"

Une extension du modèle de base NDF proposée par kassab et al, [26], [27] est le modèle d'apprentissage incrémental ILoNDF. Cette extension a été guidée par des réflexions sur le problème du modèle NDF. L'adaptation qui a été envisagée est directement liée au fait que l'apprentissage du modèle est guidé uniquement par la nouveauté : L'adaptation de l'état interne du modèle NDF décrite par la formule prcécedente ne considère que les variables représentant la nouveauté qu'apportent les données d'entrée. Donc pour faire participer toutes les variables des données d'entrée au processus d'apprentissage, la stratégie adoptée, consiste à introduire la matrice identité à chaque étape de l'apprentissage et de projeter les données d'entrée à la fois sur la matrice identité et sur la matrice du filtre. La nouvelle règle d'apprentissage s'écrit sous la forme suivante :

$$\Phi_k = \mathbf{I} + \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T}{\| \tilde{\mathbf{x}}_k \|^2} \tag{3.16}$$

où $\tilde{\mathbf{x}}_k = (\mathbf{I} + \Phi_{k-1})\mathbf{x}_k$ et Φ_0 est une matrice nulle.
Par la projection des données d'apprentissage sur la matrice identité, toutes les variables présentes dans ces données contribuent au processus de mise à jour de l'état interne du filtre, indépendamment de la quantité de nouveauté ou de la redondance qu'elles exhibaient. Aussi en projetant simultanément les données d'entrée x_k sur la matrice du filtre correspondant aux données antérieures, durant l'apprentissage et pendant qu'il progresse, les variables qui apparaissent fréquemment dans les données deviennent plus habituées par rapport aux autre moins fréquentes. Le grand avantage de la modification de la règle d'apprentissage du modèle original est donc lié à la capacité du modèle ILoNDF d'acquérir constamment de nouvelles connaissances relatives aux fréquences d'occurrence des variables et à leurs dépendances de co-occurrence dans les données utilisées pour faire l'apprentissage, ce qui rend le modèle robuste au bruit qui peut être présent dans la description des données d'apprentissage. En outre, le modèle ILoNDF

ne comporte aucun paramètre à régler avant ou pendant l'apprentissage ; il n'a donc aucun besoin de faire des calculs supplémentaires et coûteux en matière d'optimisation de paramètres.

Pour le problème de détection de nouveauté, deux proportions peuvent être calculées :

– *La proportion de nouveauté* : C'est une mesure qui permet de quantifier la nouveauté apportée par une donnée par rapport à un ensemble de données déjà apprises par le filtre.

$$N_{\mathbf{x}_i} = \frac{\parallel \tilde{\mathbf{x}}_i \parallel}{L \times \parallel \mathbf{x}_i \parallel} \tag{3.17}$$

où L est le nombre d'exemples utilisés pour l'apprentissage.

– *La proportion d'habituation* : C'est un indicateur de similarité d'un exemple vis-à-vis des données précédemment apprises :

$$H_{\mathbf{x}_i} = 1 - N_{\mathbf{x}_i} \tag{3.18}$$

Cette proportion pourrait être considérée comme le score de classification d'un exemple x_i. Elle indique la probabilité que x_i appartient à la nouvelle classe. Il est également possible de calculer un vecteur P_Φ qui représente les exemples utilisés pour l'apprentissage. Ainsi, le score de classification des données d'entrée peut être calculé comme expliqué ci-dessous :

$$P_\Phi = \sum_{f \in F} H_f \overrightarrow{u_f} \tag{3.19}$$

où $\overrightarrow{u_f}$ est le vecteur unité associé à chaque variable f dans F (l'ensemble des variables) et H_f est la proportion d'habituation d'une variable f calculée comme suit :

$$H_f = 1 - \frac{\parallel \Phi \overrightarrow{u_f} \parallel}{n \times \parallel \overrightarrow{u_f} \parallel} \tag{3.20}$$

Le vecteur P_Φ est défini comme un vecteur des poids. Les composantes de P_Φ représentent la proportion d'habituation de chaque variable dans l'espace de données.

3.2 Nos contributions

3.2.1 Le modèle RS-NDF : " Random Subspace Novelty Detection Filter"

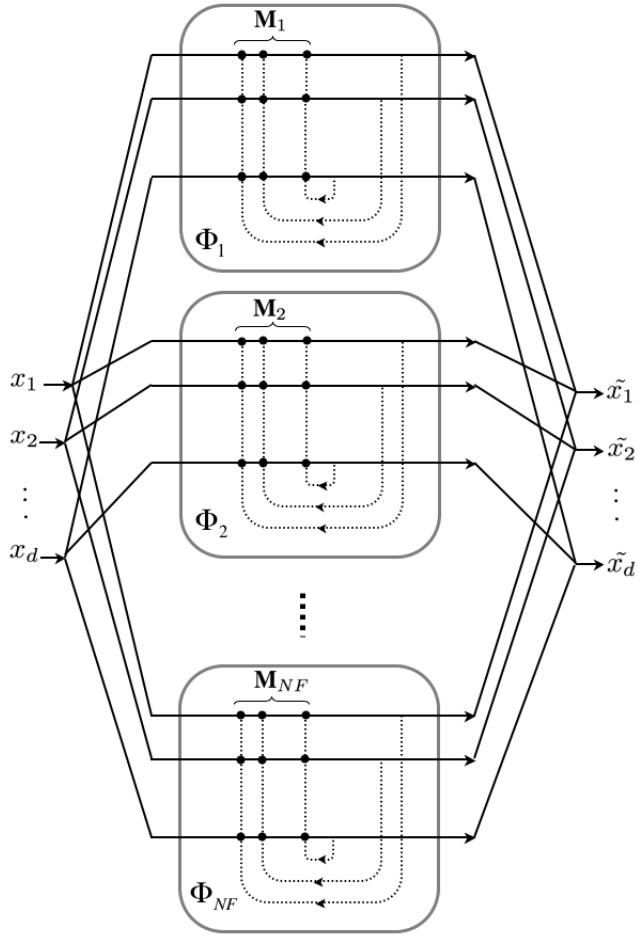


FIGURE 3.3 – L'architecture neuronale du modèle RS-NDF

Les approches ensemble ont été proposée pour traiter le problème de déséquilibre des classes. Le principe de ces techniques permet de rendre n'importe quel type d'algorithme sensible à l'asymétrie, notamment par des méthodes de boosting ou du bagging. Le Bagging (Bootstrap aggregating) est basé sur un processus stochastique de modification de l'échantillon d'apprentissage A pour créer un ensemble de classifieurs diversifié. Cette méthode consiste à construire chaque hypothèse à partir d'un ré-échantillonnage par bootstrap des données d'apprentissage, sachant qu'un bootstrap est le tirage aléatoire avec remise, sur un échantillon de taille n d'un échantillon initial de même taille (c'est à dire n). Les hypothéses ainsi

construites sont ensuite combinées par un vote majoritaire.
RS-NDF est basée sur un ensemble de $ILoNDF$ calculés sur plusieurs échantillons. Ces derniers sont obtenus par un double bootstrap (Un tirage aléatoire avec remise sur les données et un tirage aléatoire sans remise pour les variables). La prédiction est faite en agrégeant les prévisions de l'ensemble. Donc l'idée de notre algorithme $RS-NDF$ combine les opérateurs de projection orthogonale utilisées par Kohonen et Oja dans le modèle NDF [22], la technique de bootstrap et le principe de l'apprentissage d'ensemble. L'architecture de ce système est présentée par la figure (3.3).

L'algorithme d'apprentissage $RS-NDF$ est le suivant :

Algorithme 1: Random Subspace Novelty Detection Filter

Entrées :
$sD = x_1, x_2, ..., x_L$ ensemble de données d'apprentissage.
$sT = x_1, x_2, ..., x_M$ ensemble de données de test.
$F = f_1, f_2, ..., f_n$ ensemble de variales.
NF nombre de filtres.
Φ_0 la matrice initial.
Sorties :
NF filtres (Φ_i).
NF vecteurs qui représentent la classe (Pv_i).
D matrice de détection de nouveauté.
Begin
for $i = 1$ jusqu'à NF **do**
  Un tirage aléatoire avec remise sur les données d'apprentissage.
  Un tirage aléatoire sans remise sur les variables.
  Appliquer RS-NDF Φ_i avec les variables choisies aléatoirement (formule (3.16))
end for
for $i = 1$ jusqu'à M **do**
  for $k = 1$ jusqu'à NF **do**
    Calculer la proportion d'habituation $H_{\mathbf{x}_i}$ (formule (3.18)) pour $\mathbf{x}_i \in sT$ en utilisant le filtre Φ_k.
  end for
  Agréger les prédictions de l'ensemble des filtres et mettre la prédiction finale dans D.
end for
End

Pour déterminer le seuil de détection de chaque filtre, nous avons appliqué les règles suivantes :

– Les scores (sorties du filtre) attribués aux données d'apprentissage peuvent être utilisés comme un bon indicateur des scores de données qui peuvent être positives et qui sont faciles à détecter car elles sont fortement similaires aux données d'apprentissage. Par conséquent, la moyenne de ces scores peut être admise comme une limite supérieure pour le seuil de détection.

– Les scores attribués aux données disponibles pour l'apprentissage avant leur utilisation, peuvent être utilisés comme un bon indicateur des scores de données qui sont positives mais qui sont moins faciles à détecter. Par conséquent, la moyenne de ces scores peut être admise comme une limite inférieure pour le seuil de détection.

3.2.2 Le modèle SRS-NDF : "Selected Random Subspace Novelty Detection Filter"

L'approche RS-NDF proposée consiste à construire un ensemble de filtres adaptatifs. Chacun des filtres est obligatoirement censé donner une réponse face à l'arrivée d'une nouvelle observation. Nous avons donc opté pour la methode la plus "naturelle" qui consiste à évaluer tous les sous-modèles. Afin d'améliorer l'algorithme $RS-NDF$ pour de meilleurs performances en détection de nouveauté, nous avons proposé une stratégie, plus économe, permettant de réduire le nombre de modèles à estimer. L'objectif est de choisir parmi l'ensemble des modèle (filtres), le sous ensemble qui permet d'atteindre les meilleurs performances. La nouvelle approche SRS-NDF pour "Selected Random Subspace Novelty Detection Filter" consiste à sélectionner, parmi tous les filtres calculés auparavant par RS-NDF, ceux qui donnent des performances meilleures en terme de pertinence. Il s'agit donc d'une approche de sélection de modèles qui classe d'abord l'ensemble des filtres d'origine, c'est à dire tous les filtres obtenus par RS-NDF, selon le critère de pertinence et dans un ordre décroissant. Ensuite, nous utilisons le "scree test" pour choisir l'ensemble des modèles (filtres) les plus prometteurs.
Afin d'établir cette sélection, SRS-NDF opère par une évaluation de l'indice de précision et la diversité des filtres générés par RS-NDF. La solution finale est obtenu par la combinaison des résultats donnés par l'ensemble des filtres sélectionnés.
La pertinence de chaque filtre fl_i est calculée par la fonction suivante :

$$Pertinence_{\alpha}(fl_i) = \alpha \times IBA_{\alpha}(fl_i) + (1-\alpha) \times mean(Div_{\alpha}(fl_i); fl_i); i \in [1, NF] \tag{3.21}$$

tels que $IBA_{\alpha}(fl_i)$ et $Div_{\alpha}(fl_i)$ représentent respectivement "l'indice de précision equilibréé" de fl_i (Index of balanced accutracy) et la diversité du filtre fl_i. Le IBA est défini par le produit de deux termes : la dominance et le G-mean. Le premier terme est une mesure permettant d'évaluer la prédiction correcte de chaque filtre. Quand au second terme, il est défini par une moyenne géométrique des précisions mesurées pour chaque classe. L'avantage de l'utilisation du critère IBA est le fait que cette mesure est capable de distinguer la contribution de chaque classe dans la performance globale du système. La diversité de deux classificateurs consiste à attribuer des étiquettes différentes aux mêmes exemples. Plusieurs mesures ont été proposées pour estimer la diversité entre deux classificateurs. Pour notre approche nous avons retenu la distance moyenne Frobeinius entre la matrice de transfert du filtre fl_i et les autres filtres de $RS - NDF$. Le coefficient α, $0 \leq \alpha \leq 1$, est un parammètre de contrôle qui sert à équilibrer la précision et la diversité. En effet, pour chaque valeur de α, les pertinences de tous les filtres sont calculées. Ensuite l'ensemble des filtres les plus "pertinents" est sélectionné par la méthode statistique : scree test.

Le test Statistique (Scree test) a été développé pour fournir une visualisation permettant de sélectionner les valeurs propres pour l'analyse en composantes principales. L'idée principale consiste à générer une courbe de valeurs propres permettant d'identifier un comportement aléatoire. Le nombre de composantes retenues est égale au nombre de valeur précédant le "scree". Souvent le "scree" apparaît quand la pente de la courbe change radicalement. Il faut donc identifier le point représentant le maximum de ralentissement dans la courbe.
Supposant que nous disposons du vecteur de pertinence :
$\mathbf{Per}_{\alpha} = (Per_{1\alpha}, Per_{2\alpha}..., Per_{j\alpha}, ..., Per_{NF\alpha})$.
Le fonctionnement du scree test est le suivant :

Algorithme 2: Scree Test Acceleration Factor

1) Classer les pertinences $\mathbf{Per}_\alpha$ par ordre décroissant. Ensuite nous avons un nouvel ordre $\mathbf{Per}_\alpha = (Per^1_{\cdot\cdot}, Per^2_{\cdot\cdot}..., Per^i_{\cdot\cdot}, ..., Per^{NF}_{\cdot\cdot})$; tel que Per^i indique l'ordre de l'indice.
2) Calculer la première différence $df_i = Per^i_{\cdot\cdot} - Per^{i+1}_{\cdot\cdot}$;
3) Calculer la deuxième différence (acceleration) $acc_i = df_i - df_{i+1}$
4) Trouver le scree : $\max_i (abs(acc_i) + abs(acc_{i+1}))$
5) Couper et considérer tous les filtres jusqu' àu le scree ; (utiliser les indices initiaux des filtres avant leur classement)

L'algorithme SRS-NDF :

Algorithme 3: Selected Random Subspace Novelty Detection Filter

Répéter pour chaque valeur de $\alpha \in [0, 1]$

1) Construire le $RS - NDF$ avec NF filtres.
2) Calculer la valeur de IBA des filtres $= IBA_\alpha(1), ..., IBA_\alpha(NF)$.
3) Calculer la divérsté des filtres $= Div_\alpha(1), ..., Div_\alpha(NF)$.
4) Calculer la pertinence des filtres $= Pertinence_\alpha(1), ..., Pertinence_\alpha(NF)$. $Per_\alpha(fl_i) = \alpha \times IBA_\alpha(fl_i) + (1 - \alpha) \times meanDiv_\alpha(fl_i); fl_i; i \in [1, NF]$
5) Sélectionner le sous-ensemble des modèles en utilisant le ScreeTest $= SelectedFilters_\alpha$
6) Calculer la valeur de IBA pour les filtres séléctionnés $= IBA_{SelectedFilters_\alpha}$
7) $\alpha = \alpha + 0, 1$

Jusqu' à $\alpha = 1$

– Sélectionner l'ensemble avec la meilleure valeur de IBA
– Agréger les prédictions de l'ensemble sélectionné et sauvegarder le résultat de nouveauté dans D.

3.3 Vers une description unifiée des différentes approches

Si nous considérons notre problème de détection de nouveauté comme un problème de filtrage, les différentes techniques d'apprentissage citées précédemment seront considérées comme des filtres avec des fonctions de transfert (fonctions de filtrage)

F, une mesure de nouveauté Q et un critère de décision D. Pour chacune des techniques, après son apprentissage, si une nouvelle donnée x se présente au modèle, la sortie du système s'écrit sous la forme suivante :

$$\tilde{x} = Fx \tag{3.22}$$

où $\tilde{x}$ représente la nouveauté apportée par la donnée x par rapport à l'ensemble d'apprentissage.

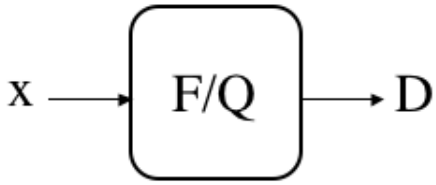


FIGURE 3.4 – La fonction de filtrage

Une proportion de nouveauté Q est également calculée pour chaque donnée. La valeur de Q représente une indication sur la proportion de nouveauté apportée par la donnée x par rapport à l'ensemble d'apprentissage et nous permet aussi de conclure une décision D binaire qui nous indique si la donnée est nouvelle ou non. Pour les différentes techniques proposées nous pourrons les décrire d'une manière unifiée de la façon suivante : La "fonction de filtrage" F et la "Proportion de nouveauté" Q.

L'Analyse en Composante Principale : ACP

-Fonction de filtrage pour l'ACP :

$$F = I - UP = I - UU^T x \tag{3.23}$$

-Proportion de nouveauté : L'erreur de prévision du résiduel définie

$$Q = ||\tilde{x}||^2 = x_i^T (I - P_k P_k^T) x_i \tag{3.24}$$

Le modèle NDF

-Fonction de filtrage pour le NDF :

$$F = \Phi_k = \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T}{\| \tilde{\mathbf{x}}_k \|^2} \tag{3.25}$$

- Proportion de nouveauté :

$$Q = N_{\mathbf{x}_i} = \frac{\parallel \tilde{\mathbf{x}}_i \parallel}{L \times \parallel \mathbf{x}_i \parallel} \tag{3.26}$$

où L est le nombre d'exemples utilisés pour l'apprentissage.

Le modèle RS-NDF : Opérateurs de projection orthogonale pour la détection de nouveauté

-Fonction de filtrage pour le RS-NDF :

$$F = [\mathbf{I} + \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T}{\parallel \tilde{\mathbf{x}}_k \parallel^2}]_{i=1}^{NF} \tag{3.27}$$

- Proportion de nouveauté :

$$Q = [N_{\mathbf{x}_i}]_{i=1}^{NF} = [\frac{\parallel \tilde{\mathbf{x}}_i \parallel}{L \times \parallel \mathbf{x}_i \parallel}]_{i=1}^{NF} \tag{3.28}$$

Réseaux de neurones auto-associatifs de type MLP

-Fonction de filtrage pour le MLP :

$$\hat{x} = F(x) = f\left[w_0^{(2)} + \sum_j w_j^{(2)} f(w_0^{(1)} + \sum_i w_i^{(1)} x_i)\right] \tag{3.29}$$

-Proportion de nouveauté : L'erreur quadratique (MSE) entre les sorties désirées et les sorties observées :

$$Q = 1/L \sum_i ||x_i - \hat{x}_i||^2 \tag{3.30}$$

Les séparateurs à vastes marges : SVM

-Fonction de filtrage pour les SVM

$$F = \sum_i \alpha_i K(x_i, x) \tag{3.31}$$

-Proportion de nouveauté :

$$Q = sgn(\sum_i \alpha_i K(x_i, x) - \rho) \tag{3.32}$$

3.4 Validation

3.4.1 Description des bases de données

Nous avons utilisé plusieurs jeux de données [29] disponibles sur :
http : *//archive.ics.uci.edu/ml/datasets.html*.
Pour notre expérimentation, nous avons adapté les différentes bases de données utilisées au contexte de la détection de nouveauté et l'apprentissage à partir d'une seule classe. Une classe a été choisie aléatoirement dans chaque base de données. Cette classe a été considérée comme étant la classe nouveauté. Les autres classes restantes sont fusionnées pour fournir la classe normale pour l'apprentissage. La description des données est résumée dans le tableau suivant :

Base de données	Dimension	Taille	Taille de la classe nouveauté
Glass	9	214	70
Ionosphere	34	351	225
Oil	48	937	41
Spectf	44	187	15
Waveform	21	5000	1647
WDBC	30	569	212
Vin	13	178	59
Yeast	8	1484	244

TABLE 3.1 – Description des bases

– La base Ionosphère : cette base radar est composée de 351 observations décrites par 34 variables et divisées en 2 classes : *Bien* et *PasBien*. La figure 3.5 montre une projection ACP du nuage de données Ionosphère dont les cercles représentent les données "nouveauté".

– La base Oil : cette base est composée de 937 observations divisées en deux classes [34]. Une classe qui contient 41 exemples et une autre de 896 exemples. A l'origine cet ensemble de données dispose de 50 variables.

– La base Spectf : les données sont divisées en un ensemble d'apprentissage et un ensemble de test. Nous avons utilisé seulement les données de test composées de 187 observations. Chacun des patients (observations) est classé en deux catégories : normale et anormale.

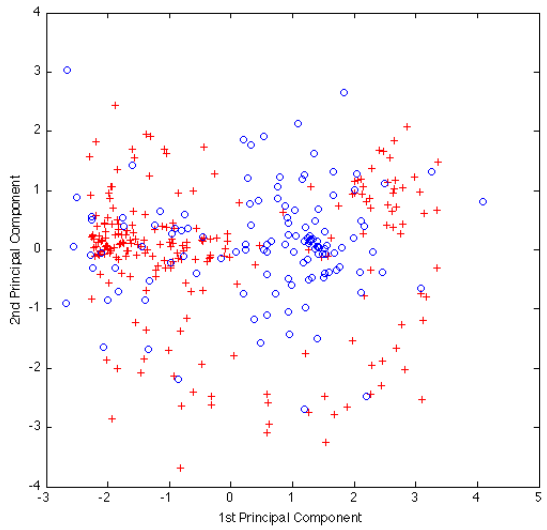


FIGURE 3.5 – Projection ACP de la base Ionosphere

– La base Waveform [36] est composée de 5000 exemples divisés en 3 classes. La base originale comportait 21 variables. Chaque observation a été générée comme une combinaison de 2 sur 3 vagues. La figures 3.6 montre une projection ACP du nuage de données de cette base.

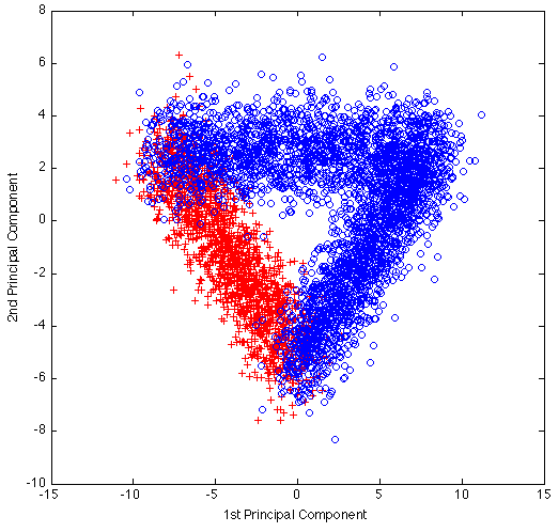


FIGURE 3.6 – Projection ACP de la base Waveform

– La base Wisconsin Diagnostic Breast Cancer (WDBC) : ce jeu de données contient 569 individus qui sont décrits par 32 variables. 357 individus sont atteint de cancer bénin et les 212 autres ont des cancers malin. Les variables décrivent les caractéristiques des noyaux de cellules présentes dans l'image numérique.
– La base Vin : ces données sont les résultats d'une analyse chimique du vin produit dans une même région en Italie [37]. Cet ensemble de données se compose de 178 observations décrites par 13 variables et divisées en trois classes.

– La base Yeast : ce jeu de données [38] se compose de 1484 observations représentant dix classes et décrites par 9 attributs. Les classes sont difficiles à séparer dans ces données. La figure 3.7 montre une projection ACP du nuage de données de cette base.

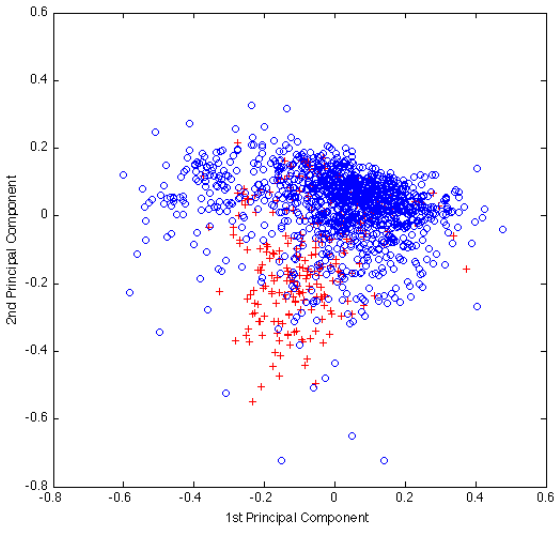


FIGURE 3.7 – Projection ACP de la base Yeast

3.4.2 Les mesures de performances et le protocole expérimental

Différentes mesures d'évaluation ont été proposées afin d'évaluer les performances des différents algorithmes d'apprentissage. L'évaluation empirique des algorithmes est une question qui occupe plusieurs chercheurs. La plupart des mesures que nous avons utilisé dans notre travail se concentrent sur la capacité

des classifieurs à identifier correctement les observations de chaque classe. Les mesures de qualité utilisées dans notre travail sont calculées à partir de la matrice de confusion décrite dans le chapitre précédent et qui sont les suivantes : $Acc-$, $Acc+$, $Precision$, $Fmeasure$, $G-mean$ et AUC.

$$Precision = \frac{VP}{VP + FP} \tag{3.33}$$

$$Fmeasure = \frac{2 * Precision * Rappel}{Precision + Rappel} \tag{3.34}$$

$$G - mean = (Acc - *Acc+)^{1/2} \tag{3.35}$$

Nous avons utilisé toutes ces mesures pour comparer nos méthodes $RS-NDF$ et $SRS-NDF$ avec des approches couramment utilisées pour le problème d'apprentissage à partir d'une seule classe. Les résultats de ces différents paramètres de performance sont obtenus suite à une validation croisée (10 fois).

3.4.3 Résultats

3.4.3.1 Résultats pour RS-NDF

Pour chaque base de données, les cinq approches ont été testées et leurs résultats ont été évalués en termes des différentes mesures de performances citées. Le tableau 3.2 montre les performances des différents algorithmes sur la base Vin.

	$Acc-$	$Acc+$ $(Rappel)$	$Prec$	$F-$ $measure$	AUC_b	$G-$ $mean$
MLP	0,78	0,68	0,83	0,81	0,73	0,73
ACP	0,60	0,69	0,80	0,68	0,65	0,64
SVM-1C	0,68	0,76	0,81	0,73	0,72	0,71
NDF	0,84	0,78	0,88	0,86	0,81	0,81
RS-NDF	**0,87**	**0,85**	**0,92**	**0,89**	**0,86**	**0,86**

TABLE 3.2 – Comparaison des performances sur la base de données Vin

À partir des résultats obtenus, notre approche $RS-NDF$ montre un fonctionnement meilleur par rapport à toutes les autres méthodes : MLP, ACP, $SVM-1C$ et NDF. Cette amélioration a touché les différents critères d'évaluation.
Les tableaux 3.3 et 3.4 montrent respectivement les performances de $RS-NDF$ et les différents algorithmes utilisés sur la base $Ionosphere$ et la base $Glass$. Ces

résultats montrent que notre approche dépasse le MLP, ACP et $SVM-1C$ avec toutes les mesures de performances utilisées ($Acc+$, $Precision(Prec)$, $F-mesure$, $G-mean$ et AUC_b) seulement pour l'$Acc-$, notre approche donne un résultat légèrement inférieur à celui de NDF pour la base *ionosphere* et à MLP, $SVM-1C$ et ACP pour la base $Glass$. Ces diférents résultats montrent la capacité de notre approche à détecter la classe positive.

