Apprentissage incrémental en données : Application à la reconnaissance d'émotions personnalisée

Jordan Gonzalez

HESAM UNIVERSITÉ

Arts et Métiers Sciences et Technologies

ÉCOLE DOCTORALE SCIENCES ET MÉTIERS DE L'INGÉNIEUR
Laboratoire Learning Data Robotics (LDR) - ESIEA

THÈSE

présentée par : **Jordan GONZALEZ**
soutenue le : **25 octobre 2022**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**

Section CNU : **27 - Informatique**

Apprentissage incrémental en données : Application à la reconnaissance d'émotions personnalisée

Jury

M. Renaud SÉGUIER	Professeur, CentraleSupélec	Président du jury
M. Éric ANQUETIL	Professeur des Universités, INSA Rennes	Rapporteur
M. Laurent HEUTTE	Professeur des Universités, Université de Rouen Normandie,	Rapporteur
M. Simon RICHIR	Professeur des Universités, ENSAM	Examinateur
M. Lionel PREVOST	Professeur des Universités, ESIEA	Directeur de thèse
Mme Aurelia DESHAYES	Maître de conférences, Université Paris-Est Créteil	Co-encadrante

Remerciements

Tout d'abord, je souhaite exprimer ma gratitude au jury de cette thèse, pour le temps précieux et l'intérêt qu'ils ont porté à mes travaux de recherche. Ainsi, je remercie Éric Anquetil, Professeur à l'INSA de Rennes et Laurent Heutte, Professeur à l'INSA de Rouen d'avoir accepté d'être rapporteurs. Je remercie également Simon Richir, Professeur à l'ENSAM, et Renaud Séguier, président du jury, Professeur à CentraleSupelec, d'avoir accepté de juger mon travail en tant qu'examinateurs.

Je tiens à remercier les membres du consortium TEEC pour leur bienveillance, l'ESIEA qui m'a permis de ne jamais manquer de rien, ainsi que mon directeur de thèse Lionel Prevost sans qui cette thèse n'aurait pas vu le jour. Merci à Lionel pour l'opportunité qu'il m'a offerte, j'admire sa passion pour la science et son enthousiasme à explorer de nouvelles idées scientifiques. Ses conseils et les nombreux échanges que l'on a eu m'ont permis de grandir ces dernières années.

Merci à ma co-encadrante Aurelia et Alexandre pour leur humeur et partage, deux excellentes rencontres qui ont fait de ces années au laboratoire des moments très agréables. Je leur suis grandement reconnaissant pour toutes les relectures qu'ils ont accepté de faire pour moi ces derniers mois. Je remercie également mon co-bureau Thibault avec qui j'ai eu des échanges enrichissants et lui souhaite beaucoup de réussite dans ses travaux.

Je remercie évidemment mes proches, plus particulièrement, Muriel, ma mère, qui attend avec impatience que je retourne lui rendre visite plus régulièrement, et bien entendu, Maxence, mon frère, Emma, ma soeur et Raphaël, mon père, qui m'ont toujours encouragé.

Enfin, mes pensées se dirigent vers Diana, ma conjointe, qui m'a apporté un soutien complet ces dernières années et qui m'a toujours accompagné quand j'en avais le plus besoin. Pour avoir participé avec courage et enthousiasme à toutes ces nuits blanches et nombreuses réflexions scientifiques, et pour sa patience incommensurable, je lui en suis profondément reconnaissant.

Résumé

Le domaine de l'informatique affective, en plein essor depuis quelques décennies, a pour objectif de créer de nouveaux systèmes interactifs capables de percevoir l'état émotionnel de leurs interlocuteurs humains et de s'y adapter automatiquement. Il est de plus en plus fréquent que les datasets ne soient plus disponibles de manière complète et que les données arrivent au fur et à mesure, sans être forcément labélisées. De plus, bien que bon nombre de solutions aient été présentées pour caractériser de mieux en mieux les données, les performances des modèles pour la reconnaissance d'émotions demeurent souvent dépendantes de la variabilité individuelle chez les individus. En effet, certaines personnes sont susceptibles d'avoir des morphologies plus ou moins différentes selon l'âge, le genre, etc. D'autant plus que d'un point de vue comportemental, deux individus sont susceptibles d'exprimer un même état émotionnel différemment. Ces biais morphologiques et comportementaux sont réunis sous le terme de biais d'identité qui représente un des défis majeurs pour ce domaine.