	$Acc-$	$Acc+$ ($Rappel$)	$Prec$	$F-measure$	AUC_b	$G-mean$
MLP	0,63	0,64	0,50	0,55	0,64	0,64
ACP	0,64	0,55	0,45	0,52	0,59	0,59
1-SVM	0,66	0,56	0,45	0,54	0,61	0,60
NDF	**0,90**	0,61	0,57	0,70	0,76	0,74
RS-NDF	0,87	**0,74**	**0,65**	**0,74**	**0,80**	**0,80**

TABLE 3.3 – Comparaison des performances sur la base de données Ionosphere

	$Acc-$	$Acc+$ ($Rappel$)	$Prec$	$F-measure$	AUC_b	$G-mean$
MLP	**0,96**	0,49	0,49	0,65	0,73	0,69
ACP	**0,96**	0,43	0,46	0,62	0,69	0,64
1-SVM	0,89	0,54	0,80	0,84	0,72	0,69
NDF	0,82	0,69	0,84	0,83	0,75	0,75
RS-NDF	0,87	**0,84**	**0,93**	**0,90**	**0,86**	**0,85**

TABLE 3.4 – Comparaison des performances sur la base de données Glass

Dans les tableaux ci-dessus, nous avons montré l'amélioration apportée par notre approche $RS-NDF$ par rapport aux méthodes utilisées. Les résultats sont également confirmés par une inspection visuelle. En regardant les graphiques des quatre radars (figure (3.8)), certaines conclusions peuvent être tirées. En général $RS-NDF$ donne de meilleurs résultats par rapport aux autres méthodes et avec tous les critères de performances utilisés. Dans le pire des cas, les résultats obtenus par NDF, ACP, MLP et $SVM-1C$ sont légèrement supérieurs aux résultats de notre approche. Pour la base de données $Waveform$, $RS-NDF$ dépasse les autres algorithmes en utilisant la $Precision$, $F-mesure$, $G-mean$ et AUC. La précision est définie comme le pourcentage d'exemples correctement

étiquetés comme positif. Nous pouvons donc conclure que notre approche classe les exemples positifs mieux que les autres méthodes. Aussi pour les critères $G-mean$, et AUC, $RS-NDF$ donne de meilleurs résultats comparé à NDF, MLP, ACP et $SVM-1C$.

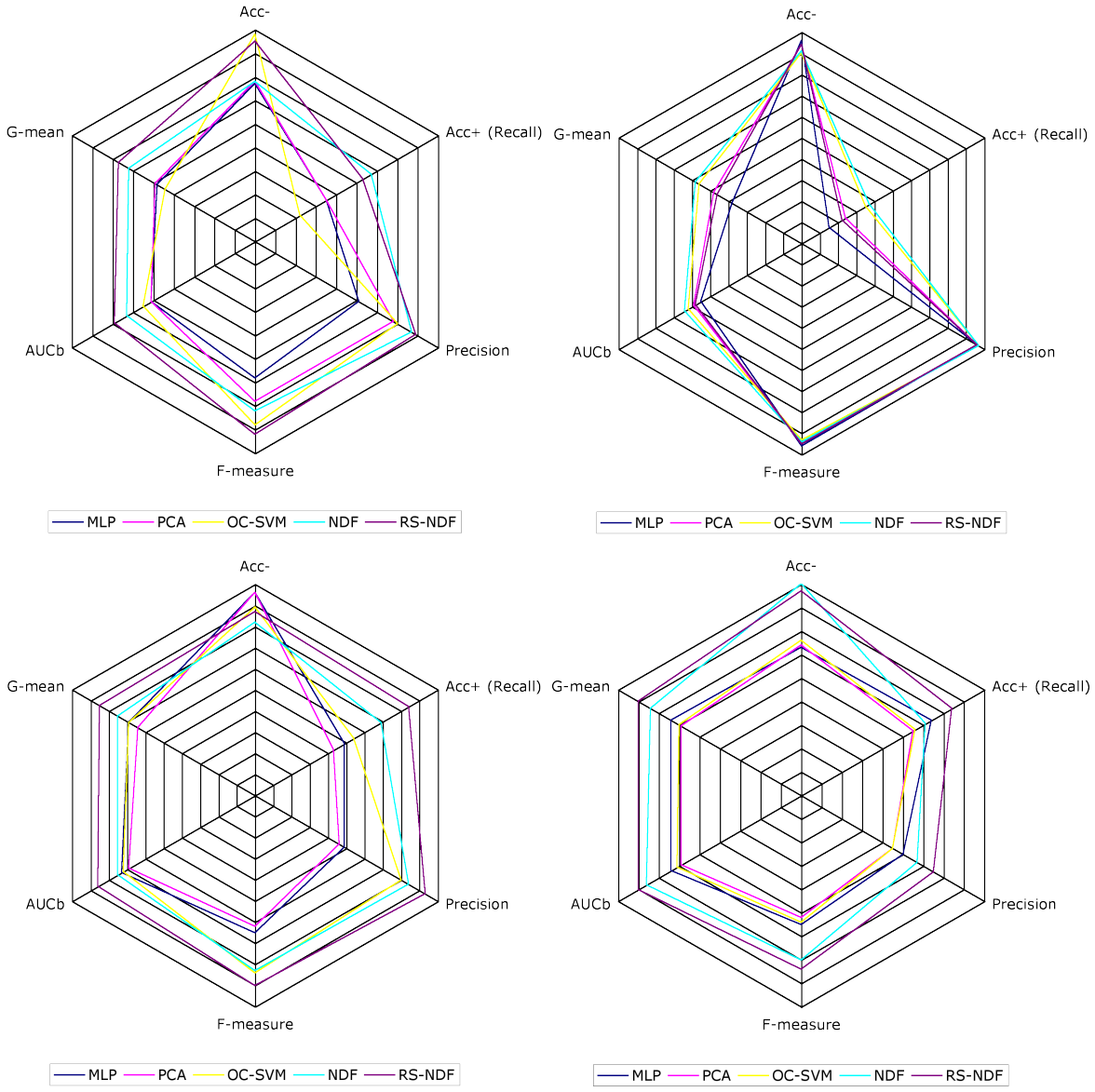


FIGURE 3.8 – Radars des bases : Waveform, Oil, Glass et Ionosphère

Aussi les graphiques des radars des bases de données Vin, $Yeast$, $WDBC$ et $Spectf$ (figure (3.9)) confirment les bonnes performances de notre approche $RS-NDF$. En effet les valeurs de AUC montrent que notre approche est en mesure de donner des résultats intéressants. Cette mesure de performance

est une représentation quantitative de la courbe ROC. La courbe ROC est largement utilisée pour l'évaluation des classifieurs, c'est un outil de visualisation, organisation et sélection des algorithmes tout en se basant sur le compromis entre les vrais positifs et les faux négatifs.

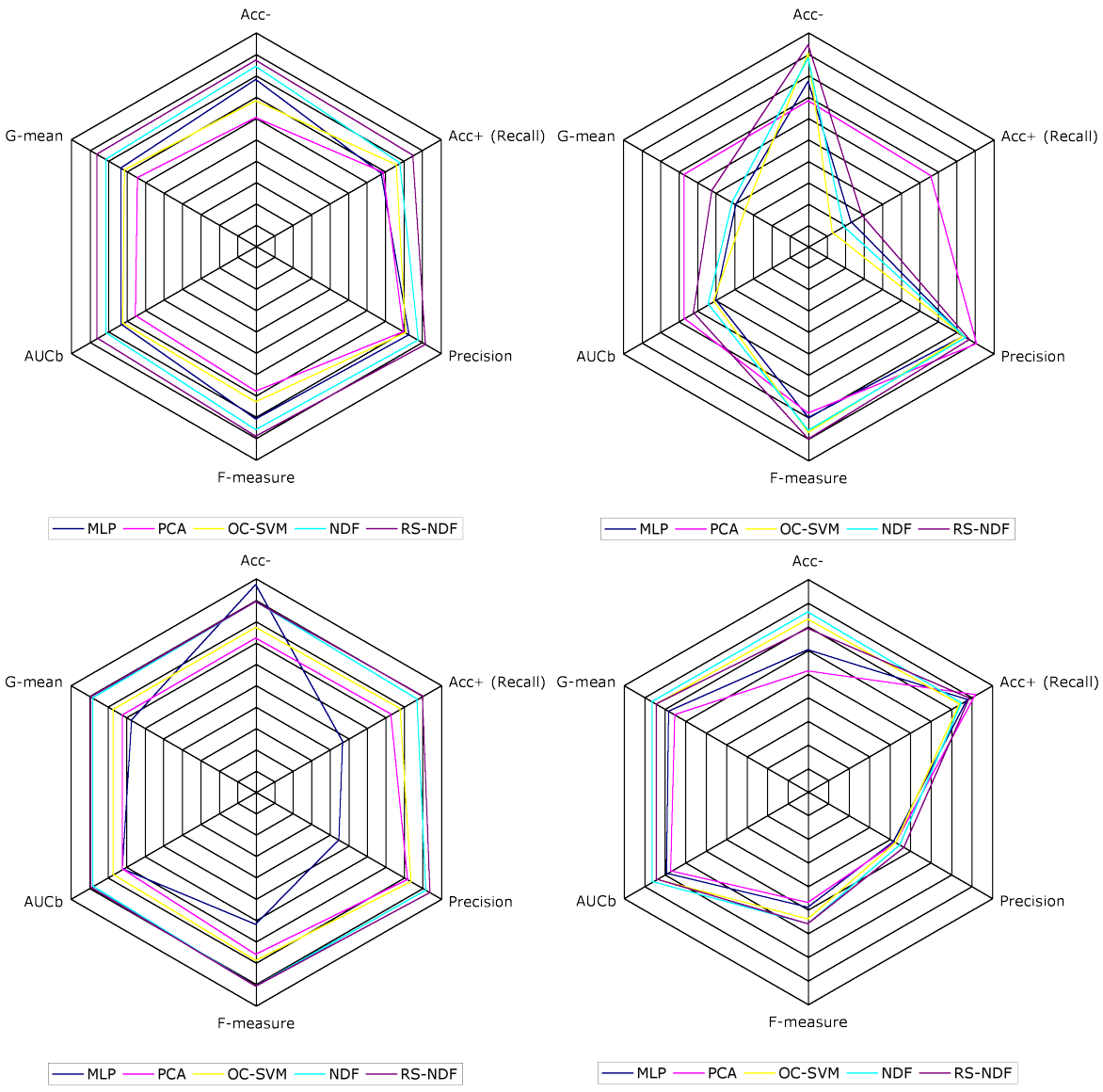


FIGURE 3.9 – Radars des bases : Vin, Yeast, WDBC et Spectf

La figure 3.10 montre la sortie de NDF (signal en bleu (en bas)) ainsi que celle de $RS-NDF$ (signal de couleur rouge (en haut)). On peut remarquer que les réponses fournies par le $RS-NDF$ sont plus importantes que celles fournies par l'approche NDF. Il y a une différence visible de variations entre les signaux produits par $RS-NDF$ et NDF. En fait, $RS-NDF$ montre une plus grande

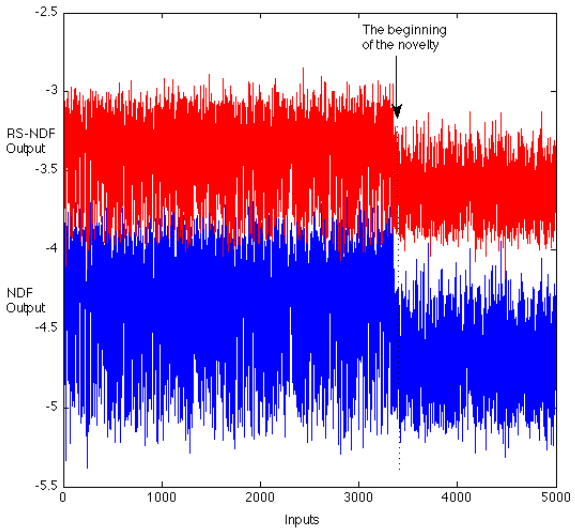


FIGURE 3.10 – Le signal de sortie de RS-NDF et NDF

stabilité dans les réponses de détection de nouveauté que l'approche NDF. Il est possible aussi de distinguer entre les deux classes.

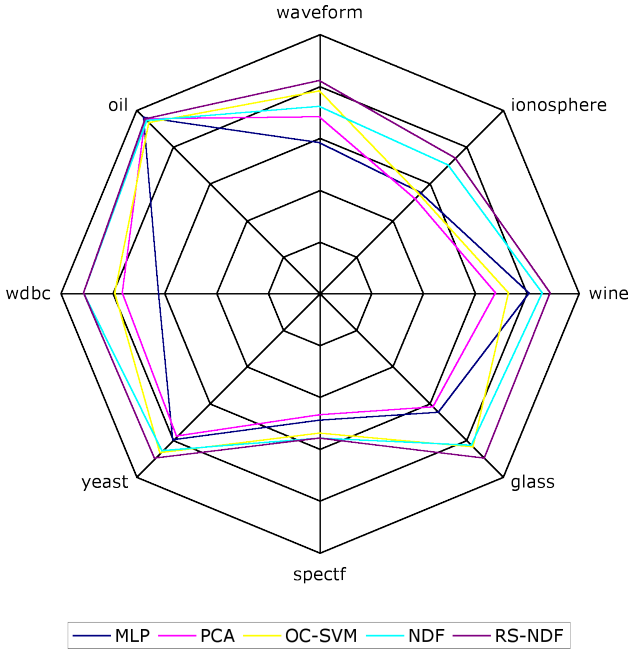


FIGURE 3.11 – La F-measure sur les différentes bases de données

D'après la figure 3.11 qui montre le résultat de F-measure pour toutes les bases de données, nous remarquons que l'algorithme $RS-NDF$ donne d'excellents résul-

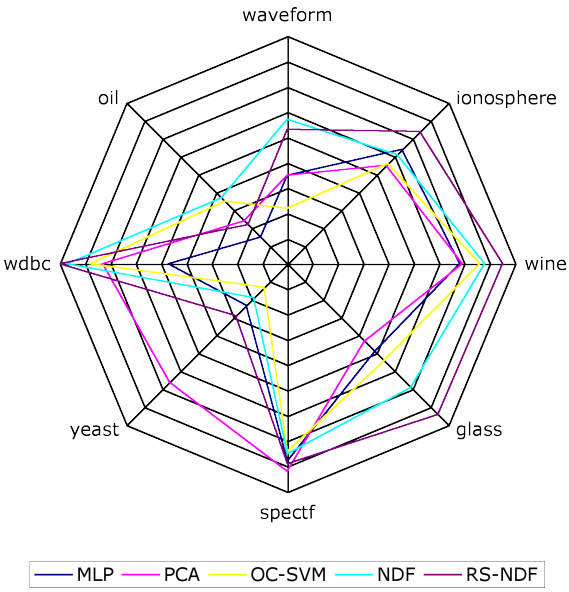


FIGURE 3.12 – La mesure Acc+ sur les différentes base de données

tats et dépasse toutes les autres méthodes pour les bases Vin, $Yeast$, $Ionosphre$ et $Spect$. Par contre il donne un résultat légèrement inférieur (0,95) à MLP (0,96) sur la base Oil et le même résultat que NDF (0,91) sur la base $WDBC$. La figure 3.12 montre le radar de la mesure $Acc+$ qui représente la capacité des modèles à détecter la classe nouveauté. Nous avons choisi cette métrique pour montrer la bonne capacité de $RS-NDF$ à détecter la classe nouveauté. Ainsi, nous pouvons voir clairement que notre approche donne de meilleurs résultats comparé aux autres méthodes.

3.4.3.2 Résultats pour SRS-NDF

Pour les bases de données Vin, Waveform, Spectf, Yeast et Oil, notre approche $SRS-NDF$ a été évaluée sur les différentes métriques de performances. Les résultats obtenus ont été comparés avec les approches citées auparavant. Dans le tableau 3.5, nous montrons les différentes valeurs obtenues pour chaque technique. Ces résultats montrent le bon fonctionnement de l'approche $SRS-NDF$, elle arrive à trouver des meilleurs performances surtout pour le critère $Acc+$. En effet, ce critère d'évaluation est très important dans la mesure où il nous montre la capacité du modèle à détecter les données nouveautés. Notre approche arrive aussi à dépasser toutes les autres techniques utilisées pour la base Vin à titre d'exemple et trouve aussi des bons résultats sur les autres bases utilisées.
Nous constatons clairement que notre approche fourni des résultats stables et dans certains cas de meilleurs résultats. Nous observons par exemple que les

$SVM - 1C$ fournit des résultats comparables mais avec une forte baisse pour certaines bases difficiles : $Waveform$, $Yeast$ et Oil. Par contre pour nos deux contributions $RS - NDF$ et $SRS - NDF$, elles présentent toujours des valeurs stables en les comparant deux à deux et même dans le cas des bases difficiles.

3.5 Conclusion

Le problème de l'apprentissage à partir d'une seule classe diffère sur un point essentiel de la classification conventionnelle multi-classes. En effet, en apprentissage mono-classe, on suppose que seule l'information de l'une des classes est disponible. Les données de cette classe sont appelées les données cibles. Toutes les autres données sont par définition des exemples atypiques par rapport à la classe cible. Cela signifie que seules les exemples de la classe cible peuvent être utilisés et qu'aucune information sur les autres classes n'est présente en général. La frontière entre les classes doit être estimée uniquement à partir des données de la classe cible. La tâche consiste donc à définir une frontière autour de la classe cible, permettant d'accepter les données cibles possibles tout en minimisant le risque d'accepter les autres données. De toute évidence, la première application dans ce type de contexte est la détection des valeurs aberrantes, pour détecter des formes inhabituelles à partir d'un ensemble de données, des exemples qui ne ressemblent pas à l'essentiel des données cibles d'une certaine façon. Ces valeurs aberrantes dans les données peuvent être aussi causées par des erreurs de mesure des caractéristiques, résultant des caractéristiques exceptionnellement élevées ou faibles en comparaison avec les exemples d'apprentissage. Ce problème est aussi lié à celui de la détection de nouveauté qui nous intéresse dans cette étude. Dans ce chapitre, nous avons proposé deux nouvelles approches permettant d'apporter une solution au problème de l'apprentissage à partir d'une seule classe. L'approche RS-NDF qui est basée sur les opérateurs de projection orthogonale, la technique de bootstrap et le principe de l'apprentissage d'ensemble. L'approche proposée RS- NDF est fondée sur un ensemble de NDF, induit à partir d'échantillons bootstrap sur les données et une sélection aléatoire des variables dans le processus d'induction du modèle NDF. La prédiction a été faite en agrégeant les prévisions de l'ensemble des filtres. Les mesures de performance telles que la précision, le rappel, le taux de faux positifs, le taux de faux négatifs, F-mesure et l'AUC sont calculées à travers une validation croisée sur des bases de données publics. Des améliorations significatives dans la précision ont été obtenues en utilisant notre méthode. L'approche RS-NDF présente généralement une amélioration substantielle des performances par rapport aux algorithmes existants de l'état de l'art. Grâce à un algorithme d'apprentissage en ligne, l'approche

Vin	$Acc-$ ($Recall$)	$Acc+$	$Prec$	$F-$ $measure$	AUC_b	$G-$ $mean$
MLP	0,78	0,68	0,83	0,81	0,73	0,73
ACP	0,60	0,69	0,80	0,68	0,65	0,64
SVM-1C	0,68	0,76	0,81	0,73	0,72	0,71
NDF	0,84	0,78	0,88	0,86	0,81	0,81
RS-NDF	0,87	0,85	0,92	0,89	0,86	0,86
SRS-NDF	**0,90**	**0,93**	**0,93**	**0,91**	**0,92**	**0,92**
Waveform						
MLP	0,67	0,35	0,51	0,58	0,51	0,48
ACP	0,68	0,35	0,68	0,68	0,51	0,49
SVM-1C	0,88	0,22	0,70	0,78	0,55	0,44
NDF	0,68	0,57	0,77	0,72	0,63	0,62
RS-NDF	0,85	0,53	**0,79**	**0,82**	0,69	0,67
SRS-NDF	**0,90**	**0,60**	0,70	0,78	**0,75**	**0,70**
Spectf						
MLP	0,60	0,78	0,42	0,49	0,69	0,68
ACP	0,51	0,82	0,43	0,47	0,67	0,65
1-SVM	0,73	0,74	0,43	0,54	0,74	0,74
NDF	**0,76**	0,75	0,45	0,56	**0,76**	**0,76**
RS-NDF	0,69	0,79	**0,47**	0,56	0,74	0,74
SRS-NDF	0,64	**0,84**	0,44	**0,58**	0,74	0,74
Yeast						
MLP	0,77	0,23	0,84	0,80	0,50	0,39
ACP	0,68	0,66	0,91	0,78	0,67	0,67
SVM-1C	0,90	0,13	0,84	0,87	0,51	0,34
NDF	0,88	0,19	0,85	0,86	0,54	0,41
RS-NDF	**0,94**	0,29	0,87	**0,90**	0,62	0,52
SRS-NDF	0,93	**0,53**	**0,93**	**0,90**	**0,72**	**0,70**
Oil						
MLP	0,96	0,15	0,96	0,96	0,55	0,38
ACP	0,94	0,24	0,96	0,95	0,59	0,48
SVM-1C	0,90	0,35	**0,97**	0,93	0,62	0,56
NDF	0,91	0,37	**0,97**	0,94	0,64	0,58
RS-NDF	0,94	0,22	0,96	0,95	0,58	0,46
SRS-NDF	**0,98**	**0,46**	0,95	**0,97**	**0,72**	**0,69**

TABLE 3.5 – Comparaison des performances

RS-NDF est également en mesure de suivre les changements dans les données au fil du temps (objet du chapitre 4). Enfin, la deuxième approche SRS-NDF est une extension de la précédente. C'est une stratégie plus économe que RS-NDF. Elle permet de réduire le nombre de modèles (filtres) à estimer. L'objectif est de choisir parmi l'ensemble des modèles, le sous-ensemble qui permet d'atteindre les meilleurs performances. Les différents résultats obtenus suite à l'application de notre modèle sont encourageants.

Chapitre 4

Détection de la dérive de concept

Introduction

Ces dernière années, la quantité de données à traiter a considérablement augmenté dans de nombreux domaines. Toutefois des progrès innombrables ont été réalisés dans le domaine de l'apprentissage automatique et de l'analyse de données afin d'aborder ce type de données massives [68]. Néanmoins, la majorité de ces méthodes est adaptée à l'analyse de données homogènes et stationnaires [69]. En effet, la plupart des approches automatiques suppose que les ressources matérielles disponibles tels que la mémoire RAM et CPU soient illimitées et que l'ordre d'arrivée des données et leurs débits soient contrôlés. Cependant les bases de données sont en pérpétuelle évolutions, elles sont caractérisées par une structure variable dans le temps, de nouvelles données arrivant constamment. Parfois, la masse des données est tellement importante qu'il est impossible de la stocker dans une base et que seule une analyse "à la volée" est possible. Donc l'analyse de flux de données (ou "Data stream mininng") et une réponse possible au traitement des données massives. Les techniques d'analyse des flux de données ont fait l'objet de nombreux travaux ces dernières années du fait du nombre important d'applications possibles dans de nombreux domaines. Comme par exemple l'analyse de données WEB.
Le problème de la classification dans les flux de données a été largement étudié par les chercheurs pendant la dernière décennie. En effet, le caractère dynamique lié à la dimension temporelle intégrée dans l'analyse de données implique un certain nombre d'aspects qui ont été étudiés, parfois individuellement, sous différents angles. Principalement, il s'agit de pouvoir traiter, de manière simultanée, l'analyse des régularités liées nécessairement aux flux de données et celle des nouveautés, exceptions, ou changements survenant dans un flux de données au cours du temps.

Les flux de données posent plusieurs problèmes qui rendent l'application de techniques standards d'analyse de données inadaptée. En effet ces bases de données sont en ligne, grossissant au fur et à mesure de l'arrivée de nouvelles données. De ce fait, les algorithmes efficaces doivent être capables de travailler avec une occupation mémoire constante, malgré l'évolution des données : la base entière ne pouvant être conservée en mémoire. Cela implique éventuellement l'oubli d'informations au cours du temps. La quantification et la détéction de changements est ainsi l'un des défis majeurs dans le traitement de flux de données qui touche aux questions fondamentales sur la nature des changements, leur importantce et leur effet potentiel sur les modèles conceptuels des flux.
La détection de ces changments peut être abordée de deux manières différentes. La première consiste à détecter les éventuels changements dans la distribution des données. Tandis que la deuxième, consiste à l'identification d'une anomalie qui est entrain de se produire dans le système observé.
Quand le phénomène observé dérive naturellement en raison d'un changement de contexte qui n'est pas explicitement décrit par les variables explicatives, nous sommes face à un problème de dérive de concept. Toutefois l'algorithme d'apprentissage à partir de flux utilisé, devrait être capable de détecter et gérer ce type de situation. La dérive de concept signifie l'apparition de nouveaux concept, la fusion ou, au contraire, la division de concepts. Ces changements peuvent apparaitre dans des données issues de la vie réelle. Par exemple, dans la détection de l'intrusion, l'apparition d'un nouveau type d'attaques.
C'est dans ce contexte que se situe notre travail, effectué dans le cadre de cette thèse. Dans ce chapitre, nous présentons dans un premier temps plus en détail la notion de dérive de concept et les problémes qu'elle entraîne pour la fouille de données. Nous présentons dans une deuxième section les principales méthodes de détection et de prise en charge de la dérive de concept. Ensuite, nous détaillons notre contribution qui consiste à mettre en place une approche de détection de dérive de concept basée sur un modéle détecteur de nouveauté "$RS - NDF$" ainsi que quelques expérimentations

4.1 Définition et problématique

4.1.1 Définition de la dérive de concept

L'évolution des données, sous forme de séquences spatio-temporelles, fait apparaître la notion de dérive de concept. Un concept désigne une classe de données ayant des caractères communs, par exemple un certain type de phénomènes. La dérive de concept caractérise l'évolution dans le temps des données observées.

de leurs étiquettes, et de leurs relations. Les modèles de classification étant à la base de la distribution des données et de leurs étiquettes, il est indispensable de les adapter régulièrement aux changement survenus dans la structure de ces données afin de prévoir correctement des classes effectivement présentes dans les observations au temps de la prévision. Très souvent la cause du changement n'est pas connue a priori, ce qui rend le problème de classification encore difficile.
Dans un problème de classification supervisé, une séquence d'exemples munis de leurs étiquettes d'appartenance à une classe est observée par l'algorithme d'apprentissage, qui en infère un modèle prédictif de relation entre les observations et les classes. En apprentissage non supervisé le principe est similaire, mais les observations sont données sans leurs étiquettes de classes. On appelle concept cible pour un exemple x_i et une classe Y_i la probabilité jointe $P(x_i, Y_i) = P(x_i)(P(x_i|Y_i)$. Si le processus qui génère les données n'est pas stationnaire, le concept à apprendre n'est pas stable au cours du temps. Cette instabilité peut être présente sous deux formes principales de dérive : brusque ou progressive.

Soient deux concepts cibles, Y et Z, et une séquence d'instances $\mathbf{x}_1...\mathbf{x}_n$. Avant une certaine instance $x_i(1 < i < n)$ le concept Y est stable. Après un certain nombre d'instances $\mathbf{\Delta}$ x au delà de $\mathbf{x}_i$, le concept revient stable, mais cette fois le concept cible est Z.