Les objectifs de cette thèse sont d'adapter automatiquement des modèles de *machine learning* aux traits morphologiques et comportementaux de l'individu afin de réduire le biais d'identité, et ainsi améliorer la reconnaissance de leurs états émotionnels. Pour cela, nous proposons d'appliquer l'apprentissage incrémental en données de modèles inspirés des forêts aléatoires afin de créer une méthode qui permette au modèle de s'adapter à différentes bases de données. L'une des principales caractéristiques des techniques incrémentales est la possibilité de mettre à jour les modèles en utilisant uniquement des données récentes. C'est souvent la seule solution pratique lorsqu'il s'agit d'apprendre des données "à la volée", car il serait impossible de garder en mémoire et de réapprendre à partir de zéro chaque fois que de nouvelles informations sont disponibles.

Dans une première partie, nous présentons une analyse réalisée au début de la thèse sur les données du projet TEEC. Les données sont collectées à partir des interactions par visioconférence impliquant deux groupes d'apprenants effectuant des tâches d'apprentissage collaboratif à distance. Du fait qu'ils

vivent dans deux pays différents, leurs représentations mentales sur divers thèmes sont différentes et produisent ce que nous appelons un effet de contexte ou conflit socio-cognitif. Nous étudions ces interactions et essayons de trouver une corrélation entre les états affectifs non verbaux et les échanges verbaux, reflétant le degré de (mauvaise) compréhension.

Dans une deuxième partie, nous proposons de spécialiser, via l'incrémentation, un modèle générique à un ensemble fini d'individus à l'aide de données labélisées (de façon supervisée) pour gérer la problématique du biais d'identité. Le modèle Nearest Class Mean Forest (NCMF) peut faire de l'apprentissage incrémental, cependant, les stratégies existantes ne permettent pas de gérer des données complexes dont les distributions conditionnellement aux classes comportent plusieurs modes. À cet effet, nous proposons une stratégie incrémentale s'appuyant sur un critère statistique de séparabilité des modes de la distribution locale, qui va permettre de mettre à jour le centroïde du nœud ou d'en créer un nouveau, si cela s'avère nécessaire.

Nous explorons également la possibilité de pouvoir intégrer un pipeline permettant de réaliser l'incrémentation de manière semi-supervisée afin de répondre à la problématique de la labélisation. Nous proposons à cet égard une méthode hybride qui combine le co-training et l'apprentissage incrémental, permettant à deux modèles de collaborer et de partager leurs connaissances. Contrairement à la méthode classique de cotraining qui effectue un ré-entraînement à partir de zéro à chaque itération de l'algorithme, notre approche effectue une incrémentation du modèle en continu sur de nouvelles observations.

Mots-clés : informatique affective, apprentissage incrémental, apprentissage semi-supervisé

Abstract

The field of affective computing, which has been growing rapidly over the last few decades, aims to create new interactive systems that are able to perceive the emotional state of their human interlocutors and adapt to them automatically. It is quite common that datasets are no longer fully available, and that data arrives gradually, without necessarily being labeled. Moreover, although many solutions have been presented to better describe the data, the performance of the models for emotion recognition is often dependent on individual variability in the people. Indeed, some individuals are likely to have different morphologies according to age, gender, etc. Moreover, from a behavioral point of view, two individuals are likely to express the same emotional state differently. These morphological and behavioral biases are grouped under the term identity bias which represents one of the major challenges for this field.

The objectives of this thesis are to automatically adapt machine learning models to the morphological and behavioral traits of the individual in order to reduce the identity bias and thus improve the recognition of their emotional states. For this purpose, we propose to apply incremental learning in random forest-based models in order to create a method that allows the model to adapt to different databases. A key feature of incremental techniques is the ability to update models using only recent data. This is often the only practical solution when it comes to learning streaming data, as it would be impossible to keep track of and relearn from scratch every time new information becomes available.

In a first part, we present an analysis performed at the beginning of the thesis on the TEEC project data. The data is collected from video conference interactions involving two groups of learners performing collaborative distance learning tasks. Because they live in two different countries, their mental representations on various topics are different and produce what we call a context effect or social cognitive conflict. We study these interactions and try to find a correlation between non-verbal affective states and verbal exchanges, reflecting the degree of (mis)understanding.

In a second part, we propose to specialize, through incrementation, a generic model to a finite set of individuals, using labeled data (in a supervised way) to deal with the identity bias problem. The Nearest Class Mean Forest (NCMF) model can do incremental learning ; however, existing strategies do not handle complex data whose class-conditional distributions have multiple modes. To this end, we propose an incremental strategy based on a statistical criterion of separability of the modes of the local distribution, which will update the centroid of the node or create a new one, if necessary.