Entre les instances $\mathbf{x}_i$ et $\mathbf{x}_i + \Delta$ x, le concept change en passant de Y vers Z selon une certaine distribution, créant ainsi une zone d'ambigüité et d'incertitude.
Selon la valeur de $\mathbf{\Delta}$ x, on distingue deux types de dérive de concept (figure 4.1) [70] :

Notons qu'avant la ligne pointillée c'est le concept 1 qui domine sur le concept 2, et vice-versa après la ligne.
- Changement brusque (Sudden shift) : ce type corresponds à $\mathbf{\Delta}$ $x = 1$: Le concept change instantanément de Y vers Z.
- La dérive progressive (Gradual drift) : Si $\mathbf{\Delta}$ $x > 1$, le concept change alors progressivement à travers le temps.

4.1.2 Problèmatique

Souvent les différentes méthodes de classification sont établies hors ligne à partir de données statiques sur lesquelles plusieurs passages sont possibles. Or, dans le contexte des flux de données l'ensemble d'exemples d'apprentissage n'est lu

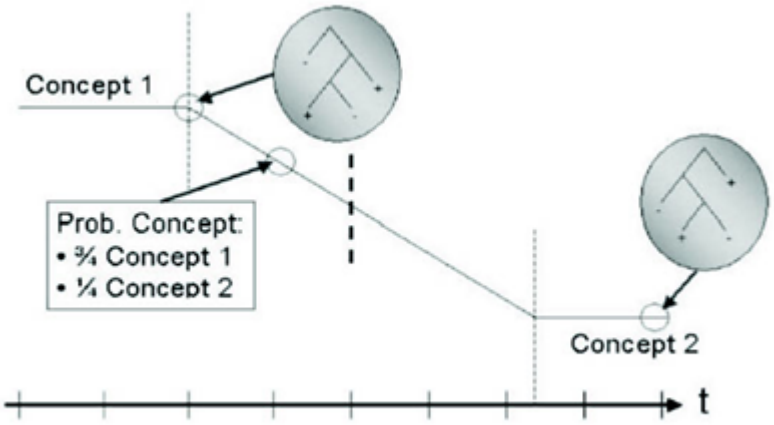


FIGURE 4.1 – Exemple d'une dérive de concept progressive, passant du concept 1 au concept 2 [70].

qu'une seule fois, de manière séquentielle. Donc le paradigme des flux de données propose de franchir une étape supplémentaire qui consiste à analyser les données à la volée, en s'affranchissant de leur stockage exhaustif. Les méthodes de classification doivent donc faire face à plusieurs défis afin de mener à bien leur tâche. Ces défis se rapportent principalement à [70] :

- La classification est effectuée en ligne ;
- Les flux doivent être traités au moins aussi rapidement que leur vitesse d'arrivée. Ce qui exige que le temps de traitement de chaque flux doit être minimal et aussi doit s'adapter au débit des flux d'entrée ;
- La nécessité de disposer d'un volume important de données d'apprentissage ;
- Le problème de dérive de concept dû au caractère dynamique de flux de données.

Comme nous l'avons mentionné auparavant, très souvent la cause du changement n'est pas connue a priori, ce qui rend le problème de classification encore plus difficile.

Nous pouvons donc identifier trois sous-problèmes qui doivent être pris en compte dans la gestion de la dérive de concept :

- Détection d'un changement ;
- Analyse du type de changement ;
- Adaptation du modèle au changement .

4.1.2.1 Détection d'un changement

Dans la littérature, il existe plusieurs manières d'aborder le problème de la détection de changements. Dans un flux de données elle consiste à comparer la distribution des tuples (tuple : les données élémentaires émises dans un flux) observés sur deux fenêtres temporelles distinctes : la "référence" et la "courante". Deux type de détections se distinguent donc :

1) La détection de changements par rapport à un régime normal qui implique une fenêtre de référence fixe ;
2) La détection de rupture qui implique une fenêtre de référence glissante

Une autre voie implique la surveillance de la distribution des exemples [71], elle propose d'utiliser le test statistique de Page-Hinkley [72] en le modifiant pour rajouter des facteurs d'oubli. D'autres approches basées sur la méthode des plus proches voisin [73], ou encore, sur des distances entre distributions ([74] et [75]) ont été développées. enfin, des travaux sur la détection de ruptures ont été réalisés ([76] et [77]).

4.1.2.2 Analyse du type de changement

Il existe quatre types de dérive de concept. Chaque type devrait pouvoir être géré par l'algorithme d'apprentissage.

- Dérive soudaine ("sudden drift ") : Au temps t la source de données change radicalement dans le flux (le concept change instantanément de Y vers Z) ;
- Dérive graduelle ("gradual drift") : Le concept change progressivement à travers le temps ;
- Dérive Incréméntale ("Incremental drift") : Le concept Y évolue au cours du temps, induisant une modification lente de la distribution des données ;
- Contextes réccurents ("recording contexts") : Les données captées par le système sont produites en alternance par différentes sources, sans pour autant qu'il y ai une périodicité définie.

La figure ci-dessous (figure 4.2) illustre ces différents types de dérive [70].

4.2 L'apprentissage avec la dérive de concept

En 1986 Schlimmer et Granger [78] ont proposé le terme "dérive de concept" en formulant le problème de l'apprentissage incrémental à partir de données bruitées. Depuis, plusieurs études sur la dérive de concept ont été publiées. Nous pouvons noter trois "pics" d'intérêts sur le sujet : le premier autour de 1998

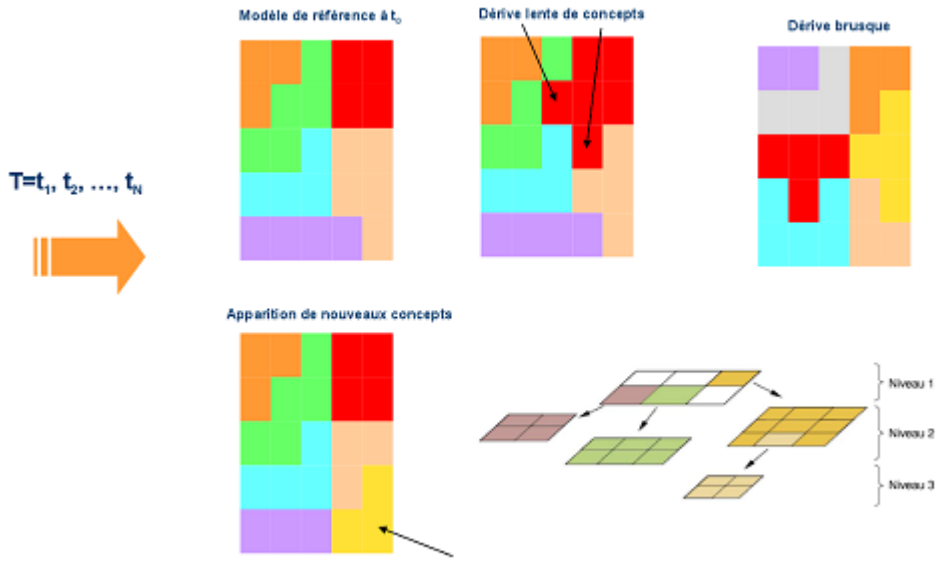


FIGURE 4.2 – Dérive lente et dérive brusque de concept.

avec l'apparition d'une édition spécial du journal "Machine Learning" [79], le second autour de 2004 [80] avec l'apparition d'une édition spécial du journal "Intelligent Data Analysis", le dernier commence en 2007 et continue actuellement, en réponse à l'augmentation des quantités de données et des ressources de calcul. Donc vu la dynamicité des flux de données et leurs tailles infinies, deux grandes familles d'approches peuvent se présenter. La première représente les approches qui s'adaptent au changement. En effet le processus d'apprentissage doit être capable d'adapter son modèle au changement détecté. Cela peut se faire par différents moyens. Soit l'algorithme d'apprentissage s'adapte directement aux données ou adapte ses paramètres ([81] et [82]). Soit l'échantillon de données choisi pour l'apprentissage est manipulé au cours du temps, par sélection ou transformation de données ou de variables, par l'utilisation de fenêtre temporelle, l'ajout de bruit ou des méthodes de bootstrap. La deuxième famille d'approches représente les mèthodes d'adaptation à la dérive de concept que ce soit par des méthodes à base de déclencheur ou des méthodes évolutives.

4.2.1 Fenêtre d'apprentissage

Le processus d'apprentissage qui adapte son modèle au changement suppose que l'échantillon de données choisi pour l'apprentissage soit manipulé au cours du temps, par sélection ou transformation de données ou de variables, par l'utilisation de fenêtres temporelles. Cela impose un réapprentissage du modèle à partir

de zéro. Dans ce cas l'apprentissage s'appuie alors soit sur une mémoire partielle des exemples, soit un nombre fini des derniers exemples, soit un nombre adaptatif des derniers exemples et enfin soit sur un résumé des derniers exemples. La taille de la fenêtre des derniers exemples utilisée ne peut en aucun cas excéder l'horizon de l'avant dernière dérive détectée [83].

Pour les méthodes à base de déclencheur les plus populaires sont basées sur des détections de changement. En générale implicitement adaptée aux cas des dérives soudaines. La détection de changement peu être basée sur la surveillance des données brutes [84], les paramètres des algorithmes d'apprentissage [78] ou les erreurs de classifieurs [85]. [86] propose une méthode de détection de changement à chacune de ces catégories. La méthode de détection est en général utilisée pour couper la fenêtre d'apprentissage au niveau du changement. D'autres techniques utilisent des heuristiques pour déterminer la taille des fenêres d'apprentissage [87]. Ces heuristiques sont basées sur le suivi des erreurs. La fenêtre d'apprentissage est déterminée selon une table donnant une action à effectuer pour chaque valeur du déclencheur. Parfois la taille de la fenêtre est déterminée en fonction de l'historique de la précision des modèles.

4.2.2 Méthodes Ensemblistes

Les méthodes qui décident d'adapter le modèle de classification peuvent s'appuyer sur plusieurs modèles de classification utilisés en même temps. Comme par exemple le bagging qui consiste à uiliser plusieurs classifieurs en même temps afin d'améliorer les capacités de prédiction de ceux-ci. Nous pouvons aussi citer l'algorithme SEA [88] : "Streaming Ensemble Algorithm" qui utilise un ensemble de classifieurs sur les flux. Le flux remplit un tampon de taille définie et dès que le tampon est plein l'algorithme C4.5 est lancé dessus. On se retrouve donc avec une suite de classifieurs générés que l'on met dans un pool. Une fois le pool plein, si le nouveau classifieur améliore la prédiction du pool alors il est conservé, sinon il est rejeté. Dans [89] la même technique est utilisée mais différents classifieurs composent l'ensemble : bayésien naif, C4.5, RIPPER, etc. Une autre technique proposée dans [90] $ADWIN$ (ADaptative WINdowing). Elle est basée sur la détection des changements de concept à l'aide d'un réservoir de taille variable W. Le réservoir s'agrandit quand il n'y a pas de changements de concept et diminue lorsqu'il rencontre un. Cette approche ne consomme en mémoire que log(W) et ne nécessite pas de paramétre. Quand un changement est détecté le plus mauvais classifieur est retiré et un nouveau classifieur est ajouté à l'ensemble.

En effet les méthodes les plus populaires actuellement sont les méthodes évolutives : Ensembles de classifieurs adaptatifs. Les sorties de nombreux classifieurs sont combinées ou sélectionnées pour obtenir une décision finale. Les règles permettant la combinaison ou la sélection sont souvent appelées "règles de fusions". De nombreuses méthodes d'ensemble de classifieurs ont été proposé pour gérer la dérive de concept. En général ces algorithmes ne dépendent pas d'un type particulier de classifieur ; cependant certains auteurs utilisent préférentiellement un modèle particulier, en proposant des règles de fusion adaptées : à titre d'exemple pour les SVM dans [91], [92] pour les modèles de mélange Gaussien, [93] avec perceptrons et [94] avec KNN. Dans tout les cas l'adaptation est prise en charge par les règles de fusion, qui définissent comment des poids sont assignés aux différents modèles à chaque instant. Dans un cas discret, un seul modèle est sélectionné (les autres modèles ont alors un poids de zéro). Dans le cas général, le poids de chaque modèle correspond à une estimation du pouvoir prédictif du modèle dans le futur proche. Ce poids est souvent une fonction de l'historique des performances du modèle dans le passé, ou bien une estimation par validation croisée, ou encore une méthode d'estimation spécifique au type de classifieur employé. Les validations historiques sont restreintes aux cas de dérives soudaines ou incrémentales, alors que l'utilisation de validation croisée prend en compte les dérives graduelles et les contextes réccurents.

4.2.3 Pondération des données

Il est aussi possible de pondérer au cours du temps les exemples utilisés pour apprendre le modèle. Depuis 1980 des solutions comme les fenêtres et les facteurs d'oubli ont été proposées [95] : quelques propositions de pondération concernant l'adaptation du modèle au changement :

- TWF (Time Weighted Forgetting) : plus l'exemple est ancien plus son poids est faible ;
- LWF (Localy Weighed Forgetting) : à l'arrivée d'un nouvel exemple on augmente le poids des exemples qui sont proches de lui et on baisse ceux des autres. Les régions ayant des exemples récents sont ainsi conservées (ou créées si elles n'existaient pas) et les régions en ayant peu ou pas sont supprimées ;
- PECS (Prediction Error Context Switching) : l'idée de LWF est reprise mais tous les exemples sont gardés en mémoire et une vérification des étiquettes des nouveaux exemples par rapport aux anciens est effectuée. Une probabilité des exemples est calculée grâce aux comptes des exemples qui sont utilisés pour réaliser l'apprentissage.

Concernant les méthodes évolutives, l'adaptation est mise en oeuvre par une pondération de données. L'idée principale est d'utiliser un échantillonnage par boostrap qui accorde plus d'importance aux données mal classées par le modèle. Dans cette catégorie nous trouvons à la fois des méthodes à la base d'unique classifieur [96], comme des méthodes d'ensemble de classifieurs [97].

4.3 Notre contribution à la dérive de concept

Pour être efficace, le traitement des flux de données massives requiert généralement des algorithmes mono-classe. Dans cette section, nous présentons notre contribution. Nous avons développé une approche de détection de dérive de concept à partir du calcul de la similarité entre les filtres NDF calculés sur différents sous-ensembles de données. La détection de dérive de concept s'effectue en calculant la distance entre les filtres voisins. La distance entre deux matrices est définie par la norme de la différence entre ces deux matrices 4.1.

$$dist(A, B) = \| A - B \| \tag{4.1}$$

Pour cela, nous avons effectué une étude des différentes normes matricielles dont les plus connues sont détaillées dans la section suivante.

4.3.1 Les normes matricielles

Les normes matricielles les plus utilisées sont issues de deux grandes familles de normes : Les normes matricielles de Schatten (Schatten-Von-Neumann) dites naturelles, et les normes Entrywise.

Normes naturelles :

Les normes de schatten sont des normes subordonnées aux normes vectorielles naturelles. Ainsi à chaque norme vectorielle p, on peut lui faire correspondre une norme matricielle p de la façon suivante :

$$\| A \|_p = (\sum_{i=1}^{min(m,n)} \sigma_i^p)^{1/\mathrm{p}} \tag{4.2}$$

où σ_i : valeur singulière.
En réalité, cette relation signifie que la norme d'une matrice correspond au maximum de la norme du vecteur résultant de la multiplication de la matrice A par tous les vecteurs de norme égale à 1. Parmi les normes matricielles de schatten les plus connues, on cite :

– Norme nucléaire connue aussi sous le nom de norme de trace : elle correspond à $p = 1$ et définie par :

$$\| A \| = trace(\sqrt{(A^{\mathrm{t}}A)} = \sum_{i=1}^{min(m,n)} \sigma_i \tag{4.3}$$

où σ_i est la i$^{\text{ème}}$ valeur propre de A

– Norme spectrale ou norme 2 : elle est définie par la valeur propre maximale de la matrice :

$$\| A \|_2 = \sqrt{\lambda_{max}(A^{\mathrm{t}}A)} = \sigma_{max}(A) \tag{4.4}$$

où σ_{max} est le rayon spectral de la matrice A, c'est à dire le plus grand module des valeurs propres de A.

Normes Entrywise :

Le deuxième type de normes, dit Entrywise, considèrent les matrices de taille $n * m$ et utilisent des normes vectorielles connues. Les normes Entrywise les plus utilisées sont :

– Norme de Frobenius : elle découle de la norme euclidienne et peut être exprimée par :

$$\| A \|_F = \sqrt{trace(A^{\mathrm{t}}A)} = \sqrt{\sum_{i=1}^{min(m,n)} {\sigma_i}^2} \tag{4.5}$$

– Norme de la valeur maximale : cette norme correspond à la norme vectorielle qui porte le même nom. Elle est définie comme le maximum des valeurs absolues de tous les éléments de la matrice :

$$\| A \|_{max} = max\{| a_{ij} |\}. \tag{4.6}$$

Nous avons testé les distances induites par ces normes sur un ensemble de matrices analogues à celle des filtres, et nous avons constaté que la norme spectrale est celle qui met le plus en valeur la différence entre les matrices. C'est la raison pour laquelle nous avons utilisé cette distance dans nos expérimentations.

4.3.2 Fonctionnement de notre approche

Notre approche est une méthode de détection de dérive de concept basée sur les filtres détecteurs de nouveauté. La sélection de données se fait par la technique de "fenêtrage". Cette technique consiste à considérer les N données les plus récentes dans le flux. Avec le temps, de nouvelles données font leur apparition dans la fenêtre, et d'autres données, plus anciennes, disparaissent (figure 4.3). Dans notre travail, nous avons utilisé les fenêtres glissantes (Sliding Window) où le pas de décalage est inférieur à la taille de la fenêtre.

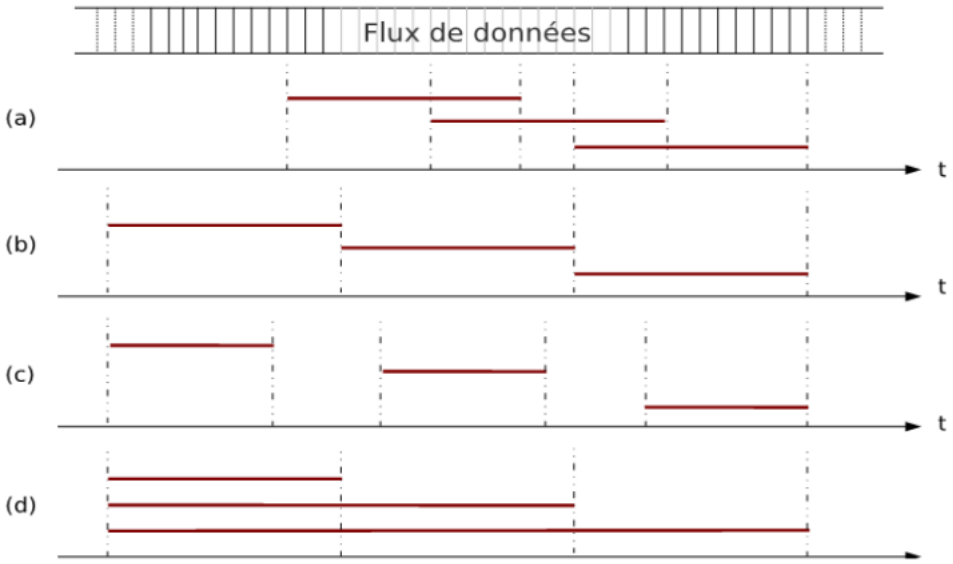


FIGURE 4.3 – Exemples de différents types de fenêtrage.
(a)fenêtres glissantes(b)fenêtres sautantes (c)fenêtres bondissantes(d)fenêtres fixes avec points de repère.

Il s'agit de construire, pour chaque fenêtre centrée sur les données récemment vues, un filtre NDF afin de le comparer avec le filtre construit à partir du batch précédent dans le but de détecter une éventuelle dérive de concept. Une dérive de concept peut être signalée grâce à une valeur élevée de la distance entre deux filtres. Pour détecter les éventuels points de dérive, nous avons appliqué le "Scree Test" à l'ensemble des valeurs des distances récemment calculées. Le

fonctionnement de l'approche de détection de dérive de concept est décrit dans l'algorithme suivant :

Algorithme 4: Concept Drift Detection Via Novelty Detection

Fixer la taille de la fenêtre glissante à n.
Répéter :
1. Calculer le filtre NDF fl_i à partir des données de la fenêtre i en cours.
2. Calculer la distance entre le filtre fl_i et le filtre fl_{i-1}.
3. Aplliquer le Scree Test pour sélectionner les éventuels points de dérive.
4. Vérifier s'il s'agit bien d'une dérive de concept.
5. Faire glisser la fenêtre de $n/2$ données.
Jusqu'à épuisement des données.

4.4 Validation

Afin de vérifier l'efficacité de notre approche pour la détection de la dynamicité des données, nous l'avons testé sur plusieurs jeux de données. La première partie de cette section contient une description des données utilisées dans nos expérimentations. La deuxième partie présente les résultats de ces expérimentations ainsi qu'une comparaison avec d'autres modèles analogues.

4.4.1 Présentation des données

Pour nos expérimentations, nous avons adapté les différentes bases de données utilisées $Waveform$ et $Wine$ au contexte de la détection de dérive de concept. En effet, nous avons ordonné les observations des deux bases suivant leurs classes afin de les présenter à notre modèle. Aussi nous avons testé la validité de la méthode proposée sur des flux de données, nous avons choisi le domaine de la détection d'intrusions. En effet, ces intrusions sont extraites de ***Base Kdd- cup1999***. Elles ont été préparées et contrôlées par le laboratoire MIT Lincoln pour le programme d'évaluation de détection d'intrusion DARPA 1998. Ces données ont aussi été utilisées pour le concours de détection d'intrusions de la conférence KDD 1999. Chaque connexion est étiquetée en tant que connexion normale ou attaque, avec le type spécifique d'attaque. Les attaques trouvées sont classées selon quatre catégories principales : DOS (déni de service), R2L (accès non autorisé d'une machine à distance, par exemple devinant le mot de passe), U2R (accès non autorisé aux privilèges d'un super-utilisateur tel que buffer overflow)

et Probe (sondage et surveillance tel que port scanning). Les différentes attaques existantes, classées en catégories, sont décrites dans le tableau (4.1).

Type d'attaque	catégorie d'attaque
neptune, back, land, pod, smurf, teardrop	DoS
buffer_overflow, loadmodule, perl, rootkit	U2R
ftp_write, guess_passwd, imap, multihop, phf, warezmaster	R2L
ipsweep, nmap, portsweep, satan	Probe

TABLE 4.1 – Types et catégories d'attaques

Ces données qui correspondent à environ quatre "GO" de données binaires TCP-dump compressées, contiennent sept semaines du trafic de réseau. Ceci a été transformé en environ cinq millions de connexions. Les données d'apprentissage KDD-Cup 99 possèdent 4 900 000 connexions étiquetées normale ou attaque. Chaque connexion contient 41 variables descriptives. Le tableau (4.2) donne la répartition exacte d'un échantillon de 10% des données utilisées lors de la compétition, nommée LS-10%.

Classe	Données d'apprentissage LS-10%	Données test
Normal	97278	60593
Probe	4107	4166
DoS	391458	229853
U2R	52	228
R2L	1126	16189

TABLE 4.2 – Répartition des données KDD-Cup 1999

En raison de problèmes de mémoire posés par le logiciel utilisé dans notre implémentation (Matlab), nous avons réduit la taille de l'échantillon d'apprentissage. C'est ainsi que nous nous sommes limité à un sous-échantillon de l'échantillon LS-10%, stratifié suivant les étiquettes de classe de connexion, sauf pour l'étiquette 3 (U2R), trop peu fréquente, nous avons conservé la totalité des 52 connexions relevant de l'étiquette 3 :U2R et nous avons pris au hasard 5% des connexions de chacune des autres étiquettes (tableau (4.3)).

Classe	Données d'apprentissage LS-10%
Normal	4864
Probe	205
DoS	19572
U2R	52
R2L	57

TABLE 4.3 – Répartition des données d'apprentissage

4.4.2 Résultats

Pour évaluer la performance de notre approche, nous avons utilisé les matrices de confusion. La matrice de confusion sert à évaluer la qualité d'une classification. Dans la matrice de confusion, les colonnes de la matrice sont les classes actuelles, et ses lignes les classes prédites (tableau 4.4.2).

Réel/Prédit	Classe_1	Classe_i	Classe_n	Total lignes
Classe_1	x_{11}	x_{1i}	x_{1n}	N_1
Classe_i	x_{21}	x_{2i}	x_{2n}	N_2
Classe_n	x_{n1}	x_{ni}	x_{nn}	N_n
Total Colonnes	M_1	M_i	M_n	N

TABLE 4.4 – Matrice de confusion

Comme nous l'avons mentionné auparavant, les données de la base $Waveform$ par exemple ont été ordonnées suivant leurs classes (données de la première classe ensuite de la deuxième classe et enfin de la troisième classe). Après présentation des données au système, les distances entre filtres sont calculées comme l'indique la figure (Fig : 4.4) (ou (Fig : 4.5) pour la base Vin). Cette figure représente l'évolution de la distance entre les filtres au cours du temps. Les valeurs élevées de cette distance peuvent indiquer de potentiels points de dérive de concept.

Nous constatons que la distance entre les filtres varie peu quand le concept est stable. Par contre, la dérive de concept est marquée par des variations plus significatives.
La matrice de confusion de la base $Waveform$ (tableau (3.1.5)) montre que cette approche arrive à bien détecter les deux dérives de concept survenues dans les données avec une éstimation de l'erreur générale de 3,52%. Ce résultat est prometteur du fait qu'il signifie que les nouveaux concepts sont détectés juste après leurs arrivées.

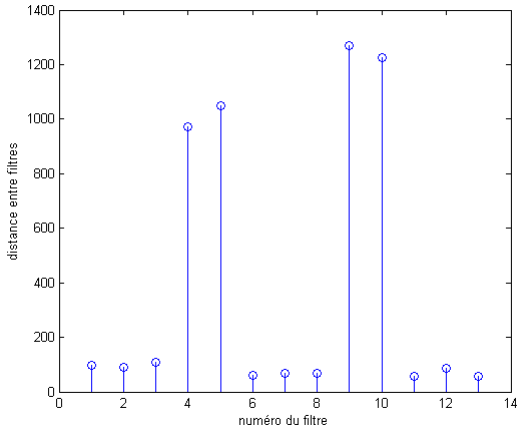


FIGURE 4.4 – L'évolution de la distance entre les filtres pour Waveform

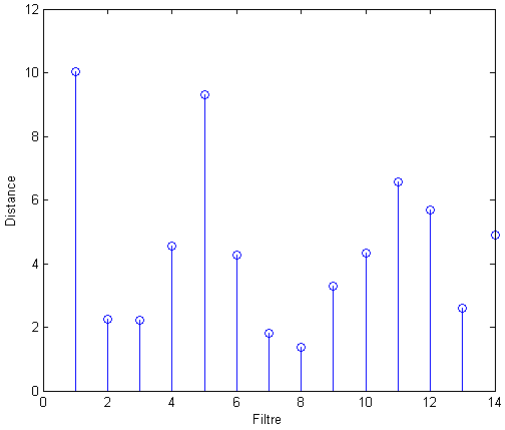


FIGURE 4.5 – L'évolution de la distance entre les filtres pour Wine

Réel/Prédit	Classe_1	Classe_2	Classe_3	Total lignes
Classe_1	1657	0	0	1657
Classe_2	13	1471	163	1647
Classe_3	0	0	1696	1696
Total Colonnes	1670	1471	1859	5000

TABLE 4.5 – Matrice de Confusion de WaveForm

Par contre pour la base *Wine* (Tableau (3.1.5)), malgré la détection des deux dérives de concept, le taux d'érreur géneral du modèle, éstimé à 21,91%, est élevé. Ce qui signifie que le nouveau concept a été détecté avec un retard par rapport à son début.

Réel/Prédit	$Classe_1$	$Classe_2$	$Classe_3$	Total lignes
$Classe_1$	50	9	0	59
$Classe_2$	5	41	25	71
$Classe_3$	0	0	48	48
Total Colonnes	55	50	73	178

TABLE 4.6 – Matrice de Confusion de Wine

Résultats sur la base KDD

Nous avons testé cette approche sur l'échantillon d'apprentissage précédemment décrit de la base d'intrusion kdd-cup1999 afin de montrer son efficacité sur les données du monde réel. La matrice de confusion correspondant aux résultats des tests est détaillée dans le tableau (4.7) :

Réel/Prédit	0	1	2	3	4	*Total*
0 :Normal	4864	0	0	0	0	4864
1 : DoS	12	19560	0	0	0	19572
2 : U2R	0	52	0	0	0	52
3 : R2L	0	57	0	0	0	57
4 :Probe	0	11	0	0	194	205
Total	4876	19680	0	0	194	024750

TABLE 4.7 – Matrice de confusion de notre approche

Le taux d'erreur de ce modèle, pour la base kdd-cup1999, est éstimé à seulement 0,53%. Ce résultat global est trompeur. En effet, Ce modéle ne prédit jamais les classes les plus minoritaires, U2R, R2L, à cause de leur faible effectif (respectivement 52 et 57 connexions). Seule les classes Normal, DOS et Probe, plus nombreuses (resp. 4864, 19572 et 205 connexions), donnent lieu à des prédictions. Afin de pouvoir comparer notre approche avec les autres approches existantes de dérive de concept, nous avons utilisé la matrice de confusion de l'approche AdaBoostM1(4.8). Il est indispensable de noter que les données d'apprentissage de cet algorithme ont été tirées aléatoirement de

l'échantillon LS-10%, avec les mêmes proportions que nous avons utilisé pour construire notre échantillon de données.

Réel/Prédit	0	1	2	3	4	*Total*
0 :Normal	3988	876	0	0	0	4864
1 : DoS	641	18931	0	0	0	19572
2 : U2R	32	20	0	0	0	52
3 : R2L	35	22	0	0	0	57
4 :Probe	134	71	0	0	0	205
Total	4830	19920	0	0	0	024750

TABLE 4.8 – Matrice de l'approche AdaBoostM1

Les tableaux (4.8) et (4.7) montrent que pour chaque classe, qu'il s'agisse des classes d'attaque ou de la classe normale, c'est notre approche qui obtient le meilleur nombre de bonnes prédictions. En effet, le modèle AdaBoostM1 ne détecte que deux classes (les classes majoritaires DoS et normal). Par contre, notre mèthode permet de détecter ces deux mêmes classes en plus de la classe Probe. De plus, en constatant le nombre de données mal classées, le taux d'erreur général de AdaBoostM1 est estimé à 7,4%, ce qui indique que les performances (la précision) de notre modèle s'avèrent nettement meilleures.

Conclusion

Dans de nombreux problèmes du monde réel de classification, le concept modélisé peut ne pas être statique, mais changer au fil du temps. Ces changements peuvent être dus à des événements externes, des phénomènes cachés ou même des changements dans la distribution sous-jacente des données. Des exemples de l'évolution des concepts peuvent être vus dans une variété d'applications du monde réel. Les prévisions météorologiques sont influencées par les variations climatiques saisonnières, le comportement d'achat des clients et leurs préférences peuvent être corrélés avec les tendances de la mode ou des inclinaisons saisonniers, etc. La dérive de concept serait sans doute présente dans les exemples ci-dessus, mais il serait difficile de déterminer si le concept a changé, et à quel taux il a changé. Sans cette information cruciale, évaluer un algorithme pour traiter la dérive de concept peut être problématique. Les recherches sur l'évolution des concepts ont été entravées par un manque de disponibilité des bases de

données du monde réel. Les chercheurs utilisent leurs propres ensembles de données (qui ne sont souvent pas disponibles en raison de problèmes de confidentialité) ou des ensembles publics de données artificielles.

Chapitre 5

Conclusion et perspectives

5.1 Conclusion

Les apports de cette thèse ont été présentés dans les chapitres 3, 4 et 5. Nos contributions à l'apprentissage en distributions déséquilibrées portent sur trois points : une technique d'échantillonnage agissant au niveau des données, une technique agissant au niveau algorithmique par un apprentissage à partir de données d'une seule classe et enfin des premiers travaux sur la dérive de concept. Celles-ci sont rappelées ci-dessous.

Sous-échantillonnage structurel adaptatif

La première contribution de cette thèse concerne les techniques d'échantillonnage. Nous avons proposé une nouvelle approche de sélection des données basée sur la structure sous-jacente et la notion de voisinage en particulier autour des frontières entre classes. Cette méthode est semi supervisée, elle utilise un échantillon de la classe minoritaire pour guider le processus du sous échantillonnage. C'est un moyen pour rééquilibrer les jeux de données en supprimant un certain nombre d'individus appartenant à la classe majoritaire jugés non pertinents. Cette mèthode utilise la régle des plus proches voisins de Wilson pour supprimer des individus de la classe majoritaire ("Neighborhood Cleaning Rule" (NCR)). Afin de mieux guider le sous-échantillonnage et de le rendre moins aveugle, l'utilisation des cartes auto-organisatrices SOM, nous paraît une solution efficace pour choisir d'une façon intelligente les données à supprimer de la classe majoritaire en tenant compte de leurs topologies. La mèthode proposée consiste

à modifier l'algorithme d'apprentissage en proposant de supprimer à chaque itération les observations qui "gênent" les données minoritaires.

Dans cette proposition l'algorithme SOM est utilisé dans le cas semi-supervisé, puisque les étiquettes associées à la classe positive ("+", minoritaire) sont utilisées. Ces étiquettes ne sont pas utilisées comme variable de la base, mais uniquement dans la phase de nettoyage. L'élimination au cours de l'apprentissage d'observations implique la modification de la base d'apprentissage $\mathcal{A}$ qui diminue au fur et à mesure des itérations. Chaque observation $\mathbf{x}$ dispose d'une étiquette $label(\mathbf{x})$ positive ("+", données minoritaires) ou nègative ("-", données majoritaires). L'étiquette nègative "-" est utilisée uniquement dans la régle de nettoyage. L'approche que nous avons proposée est donc une approche hybride : une action sur les données avec la phase de nettoyage par voisinage et une modification algorithmique de SOM. A l'opposé de NCR qui est dans l'obligation de supprimer des données de la classe majoritaire, notre approche SNCR ne l'est pas puisque le nettoyage par voisinage s'applique d'une maniére locale au niveau de chaque cellule.

L'apprentissage à partir d'une seule classe pour le problème de détection de nouveauté

Les différents travaux élaborés pour cette problèmatique pourraient être résumés en deux propositions :

– Une première contribution, concerne l'apprentissage à partir d'une seule classe. C'est un moyen permettant de contourner le problème des distributions déséquilibrées vers un problème de détection de nouveauté. Le but essentiel de la détection de nouveauté est de repérer la nouveauté apportée par des données encore inconnues, en exploitant la connaissance extraite à partir d'un ensemble de données de référence (données d'apprentissage). Les données de référence se limitent à des exemples positifs des données normales ou familières, du fait de la difficulté, voire l'impossibilité dans certains cas, d'identifier a priori ce qui constituerait une nouveauté par rapport aux données déjà connues (ce qui amène au problème de la classification à partir d'une seule classe). La détection de nouveauté est particulièment utile quand une classe importante est sous représentée dans les données. Un exemple typique de ce problème est la détection des fraudes où il peut y avoir un intervalle de plusieurs heures entre deux transactions frauduleuses. L'idée fondamentale est donc d'apprendre un modèle des données normales disponibles et de l'employer pour identifier des données entrantes. Le nouveau modèle d'apprentissage que nous avons proposé est une extension d'un modèle de détection de

nouveauté proposé dès les années 1976. L'idée est basée sur les opérateurs de projection orthogonale utilisés par Kohonen et Oja dans le modèle NDF, la technique de bootstrap et le principe de l'apprentissage d'ensemble. Notre approche appelée Random Subspace Novelty Detection Filter ($RS-NDF$) est un ensemble de NDF calculés sur plusieurs échantillons. Ces derniers sont obtenus par un double bootstrap. La prédiction est faite en agrégeant les prévisions de l'ensemble.

– Une deuxième contribution proposée pour cette problèmatique, consiste à améliorer l'algorithme $RS-NDF$ pour de meilleurs performances en détection de nouveauté. Nous avons proposé une stratégie, plus économe, ayant pour objectif de réduire le nombre de modèles à prendre en compte lors de la décision afin de choisir l'ensemble de modèles (filtres) permettant d'atteindre de meilleurs performance. $SRS-NDF$ est une approche de sélection de modèles qui classe d'abord l'ensemble des filtres d'origine, c'est à dire tous les filtres obtenus par $RS-NDF$, selon le critère de pertinence et dans un ordre décroissant. Ensuite, nous avons utilisé le test statistique pour choisir l'ensemble des modèles (filtres) les plus prometteurs.

Les deux propositions ont été testées sur plusieurs bases de données publiques. Elles ont été aussi comparées à des approches très utilisées pour le problème de classification à partir d'une seule classe. Les résultats obtenus sont encourageants et prouvent le bon fonctionnement de nos méthodes.

Détection de la dérive de concept

Une dernière piste a été également explorée dans le cas de la dérive de concept. L'évolution des données, sous forme de séquences spatio-temporelles, fait apparaître la notion de dérive de concept. Un concept désigne une classe de données ayant des caractères communs, par exemple un certain type de phénomènes. La dérive de concept caractérise l'évolution dans le temps des données observées, de leurs étiquettes, et de leurs relations. Les modèles de classification étant à la base de la distribution des données et de leurs étiquettes, il est indispensable de les adapter régulièrement aux changements survenus dans la structure de ces données afin de prévoir correctement des classes effectivement présentes dans les observations au temps de la prévision. Très souvent la cause du changement n'est pas connue a priori, ce qui rend le problème de classification encore difficile.
Nous avons détaillé notre contribution qui consite à mettre en place une approche de détection de dérive de concept basée sur un modéle détecteur de nouveauté NDF ainsi que quelques expérimentations. Cette approche propose d'utiliser une distance entre

matrices représentant les filtres.

5.2 Perspectives

De nombreuses perspectives peuvent être envisagées suite à cette thèse, notamment :

- Étendre le travail du sous-échantillonnage structurel adaptatif afin de permettre un nettoyage plus large. L'approche SNCR consiste à éliminer des données "négatifs" qui se situent particulièrement autour des frontières entre classes. Nous désirons donc améliorer l'approche pour un meilleur nettoyage qui ne touche pas seulement les données situées dans les frontières.

- Nous souhaitons également rajouter un critère d'évaluation à notre algorithme $SNCR$ permettant d'évaluer l'échantillon obtenu après chaque suppression d'observations appartenant à la classe majoritaire. En effet, le nettoyage s'arrête soit quand nos règles ne sont plus respectées ou soit, dans certains cas, nous sommes arrivés au taux de suppression souhaité. Cela peut induire à des échantillons moins représentatifs (moins "bons").

- Pour les approches à base de filtres NDF, nous comptons améliorer d'une part le choix des sous espaces aléatoires et d'autre part perfectionner la phase de décision en sppuyant par exemple sur la théorie du consensus, qui a bénéficié d'un intérêt considérable dans le domaine social et sciences de gestion mais qui est restée peu connu ailleurs. Une telle théorie permettra d'avoir des règles de consensus qui définissent la façon dont la combinaison des différentes décisions est mise en oeuvre. L'approche adoptée pourra attribuer des poids de consensus pour chaque filtre NDF sur la base de son importance dans le mélange de filtres. Nous souhaitons aussi améliorer la phase de sélection des filtres dans l'approche SRS-NDF. Pour ce faire, nous étudierons d'autres mesures de pertinence utilisant d'autres critères. Nous pensons par exemple utiliser une approche de type AdaBoost pour estimer la pertinence de chaque filtre. AdaBoost est un algorithme bien établi qui pondère de manière itérative des échantillons afin de réduire l'erreur cumulée du classificateur, ce qui permet d'identifier les exemples importants parmi les données d'apprentissage. Nous envisageons une extension de ce type d'algorithme au niveau du modèle : estimer la pertinence de chaque modèle sous forme de pondération. Nous comptons donc intégrer un tel paradigme dans notre approche SRS-NDF.

– Finalement, pour la dérive de concept, il nous reste beaucoup de travail à faire. Ce que nous avons présenté sur cette problématique n'était qu'un début de travail. Nous envisageons d'apporter plus de contributions pour ce problème et sur plusieurs niveaux. En effet, notre approche est plus adaptée aux dérives soudaines de concept, donc nous souhaitons aborder les autres types de dérive de concept. Aussi nous avons utilisé les fenêtres glissantes (Sliding Window) où le pas de décalage est inférieur à la taille de la fenêtre, nous projetons de tester le fonctionnement de l'approche avec les autres techniques de fenêtrage.

Liste des publications

- Papier en cours de préparation : " Selected Random Subspace Novelty Detection Filters"
- HAMDI F., BENNANI Y. (2012), pprentissage d'ensemble d'opérateurs de projection orthogonale pour la détection de nouveauté EGC2012, Bordeaux, Janvier 2012
- HAMDI F., BENNANI Y. (2011), Learning Random Subspace Novelty Detection Filters, in Proc. IJCNN'11, IEEE International Joint Conference on Neural Network, San Jose, California-July 31 - August 5, 2011.
- HAMDI F., LEBBAH M., BENNANI Y. (2010), Topographic Under-Sampling for Unbalanced Distributions, in Proc. IJCNN'10, IEEE International Joint Conference on Neural Network, 18-23 July 2010, Barcelona, Spain.
- HAMDI F., ELGHAZEL H., BENABDESLEM K. (2010), Approche graphique pour l'agrégation de classifications non-supervisées, Atelier : Fouille de données complexes dans le cadre de la conférence EGC2010, Hammamet, Janvier 2010.
- ELGHAZEL H., BENABDESLEM K., HAMDI F., (2010), Consensus clustering by graph based approach, ESANN'10 proceedings - European Symposium on Artificial Neural Networks, pp 493-498, Bruges (Belgium), 28-30 April 2010

Annexe 1

Learning Random Subspace Novelty Detection Filters

Fatma Hamdi and Younès Bennani

***Abstract*— In this paper we propose a novelty detection framework based on the orthogonal projection operators and the bootstrap idea. Our approach called Random Subspace Novelty Detection Filter ($RS-NDF$) combines the sampling technique and the ensemble idea. $RS-NDF$ is an ensemble of NDF, induced from bootstrap samples of the training data, using random feature selection in the NDF induction process. Prediction is made by aggregating the predictions of the ensemble. $RS-NDF$ generally exhibits a substantial performance improvement over the single NDF. Thanks to an online learning algorithm, the $RS-NDF$ approach is also able to track changes in data over time. The $RS-NDF$ method is compared to single NDF and other novelty detection methods with tenfold cross-validation experiments on publicly available datasets, where the methods superiority is demonstrated. Performance metrics such as precision and recall, false positive rate and false negative rate, F-measure, AUC and G-mean are computed. The proposed approach is shown to improve the prediction accuracy of the novelty detection, and have favorable performance compared to the existing algorithms.**

I. Introduction

We consider the problem of determining whether new data is similar to the data used during the training process or if it is novel. Understanding when new data cases are novel can be extremely important in qualifying the confidence of other predictive modeling methods and identifying behavior in data that has not been previously encountered. Learning to detect novelty from a given data is a very challenging task. The main objective of novelty detection is to emphasize the novelty in yet unseen data with respect to previously learned ones. The basic idea is to learn a model or set of models of available data and to use it for identifying the dissimilar data (novelty). Novelty detection is usually defined as the task of detecting a signal or pattern that a learning system is not aware of during training. Broad reviews of the subject can be found in [1], [2] and [3].

Novelty detection is a research area with a wide variety of applications and with even more methodologies and a diverse nomenclature. In literature, novelty detection is often called fault detection or abnormality detection. The term fault detection is often used in the mathematical systems theory statistics and electrical engineering branch. Abnormality detection appears to be the more general term of novelty detection. The term novelty detection itself is typically used in relation with neural networks. In this paper mostly the last term is used, as this paper discusses novelty detection with neural networks. Novelty detection has a wide variety of applications. Applications exist in fraud detection, process drift/fault detection, homeland security, preventative maintenance, model lifecycle management, network intrusion, rare disease diagnosing, and many other areas. Most paradigms and methodologies of novelty detection are stated in the field of mathematical systems and electrical engineering. The research on novelty detection took in the 90's a flight in the area of computational intelligence, specifically the neural networks branch. Many people noted the benefits of neural networks over classical statistics. In this paper the emphasis is at neural networks. In statistics and machine learning, ensemble methods use multiple models to obtain better predictive performance than could be obtained from any of the constituent models. Thus, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, let them over-fit the training data more than a single model would do, but in practice, some ensemble techniques (especially bagging) tend to reduce problems related to over-fitting of the training data.

In this paper we propose a novelty detection framework based on the orthogonal projection operators used by Kohonen and Oja in the NDF model [4], the bootstrap idea and the ensemble learning paradigm. Our approach called Random Subspace Novelty Detection Filter ($RS-NDF$) combines the sampling technique and the ensemble idea. $RS-NDF$ is an ensemble of NDF, induced from bootstrap samples of the training data, using random feature selection in the NDF induction process. Prediction is made by aggregating the predictions of the ensemble.

An interesting feature of the NDF model is its simple implementation by a neural network architecture [4] taking advantage of the capabilities of massive parallelism, adaptive learning and noise robustness.

This paper is organized as follows: Section II introduces the basic concepts of the Random Subspace Novelty Detection Filte. Section III describes the validation databases and experimental protocol. In section IV we show validation results and their evaluation. Conclusion and future work are given in section V.

II. Random Subspace Novelty Detection Filters

Bagging [5], a name derived from bootstrap aggregation, was the first effective method of ensemble learning and is one of the simplest methods of arching. The meta-algorithm, which is a special case of model averaging, was originally designed for classification and it is usually applied on decision tree models, but it can be used with any type of model, whether for classification or regression.

The random subspace principle is an interesting method of combining models. Learning machines are trained on

Fatma Hamdi and Younès Bennani are with the LIPN - UMR 7030 - Université Paris 13 - CNRS, 99, Avenue J-B. Clément - 93430 Villetaneuse - FRANCE (email: {Firstname.Lastname}@lipn.univ-paris13.fr).

This work was funded by the ANR (Agence Nationale de la Recherche) under the $E-Fraud$ project .

randomly chosen subspaces of the original input space (i.e. the training set is sampled in the feature space). The outputs of the models are then combined, usually by a simple majority vote.

Several researchers have tried to use this principle for many classifiers. Fast algorithms, such as Decision trees are commonly used with ensembles, although slower algorithms can benefit from ensemble techniques as well. Examples of such approach can be found in [6] and [10], where the classifier consists of multiple trees constructed systematically by pseudorandomly selecting subsets of components of the feature vector, that is, trees constructed in randomly chosen subspaces. The essence of the method is to build multiple trees in randomly selected subspaces of the feature space (Random Forest). Trees in different subspaces generalize their classification in complementary ways, and their combined classification can be monotonically improved.

The main idea behind the proposed approach $RS-NDF$ is to combine the orthogonal projection operators power [4] in randomly selected subspaces of the feature space, the bootstrap idea and the ensemble learning paradigm. The $RS-NDF$ approach uses multiple versions of a training set by using a double bootstrap, i.e. sampling with replacement on examples and sampling without replacement on features. Each of these data sets is used to train a different NDF model. The $RS-NDF$ is then an ensemble of NDF, induced from bootstrap samples of the training data, using random features and examples selection in the model induction process. Prediction is made by aggregating (majority vote) the predictions of the ensemble to create a single output.

A. Principle of the Kohonen and Oja's Novelty Filter

In 1976, Kohonen and Oja [4] introduced an orthogonalising filter which extracts the parts of an input vector that are new, with respect to previously learned patterns. This is the desired functionality of a novelty filter. Another description of the filter is given in [7]. The novelty filter shows the novelties in an input pattern with respect to previously learned patterns. Furthermore, the novelty filter can distinguish the missing parts from the added parts in the input pattern with respect to the previously learned patterns.

The novelty filter may be implemented by a simple neural network architecture made by a single layer of linear units interconnected by a feedback "gain" time-variable matrix $\mathbf{M} = (m_{ij}) \in \mathbb{R}^{d\times d}$ (see fig. 1). Let us denote the set of input signals by a vector $\mathbf{x} = [x_1, x_2, ..., x_d]^T$ and let $\tilde{\mathbf{x}} = [\tilde{x_1}, \tilde{x_2}, ..., \tilde{x_d}]^T$ be the vector of output signals. Every element of the output $\tilde{\mathbf{x}}$, denoted by $\tilde{x_i}$, is assumed to receive a feedback from the other elements $\tilde{x_j}$ through connection weights m_{ij}. The network output signals $\tilde{x_i}$ are assumed to be linear combinations of the input signals x_i and the feedback signals as :

$$\tilde{x_i} = x_i + \sum_j m_{ij}\tilde{x_j} \tag{1}$$

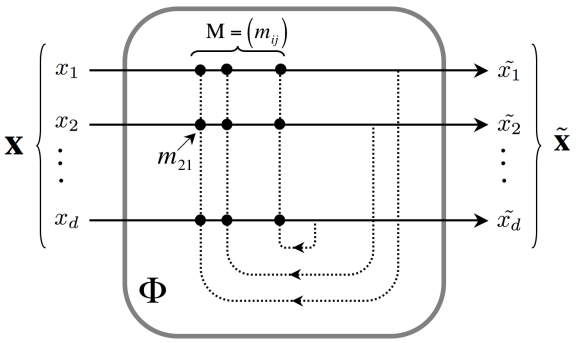


Fig. 1. The novelty filter implemented by a simple neural network architecture made by a single layer of linear units interconnected by a feedback connection

Equation (1) can be written in matrix notation as:

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{M}\tilde{\mathbf{x}} \tag{2}$$

The weights of the feedback connections m_{ij} characterize the internal state of the network. They are initialized at zero and then updated during learning. The learning phase follows an anti-Hebbian rule. The feedback matrix has the following state equation:

$$\frac{d\mathbf{M}}{dt} = -\alpha\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T \tag{3}$$

where the parameter α is adaptively modified during the learning phase, it may be interpreted as a measure of the similarity of the pattern under consideration with respect to the previously learned patterns.

The $overall$ network transfer function $\Phi \in \mathbb{R}^{d\times d}$ can be solved from implicit feedback equations:

$$\tilde{\mathbf{x}} = (\mathbf{I} - \mathbf{M})^{-1}\mathbf{x} = \Phi\mathbf{x} \tag{4}$$

where $\mathbf{I}$ denotes the identity matrix.

The differential equation for Φ is obtained as follows:

$$\begin{aligned} \frac{d\Phi^{-1}}{dt} &= -\Phi^{-1}\frac{d\Phi}{dt}\Phi^{-1} = -\frac{d\mathbf{M}}{dt} \\ \frac{d\Phi}{dt} &= -\alpha\Phi^2\mathbf{x}\mathbf{x}^T\Phi^T\Phi \end{aligned} \tag{5}$$

This is a matrix $Bernoulli$ equation. The Greville's theorem [24] gives a recursive expression to estimate the transfer function of the network as follows:

$$\Phi_k = \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^T}{\| \tilde{\mathbf{x}}_k \|^2} \tag{6}$$

where $\mathbf{x}_k = [x_1, x_2, ..., x_d]^T$ is a d-dimensional vector from the reference data matrix .

An interesting alternative approach was given by Kassab and al. [8], [9], who introduces the identity matrix in the learning formula for considering separately all training

examples, and consequently all their features. During learning phase, features which frequently appear in the training examples become more and more habituated as compared to the less frequent ones. This helps to more discriminate the relevant and irrelevant examples.The new learning rule is then defined as:

$$\Phi_k = \mathbf{I} + \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T}{\| \tilde{\mathbf{x}}_k \|^2} \quad (7)$$

where $\tilde{\mathbf{x}}_k = (\mathbf{I} + \Phi_{k-1})\mathbf{x}_k$ and Φ_0 is zero, or null matrix. The work described in this paper uses this new learning rule.

For the novelty detection problem, two proportions can be computed:

- *Novelty proportion*: this measure quantifies the novelty of an input data with respect to data that has been previously seen during the training.

$$N_{\mathbf{x}_i} = \frac{\| \tilde{\mathbf{x}}_i \|}{L \times \| \mathbf{x}_i \|} \quad (8)$$

where L is the number of examples used for the training.

- *The habituation proportion*: this measure calculates the similarity of an example with the previously learned one:

$$H_{\mathbf{x}_i} = 1 - N_{\mathbf{x}_i} \quad (9)$$

This proportion could be considered as the classification score of an example x_i. It indicates the probability that x_i belongs to the novel class. It is also possible to compute a vector P_Φ that represents the examples used for the training. So the classification score of the input data can be computed by comparing with this vector as explained below:

$$P_\Phi = \sum_{f \in F} H_f \overrightarrow{u_f} \quad (10)$$

where $\overrightarrow{u_f}$ is the unit vector associated with each feature f in F (set of all features) on the space spanned by the NDF, H_f is the habituation proportion of a feature f calculated as below:

$$H_f = 1 - \frac{\| \Phi \overrightarrow{u_f} \|}{L \times \| \overrightarrow{u_f} \|} \quad (11)$$

The vector P_Φ is defined as a weight vector. The components corresponding to P_Φ represent the habituation proportions of features in the data space.

To determine a detection threshold for each filter, we used the following principle:

- Scores (output's filter) attributed to the learning data can be used as a good indicator of the scores of data which can be positive and which are easy to detect because they are strongly similar to the data used for the learning. Consequently, the average of these scores can be admitted as a higher limit for the detection threshold.
- The scores attributed to available data for learning before their use, can be used as a good indicator of the scores of data which are positive but which are less easy to detect. Consequently, the average of these scores can be admitted as a lower limit for the detection threshold.