We also explore the possibility of integrating a pipeline allowing to perform the incrementation in a semi-supervised way to address the problem of labeling. In this respect, we propose a hybrid method that combines co-training and incremental learning, allowing two models to collaborate and share their knowledge. Contrary to the classical co-training method which performs a re-training from scratch at each iteration of the algorithm, our approach performs a continuous incrementation of the model on new observations.

Keywords : affective computing, incremental learning, semi-supervised learning.

Table des matières

Remerciements **3**

Résumé **5**

Abstract **9**

Liste des tableaux **20**

Liste des figures **24**

1 Introduction **25**

1.1 Genèse du projet 26

1.2 Intérêt de l'informatique affective pour TEEC 27

1.2.1 Défis liés à la modalité visuelle 28

1.2.2 Les contraintes liées à la collecte de données dans le projet TEEC 28

1.3 Biais d'identité et personnalisation 29

1.3.1 Définition 29

1.3.2 Généralisation *vs* personnalisation 31

1.3.2.1 Taxonomie 31

1.3.2.2 Solutions de personnalisation 32

1.3.3 Apprentissage incrémental : principes et défis 34

1.4 Organisation du mémoire et contributions 34

2 Concepts **37**

2.1 Apprentissage supervisé 38

2.1.1 Dataset 38

2.1.1.1 Définition 38

2.1.1.2 Mesure de qualité des données 39

2.1.2 Classifieur 40

2.1.2.1 Apprentissage et généralisation 40

2.1.2.2 Dilemme Biais-Variance 41

2.1.2.3 Bruit et Signal 42

2.1.2.4 *k-fold cross validation* 42

2.1.3 Métriques d'évaluation 43

2.1.3.1 Probabilités d'appartenance à une classe 43

2.1.3.2 *Accuracy* 43

2.1.3.3 Matrice de confusion 44

2.2 Algorithmes usuels 44

2.2.1 Arbre de décision 44

2.2.1.1 Apprentissage 45

2.2.1.2 Inférence 47

2.2.2 Forêt aléatoire 49

2.2.2.1 Bagging et Out-Of-Bag 49

2.2.2.2 Random Feature Selection 50

2.2.2.3 Capacité de généralisation 50

2.2.3 Calcul des probabilités et prédictions 51

2.2.3.1 Méthode classique 51

2.2.3.2 Méthode proposée 51

2.2.4 Modèle *Nearest Class Mean* 53
2.2.5 Forêt NCM (*NCMF*) 53
2.2.5.1 Apprentissage 54
2.2.5.2 Inférence 54
2.2.5.3 Intérêt du modèle 55
2.2.6 Réseaux de neurones 55
2.2.6.1 Apprentissage d'un neurone formel 56
2.2.6.2 Réseaux de neurones monocouche 56
2.2.6.3 Réseaux de neurones multicouche 57
2.2.6.4 Réseaux de neurones à convolution 57

3 État de l'art **59**

3.1 Reconnaissance des émotions faciales 60
3.1.1 Informatique affective : modélisation des émotions 61
3.1.1.1 Le système catégoriel subjectif 61
3.1.1.2 Le système dimensionnel subjectif 62
3.1.1.3 Le système FACS objectif 63
3.1.2 Intérêt pour l'Éducation 65
3.2 Apprentissage incrémental 66
3.2.1 Réseaux de neurones et oubli catastrophique 68
3.2.1.1 Les stratégies de répétition (*rehearsal*) 68
3.2.2 Méthodes basées sur les forêts aléatoires 70
3.3 Apprentissage semi-supervisé 72
3.3.1 Contexte 72
3.3.2 Self-training 73
3.3.3 Co-training 74