B. RS-NDF algorithm

The RS-NDF learning algorithm is shown below:

Algorithme 1 : Random Subspace Novelty Detection Filters

Inputs:
$sD = x_1, x_2, ..., x_L$ the traning dataset.
$sT = x_1, x_2, ..., x_M$ the testing dataset.
$F = f_1, f_2, , f_n$ the set of features.
NF the number of filters.
Φ_0 the initial matrix.
Outputs:
NF filters (Φ_i).
NF vectors that represent the target classes (P_{Φ_i}).
D Matrix of novelty detection.
Begin
for $i = 1$ to NF **do**
- Draw a bootstrap with replacement sample from the learning data sD.
- Draw a bootstrap without replacement on features in F.
- Induce a RS-NDF Φ_i using the randomly selected features (formula (7))

end for
for $i = 1$ to M **do**
for $k = 1$ to NF **do**
- Compute the habituation proportion $H_{\mathbf{x}_i}$ (formula (9)) for $\mathbf{x}_i \in sT$ using filter Φ_k.

end for
- Aggregate the predictions of the ensemble and save the novelty detection results in D.

end for
end

III. Experiments

A. Databases description

To demonstrate the effectiveness of the proposed method, we experimented it with 8 data sets using a tenfold cross-validation. These data sets are summarized in table 2. All of which were reported in [11] and available at $http : //archive.ics.uci.edu/ml/datasets.html$. These data sets are publicly available and have been studied before by

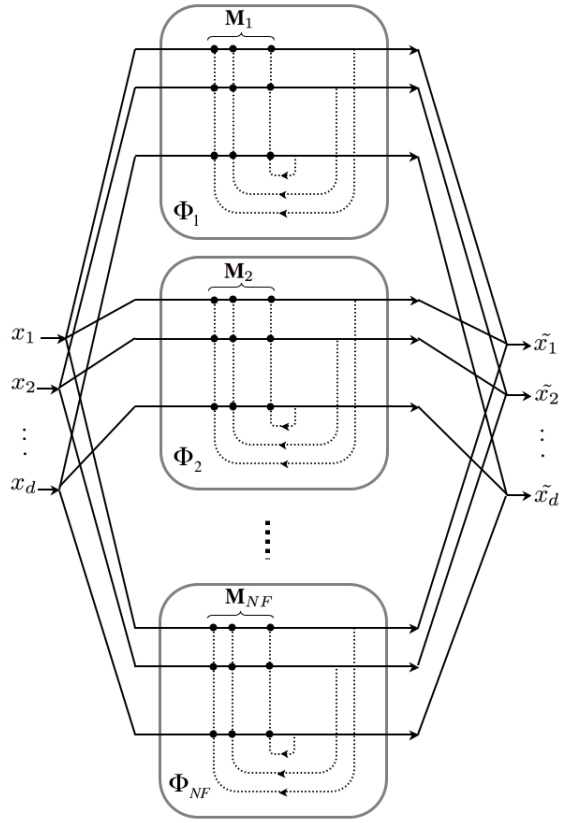


Fig. 2. The RS-NDF network architecture

many researchers with different methods. We compare our proposed method with the existing methods. Since all of the datasets are for binary or multi-class classication problems, they were transformed into novelty detection context. A class is randomly chosen among all classes and assigned as the novelty class. Other remaining classes are merged to obtain the normal class for learning.

TABLE I
DATA SET SUMMARY

Dataset	Dimension	Size	Size of Novelty class
Glass	9	214	70
Ionosphere	34	351	225
Oil	48	937	41
Spectf	44	187	15
Waveform	21	5000	1647
WDBC	30	569	212
Wine	13	178	59
Yeast	8	1484	244

- *Glass dataset*: the glass identification database is composed of 214 examples with 9 variables, the data is divided in seven classes [15].

- *Ionosphere dataset*: this radar data is composed of 351 instances discribed by 34 variables and divided into two classes: good or bad. *Good* radar instances showed evidence of some type of structure in the ionosphere [14]. *Bad* instances did not, their signals passed through the ionosphere. Figure 3 shows the projection in the pca plan of the cloud of the Ionosphere dataset; where circles represent the new data (the novelty).

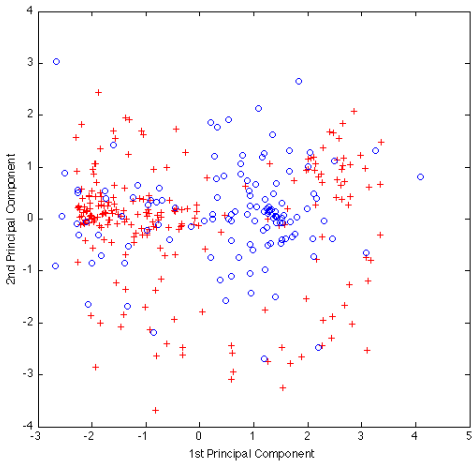


Fig. 3. The plot shows in the principal component subspace, data from Ionosphere dataset. Training data points are denoted by blue circles (o), and testing data (novelty) by red crosses (+).

- *Oil dataset*: this dataset is composed of 937 observations divided in two classes [16]. It has 41 oil slick samples and 896 non-slick samples. Originally this dataset has 50 features.
- *Spectf dataset*: this dataset discribes diagnosing of cardiac Single Proton Emission Computed Tomography image [17]. The data was split into training set and test set. We only used the testing dataset composed of 187 observations. Each of the patients (observations) is classified into two categories: normal and abnormal.

- *Waveform dataset*: this dataset consists of 5000 observations divided into three classes [18]. The original dataset included 40 features, 19 of which were attributed to noise, with mean 0 and variance 1. Each class was generated from a combination of 2 of 3 *base* waves. Figure 4 shows a projection in the pca plan of the cloud of the waveform dataset; crosses represent the learning data and circles represent the new data (the novelty).

- *WDBC dataset*: the Wisconsin Diagnostic Breast Cancer includes 569 observations with 32 features (ID, diagnostic, 30 real-valued input features). Each observation is labeled as benign (357) or malignant

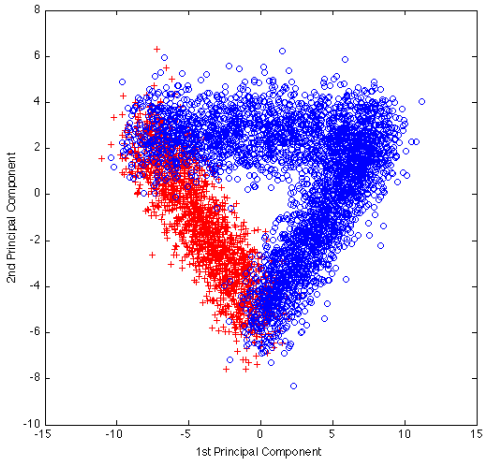


Fig. 4. The plot shows in the principal component subspace, data from Waveform dataset. Training data points are denoted by blue circles (o), and testing data (novelty) by red crosses (+).

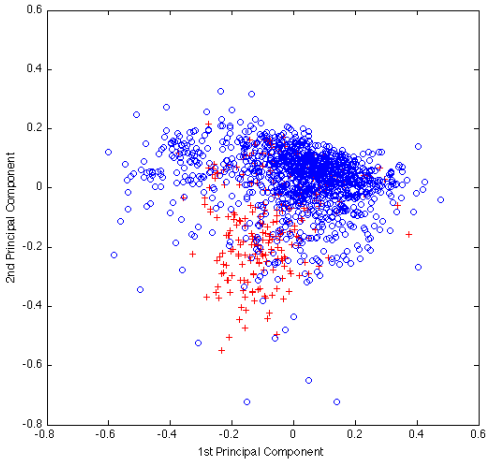


Fig. 5. The plot shows in the principal component subspace, data from Yeast dataset. Training data points are denoted by blue circles (o), and testing data (novelty) by red crosses (+).

(212). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

- *Wine dataset*: these data are the results of a chemical analysis of wines grown in a same region in Italy but derived from three different cultivars [19]. This dataset consist of 178 instances discribed by 13 features and divided in three classes.

- *Yeast dataset*: this dataset consists of 1484 instances representing ten classes discribed by 9 attributes (see figure 5). Note that the goal is not to classify; it is known classes are hardly separable in such data. Instead, we want to explore the novelty problem with this data. The data set [20] consists of measurements of the expression of each yeast gene in 300 knock-out mutation experiments. After leaving out all genes and experiments without significant expression.

B. Performance Measurement and experimental protocol

Different evaluation measures assess different characteristics of machine learning algorithms. The empirical evaluation of algorithms and classifiers is a matter of on-going debate among researchers. Most measures we use today focus on classifiers ability to identify classes correctly. Measures of quality of classification are built from a confusion matrix which records correctly and incorrectly recognized examples for each class. Table 1 presents a confusion matrix for binary classification, where TP are true positive, FP false positive, FN false negative, and TN true negative counts.

TABLE II
A CONFUSION MATRIX FOR BINARY CLASSIFICATION.

	Predicted Positive Class	Predicted Negative Class
Actual Positive Class	TP (True Positive)	FN (False Negative)
Actual Negative Class	FP (False Positive)	TN (True Negative)

$$TrueNegativeRate(Acc-) = \frac{TN}{TN+FP} \quad (12)$$

$$TruepositiveRate(Acc+)(Recall) = \frac{TP}{TP+FN} \quad (13)$$

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (15)$$

$$Fmesure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (16)$$

$$G - mean = (Acc - *Acc+)^{1/2} \quad (17)$$

We also used the Area under the ROC Curve (AUC) [13]. The ROC curve is a graphical representation of the trade off between the false negative and false positive rates for every possible cut off. There are several methods to estimate the area under the ROC curve. In the case of binary classification, the balanced AUC_b is defined as:

$$AUC_b = \frac{Acc - +Acc+}{2} \quad (18)$$

We use all these measurements to compare our method $RS - NDF$ with some published approaches commonly

used for the problem of novelty detection. Ten-fold-cross-validations were carried out to obtain all the performance metrics. Note that the most used empirical measure, $Accuracy$, does not distinguish between the number of correct labels of different classes.

We compared the performances of the $RS-NDF$ algorithm in term of performance and the traditional novelty detection methods. The selected algorithms to compare with are: The Principal Components Analysis (PCA) [12], The one class Support Vector Machines ($OC-SVM$) [23], The auto-associative Multi Layers Perceptron (MLP) [21], and the basic model NDF. The standard statistical technique (PCA) was used for detecting outliers [12]. The problem of novelty detection is closely related to that of statistical outlier detection. The novelty detection method tries to identify outliers that differ from the distribution of ordinary data. This technique consists in extracting structure from a dataset by performing an orthogonal basis transformation to the coordinate system in which the data is described. This can reduce the number of features needed for effective data representation. After applying the PCA transformation, the subspace representing the model (which is generated by the k first principal components) is associated with systematic variations in data. The projection of $x_i \in \mathbb{R}^n$ on the orthogonal space, induced $x \in \mathbb{R}^p, (p \ll n)$ with a reconstruction error.

We also compared the results of our approach with the $OC-SVM$. The purpose is to model the support of a data distribution, a binary valued function that is positive in those parts of the input space where the data lies, and negative otherwise. So the $OC-SVM$ model, can then detect inputs that were not in the training set (novel input). The $OC-SVM$ operates by selecting the optimal hyperplane that maximises the minimum distance to the training points that are closest to the hyperplanes.The support vectors are derived only from a data belonging to the same class, which requires a large number of training data in order to define a clear separation border. In addition, we used a technique of the auto-associative neural network: the Multi Layer Perceptron (MLP). These networks are trained to produce an approximation of a mapping function between the inputs and outputs of the network (target=input). The auto-associative neural network consists of an input neurons layer, followed by a hidden layers, and output neurons layer having the same size of the input layer.

IV. Results

For each database, the five approaches have been used and their results have been evaluated in terms of the seven performances metrics. Table 3 shows the performance of the different algorithms on $Wine$ data. In this results $RS-NDF$ shows great improvement over MLP, PCA, $OC-SVM$ and NDF on all metrics.

Tables 4 and 5 show the performance of $RS-NDF$ with the other algorithms applied on $Ionosphere$ database and $Glass$ database. As we can see, $RS-NDF$ outperforms the results obtained by MLP, PCA and $OC-SVM$ on all metrics ($Acc-$, $Precision(Prec)$, $F-measure$, $G-mean$ and AUC_b) except for the $Acc-$ measure, the NDF gives a slight better result on $Ionosphere$ database.

TABLE III
PERFORMANCE COMPARISON ON WINE DATA SET

	$Acc-$	$Acc+$ $(Recall)$	$Prec$	$F-$ $measure$	AUC_b	$G-$ $mean$
MLP	0,78	0,68	0,83	0,81	0,73	0,73
PCA	0,60	0,69	0,80	0,68	0,65	0,64
OC-SVM	0,68	0,76	0,81	0,73	0,72	0,71
NDF	0,84	0,78	0,88	0,86	0,81	0,81
RS-NDF	**0,87**	**0,85**	**0,92**	**0,89**	**0,86**	**0,86**

TABLE IV
PERFORMANCE COMPARISON ON IONOSPHERE DATA SET

	$Acc-$	$Acc+$ $(Recall)$	$Prec$	$F-$ $measure$	AUC_b	$G-$ $mean$
MLP	0,63	0,64	0,50	0,55	0,64	0,64
PCA	0,64	0,55	0,45	0,52	0,59	0,59
OC-SVM	0,66	0,56	0,45	0,54	0,61	0,60
NDF	**0,90**	0,61	0,57	0,70	0,76	0,74
RS-NDF	0,87	**0,74**	**0,65**	**0,74**	**0,80**	**0,80**

TABLE V
PERFORMANCE COMPARISON ON GLASS DATA SET

	$Acc-$	$Acc+$ $(Recall)$	$Prec$	$F-$ $measure$	AUC_b	$G-$ $mean$
MLP	**0,96**	0,49	0,49	0,65	0,73	0,69
PCA	**0,96**	0,43	0,46	0,62	0,69	0,64
OC-SVM	0,89	0,54	0,80	0,84	0,72	0,69
NDF	0,82	0,69	0,84	0,83	0,75	0,75
RS-NDF	0,87	**0,84**	**0,93**	**0,90**	**0,86**	**0,85**

In the tables above, we have shown that the $RS-NDF$ have favorable improvement over existing methods. We will use F-measure criteria and the signal output of NDF and $RS-NDF$ to further investigate the performances.

Table 6 shows the performance of the different algorithms in terms of accuracy as a 95% confidence interval for each dataset from a 10-fold cross validation.

TABLE VI
ACCURACY AS A 95% CONFIDENCE INTERVAL FOR EACH DATASET FROM 10-FOLD CROSS VALIDATION

Dataset	MLP	PCA	OC-SVM	NDF	RS-NDF
Glass	[58.35 71.03]	[54.07 67.05]	[71.52 82.64]	[71.52 82.64]	[81.14 90.01]
Ionosphere	[58.67 68.67]	[52.95 63.12]	[54.05 64.27]	[66.87 76.25]	[73.74 82.34]
Oil	[90.67 94.04]	[88.92 92.60]	[84.97 89.24]	[86.50 90.56]	[88.41 92.18]
Spectf	[68.53 79.06]	[70.13 80.46]	[68.35 78.87]	[69.73 80.11]	[71.74 81.85]
Waveform	[49.16 52.56]	[55.60 58.34]	[65.00 67.62]	[63.23 65.88]	[73.07 75.49]
WDBC	[58.34 66.27]	[68.47 75.80]	[74.02 80.37]	[85.60 90.86]	[86.47 91.56]
Wine	[67.86 80.54]	[55.62 69.67]	[64.53 76.61]	[75.72 86.97]	[80.35 90.48]
Yeast	[66.02 70.73]	[65.44 70.18]	[74.88 79.15]	[74.66 78.95]	[81.38 85.16]

The results are also confirmed by visual inspection. Note that to better understand the figures, the results should be analyzed in a color mode. Looking to the four radars graphics showed above, some conclusions can be drawn. In general $RS-NDF$ gives better results compared to other methods on all metrics and in the worst case the results obtained by NDF, PCA, MLP and $OC-SVM$ are slightly superior. For the waveform dataset, $RS-NDF$ outperforms the other algorithms using $Precision$, $F-measure$, $G-mean$ and AUC_b measures. The $Precision$ is defined as the percentage

of samples wich are correctly labeled as positive. So we can conclude that our approach classifies the novelty examples better than the other methods. Also for the $G-mean$ and AUC_b metrics,$RS-NDF$ gives a good results comparing to NDF MLP, PCA and $OC-SVM$. The $G-mean$ of accuracies, measured separately on each class, is associated to a point in the ROC curve and the idea is to maximize the accuracies of both classes while keeping them balanced.

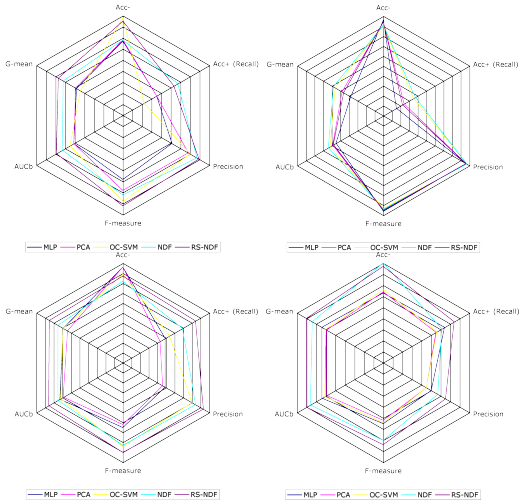


Fig. 6. Top left : Waveform, top right: Oil, down left: Glass, down right: ionosphere

The radars graphics of $Wine$, $Yeast$, $WDBC$ and $Spectf$ databases confirmed the good performances of our approach $RS-NDF$. The AUC_b shows that our approach is able to give interesting results. This metric is a quantitative representation of a ROC curve. The ROC is widely used for the evaluation of classifiers, it is a tool for visualizating, organizing and selecting classifiers based on their trade-offs between benefits (true positives) and costs (false negatives).

Figure 8 shows the outputs of the NDF approach (signal in blue (down)) and of the $RS-NDF$ approach (signal in red (top)). We can notice that the answers supplied by the $RS-NDF$ approach are more important than those supplied by the NDF approach. Note that there is a visible difference of variations between the signals produced by $RS-NDF$ and NDF. In fact, $RS-NDF$ shows more stability in the novelty detection answers than the single NDF. At the cross point it is possible to distinguish between the two classes.

Considering the F-measure, figure 9, $RS-NDF$ algorithm shows excellent results for all datasets. This metric, that combines the precision and recall, is commonly used in the information retrieval area as performance measure. As for $Wine$, $Yeast$, $Waveform$, $Ionosphere$, $Glass$ and $Spectf$ databases, our proposed algorithm outperforms all the other methods, except Oil and $WDBC$ datasets. $RS-NDF$ gives a slightly inferior result (0.95) comparing to MLP (0.96) applied on Oil database and the same result

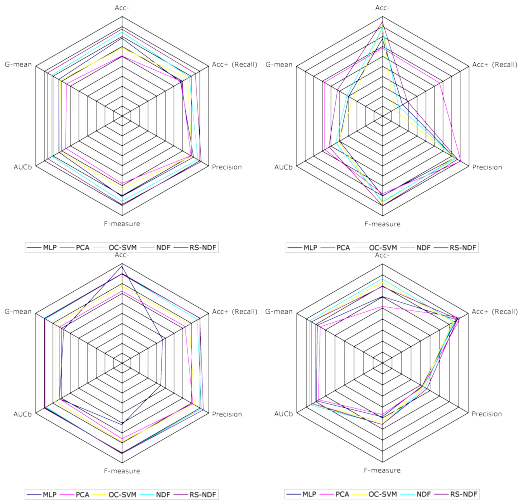


Fig. 7. Top left : Wine, top right: Yeast, down left: WDBC, down right: Spectf

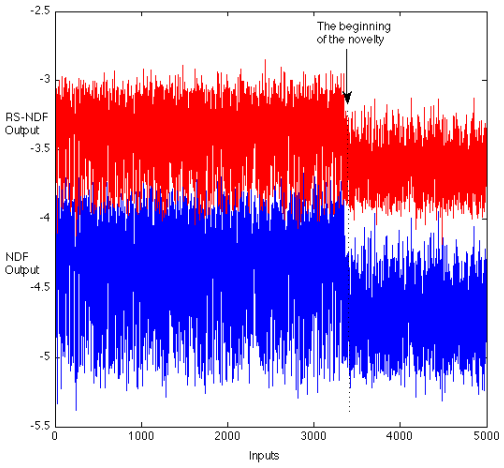


Fig. 8. Outputs of RS-NDF and NDF approaches

(0.91) on the $WDBC$ dataset. Figure 10 presents the Radar of $Acc+$ on all datasets. The $Acc+$ measure represents the models capacity to detect the novelty class. We chose this metric in purpose to show the good capacity of $RS-NDF$ to detect the novelty class. So we can clearly see that our approach gives better results comparing to the other methods.

V. Conclusion and Future Work

We have proposed a new approach to the novelty detection problem. It is based on the orthogonal projection operators, the bootstrap method and the ensemble learning paradigm. The proposed approach $RS-NDF$ is an ensemble of NDF, induced from bootstrap samples of the training data,

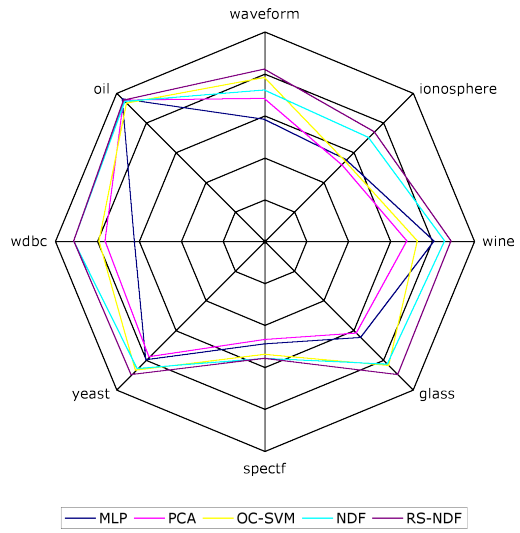


Fig. 9. The F-measure on different data sets

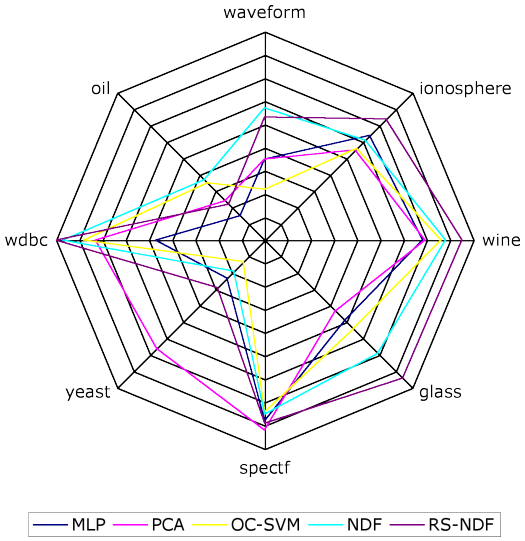


Fig. 10. The Acc+ measure on different data sets

using random feature selection in the NDF induction process. Prediction was made by aggregating the predictions of the ensemble. Performance metrics such as precision and recall, false positive rate and false negative rate, F-measure and AUC are computed using a tenfold cross-validation experiments on publicly available datasets. Significant improvements in accuracy were obtained using our method. The $RS-NDF$ generally exhibits a substantial performance improvement and have favorable performance compared to the existing algorithms . Thanks to an online learning algorithm, the RS-NDF approach is also able to track changes in data over time. Our future work will follow several interconnected avenues: find new characteristics of the algorithms which must be evaluated, consider new measures of algorithm performance, and search for others applications such learning keystroke patterns for user authentication.

REFERENCES

[1] M. Markou and S. Singh, *Novelty detection: a review - part 1: statistical approaches.* Signal Processing, 83:2481-2497, 2003.

[2] M. Markou and S. Singh, *Novelty detection: a review - part 2: neural network based approaches.* Signal Processing, 83:2499-2521, 2003.

[3] S. Marsland, *Novelty detection in learning systems.* Neural Computing, 3:157-195, 2003.

[4] Kohonen, T. and Oja, E., *Fast Adaptive Formation of Orthogonalizing Filters and Associative Memory in Recurrent Networks of Neuron-Like Elements.* Biological Cybernetics, 21:85-95, 1976.

[5] Breiman L., *Bagging Predictors.* Machine Learning, 24:(2), 123140, 1996.

[6] Ho, Tin Kam., *he Random Subspace Method for Constructing Decision Forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:(8), 832844, 1998.

[7] Kohonen, T., *Self-Organization and Associative Memory.* 3rd ed. Springer, Berlin, 1993.

[8] Kassab, R., Lamirel, J.-C. and Nauer, E., *Novelty Detection for Modeling Users Profile.* The 18th International FLAIRS Conference. 830831, 2005.

[9] Kassab, R., et Alexandre, F., *Incremental Data-driven Learning of a Novelty Detection Model for One-Class Classification Problem with Application to High-Dimensional Noisy Data.* Machine Learning. 74(2) :191234, 2009.

[10] Breiman, L., *Random forest.* Machine Learning, 45, 532, 2001.

[11] Asuncion A. and Newman D.J., *UCI Machine Learning Repository.* Irvine, CA, University of California, 2007.

[12] David M.J. Tax, Alexander Ypma and Robert P.W. Duin., *Support Vector data description applied to machine learning vibration analyses* Annual Conference of the Advanced School for computing and Imaging, pages 398405, 1999.

[13] Bradeley,P.W., *The use of area under the ROC curve in the evaluation of machine learning algorithms.* Pattern Recognition,30, 1145-1159 (1997).

[14] Sigillito, V. G., Wing, S. P., Hutton, L. V., *Classification of radar returns from the ionosphere using neural networks.* Johns Hopkins APL Technical Digest, 10, 262-266 (1989).

[15] B. German,Vina Spiehler, *Diagnostic Products Corporation.* (213) 776-0180 (ext 3014) September, 1987.

[16] Miroslav Kubat, Robert C. Holte, and Stan Matwin, *Machine Learning for the Detection of Oil Spills in Satellite Radar Images.* Machine Learning, volume 30, pp. 195-215, (1998).

[17] Kurgan, L.A., Cios, K.J., Tadeusiewicz, R., Ogiela, M. Goodenday, L.S., *Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis.* Artificial Intelligence in Medicine, vol. 23:2, pp 149-169, Oct 2001.

[18] Breiman,L., Friedman,J.H., Olshen,R.A., Stone,C.J. *Classification and Regression Trees. Wadsworth International.* Wadsworth International Group: Belmont, California. (see pages 43-49)(1984).

[19] Forina, M., *An Extendible Package for Data Exploration, Classification and Correlation.* Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno,16147 Genoa, Italy.

[20] Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, Bennett HA, Coffey E, Dai H, He YD, Kidd MJ, King AM, Meyer MR, Slade D, Lum PY, Stepaniants SB, Shoemaker DD, Gachotte D, Chakraburtty K, Simon J, Bard M, Friend SH., *Functional discovery via a compendium of expression profiles.* Cell. Jul 7;102(1):109-26. (2000).

[21] Rumelhart, D. E., Hinton, G. E., et Williams, R. J., *Learning internal representation by error propagation.* Parallel Distributed Processing : Explorations in the microstructures of cognition, MIT Press. 318362.(1986).

[22] I.T. Jolliffe. *Principal Component Analysis.* Springer Series in Statistics. Springer-Verlag, Berlin, Germany,1986.

[23] Scholkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., et Williamson, R. C., *Estimating the support of a high-dimensional distribution.* Neural Computation. 13(7) :14431471.(1999).

[24] Greville, T. N. E. *Some applications of the pseudoinverse of a matrix.* SIAM Rev. 2 :1522. (1960).

Topographic Under-Sampling for Unbalanced Distributions

Fatma Hamdi, Mustapha Lebbah and Younès Bennani

***Abstract*— Several aspects could affect the existing machine learning algorithms. One of these aspects is related to unbalanced classes in which the number of observations belonging to a class, greatly exceeds the observations in other classes. We propose in this paper an under-sampling method which uses self-organizing map to cluster the majority class guided with minority class. The proposed approach has been validated on multiple data sets using decision trees as a classifier with cross validation. The experimental results showed that elimination from majority class by integrating Neighborhood Cleaning Rule in SOM algorithm, produce high and very promising performance.**

I. Introduction

Many learning approaches assume that the classes share similar prior probabilities. However, in many real-world tasks this assumption is grossly violated. Often, the ratios of prior probabilities between classes are significantly skewed. This situation is known as the unbalance problem. The unbalanced distribution of classes constitutes a difficulty for standard learning algorithms because they are biased toward the majority classes. Data set is defined as unbalanced when one of the classes (the minority one) is under-represented comparing to the other class (the majority one). We can illustrate a simple example taken frequently in literature: if 99% data belong to one class, it will be difficult to do better than the 1% error obtained by classifying all individuals in this class. Therefore it is necessary to find alternatives and an appropriate assumptions to resolve the problem of unbalanced data.

In [1], [2] the authors propose to distinguish six categories of problems associated with unbalanced data, and learning with rare classes. (**a**) inappropriate distance: in this case the measure used in the learning step is not adapted to unbalanced classes. (**b**) "absolute" lack: this problem is observed when the available data are not enough to define the boundaries of classes. (**c**) "relative" lack : it is similar to the absolute lack, except that in this case it depend on the size of the majority class. (**d**) Fragmentation of data: this problem is related to algorithms with a top-down approach, such as decision trees or LBG clustering algorithm "Linde-Buzo-Gray", [3]. (**e**) inappropriate induction margin: this is the margin applied to the rule learned on the training data in order to generalize. (**f**) data noisy: the noise has more impact on the minor classes than frequent classes. Therefore, the model is unable to distinguish the sound of rare examples. The unbalanced distribution of classes is not only the problem responsible for the failure of learning algorithms. There are other difficulties that can disrupt the learning of the minority class like overlapping or duplication of examples.

Fatma Hamdi, Mustapha Lebbah and Younès Bennani are with the LIPN-UMR 7030 - CNRS, Université Paris 13, 99, avenue Jean-Baptiste Clément 93430 Villetaneuse, France. Email: firstname.secondname@lipn.univ-paris13.fr.

Several methods have been proposed to address the problem of unbalanced data that can be defined in two categories [4]:

- **Algorithmic level**. We find in this case methods that take into account the inherent unbalance using a matrix of costs. In [5] the authors propose the introduction of bias using a weighted distance for k nearest neighbors $(k\text{-}nn)$. The purpose is to offset the unbalance of data without altering the class distribution. In [6] the authors propose also a weighted distance and provide a comparative study about the influence of overlapping. The majority of algorithms provide for each example a probability to belong to a particular class. Therefore the decision is taken by setting a threshold. This is the case of naive Bayes, or some neural networks. Therefore it is possible to take into account the unbalance of data by decreasing the decision threshold for the minority class (or increase the threshold for the majority class). Thus, the sensitivity is improved on the minority class.
- **Data level**. In this case we find sampling methods that modify the distributions of original class by preprocessing. Two categories are considered: under-sampling the majority class and the over-sampling the minority class. The over-sampling aims to rebalance the data by increasing the number of examples belonging to the minority class. Methods such as on random-sampling algorithm or SMOTE (Synthetic Minority Over-sampling Technique), allow to generate artificial observation in the minority class [7].
  Unlike the over-sampling, to rebalance data we can remove some examples belonging to the majority class. In this paper we focus on this categorie of algorithms. Several methods as NCR (Neighborhood Cleaning Rule), Tomek links or their combinations were experimentally shown to work well [8].
  - Tomek links [9]: given two examples $\mathbf{x}_i$ and $\mathbf{x}_j$ belonging to different classes, and $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. A pair $(\mathbf{x}_i, \mathbf{x}_j)$ is called a Tomek link if there is not an example $\mathbf{x}_k$, such that $d(\mathbf{x}_i, \mathbf{x}_k) < d(\mathbf{x}_i, \mathbf{x}_j)$ or $d(\mathbf{x}_j, \mathbf{x}_k) < d(\mathbf{x}_i, \mathbf{x}_j)$. If two examples form a Tomek link, then either one of these examples is noise or both examples are borderline. Tomek links can be used as an under-sampling method or as a data cleaning method. As an under-sampling method, only examples belonging to the majority class are eliminated, and as a data cleaning method, examples of both

classes are removed.

- Condensed Nearest Neighbor Rule (CNN) is used to find a consistent subset of examples. A subset $\hat{A} \subset A$ is consistent with A if using a 1-nearest neighbor (1-NN), $\hat{A}$ correctly classifies the examples in A. An algorithm to create a subset $\hat{A}$ from A as an under-sampling method is the following : First, randomly draw one majority class example and all examples from the minority class and put these examples in $\hat{A}$. Afterwards, use a 1-NN over the examples in $\hat{A}$ to classify the examples in A. Every misclassified example from A is moved to $\hat{A}$. It is important to note that this procedure does not find the smallest consistent subset from A. The idea behind this implementation of a consistent subset is to eliminate the examples from the majority class that are distant from the decision border, since these examples might be considered less relevant for learning.
- NCR: Neighborhood Cleaning Rule [10] uses the Wilson's Edited Nearest Neighbor Rule (ENN) [11] to remove majority class examples. ENN removes any example whose class label differs from the class of at least two of its three nearest neighbors. NCR modifies the ENN in order to increase the data cleaning. For a two-class problem the algorithm can be described in the following way: for each example $\mathbf{x}_i$ in the training set, its 3-NN are found. If $\mathbf{x}_i$ belongs to the majority class and the classification given by its 3-NN contradicts the original class of $\mathbf{x}_i$, then $\mathbf{x}_i$ is removed. If $\mathbf{x}_i$ belongs to the minority class and its 3-NN misclassify $\mathbf{x}_i$, then the 1-NN that belong to the majority class are removed.

Other methods propose combining different under-sampling algorithms as "One sided selection OSS" (Tomek links + CNN) [8]. In [7] authors propose algorithm that adds artificial observations using SMOTE method instead of increasing the weight of examples of the minority class, [1]. [12] offer a new approach to pre-selective treatment of unbalanced data which combines local over-sampling minority class with data filtering difficult for the majority class.

II. Topographic Neighborhood Cleaning Rule (TNCR)

A. Topological Quantization and adaptive Cleaning

In order to better guide the under-sampling we use self-organizing maps (SOM). We propose to modify the learning algorithm by removing at each iteration, examples which "disturb" the minority data. The idea consists on integrating the rule of cleaning algorithm NCR as a third step in the SOM learning algorithm. This rule will be applied locally at each cell of the map. Therefore, the SOM algorithm is used in a "pseudo-unsupervised" case, where only positive labels ("+", minority) are used. The cleaning phase implies changing in the training set $\mathcal{A}$ which decreases gradually with iterations. The approach we propose is a hybrid approach: an action on the data with the cleaning phase and modified SOM algorithm.

Our approach is titled TNCR (Topographic Neighborhood Cleaning Rule) will have the opportunity to delete multiple negative examples during the learning phase of the SOM. In contrast to NCR, which removes observation from the majority class, our approach applies a local cleaning in each cell. The self-organizing maps presented by [13] have been widely used for clustering and visualization of multidimensional data set. This model consists of clustering of data sets $\mathcal{A} = \{\mathbf{x}_i \in \mathcal{R}^n, i = 1 : N\}$ where $\mathbf{x}_i = (x_{i1}, x_{i2}..., x_{ij}, ..., x_{in})$. Each example $\mathbf{x}$ is associated with positive $label$ ("+" minority classes) or negative ("-", majority classes). The negative label "-" is used only in the rule of cleaning. We denote by $\mathcal{A}^+$ the set of positive examples and by $\mathcal{A}^-$ negative data set ($\mathcal{A} = \mathcal{A}^+ \cup \mathcal{A}^-$).

As with a traditional self-organizing map, we assume that the lattice $\mathcal{C}$ (map) has a discrete topology defined by an indirect graph. Usually, this graph is a regular grid in one or two dimensions. For each pair of cells (c,r) on the map, the distance $\delta(c, r)$ is defined as the length of the shortest chain linking cells r and c. For each cell c this distance defines a neighboring cell; a kernel positive function $\mathcal{K}$ ($\mathcal{K} \geq 0$ and $\lim_{|y|\to\infty} \mathcal{K}(y) = 0$) is introduced to determine the neighboring area. We define the mutual influence of two cells c and r by $\mathcal{K}(\delta(c, r))$. In practice, as for classical topological maps, we use a smooth function to determine the size of the neighboring area: $\mathcal{K}^T(\delta(c, r)) = \exp(\frac{-\delta(c,r)}{T})$. Using this kernel function, T becomes a parameter of the model. As in the Kohonen algorithm, we decrease T from an initial value T_{max} to a final value T_{min}. For each cell c of the grid (map), a referent vector $\mathbf{w}_c = (w_{c1}, w_{c2}..., w_{ck}, ..., w_{cn})$ is associated. The set of referent vectors is denoted by $\mathcal{W} = \{\mathbf{w}_c, \mathbf{w}_c \in \mathcal{R}^n\}_{c=1}^{|\mathcal{W}|}$. Each referent $\mathbf{w}_c$ is associated with an assigned subset of data denoted by P_c (cluster). The set of cluster is denoted by $\mathcal{P} = \{P_1, ..., P_c, ..., P_{|W|}\}$. The set of parameter $\mathcal{W}$ and the size of the cleaned data set have to be estimated iteratively by minimizing the classical objective function defined as follows:

$$\begin{aligned}\mathcal{R}(\mathcal{A}_t, \chi, \mathcal{W}) &= \sum_{\mathbf{x}_i \in \mathcal{A}_t} \sum_{r \in C} \mathcal{K}^T(\delta(\chi(\mathbf{x}_i), r))||\mathbf{x}_i - \mathbf{w}_r||^2 \\ &= \sum_{\mathbf{x}_i \in \mathcal{A}_t^-} \sum_{r \in C} \mathcal{K}^T(\delta(\chi(\mathbf{x}_i), r))||\mathbf{x}_i - \mathbf{w}_r||^2 \\ &+ \sum_{\mathbf{x}_i \in \mathcal{A}^+} \sum_{r \in C} \mathcal{K}^T(\delta(\chi(\mathbf{x}_i), r))||\mathbf{x}_i - \mathbf{w}_r||^2 \end{aligned} \quad (1)$$

where χ assigns each observation $\mathbf{x}_i$ to a single cell in the map $\mathcal{C}$.

The principle phases of our algorithm is presented as follows:

- **Input**:
  1) Learning data set, $\mathcal{A}_{Init} = \mathcal{A}_{Init}^- \cup \mathcal{A}_{Init}^+$
  2) k: size of neighborhood associated to local cleaning

- **Output** :
  1) Referent vectors $\mathcal{W}$.
  2) Under-sampled data set $\mathcal{A}_{final} = \mathcal{A}^+_{final} \cup \mathcal{A}^-_{final}$ ($|\mathcal{A}_{final}| \leq |\mathcal{A}_{Init}|$), where $|\mathcal{A}^+_{final}| = |\mathcal{A}^+_{Init}|$.

- **Assignment phase**: Each example $\mathbf{x}_i$ is assigned to the closest prototype $\mathbf{w}_c$ using the assignment function, defined as follows:

$$\chi(\mathbf{x}_i) = \arg\min_c \left(\|\mathbf{x}_i - \mathbf{w}_c\|^2\right) \quad (2)$$

  This equation minimizes the cost function with respect to χ.
- **Quantization phase**: Minimize $\mathcal{R}(\mathcal{A}_t, \chi, \mathcal{W})$ with respect to $\mathcal{W}$ by fixing $\mathcal{A}^-$ and χ. The prototype vectors are updated using following expression:

$$\mathbf{w}_c = \frac{\sum_{r \in C} \mathcal{K}^T(\delta(c,r)) \sum_{\mathbf{x}_i \in \mathcal{A}_t, \chi(\mathbf{x}_i)=r} \mathbf{x}_i}{\sum_{r \in C} \mathcal{K}^T(\delta(c,r))|P_r|} \quad (3)$$

- **Cleaning Phase** : At iteration t, for each cell $c \in \mathcal{C}$
  - if $\mathbf{x}_i \in \mathcal{A}^-_t \wedge k\text{-}nn(\mathbf{x}_i) \subset P_{\chi(\mathbf{x}_i)} \subset A^+_t$ then $A_{t+1} \leftarrow A_t - \{\mathbf{x}_i\}$;
  - if $\mathbf{x}_i \in \mathcal{A}^+_t \wedge k\text{-}nn(\mathbf{x}_i) \subset P_{\chi(\mathbf{x}_i)} \subset A^-_t$ then $A_{t+1} \leftarrow A_t - k\text{-}nn(\mathbf{x}_i)$;

The three phases minimize the cost function (1). The first two phases are similar to the traditional SOM algorithm. The third phase will minimize the cost function with respect to $\mathcal{A}^+$. As in the traditional SOM algorithm, we decrease the value of T between two values T_{max} and T_{min}, to control the size of the neighbourhood influencing a given cell on the map. For each T value, we get a cost function $\mathcal{R}(\mathcal{A}_t, \chi, \mathcal{W})$, and therefore the expression varies with T. We apply the following formula: $T = T_{max}\left(\frac{T_{min}}{T_{max}}\right)^{\frac{t}{N_{iter}-1}}$, where N_{iter} is the number of iteration.

At the end of learning, self-organizing map provides a partition of data in $|\mathcal{W}|$ clusters. Thus the new approach TNCR provides a new learning data set less than or equal to the initial data set $\mathcal{A}_t$ ($|\mathcal{A}_{final}| \leq |\mathcal{A}_t|$, $|\mathcal{A}^+_{final}| = |\mathcal{A}^+_{Init}|$).

III. Validation

We used different data set from UCI directory, [14], which have varying degrees of unbalance. Table I shows for each data set, the number of examples, the number of attributes (quantitative and qualitative) and the distribution of minority and majority classes. For data set with more than two classes, we chose the minority class as positive class and the rest as negative class (majority class). Several assessment indices exist in literature, but for our experiments we chose to compute two measures. The first one is the conventional AUC index "Area under curve" and the second is a new index IBA called "Index of Balanced Accuracy," presented by [15]. All results presented below are obtained with the local neighborhood setting $k = 3$. Concerning the parameters and the initialization of self-organizing map, we use the "SOM Toolbox" Kohonen (*http://www.cis.hut.fi/projects/somtoolbox/*).

The different measures are derived from a 2×2 confusion matrix as that given in Table II, where TP (true positive), FN (false negative), TN (true negative) and FP (false positive) represent the number of examples belonging in each case. True positive rate (also referred to as recall or sensitivity) is the percentage of positive examples which are correctly classified, $TPrate = TP/(TP + FN)$. True negative rate (or specificity) is the percentage of negative examples which are correctly classified, $TNrate = TN/(TN + FP)$.

Data sets	Size	Dim. (quanti, quali)	Size min/maj	Class % (min, maj)
Post-operative	90	8(1,7)	24,66	26.67, 73.33
Thyroid	215	5 (5,0)	30,185	13.95, 86.04
Ecoli	336	7 (7,0)	35,301	10.42, 89.58
Satimage	6435	36 (36,0)	626,5809	9.72, 90.27
Glass	214	9 (9,0)	17,197	7.94, 92.06
Flag	194	28(10,18)	17,177	8.76, 91.24

TABLE I

DATA SET. THE COLUMN SHOWS THE NUMBER OF EXAMPLES, THE NUMBER OF ATTRIBUTES (QUANTITATIVE AND QUALITATIVE)AND THE DISTRIBUTION OF MINORITY AND MAJORITY CLASSES.

"Area under curve", AUC

One of the most widely-used techniques for the evaluation of classifiers in unbalanced cases is the ROC curve. The area under the ROC curve ("Area under curve, AUC) is a synthetic indicator of the ROC curve. There are several methods to estimate the area under the ROC curve. In the case of binary classification, the ROC curve is given by the coordinates ((0,1), (TN / (TN + FP), TP / (TP + FN)), (1,0)) and thus AUC is defined as follows ([16]):

$$AUC = (TPrate + TNrate)/2$$

However, we observe that this measure is biased with respect to the unbalanced data and to the proportions of correct and incorrect classifications.

"Index of Balanced Accuracy", IBA

The AUC minimizes the negative influence of the unbalanced of classes, but doesn't distinguish the contribution of each class for overall performance. This means that different combinations $TPrate$ and $TNrate$ provide the same AUC value. To address this constraint by giving the pertinence to

		predicted	
		class (+)	class (-)
real class	class (+)	TP	FN
	class (-)	FP	TN

TABLE II

CONFUSION MATRIX FOR A TWO-CLASS PROBLEM

Data sets	AUC (%)		IBA (%)	
	NCR	TNCR	NCR	TNCR
Post-operative	45.62 (16.49)	**47.89** (18.46)	9.89 (10.92)	**12.08** (18.56)
Thyroid	**95.84** (6.51)	95.12 (8.58)	80.64 (27.95)	**81.85** (27.71)
Ecoli	83.70 (12.24)	**84.46** (12.33)	25.42 (19.37)	**26.69** (18.78)
Satimage	82.30 (2.42)	**83.42** (2.21)	24.05 (5.21)	**24.92** (4.87)
Flag	**61.61** (19.32)	61.18 (18.66)	12.10 (23.10)	**14.20** (25.84)
Glass	97.37 (7.58)	**97.62** (7.61)	85.92 (28.44)	**91.26** (24.84)

TABLE III

AUC (%) AND IBA (%) COMPUTED WITH DECISION TREES CLASSIFIER AFTER CROSS VALIDATION. AUC: AREA UNDER CURVE. IBA: INDEX OF BALANCED ACCURACY

Data sets	RR(%)	
	NCR	TNCR
Post-operative	75.76	46.97
Thyroid	4.87	7.03
Ecoli	14.29	19.61
Satimage	10.61	13.74
Flag	25.43	38.42
Glass	7.11	12.7

TABLE IV

RR (%):REMOVAL RATE. 100% CORRESPONDS TO THE NEGATIVE CLASS (MAJORITY CLASS)

the positive class, in [15] the authors propose a new measure, denoted by (IBA), which computes the area of a rectangular region in a two-dimensional space called "Balanced Accuracy Graph". The measure is defined as follows :

$$IBA = (1 + Dominance) \times Gmean^2$$

This index is defined by the product of two terms. The first term is a simple measure evaluating the relationship between the TPrate and TNrate,

$$Dominance = TPrate - TNrate$$

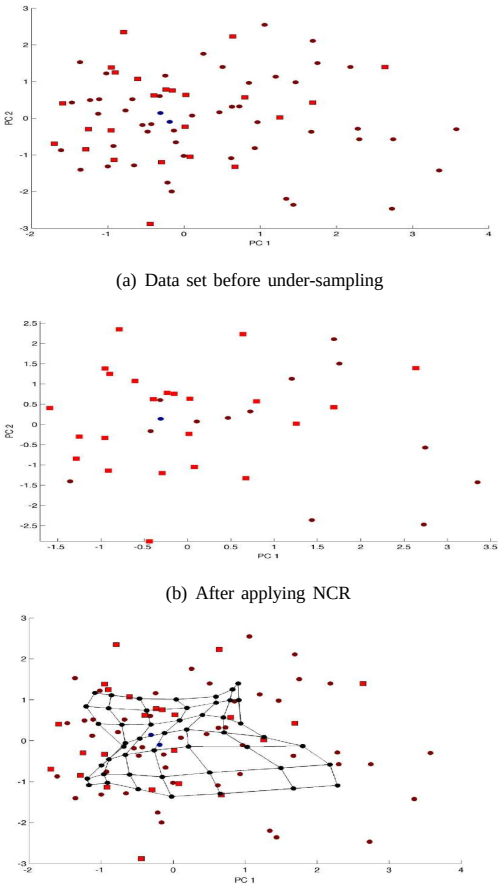
The second term is the geometric mean of accuracies measured separately on each class, $Gmean = \sqrt{TPrate \times TNrate}$, [17]. A theoretical and experimental study is presented in the paper [15].

A. Experiments

Table III presents AUC and IBA performances applying decision trees classifiers. The number in parenthesis indicates the standard deviation computed over 100 experiments corresponding to a cross-validation. From results in this table some preliminary conclusions can be drawn. The association of adaptive under-sampling and topological clustering allows better performance in terms of the AUC or IBA. Analyzing only the AUC performance, we observe a slight decrease with Flag and thyroid data sets. This decrease is due to the low value of $TNrate$. With Flag, $TNrate$ decreases from 93.59% (NCR) to 93.33% (TNCR). On the other side the true positive rate $TPrate$ increases from 20.82% (NCR) to 22.82% (TNCR). This implies the increase of IBA measure that provides an advantage to the positive classes.

To better understand the behavior of TNCR and NCR methods, we compute the removal rate (RR) obtained at the end of the learning process (Table IV). We show clearly that the TNCR method provides high removal rate of negative examples comparing to the rate obtained with NCR. Thus, TNCR achieves similar performance or better performance with less examples than the classical NCR method. On most data set we observe a good performance (AUC and IBA RR) with TNCR, except the "post-operative" where we obtain a lower elimination rate RR.

Indeed, the data set Post-operative has a distribution of



(a) Data set before under-sampling

(b) After applying NCR

(c) After applying TNCR

Fig. 1. NCR and TNCR results with Post-operative data set. The positive class is shown with red square, and the negative class with colored circles (each color corresponds to one class).

66 (positive class, +) and 24 (negative class, -). TNCR achieves an AUC of 47.89% and IBA of 12.08% with RR

rate equal to 46.97% which is less than 75.76% (NCR). The removing action associated to TNCR method, it is under no obligation to delete example, unlike the NCR algorithm which is in the obligation to delete data. Figures 1 (a, b, c) show a PCA projection of original data and those obtained after under-sampling using NCR and TNCR. The minority class is shown with red squares, and the majority class with colored circles (each color corresponds to one class). We observe clearly that NCR deletes examples belonging to a far neighborhood, which are considered outside the local neighborhood considered by the TNCR method.

As in the topological clustering algorithm, we have to decrease the temperature T between two values T_{max} and T_{min}, to control the size of the neighbourhood influencing a given cell of the map. For each T value, we get a cost function $\mathcal{R}^T$, and therefore the expression varies with T. When decreasing T, the model of TNCR will be defined with two phases which are observed in all figures (Fig. 2) :

- Cleaning phase : This step corresponds to high values of T and provides the topological order of TNCR. During this step, the algorithm reduces the size of the negative examples and converge to optimal size of data set.
- Quantization phase: The second step corresponds to small T. Therefore, the adaptation is very local. The parameters are accurately computed from the local density of the data. We observe in this phase that the cleaning phase is stopped and the size of data set is stable along this phase (Fig. 2).

IV. Conclusions and perspectives

We are interested in this work on the problem of preprocessing of unbalanced data. We presented an under-sampling approach which, is based on topological maps. This solution guides the choice of examples to be deleted using a local neighborhood, which take into account the distribution in a topological clustering. Experiments are presented to validate the proposed method. The results are compared with traditional method known as famous under-sampling method. Empirical studies have shown the robustness and advantages of IBA with respect to the well-known AUC performance measure. Future work will primarily compare our approach with other under-sampling methods and study the influence of overlapping classes. Also we will study performance of the TNCR method using other classifier method as SVM. Other performance measure will also be studied [18].

Acknowledgements

This work has been supported by French research agency ANR (Agence National de la Recherche) under e-FRAUD grants ANR-09-SECU-03-03.

References

[1] S. Marcellin, D. A. Zighed, and G. Ritschard, “Evaluating decision trees grown with asymmetric entropies,” in *ISMIS*, ser. Lecture Notes in Computer Science, A. An, S. Matwin, Z. W. Ras, and D. Slezak, Eds., vol. 4994. Springer, 2008, pp. 58–67.

[2] G. M. Weiss, “The effect of small disjuncts and class distribution on decision tree learning,” Ph.D. dissertation, New Brunswick, NJ, USA, 2003, director-Hirsh, Haym.

[3] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84–95, 1980.

[4] G. M. Weiss, “Mining with rarity: a unifying framework,” *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 7–19, 2004.

[5] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, “Strategies for learning in class imbalance problems,” *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[6] V. García, R. Alejo, J. S. Sánchez, J. M. Sotoca, and R. A. Mollineda, “Combined effects of class imbalance and class overlap on instance-based classification,” in *IDEAL*, ser. Lecture Notes in Computer Science, E. Corchado, H. Yin, V. J. Botti, and C. Fyfe, Eds., vol. 4224. Springer, 2006, pp. 371–378.

[7] N. V. Chawla, K. W. Bowyer, and P. W. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.5547

[8] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.

[9] I. Tomek, “An experiment with the edited nearest-neighbor rule,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 448–452, 1976.

[10] J. Laurikkala, “Improving identification of difficult small classes by balancing class distribution,” in *AIME ’01: Proceedings of the 8th Conference on AI in Medicine in Europe*. London, UK: Springer-Verlag, 2001, pp. 63–66.

[11] D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *IEEE Transactions on Systems, Man, and Communications 2*, vol. 3, pp. 408–421, 1972.

[12] J. Stefanowski and S. Wilk, “Selective pre-processing of imbalanced data for improving classification performance,” in *DaWaK ’08: Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 283–292.

[13] T. Kohonen, *Self-organizing Maps*. Springer Berlin, 2001.

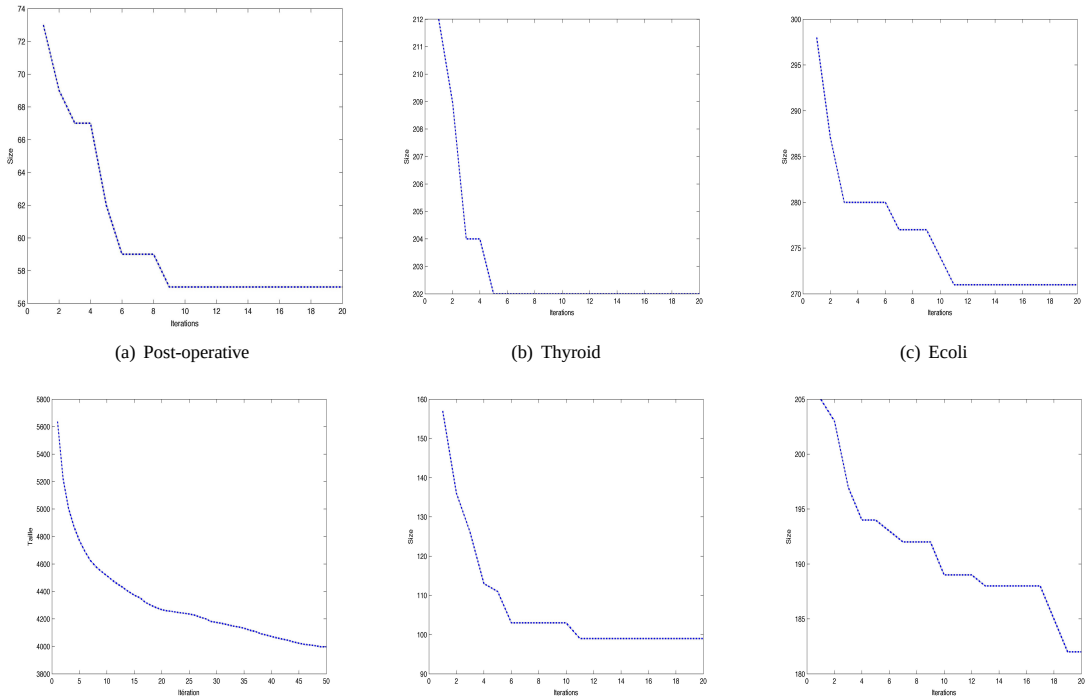
[14] A. Asuncion and D. Newman, “UCI machine learning repository,” http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[15] V. García, R. A. Mollineda, and J. S. Sánchez, “Index of balanced accuracy: A performance measure for skewed class distributions,” in *IbPRIA*, ser. Lecture Notes in Computer Science, H. Araújo, A. M. Mendonça, A. J. Pinho, and M. I. Torres, Eds., vol. 5524. Springer, 2009, pp. 441–448.