3.4 Discussion 75

4 Travaux préliminaires réalisés sur les données TEEC **77**

4.1 Travaux connexes 78

4.2 Méthodologie DBR et collecte de données 79

4.3 Effet Eurêka 80

4.4 Analyse de l'itération "Langues" 83

4.5 Méthodologie 84

4.5.1 Interactions verbales et élaboration des connaissances 84

4.5.2 Analyse des interactions non verbales (affectives) 85

4.6 Résultats : Analyse conjointe 86

4.7 Conclusion 89

5 Datasets **91**

5.1 Extraction des features 92

5.1.1 Prétraitements réalisés par la librairie OpenFace 92

5.1.2 Unités d'actions (AUs) 93

5.1.3 Textures (TX) 94

5.2 Extended Cohn-Kanade Dataset CK+ 94

5.2.1 Description des sujets 94

5.2.2 Description des données 94

5.2.3 Description des datasets utilisés 95

5.3 Compound Facial Expressions of Emotion (CFEE) 96

5.3.1 Description des sujets 96

5.3.2 Description des données 96

5.3.3 Description des datasets utilisés 96

6 Apprentissage incrémental supervisé en données **97**

6.1 Avantages et limites du classifieur NCM 98

6.1.1 Évaluation 98

6.1.2 Personnalisation 99

6.2 Stratégies d'apprentissage incrémental dans la NCMF 100

6.2.1 Stratégie mettant à jour les feuilles 100

6.2.2 Stratégie créant de nouveaux noeuds 101

6.3 Stratégie proposée de création conditionnelle d'un nouveau mode 102

6.3.1 Stratégie UpdateCentroïd (UC) 103

6.3.2 Stratégie AddCentroïd (AC) 104

6.3.3 Critère de choix de la solution optimale 104

6.4 Protocole expérimental 110

6.4.1 Préparation des datasets 110

6.4.1.1 Données d'apprentissage CFEE 110

6.4.1.2 Données d'évaluation intermédiaires CFEE 111

6.4.1.3 Données d'incrémentation CK+ 114

6.4.1.4 Données d'évaluation CK+ 114

6.4.2 Choix des hyperparamètres du modèle générique 114

6.4.3 Architecture du pipeline de personnalisation 117

6.5 Résultats 119

6.5.1 Qualité de la baseline 119

6.5.2 Comparaison des stratégies incrémentales sur CK+ 120

6.5.3 Performances par classe 122

6.5.4 Statistiques sur les stratégies d'IGTC 122

6.5.5 Performances inter-groupes des modèles personnalisés 124

6.5.6 Performances sur des expressions plus subtiles 125
6.6 Conclusions 127

7 Apprentissage incrémental semi-supervisé en données 129

7.1 Travaux préliminaires sur l'apprentissage incrémental semi-supervisé 130
7.1.1 One Pass Incrementation 130
7.1.2 Two Pass Incrementation 131
7.1.3 Continuous Incrementation 132
7.1.4 Résultats expérimentaux 132
7.2 Algorithme de co-incrementation 133
7.2.1 Algorithme original de co-training 133
7.2.2 Algorithme de co-training incrémental (EBSICO) 134
7.3 Pipeline expérimental et résultats 137
7.3.1 Préparation des datasets 137
7.3.2 Apprentissage des modèles de référence 137
7.3.3 Personnalisation par cluster 139
7.3.4 Évaluation du modèle selon la stratégie de personnalisation choisie 139
7.3.5 Résultats expérimentaux 140
7.3.5.1 Contribution de la co-incrémentation sur les modèles individuels . . . 140
7.3.5.2 Apport de la personnalisation 142
7.3.5.3 Comparaison des performances avec la procédure de co-training classique 145
7.4 Conclusion 146

8 Conclusions et perspectives 147

8.1 Conclusions 147
8.2 Perspectives 149

Publications **152**

Bibliographie **155**

Liste des tableaux

2.1 Exemple de matrice de confusion d'un modèle faisant des erreurs de classification . . . 44

6.1 Sélection du nombre d'arbres en fonction du score oob 115

6.2 5-fold cross validation pour trouver le nombre optimal de features n_F 116

6.3 5-fold cross validation pour trouver le nombre optimal de n_{stop} 116

6.4 5-fold cross validation pour trouver le nombre optimal pour $n_{\mathcal{K}}$ 117

6.5 Récapitulatif des paramètres utilisés pour entraîner la baseline NCMF 117

6.6 Performances inter-groupes avec les modèles IGTC - chaque modèle (i), spécialisé sur son slot correspondant, a été évalué sur les autres slots (s). La dernière ligne correspond à l'évaluation du modèle générique (baseline) non spécialisé sur les différents slots. Le score le plus élevé de chaque ligne apparaît en gras. Pour chaque ligne, un score est souligné lorsque celui-ci est supérieur ou égal à celui de la baseline. 125

6.7 Performances moyennes sur d'autres images des séquences de CK+ - évaluation sur les états neutre de la frame numéro 2 et émotions de la frame numéro $(n-2)$, puis, sur les états neutre de la frame numéro 3 et émotions de la frame numéro $(n-3)$. Plus on travaille avec des images vers le milieu de la séquence vidéo, et plus les états correspondants sont subtils et résultent de transitions d'un état neutre avec une émotion. 126

6.8 Comparaison des performances moyennes des deux modèles NCMF et CNN sur les émotions subtiles et sur les données initiales avant et après incrémentation 127

7.1 Taux de reconnaissance moyen sur 8 slots en fonction du pourcentage de données non labélisées (incrémentation avec la méthode IGTC) 133