[16] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation,” *AI 2006: Advances in Artificial Intelligence*, pp. 1015–1021, 2006.

[17] M. Kubat and S. Matwin, “Addressing the curse of imbalanced training sets: One-sided selection,” in *In Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann, 1997, pp. 179–186.

[18] D. Hand, “Measuring classifier performance: a coherent alternative to the area under the roc curve,” *Machine Learning*, vol. 77, no. 1, pp. 103–123, October 2009. [Online]. Available: http://dx.doi.org/10.1007/s10994-009-5119-5



(a) Post-operative (b) Thyroid (c) Ecoli

(d) Satimage (e) Flag (f) Glass

Fig. 2. TNCR self-organizing process. The development of the under-sambling indicated by data set size with iterations. Each plot indicates the two phases : cleaning phase and quantization phase.

Consensus clustering by graph based approach

Haytham Elghazel[1], Khalid Benabdeslemi[1] and Fatma Hamdi[2]

1- University of Lyon 1, LIESP, EA4125, F-69622 Villeurbanne, Lyon, France;
{elghazel,kbenabde}@bat710.univ-lyon1.fr
2- University of Paris 13, LIPN, UMR CNRS 7030, 93430 Villetaneuse, France;
fatma.hamdi@lipn.univ-paris13.fr

Abstract. In this paper, we propose G-Cons, an extension of a graph minimal coloring paradigm for consensus clustering. Based on the co-association values between data, our approach is a graph partitioning one which yields a combined partition by maximizing an objective function given by the average mutual information between the consensus partition and all initial combined clusterings. It exhibits more important consensus clustering features (quality and computational complexity) and enables to build a combined partition by improving the stability and accuracy of clustering solutions. The proposed approach is evaluated against benchmark databases and promising results are obtained compared to another consensus clustering techniques.

1 Introduction

Consensus clustering [6], also called cluster ensemble, has received considerable attention in the statistics and machine learning communities.Different cluster ensemble approaches are considered in the literature, including graph partitioning, Voting approach, Mutual information algorithms and Co-association based functions. *Graph partitioning based methods* [1] summarize the cluster ensemble in a graph whose vertices correspond to the objects to be clustered and partition it to yield the final clustering. The *Voting Approaches* [2], also called relabeling approaches attempt to solve a correspondence problem between the labels of initial and derived clusters using a majority vote to determine the final consensus partition. *Mutual Information based approaches* [3] consider, as cluster ensemble objective function, the mutual information between the empirical probability distribution of labels in the consensus partition and the labels in the ensemble. *Co-association based functions* compute the co-association values for every pair of objects, as the number of clusters shared by these objects in the initial partitions and feed them into any reasonable similarity based clustering algorithms, such as hierarchical clustering and graph partitioning [1].

In this paper, we present a new efficient method, called GCons to solve the cluster ensemble problem. From the co-association values between objects, GCons approachs the problem by first transforming the set of initial clusterings into a graph representation and then partition it using a minimal coloring mechanism. Unlike the traditional graph partitioning methods, the main advantage to adopt the minimal coloring paradigm is its ability to ensure a high cohesion within the generated clusters that we will show its strong relation to the cluster ensemble objective function.

2 GCons : A New Graph Based Consensus Function

In this section, a *minimal coloring based consensus clustering method* is proposed. Given a data set $\boldsymbol{X} = \{x_1, \ldots, x_n\}$ and an ensemble of r clusterings (partitions) $\boldsymbol{\Pi} = \{\pi_1, \ldots, \pi_r\}$ with the *q-th* clustering π_q having k_q clusters, a *consensus function* Γ is defined as a function $\mathbb{N}^{n \times r} \longrightarrow \mathbb{N}^n$ mapping a set of clusterings to a combined (integrated) clustering λ (*i.e.* $\Gamma : \boldsymbol{\Pi} \longrightarrow \lambda$). Our main goal is to construct a consensus partition without the assistance of the original patterns in $\boldsymbol{X}$, but only from their cluster labels. As showed in [1], the optimal consensus partition should *share* as much *information* as possible with the given original r clusterings. Therefore, the optimal combined clustering λ_{opt} will be defined as the one that has ***maximal*** average mutual information with all individual clusterings π_q. As given in [1], using the definition of normalised mutual information estimate(NMI) between two clusterings (*c.f.* eq.(1)), our objective function can be written as the average of pair-wise NMI between the combined partition and initial clusterings. One can easily compute its value for a candidate partition solution λ and the ensemble of r clusterings $\boldsymbol{\Pi}$ as in equation 2.

$$\phi_{NMI}(\pi_a, \pi_b) = \frac{\sum_{h=1}^{k_a} \sum_{l=1}^{k_b} n_{h,l} \log\left(\frac{n \cdot n_{h,l}}{n_h^a \cdot n_l^b}\right)}{\sqrt{\left(\sum_{h=1}^{k_a} n_h^a \cdot \log \frac{n_h^a}{n}\right)\left(\sum_{l=1}^{k_b} n_h^b \cdot \log \frac{n_h^b}{n}\right)}} \tag{1}$$

where n_h^a is the number of objects in cluster C_h according to the partition π_a, $n_{h,l}$ denote the number of objects that are in cluster C_h according to π_a as well as in group C_l according to π_b.

$$\phi(\boldsymbol{\Pi}, \lambda) = \frac{1}{r} \sum_{q=1}^{r} \phi_{NMI}(\lambda, \pi_q) \tag{2}$$

In the remainder of this section, we present an elegant solution to the consensus problem by developping a consensus function based on graph minimal coloring algorithm. Our function approaches the problem by first transforming the set of clusterings into a graph representation. However this function needs to a definition of dissimilarity level between objects. Given r component clusterings, the overall dissimilarity matrix D for objects is just the complement of the *co-association matrix* [1], with entry $D(i, j)$ denoting the fraction of components in the ensemble in which the two objects i and j are not assigned together. Based on D, the objects set $\{x_1, x_2, \ldots, x_n\}$ can be conceived as a weighted linkage graph $G = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V} = \{v_1, v_2, \ldots, v_n\}$ is the vertex set which corresponds to the objects (v_i for x_i), and $\boldsymbol{E} = \boldsymbol{V} \times \boldsymbol{V}$ is the edge set which corresponds to a pair of vertices (v_i, v_j) weighted by the dissimilaritiy $D(i, j)$.

As said before, the optimal combined clustering should share the most information with the original clusterings. Under the dissimilarity definition, the maximization of the underlying objective function ϕ (*c.f.* eq.(2)) can be related to the minimization of the *total intracluster dissimilarity criterion* of the combined partition. Essentially, if two objects are grouped together in the combined

partition, the fraction of components that not assign them together should be small (denoting that they are considered to be *fully similar*). Therefore, an adopted definition of optimal consensus clustering is a partitioning that *minimizes* dissimilarities within clusters. This condition amount to saying that edges between two vertices within one cluster should be small weighted. The partitioning problem can be formulated as a *graph minimal coloring problem*.

In [4], Hansen and Delattre showed that the partitioning problem into k classes with a minimal *diameter* (The *diameter* of one cluster is the largest dissimilarity between two objects belonging to the same cluster.), an equivalent criterion to the *total intracluster dissimilarity* one, can be reduced to the minimal coloring problem of a *superior threshold graph* in which vertices correspond to objects and edges correspond to dissimilarities between two elements which is higher than a given threshold value θ chosen among the dissimilarity matrix D. In other words, $G_{>\theta}$ is given by $\boldsymbol{V} = \{v_1, v_2, \ldots, v_n\}$ as vertex set and $\boldsymbol{E_{>\theta}} = \{(v_i, v_j) | D(i,j) > \theta\}$ as edge set. The goal is to divide the vertex set $\boldsymbol{V}$ into a combined partition $\lambda = \{C_1, C_2, \ldots, C_k\}$ (when k is not predefined).

Despite the fact that the r clustering components are considered to be obtained from diverse clustering strategies (different clustering approaches or views of the data), they can share common informations. Indeed, a reasonnable number of objects can be clustered together in all r components of the ensemble and having then a pairwise dissimilarity of 0. Therefore, it can be analytically shown that the dissimilarity matrix D is generally a sparse matrix. Under this assumption, we propose a pre-treatment step which concern the construction of the superior threshold graph that will be presented to the minimal coloring algorithm. For that, we need to introduce the following definition:

Definition 1 A composite vertex v' is a subset of objects such that all pairs among these objects appear together in the r initial clusterings.

The superior threshold graph $G_{>\theta}$ will be transformed to $G^{'}_{>\theta} = (\boldsymbol{V'}, \boldsymbol{E}^{'}_{>\theta})$ given by the following instructions:

- Using the previous definition 1, find the overall composite vertex set $\boldsymbol{V_1'} = \{v_1^{'}, v_2^{'}, \ldots, v_{n_1}^{'}\}$ from the original vertex set $\boldsymbol{V}$. The composite vertices in $V_1^{'}$ are pairwise disjointed. In the other hand, the remaining vertices $(\boldsymbol{V} \setminus \bigcup_{i=1}^{n_1} v_i^{'})$ which are not involved in any composite vertex are affected each one to a proper composite vertex in the set $V_2^{'} = \{v_{n_1+1}^{'}, ..., v_m^{'}\}$. Finally, $V_1^{'}$ and $V_2^{'}$ are combined into $\boldsymbol{V'} = \{v_1^{'}, ..., v_m^{'}\}$ where $m < n$.

- The dissimilarity matrix D wil be reduced to a new dissimilarity matrix D'. Since each pair of objects x_i and x_j from the same composite vertex $v_i^{'}$ are always grouped together in all r clusterings, $D(x_i, x_k) = D(x_j, x_k)\ \forall x_k \in \boldsymbol{X}$. Consequently, $D^{'}(i,j)$ between $v_i^{'}$ and another composite vertex $v_j^{'}$ is given by the dissimilarity between any two pair of objects from both vertices. Likewise the construction of $\boldsymbol{E_{>\theta}}$ the edge set $\boldsymbol{E}^{'}_{>\theta}$ is given by $\{(v_i^{'}, v_j^{'}) | D'(i,j) > \theta\}$ where θ is chosen among the reduced dissimilarity matrix D'.

This pre-treatment step is very important to the consensus problem since (1) it allows to decrease the runtime of the partitioning algorithm (we are dealing with $m < n$ vertices) and (2) it offers the possibility to minimize the intracluster dissimilarity (and then to maximize our objective function ϕ) since the fully similar objects are pre-clustered together before performing any partitioning.

In such *superior threshold graph* $G^{'}_{>\theta}$, the *minimal coloring* is NP-*complete* and consists to determine the minimum number of colors (clusters) needed to color the vertices of the graph such that no two adjacent vertices (dissimilar in the sense of threshold θ) have the same color (*proper coloring*). A variety of approximations and search algorithms have been developed to solve the minimal graph coloring problem in a reasonable amount of time. The simplest and well-known graph minimal coloring algorithm is the *Largest First* (LF) one developped by *Welsh and Powell* in [5]. This algorithm, easy to implement and fast, sorts the vertices by decreasing degree. The top vertex is put in color class number one. The remaining vertices are considered in order, and each is placed in the first color class for which it has no adjacencies with the vertices already assigned to the class. If no such class exists, then a new class is created. The main problem of LF algorithm is to find the appropriate vertex to color it when there is a choice between many vertices with the same degree. For an illustration purpose, suppose that we have two adjacent vertices $v_i^{'}$ and $v_j^{'}$ having the same degree and no neighbors in one color c. Therefore, if $v_i^{'}$ is selected for coloring, it can be assigned to color c which will not be possible after for v_j, and vice versa. We note the reliance of the coloring based partitioning result to the selection manner for such vertices. As a solution, GCons constrains this choice to ***maximize*** the *intracluster homogeneity* and then our *objective function* ϕ of the returned (combined) partition λ. We propose the following strategy: when one vertex $v_i^{'}$ with degree d is selected for coloring and the first color c different from those of its neighborhood is found, the vertices not yet colored, having the same degree d and without any neighbor in c, will be simultanously considered for coloring. So the vertex whose dissimilarity with c is minimal will be the first to color with c and the remaining vertices will be considered later. GCons's complexity is $O(n^2)$ which is reduced to $O(m^2)$ $(m < n)$ after pre-treatment step,

3 Experimental Results

In this section, we illustrate our algorithm's performance on several relevant benchmark data sets [7] (*c.f.* Table 1). For our experiments, four clustering approaches are used to generate the partitions for the combination: (1) *k-means*; (2) *Agglomerative Hierarchical Classification* (AHC) in the form of *Ward-based* approach; *Self-Organizing Map* clustered based on (3) k-means, and (4) AHC. The four clusterings are then integrated using our proposed GCons approach and three other graph based consensus functions : CSPA, MCLA and HGPA [1]. We note that GCons is iterative and performs multiple runs, each of them increasing the value of the dissimilarity threshold θ. Once all threshold values passed,

the algorithm provides the best combined partition λ (corresponding to one threshold value θ_o) with the highest objective function $\phi(\mathbf{\Pi}, \lambda)$ (*c.f.* eq.(2)). For an interesting assess of the results gained with the different consensus clustering approaches the following performance indices are used:

- The *objective function* $\phi(\mathbf{\Pi}, \lambda)$ (*c.f.* eq.(2)). It gives an idea about the dependency between the combined partition and the four clusterings in the ensemble.
- A *statistical-matching* scheme given by the *Normalized Mutual Information* $\phi_{NMI}(\lambda, L)$ (*c.f.* eq.(1)). In our case, the used UCI data sets include *class information* (*label*) for each data instance. These labels are available for evaluation purposes but not visible to the clustering algorithms. Indeed, evaluation is based on this scheme in order to assess the *degree of agreement* between the combined partition λ and the correct predefined one $L(labels)$. When comparing two consensus clustering algorithms, the one that produces the greater ϕ_{NMI} should be preferred since the partition correctly identifies the underlying classes in the data set.

Data sets	*instances*	*features*	*#labels*
Wdbc	569	30	2
Rings	1000	3	2
Image Segmentation	2310	19	7
Engytime	4096	2	2

Table 1: Characteristics of used data sets.

Data sets	*CSPA*	*HGPA*	*MCLA*	*GCons*
Wdbc	0.2892	0.0001	0.7471	0.8830
Rings	0.6383	0.0010	0.6260	0.6663
Image Segmentation	0.6656	0.4713	0.6593	0.7612
Engytime	0.7952	0.0001	0.8072	0.8191

Table 2: Comparison of consensus functions in terms of the objective function.

Table 2 provides the clustering results according to the *objective function*. The reported values indicate better consensus clustering for all partitions generated by the proposed GCons approach. The combined partitions given from GCons are thus *highly related* to all individuals clusterings and share the most information with them, compared to the other consensus functions. HGPA performs the worst in these experiments which is also highlighted in [1]. This confirms the pertinence of the *graph minimal coloring technique* to offer a consensus partition with minimal diamater and then reaching a larger objective function.

Table 3 provides the clustering results according to the *normalized mutual information* with original labels. The ranking of the consensus algorithms is

Data sets	*CSPA*	*HGPA*	*MCLA*	*GCons*	*AIA**
Wdbc	0.0973	0.0007	0.3985	0.4514	0.4242
Rings	0.0705	0.0001	0.1296	0.2971	0.1703
Image Segmentation	0.4553	0.3134	0.4801	0.5760	0.4874
Engytime	0.7197	0.0001	0.7228	0.7278	0.6994

Table 3: Comparison of consensus functions in terms of their normalized mutual information with original labels. *AIA: Average Individual Algorithms

the same using this measure, with GCons best, followed by the other consensus clustering approaches: MCLA, CSPA, and HGPA worst. This indicates that the objective function we used $\phi(\mathbf{\Pi}, \lambda)$ is a suitable choice in real applications where the labels are not available. Consequently, it is observed that GCons achieves the highest correspondance with the *correct predefined partition* even when compared to the average quality of all individual algorithms. In fact, the average GCons normalized mutual information based quality $\phi_{NMI}(\lambda, L)$ over all data sets is 26% higher than the average quality of all individual clustering algorithms.

4 Conclusion

In this work we have proposed GCons, a *graph minimal coloring based approach* for consensus clustering. Two problems have been considered: 1) why the *minimal coloring paradigm* is well adapted to the cluster ensemble problem; 2) how to adopt it as best as possible in order to yield a good consensus partition. GCons is evaluated against benchmark data sets and the results of this study demonstrate that the proposed cluster ensemble approach is able to combine individual partitions in a better way than a well known graph partitioning based consensus methods and indicate the effectiveness of *minimal coloring paradigm* to offer an elegant solution to the cluster ensemble problem.

References

[1] A. Strehl and J. Ghosh, Cluster ensembles - a knowledge reuse framework for combining multiple partitions, *Journal of Machine Learning Research*, 3: 583-617, 2002.

[2] S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9): 1090-1099, 2003.

[3] A. Topchy, A. K. Jain and W. Punch, Clustering ensembles: Models of consensus and weak partitions, *IEEE Trans on Patt Anal and Mach Intell*, 27(12): 1866-1881, 2005.

[4] P. Hansen and M. Delattre, Complete-link cluster analysis by graph coloring, *Journal of the American Statistical Association*, 73: 397-403, 1978.

[5] D. J. A. Welsh and M. B. Powell, An upper bound for the chromatic number of a graph and its application to timetabling problems, *Computer Journal*, 10(1): 85-87, 1967.

[6] R. Ghaemi, N. Sulaiman, H. Ibrahim, and N. Mustapha, A survey: Clustering ensembles techniques. In *World Academy Science, Engineering and Technology*, 38, 2009.

[7] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

Selected Random Subspace Novelty Detection Filter

No Author Given

No Institute Given

Abstract. In this paper we propose a solution to deal with the problem of novelty detection. Given a set of training examples believed to come from the same class, the aim is to learn a model that will be able to distinguich examples in the future that do not belong to the same class. The proposed approach called Selected Random Subspace Novelty Detection Filter ($SRS - NDF$) is based on the bootstrap technique, the ensemble idea and model selection principle. The $SRS - NDF$ method is compared to novelty detection methods on publicly available datasets. The results show that for most datasets, this approach significantly improves performance over current techniques used for novelty detection.

1 Introduction

The task of novelty detection consists of identifing a new data that differs from those used in the training phase of a machine learning system. Several important works in the machine learning literature have addressed the issue of novelty detection and broad reviews of the subjet can be found in [6] and [7]. Novelty detection is an important learning problem, the basic idea is to build a decision rule that distinguishes *normal* from *novel* pattern. Since we can never train a machine learning system on all possible data that the system may deal with, it becomes important that it is able to detect *new* data. In order to overcome the limitations of individual learning algorithms and face the necesstiy of high classification performance specially in some critical domains, many researchers have been interested in ensemble methods. The aim of these techniques is to produce and combine multiple classifiers. Bagging [2], Boosting [9], random forest [3] and their variants are the most popular examples of this methodology. Bagging, a name derived from bootstrap aggregation, was the first effective method of ensemble learning and is one of the simplest methods of arching. Generally the ensemble methods [20] work on two steps. The first one is the production of homogeneous or heterogeneous models. Models built from the same learning algorithm are called homogeneous and others that derive from running different learning algorithms on the same data set are called heterogeneous. The second step is the agregation of the models. Several techniques here include voting, weighted voting, selection and stacking. The ensemble selection algorithms was proposed to determine the good sub ensemble of classifiers. In supervised classification, it is known that selective classifier ensembles can always achieve better

results compared to traditional ensemble methods [12]. The ensemble selection also called in the litterature ensemble prunning, ensemble overproduce or choose paradigm, consists in choosing a subset of l classifiers from the initial ensemble of size L ($l \leq L$). The selection of classifiers is based on predefined criteria. Generally the proposed approaches rearrange the initial ensemble and select a subset of ensemble members from the sorted list.
In this paper we present a learning model called Selected Random Subspace Novelty Detection Filter ($SRS-NDF$). It is a new approach to novelty detection, wich involves learning from only one class of traning example. We have a sample from one distribution *normal samples*, and our purpose is to differentiate between these normal examples and those that do not appear to come from the same distribution (*novelty*). The $SRS-NDF$ is an extension of our novelty detection model ($RS-NDF$) proposed in [12]. $SRS-NDF$ is based on the bootstrap technique, the ensemble idea [20] and model selection principle [16]. The methodologies for the production and combination of multiple predective models is a very active research area and it is commonly referred to ensemble method. The advantages of these methods are the improvement of the models estimation and the potential improvement of the scalability of their learning algorithms. The main idea of our approach is to perform classifier selection from an initial pool of filters [8] obtained with the ($RS-NDF$) algorithm. The proposed method works by evaluating the qualities of all obtained filters in terms of pertinence. Next we use the scree test [5] to choose the part of pertinent filters to build our final system.
The rest of the paper is organized as follow: Section 2 introduces the basic concepts of the Selected Random Subspace Novelty Detection Filter. Section 3 describes the databases and the experimental protocol. In section 4 we show validation results and their evaluation. Conclusion is given in section 5.

2 Selected Random Subspace Novelty Detection Filters

2.1 Principle of the Kohonen and Oja's Novelty Filter

In 1976, Kohonen and Oja [8] introduced an orthogonalising filter which extracts the parts of an input vector that are, new, with respect to previously learned patterns. This is the desired functionality of a novelty filter. The novelty filter shows the novelties in an input pattern with respect to previously learned patterns. Furthermore, the novelty filter can distinguish the missing parts from the added parts in the input pattern with respect to the previously learned patterns. The construction of the filter is based on Greville's theorem [13]. This theorem gives a recursive expression to estimate the transfer function of the network as follows:

$$\Phi_k = \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T}{\| \tilde{\mathbf{x}}_k \|^2} \tag{1}$$

where $\mathbf{x}_k = [x_1, x_2, ..., x_d]^T$ is a d-dimensional vector from the reference data matrix; $\tilde{\mathbf{x}} = \Phi_{k-1}\mathbf{x}_k$ represents the orthogonal projection of the vector $\mathbf{x}_k$ in the

subspace of novelty (Φ_{k-1}). This subspace is orthogonal to the space defined by the first $k-1$ reference data and $\Phi_0 = I$.

An interesting alternative approach was given by Kassab and al. [10], [11], that introduces the identity matrix in the learning formula for considering separately all training examples, and consequently all their features. During the learning phase, features which frequently appear in the training examples become more and more habituated as compared to the less frequent ones. This helps to more discriminate the relevant and irrelevant examples.The new learning rule is then defined as:

$$\Phi_k = \mathbf{I} + \Phi_{k-1} - \frac{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T}{\| \tilde{\mathbf{x}}_k \|^2} \tag{2}$$

where $\tilde{\mathbf{x}}_k = (\mathbf{I} + \Phi_{k-1})\mathbf{x}_k$ and Φ_0 is zero, or null matrix. The work described in this paper uses this new learning rule.
For the novelty detection problem, two proportions can be computed:

- *Novelty proportion*: this measure quantifies the novelty of an input data with respect to data that has been previously seen during the training.

$$N_{\mathbf{x}_i} = \frac{\| \tilde{\mathbf{x}}_i \|}{L \times \| \mathbf{x}_i \|} \tag{3}$$

  where L is the number of examples used for the training.
- *The habituation proportion*: this measure calculates the similarity of an example with the previously learned one: $H_{\mathbf{x}_i} = 1 - N_{\mathbf{x}_i}$.
  This proportion could be considered as the classification score of an example x_i. It indicates the probability that x_i belongs to the novel class.

To determine a detection threshold for each filter, the following principle was used:

- Scores (output's filter) attributed to the learning data can be used as a good indicator of the scores of data which can be positive and which are easy to detect because they are strongly similar to the data used for the learning. Consequently, the average of these scores can be admitted as a higher limit for the detection threshold.

- The scores attributed to available data for learning before their use, can be used as a good indicator of the scores of data which are positive but which are less easy to detect. Consequently, the average of these scores can be admitted as a lower limit for the detection threshold.

2.2 SRS-NDF algorithm

In this section, we present an extention for the novelty detection algorithm $RS-NDF$. The $RS-NDF$ [12] approach uses multiple versions of a training set

by using a double bootstrap, i.e. sampling with replacement on examples and sampling without replacement on features. Each of these data sets is used to train a different NDF model. The $RS-NDF$ is then an ensemble of NDF, induced from bootstrap samples of the training data, using random features and examples selection in the model induction process. Prediction is made by aggregating (majority vote) the predictions of the ensemble to create a single output. Our method, called $SRS-NDF$, consists in selecting the ensemble members from a set of individuals filters wich gives better results, in terms of pertinence, than the original ensemble. $SRS-NDF$ belongs to the model selection approaches that reorder the original ensemble members based on pertinence criteria and select a subset of ensemble members from the sorted list using the scree test. $SRS-NDF$ works by evaluating the "index of balanced accuracy" and "diversity" of the filters in the $RS-NDF$ and selecting the promising filters. The final solution is achieved by combining all the selected filters from the original ensemble. To study the pertinence of each filter fl we used the following function:

$$Pertinence_{\alpha}(fl_1) = \alpha \times IBA_{\alpha}(fl_i) + (1-\alpha) \times mean(Div_{\alpha}(fl_i); fl_i); i \in [1, NF] \tag{4}$$

Where $IBA_{\alpha}(fl_i)$ and $Div_{\alpha}(fl_i)$ stands respectively for the index of balanced accuracy [19] and the diversity of the filter fl_i. The index of balanced accuracy is defined by the product of two terms Dominance and Gmean [19]. The first term is a simple measure evaluating the correct predictions of each filter, the second term is the geometric mean of accuracies measured separately on each class. The asset of IBA measure is the ability to distinguish the contribution of each class for overall performance. This means that different combinations of the true positive rate and the true negative rate don't provide the same IBA value and gives the pertinence to the positive class. This measure computes the area of a rectangular region in a two-dimensional space called "Balanced Accuracy Graph". The diversity of two classifiers consist on assigning differents labels to the same examples. Many measures have been proposed to quantify the diversity between two classifiers. In our work, we propose to use the mean Frobenius distance between the transfer matrix of fl_i and the other filters in $RS-NDF$. The coefficient α, $0 \leq \alpha \leq 1$, is a control parameter that balances the accuracy and diversity. The pertinence is then defined as a weighted combination of the diversity and accuracy. Once the pertinence have been calculated for a given α, we then used an established statistical method, scree test, to select the most important filters. This statistical test was initially developed to provide a visual technique to select eigenvalues for principal components analysis.

The basic idea is to generate a curve associated with eigenvalues, allowing random behavior to be identified. The number of components retained is equal to the number of values preceding this "scree". Often the "scree" appears where the slope of the graph changes radically. We therefore needed to identify the point of maximum deceleration in the curve.
Assuming that we have pertinence vector $\mathbf{Per}_k = (Per_{1k}, Per_{2k}..., Per_{jk}, ..., w_{nk})$.
Thus we have to process the steps: **Scree Test Acceleration Factor**

1. Sort the pertinence in descending order $\mathbf{Per}_k$. The we obtain a new order $\mathbf{Per}_k = (Per^1_{..}, Per^2_{..}..., Per^i_{..}, ..., Per^n_{..})$; where Per^i indicates the index order.
2. Compute the first difference $df_i = Per^i_{..} - Per^{i+1}_{..}$;
3. Compute the second difference (acceleration) $acc_i = df_i - df_{i+1}$
4. Find the scree: $\max_i (abs(acc_i) + abs(acc_{i+1}))$
5. Cut and consider all the filters until the scree; (use initial indices of filter before sorting)

The SRS-NDF learning algorithm is shown below:

Algorithme 1: Selected Random Subspace Novelty Detection Filter

Repeate for all $\alpha \in [0, 1]$

1. Construct the $RS - NDF$ with NF filters.
2. Calculate the IBA of filters $= IBA_\alpha(1), ..., IBA_\alpha(NF)$.
3. Calculate the diversity of filters $= Div_\alpha(1), ..., Div_\alpha(NF)$.
4. Calculate the pertinence of filters $= Pertinence_\alpha(1), ..., Pertinence_\alpha(NF)$. $Pertinence_\alpha(fl_1) = \alpha \times IBA_\alpha(fl_1) + (1 - \alpha) \times meanDiv_\alpha(fl_1); fl_i; i \in [1, NF]$
5. Select the subset of models using the ScreeTest $= SelectedFilters_\alpha$
6. Calculate the IBA of the selected filters $= IBA_{SelectedFilters_\alpha}$
7. $\alpha = \alpha + 0, 1$

Until $\alpha = 1$

- Select the subset with the best value of IBA
- Aggregate the predictions of the selected ensemble and save the novelty detection results in D.

3 Experiments

3.1 Databases description

We performed several experiments on many relevant data sets: $Spectf$, $Waveform$, $Wine$ and $Yeast$ from the UCI repository [1] , and Oil [15]. These data sets are summarized in table 1. Since all of the datasets are for binary or multi-class classification problems, they were transformed into novelty detection context. We chose randomly, from each data set, a class as the novelty class and collapsed the rest of the classes into one, and use the modified datasets to evaluate the performance of our approach.

3.2 Performance Measurement and experimental protocol

To evaluate the performance of our approach we used several metrics such as true negative rate (Acc-), true positive rate (Acc+)(recall), precision, F-measure

Table 1. Data Set summary

Dataset	Dimension	Size	Size of Novelty class
Oil	48	937	41
Spectf	44	187	15
Waveform	21	5000	1647
Wine	13	178	59
Yeast	8	1484	244

and G-mean. These metrics have been widely used for comparaison.

For each dataset, the performance of the classifier ensemble obtained by $SRS-NDF$ was compared to the unpruned filter ensemble obtained from $SR-NDF$, the basic model NDF and the traditional novelty detection methods : The one class Support Vector Machines (SVM) [18], The Principal Components Analysis (PCA) [14], The auto-associative Multi Layers Perceptron (MLP) [17].

We also used the Area under the ROC Curve (AUC) [4]. There are several methods to estimate the area under the ROC curve. In the case of binary classification, the balanced AUC_b is defined as:

$$AUC_b = \frac{Acc- + Acc+}{2} \tag{5}$$

4 Results

For each database, the six approaches have been used and their results have been evaluated in terms of the six performances metrics. The table 2 below shows the performance of the different algorithms on all data sets.

Based on the table above, some conclusions can be drawn.

The results of *wine* data set shows that $SRS-NDF$ is the superior approach to novelty detection. Our approach outperforms the results obatained by all others methods on all metrics. For $Waveform$ dataset, $SRS-NDF$ shows a great improvement over all algorithms on $Acc-$, $Acc+$, AUC_b and $G-mean$ metrics. Except on $precision$ and $F-measure$, $RS-NDF$ gives the better results. For $Spectf$ dataset, our algorithm outperforms the other methods on $Acc+$ and $F-measure$ but gives a slightly inferior results on AUC_b and $G-mean$. Genarally our proposed approach $SRS-NDF$ gives better results compared to $RS-NDF$ on $Acc+$ and $F-measure$ metrics. The $Acc+$ measure represents the models capacity to detect the novelty class. We chose this metric in purpose to show the good capacity of $SRS-NDF$ to detect the novelty class. As we can see, $SRS-NDF$ shows excelent results comparing to $RS-NDF$ on all datasets. Considering the $F-measure$, our algorithm gives favorable improvement over $RS-NDF$ on Oil, $Wine$ and $Spectf$ datasets. For $Waveform$ data, $RS-NDF$ outperforms our proposed approach. This metric, that combines precision and recall measures, is commonly used in the information retrieval area as performance measure.

Table 2. Performance comparison on all data sets

Wine	$Acc-$ $(Recall)$	$Acc+$	$Prec$	$F-$ $measure$	AUC_b	$G-$ $mean$
MLP	0,78	0,68	0,83	0,81	0,73	0,73
ACP	0,60	0,69	0,80	0,68	0,65	0,64
SVM-1C	0,68	0,76	0,81	0,73	0,72	0,71
NDF	0,84	0,78	0,88	0,86	0,81	0,81
RS-NDF	0,87	0,85	0,92	0,89	0,86	0,86
SRS-NDF	**0,90**	**0,93**	**0,93**	**0,91**	**0,92**	**0,92**
Waveform						
MLP	0,67	0,35	0,51	0,58	0,51	0,48
ACP	0,68	0,35	0,68	0,68	0,51	0,49
SVM-1C	0,88	0,22	0,70	0,78	0,55	0,44
NDF	0,68	0,57	0,77	0,72	0,63	0,62
RS-NDF	0,85	0,53	**0,79**	**0,82**	0,69	0,67
SRS-NDF	**0,90**	**0,60**	0,70	0,78	**0,75**	**0,70**
Spectf						
MLP	0,60	0,78	0,42	0,49	0,69	0,68
ACP	0,51	0,82	0,43	0,47	0,67	0,65
1-SVM	0,73	0,74	0,43	0,54	0,74	0,74
NDF	**0,76**	0,75	0,45	0,56	**0,76**	**0,76**
RS-NDF	0,69	0,79	**0,47**	0,56	0,74	0,74
SRS-NDF	0,64	**0,84**	0,44	**0,58**	0,74	0,74
Yeast						
MLP	0,77	0,23	0,84	0,80	0,50	0,39
ACP	0,68	0,66	0,91	0,78	0,67	0,67
SVM-1C	0,90	0,13	0,84	0,87	0,51	0,34
NDF	0,88	0,19	0,85	0,86	0,54	0,41
RS-NDF	**0,94**	0,29	0,87	**0,90**	0,62	0,52
SRS-NDF	0,93	**0,53**	**0,93**	**0,90**	**0,72**	**0,70**
Oil						
MLP	0,96	0,15	0,96	0,96	0,55	0,38
ACP	0,94	0,24	0,96	0,95	0,59	0,48
SVM-1C	0,90	0,35	**0,97**	0,93	0,62	0,56
NDF	0,91	0,37	**0,97**	0,94	0,64	0,58
RS-NDF	0,94	0,22	0,96	0,95	0,58	0,46
SRS-NDF	**0,98**	**0,46**	0,95	**0,97**	**0,72**	**0,69**

The $G-mean$ results on all datasets confirmed the good performances of our approach $SRS-NDF$. As we can see, our method gives better results comparing to $RS-NDF$. The Gmean of accuracies, measured separatly on each class, is associated to a point in theROC curve and the idea is to maximize the accuracies of both classes while keeping them blanced.

5 Conclusion

This paper introduced a filter ensemble selection method to improve the Random Subspace Novelty Detection Filter ($RS-NDF$) by adaptively trading off diversity and accuracy according to the data. The proposed approach $SRS-NDF$ is based on the orthogonal projection operators, the bootstrap method and the ensemble selection paradigm. Several metrics are computed on publicly available datasets and significant improvements were obtained by $SRS-NDF$ comparing to existing methods.

References

1. Asuncion A. and Newman D.J., *UCI Machine Learning Repository.* Irvine, CA, University of California, 2007.
2. Breiman L., *Bagging Predictors.* Machine Learning, 24:(2),123-140, 1996.
3. Breiman, L., *Random forest.* Machine Learning, 2001.
4. Bradeley,P.W., *The use of area under the ROC curve in the evaluation of machine learning algorithms.* Pattern Recognition,30, 1145-1159 (1997).
5. Catell, R *The scree test for the number of factor.* Multivariate Behaviorial Research, 245-276 (1966).
6. M. Markou and S. Singh, *Novelty detection: a review - part 1: statistical approaches.* Signal Processing, 83:2481-2497, 2003.
7. M. Markou and S. Singh, *Novelty detection: a review - part 2: neural network based approaches.* Signal Processing, 83:2499-2521, 2003.
8. Kohonen, T. and Oja, E., *Fast Adaptive Formation of Orthogonalizing Filters and Associative Memory in Recurrent Networks of Neuron-Like Elements.* Biological Cybernetics, 21:85-95, 1976.
9. Y. Freund and Robert E. Saphire., *Experiments with a new boosting algorithm* The 13th international Conference on Machine Learning, pages 276-280, 1996.
10. Kassab, R., Lamirel, J.-C. and Nauer, E., *Novelty Detection for Modeling Users Profile.* The 18th International FLAIRS Conference. 830-831, 2005.
11. Kassab, R., et Alexandre, F., *Incremental Data-driven Learning of a Novelty Detection Model for One-Class Classification Problem with Application to High-Dimensional Noisy Data.* Machine Learning. 74(2) :191-234, 2009.
12. F. Hamdi and Y. Bennani, *Learning Random Subspace Detection Filter.* International Joint Conference in Neural Networks , IJCNN (2011).
13. Greville, T. N. E. *Some applications of the pseudoinverse of a matrix.* SIAM Rev (1960).
14. I.T. Jolliffe. *Principal Component Analysis.* Springer Series in Statistics. Springer-Verlag, Berlin, Germany,1986.
15. Miroslav Kubat, Robert C. Holte, and Stan Matwin, *Machine Learning for the Detection of Oil Spills in Satellite Radar Images.* Machine Learning, volume 30, pp. 195-215, (1998).
16. R. Caruna, A. Niculescu Mizil, G. Grew, and A. Ksikes, *Ensemble selection from librairiesof models.* The 21st International Conference on Machin Learning, (2004).
17. Rumelhart, D. E., Hinton, G. E., et Williams, R. J., *Learning internal representation by error propagation.* Parallel Distributed Processing : Explorations in the microstructures of cognition, MIT Press. 318-362.(1986).
18. Scholkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., et Williamson, R. C., *Estimating the support of a high-dimensional distribution.* Neural Computation (1999).
19. V. García, R. A. Mollineda, and J. S. Sánchez, *Index of balanced accuracy: A performance measure for skewed class distributions.* Lecture Notes in Computer Science, Springer, 2009, pp. 441–448, (2009).
20. Y. Zhang, S. Burer and W. N. Street, *Ensemble prunning via semi-denite programming.* Journal of Machin Learning Reasearch, 7:1315-1338 (2006).

Bibliographie

[1] S. Marcellin, D. A. Zighed, and G. Ritschard, "Evaluating decision trees grown with asymmetric entropies," in *ISMIS*, ser. Lecture Notes in Computer Science, A. An, S. Matwin, Z. W. Ras, and D. Slezak, Eds., vol. 4994. Springer, 2008, pp. 58–67.

[2] G. M. Weiss, "The effect of small disjuncts and class distribution on decision tree learning," Ph.D. dissertation, New Brunswick, NJ, USA, 2003, director-Hirsh, Haym.

[3] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84–95, 1980. [Online]. Available : http ://ieeexplore.ieee.org/xpls/abs$_all.jsp?arnumber$ = 1094577

[4] G. M. Weiss, "Mining with rarity : a unifying framework," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 7–19, 2004.

[5] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[6] V. García, R. Alejo, J. S. Sánchez, J. M. Sotoca, and R. A. Mollineda, "Combined effects of class imbalance and class overlap on instance-based classification," in *IDEAL*, ser. Lecture Notes in Computer Science, E. Corchado, H. Yin, V. J. Botti, and C. Fyfe, Eds., vol. 4224. Springer, 2006, pp. 371–378.

[7] N. V. Chawla, K. W. Bowyer, and P. W. Kegelmeyer, "Smote : Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. [Online]. Available : http ://citeseerx.ist.psu.edu/viewdoc/summary ?doi=10.1.1.18.5547

[8] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.

[9] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 448–452, 1976.

[10] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *AIME '01 : Proceedings of the 8th Conference on AI in Medicine in Europe.* London, UK : Springer-Verlag, 2001, pp. 63–66.

[11] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Communications 2*, vol. 3, pp. 408–421, 1972.

[12] J. Stefanowski and S. Wilk, "Selective pre-processing of imbalanced data for improving classification performance," in *DaWaK '08 : Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery.* Berlin, Heidelberg : Springer-Verlag, 2008, pp. 283–292.

[13] T. Kohonen, *Self-organizing Maps.* Springer Berlin, 2001.

[14] A. Asuncion and D. Newman, "UCI machine learning repository," http ://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[15] V. García, R. A. Mollineda, and J. S. Sánchez, "Index of balanced accuracy : A performance measure for skewed class distributions," in *IbPRIA*, ser. Lecture Notes in Computer Science, H. Araújo, A. M. Mendonça, A. J. Pinho, and M. I. Torres, Eds., vol. 5524. Springer, 2009, pp. 441–448.

[16] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc : A family of discriminant measures for performance evaluation," *AI 2006 : Advances in Artificial Intelligence*, pp. 1015–1021, 2006. [Online]. Available : http ://dx.doi.org/10.1007/11941439$_1$14

[17] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets : One-sided selection," in *In Proceedings of the Fourteenth International Conference on Machine Learning.* Morgan Kaufmann, 1997, pp. 179–186.

[18] D. Hand, "Measuring classifier performance : a coherent alternative to the area under the roc curve," *Machine Learning*, vol. 77, no. 1, pp. 103–123, October 2009. [Online]. Available : http ://dx.doi.org/10.1007/s10994-009-5119-5

[19] M. Markou and S. Singh, *Novelty detection : a review - part 1 : statistical approaches.* Signal Processing, 83 :2481-2497, 2003.

[20] M. Markou and S. Singh, *Novelty detection : a review - part 2 : neural network based approaches.* Signal Processing, 83 :2499-2521, 2003.

[21] S. Marsland, *Novelty detection in learning systems.* Neural Computing, 3 :157-195, 2003.

[22] Kohonen, T. and Oja, E., *Fast Adaptive Formation of Orthogonalizing Filters and Associative Memory in Recurrent Networks of Neuron-Like Elements.* Biological Cybernetics, 21 :85-95, 1976.

[23] Breiman L., *Bagging Predictors.* Machine Learning, 24 :(2), 123 :140, 1996.

[24] Ho, Tin Kam., *he Random Subspace Method for Constructing Decision Forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 :(8), 832–844, 1998.

[25] Kohonen, T., *Self-Organization and Associative Memory.* 3rd ed. Springer, Berlin, 1993.

[26] Kassab, R., Lamirel, J.-C. and Nauer, E., *Novelty Detection for Modeling Users Profile.* The 18th International FLAIRS Conference. 830831, 2005.

[27] Kassab, R., et Alexandre, F., *Incremental Data-driven Learning of a Novelty Detection Model for One-Class Classification Problem with Application to High-Dimensional Noisy Data.* Machine Learning. 74(2) :191234, 2009.

[28] Breiman, L., *Random forest.* Machine Learning, 45, 532, 2001.

[29] Asuncion A. and Newman D.J., *UCI Machine Learning Repository.* Irvine, CA, University of California, 2007.

[30] David M.J. Tax, Alexander Ypma and Robert P.W. Duin., *Support Vector data description applied to machine learning vibration analyses* Annual Conference of the Advanced School for computing and Imaging, pages 398–405, 1999.

[31] Bradeley,P.W., *The use of area under the ROC curve in the evaluation of machine learning algorithms.* Pattern Recognition,30, 1145-1159 (1997).

[32] Sigillito, V. G., Wing, S. P., Hutton, L. V., *Classification of radar returns from the ionosphere using neural networks.* Johns Hopkins APL Technical Digest, 10, 262-266 (1989).

[33] B. German,Vina Spiehler, *Diagnostic Products Corporation.* (213) 776-0180 (ext 3014) September, 1987.

[34] Miroslav Kubat, Robert C. Holte, and Stan Matwin, *Machine Learning for the Detection of Oil Spills in Satellite Radar Images.* Machine Learning, volume 30, pp. 195-215, (1998).

[35] Kurgan, L.A., Cios, K.J., Tadeusiewicz, R., Ogiela, M. and Goodenday, L.S., *Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis.* Artificial Intelligence in Medicine, vol. 23 :2, pp 149-169, Oct 2001.

[36] Breiman,L., Friedman,J.H., Olshen,R.A., and Stone,C.J. *Classification and Regression Trees. Wadsworth International.* Wadsworth International Group : Belmont, California. (see pages 43-49)(1984).

[37] Forina, M., *An Extendible Package for Data Exploration, Classification and Correlation.* Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno,16147 Genoa, Italy.

[38] Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, Bennett HA, Coffey E, Dai H, He YD, Kidd MJ, King AM, Meyer MR, Slade D, Lum PY, Stepaniants SB, Shoemaker DD, Gachotte D, Chakraburtty K, Simon J, Bard M, Friend SH., *Functional discovery via a compendium of expression profiles.* Cell. Jul 7 ;102(1) :109-26. (2000).

[39] Rumelhart, D. E., Hinton, G. E., et Williams, R. J., *Learning internal representation by error propagation.* Parallel Distributed Processing : Explorations in the microstructures of cognition, MIT Press. 318-362.(1986).

[40] I.T. Jolliffe. *Principal Component Analysis.* Springer Series in Statistics. Springer-Verlag, Berlin, Germany,1986.

[41] Scholkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., et Williamson, R. C., *Estimating the support of a high-dimensional distribution.* Neural Computation.(1999).

[42] Greville, T. N. E. *Some applications of the pseudoinverse of a matrix.* SIAM Rev (1960).

[43] Tuffry S *Data Mining et statistique dcisionnelle.* Editions Technip (2007).

[44] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. *Smote : Synthetic minority over-sampling technique* Journal of Artificial Intelligence and Research (2002).

[45] Nickerson A., N. Japkowiez et E. Milios *Using unsupervised learning to guide resampling in imbalanced data set* In Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics (2001).

[46] Ricardo Barandela, José Salvador Sánchez, Vicente Garcia, and E. Rangel *Strategies for learning in class imbalance problems* Pattern Recognition (2003).

[47] Bhavani Raskutti et Adam Kowalczyk *Extreme re-balancing for svms : a case study* SIGKDD Explor. Newsl, (2004).

[48] Gustavo Batista, Ronaldo Prati, Maria Carolina Monard *A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data* (2004).

[49] G. Wu and E. Y. Chang. *Class boundary alignment for ibalanced data set learning.* ICML Workshop on learning from imbalanced data set (2003).

[50] K. Verpoulos, C. Campbell, and N. Cristiani. *Controlling the sensitivity of support vctor machines.* In sixteen International Joint Conference on Artificial Intelligence, (1999).

[51] Miroslav Kubat, Robert C. Holte, and Stan Matwin. *Learning when negative examples abound.* In Maarten van Someren and Gerhard Widmer, Editor, ECML, volume 1224 of Lecture Notes in Coputer Science, pages 146-153. Springer, (1997).

[52] P. Domingos. *Metacost : A general method for making classifiers cost-sensitive.* Proceedings of the Fifth International Conference on KnowledgeDiscovery and Data Mining (KDD-99), page 155-164, (1999).

[53] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. *Adacost : misclassification sost senstive boosting.* Sixteenth International Conference on Machin Learning, pages 99-105 (1999).

[54] M. V. Joshi, V. Kumar, and R. C. Agarwal. *Evaluating boosting algorithms to classifiy rare cases : comparison and improvements.* First IEEE Internationa Conference on Data Mining, pafes 257-264, November (2001).

[55] N. V. Chwala, A. Lazarevic, L. O. Hall, and K. W. Bowyer. *Smoteboost : Improving prediction of the minority class in boosting.* Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases, pages 107-119, Dubrovnik, Croitia, (2003).

[56] Breiman, L., *Random forest.* Machine Learning, 45, 5, 2001.

[57] Chao Chen, Andy Liaw and Breiman, L., *Using Random Forest to Learn Imbalanced Data.* Machine Learning, 45, 5, 2001.

[58] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz.
Editorial : special issue on learning from imbalanced data sets. SIGKDD Explorations, 6(1) :1, 2004

[59] Sharon Summers et Linda Woolery *Postoperative Patient Data.* School of Nursing, University of Kansas Medical Center, Kansas City, KS 66160 and School of Nursing, University of Missouri, Columbia, MO 65211,1991.

[60] Kenta Nakai *Protein Localization Sites.* Institue of Molecular and Cellular Biology Osaka, University 1-3 Yamada-oka, Suita 565 Japan, 1996.

[61] Ashwin Srinivasan *Satimage database.* Department of Statistics and Modelling Science, University of Strathclyde Glasgow Scotland UK.

[62] B. German *Glass Identification Database.* Central Research Establishment Home Office Forensic Science Service Aldermaston, Reading, Berkshire RG7 4PN, 1987.

[63] Collins Publishers *Flag database.* Collected primarily from the "Collins Gem Guide to Flags" : Collins Publishers (1986).

[64] Ross Quinlan *Thyroid Disease Database.* the University of California at Irvine, Machine Learning Workshop, 1987.

[65] M. Markou and S. Singh *Novelty detection : a review - part 1 : statistical approaches.* Signal Processing, 2481–2497, 2003.

[66] M. Markou and S. Singh *Novelty detection : a review - part 2 : neural network based approaches.* Signal Processing, 2499–2521, 2003.

[67] S. Marsland *Novelty detection in learning systems.* Neural Computing, 157–195, 2003.

[68] S. Hashemi, V. Lemaire G. Dror et D. Vogel *Analysis of the kdd cup 2009 : Fast scoring on a large orange customer database. JMLR* : Work shop and conference Proceedings, 7 : 1–22, 2009.

[69] C. Salperwyck et V. Lemaire *Classification incrémentale supervisée : un panel introductif.* RTNI, 2011.

[70] R. Kassab. *Analyse des propriétés stationnaires et des propriétés émergentes dans les flux d'informations changeant au cours du temps.* PhD thesis, Universit Nancy 1, 2009.

[71] G. Gama. *Knowledge Discovery from data stream.* Chapman et HALL/CRC Press, 2010.

[72] E. Page. *Continuous inspection schemes.* Biometrika, 41(1-2 : 100, 1954.

[73] P. Hall *Permutation test for equality of distributions in high-dimensional setting.* Biometrika 89(2), 359–374, 2002.

[74] Anderson, N. H., P. Hall et D. M. Titterington *Tow sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates.* Journal of Multivariate Analysis 50(1). 41–54, 1994.

[75] Bondu. A., B. Grossin, et M. Picard *Density estimation on Data Streams : an application to change detection.* EGC (Extraction et Gestion de Connaissance, 2010.

[76] Basseville. M et L. V. Nikiforov *Detection of Abrupt Changes.* Theory and Application. Pertinence-Hall, 1993.

[77] Desobry, F. et M. Davy *Support Vector-Based Online Detection of Abrupt Changes.* In Proc. IEEE ICASSP, Hong Kong, pp.872–875, 2003.

[78] B. Su, Y. Shen et W. Xn *Modeling concept drift from the perspective of classifiers.* In Proc. of the conference on cybermetrics and intelligent System, IEEE, pp. 1055–1060, 2008.

[79] A. da Silva, Y. Lechevallier, F. Rossi et F. de Carvalho. *construction and analysis of evolving data summaries : An application on web usage data* In Proc. of the 7th Int conference on intelligent Systems Design and Applicatio, IEEE, pp. 377–380, 2007.

[80] M. Kubat, J. Ganna and P. Utgoff. *Special issue on incremental learning systems capable of dealing with concept drift* Intelligent data analysis, 8(3), 2004.

[81] G. Hulten, L. Spencer et P. Domingos. *Mining time-changing data streams.* In KDD '01 : Proc of 7th ACM SIGKDD int. conf. on Knowledge discovery and data mining, pages 97–106. ACM 2001.

[82] R. Klmkenberg et T. Joachims. *Detecting concept drift with support vector machine.* In ICML '00 : Proc of the 7th int. conf. on machin learning, pages 487–494. Morgan Kaufmann Publihers Inc., 2000.

[83] G. Widner et M. Kubat. *Effective learning in dynamic environments by explicit context tracking.* In European Conference on Machin Learning, pages 277–213. Springer-Verlag, 1993.

[84] A. Bifet et R. Gavalda. *Learning from time-custering data with adaptative windowing.* In Proc of SIAM Int. conf. on knowledge discovery and data mining (SDM '07). SIAM, 2007.

[85] K.Nishida et K. Yamauchi. *Detecting concept drift using statistical testing.* In Proc of Discovery Science. 10th Int. conf. DS 2007, volume 4755 of LNCS. pages 264–269. Springer, 2007.

[86] A. Dries et U. Ruckert. *Adaptative concept drift detection.* In SDM, pages 233–214. SIAM, 2009.

[87] S. Bash et M. Maloof. *Paired learners for concept drift.* In Proc, of the 8th IEEE Int Conf. On Data Mining. Pages 23–32. IEEE Press, 2008.

[88] N. Street et Y. Kim. *Astreaming ensemble algorithm for large-scale classification.* In KDD '01 : Proc. of the 7th ACM SIGKDD international conference on knowledge discovery and data mining, pages 377–382. ACM 2001.

[89] W. Wang, W. Fan, P. Yu et J.Ham. . *Mining concept drifting data streams using ensemble classifiers.* In KDD '03 : Proc. of the 9th ACM SIGKDD international conference on knowledge discovery and data mining, pages 326–335. ACM 2003.

[90] A. Bifet, G. Holmes. B. Pfahringer, R. Kirkby et R. Cavalda. *New ensemble method for evolving data streams.* In KDD '09 : Proc. of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pages 139–148. ACM 2009.

[91] R. Klinkenberg *Learning drifting concept : Example selection vs example weighting.* Intell. Data Anal, pp 281–300, 2004.

[92] J. Wu, D. Ding, X. Hun et B.Zghang. *Tracking concept drifting with online-optimized incremental learning framework.* In MIR '05 : Proc of the 7th ACM SIGKDD international workshop on multimedia information retrieval pages 33–40. ACM, 2005.

[93] S. Raudys et A. Mitasiunas. *Multi-agent system approach to react to sudden environmental changes.* In Proc of Machine Learning and Data Mining in Pattern Recognition, 5th Int. Conf., MLDM 2007. Volume 4571 of LNCS. Pages 810–823. Springer, 2007.

[94] Y. Law et C. Zaniolo. *An adaptative nearest neighbor classification algorithme for data stream.* In Proc of PKDD, volume 3721 of LNCS. Pages 108–120. Springer, 2005.

[95] J. Schlimer et R. Granger. *Incremental learning from noisy data.* Machine Learning, 1(3) : 317–354, 1986.

[96] L. Koyehev. *Gradual forgetting for adaptation to concept drift* In Proc. of ECAI 2000 Workshop current Issues in Spatio-Temporal Reasonning, pages 101–106, 2006.

[97] Fang chu et Carlo Zaniolo. *Fast and light boosting for adaptive mining of data streams* In PAKDD, pages 282–292, Springer Verglas, 2004.