7.2 Vue AU. Mesures de l'accuracy à différentes étapes du pipeline EBSICO avec **5%** d'observations labélisées - chaque ligne correspond à un slot selon le critère de personnalisation choisi pour séparer les sujets : un seul slot avec le critère ALL, deux slots Man et Woman avec le critère $GENDER$, et huit slots avec le critère $MORPHO$ allant de $C1$ à $C8$. 140

7.3 Vue TX. Mesures de l'accuracy à différentes étapes du pipeline EBSICO avec **5%** d'observations labélisées 141

7.4 Accuracy moyenne par slot en fonction du pourcentage de données labélisées - EBSICO sur la vue (AU) 143

7.5 Nombre moyen d'images labélisées selon le critère de personnalisation 143

7.6 Évolution de la variance intra-classe selon la séparation des sujets 145

7.7 Comparaison de la méthode de co-training classique avec notre méthode de co-incrémentation 146

Table des figures

1.1 Plusieurs rencontres avec les mêmes groupes - rencontre 1 en haut, rencontre 2 en bas 29

1.2 Dérives de concept : la représentation de la classe X évolue au cours du temps de manière (a) brutale ou (b) progressive / lente 30

1.3 Illustration du biais d'identité : neutre en haut, joie en bas 31

2.1 Dilemme biais-variance [1] 42

2.2 Exemple de frontières de décisions pour un arbre de décision classique - les frontières sont des coupes orthogonales. 47

2.3 Un arbre de décision entraîné à reconnaître les classes A, B, C et D. Les labels majoritaires dans les feuilles l_1, l_2, l_3 et l_4 sont respectivement B, A, D et C. 48

2.4 Illustration des distributions de classes au sein de feuilles d'une forêt 53

2.5 Exemple de frontières de décisions pour un arbre de décision NCM. 54

2.6 Split dans un noeud d'une NCMF - les observations sont dirigées vers le noeud fils associé au centroïde le plus proche, que ce soit lors de la phase d'apprentissage initiale (a), incrémentale ou d'évaluation (b). 55

3.1 Représentation graphique du modèle des émotions dans le circomplexe de Russell [2] Dimension horizontale : valence - Dimension verticale : arousal 63

3.2 Représentation d'une partie des unités d'actions faciales de FACS 64

3.3 Recency rehearsal - Si 20 observations sont présentes lors de l'apprentissage original, alors la file ressemble à a) pour l'incrément de l'observation 21, puis b) pour l'observation 22, et enfin c) pour l'observation 23 69

3.4 Finetuning et oubli catastrophique - le réseau se spécialise sur la dernière classe à partir de laquelle il s'est incrémenté et oublie les anciennes 70

4.1 Illustration de l'effet Eurêka avec des images issues de TEEC 82

4.2 Itération Langues - étudiants de Québec et de Guadeloupe 84

4.3 Itération Langues (Guadeloupe), résultats de l'analyse émotionnelle - chaque graphe est associé à une émotion basique, et décrit le nombre d'élèves exprimant l'émotion au même instant de la séquence vidéo 86

4.4 Analyse conjointe : verbal et non verbal - des successions de tristesse puis joie correspondent à des phases d'explications et d'accord mutuel dans la vidéo (1,2,3,5). Effet eurêka observé en (8) correspondant à une compréhension totale 88

5.1 Modèle de 68 points caractéristiques (*landmarks*) utilisés par OpenFace 93

5.2 Activation de muscles faciaux lors d'expressions faciales - *tiré de imotions.com* 93

5.3 Exemple d'une séquence vidéo de CK+ pour la surprise - la séquence commence avec le sujet à l'état neutre (0) et se termine sur l'état de surprise exprimé avec une intensité maximale (n) 95

6.1 Illustration de la mauvaise orientation de certains exemples lors de l'incrémentation - l'observation (5) sera dirigée vers le mauvais centroïde (j), le centroïde (i) peut se déplacer avec sa frontière pour le faire passer du bon côté en (a), mais ce n'est plus possible en (b) car le noeud devient multimodal. 100

6.2 Stratégies UC (a) et AC (b) utilisées lors de IGTC - le centroïde s'est déplacé vers l'observation (x) avec la stratégie UC (a), un nouveau centroïde est créé avec la stratégie AC (b) pour bien diriger l'observation (x) quand UC n'est plus applicable. 104

6.3 Choix de la stratégie basé sur la comparaison des deux indices de Calinski-Harabasz - quand $CH_1 > CH_2$, UC est appliqué (a), sinon, UC est encore appliqué si $d_1 \leq d_2$ (b), dans le cas contraire, AC est appliqué (c). 108

6.4 Illustration de la stratégie UC : (a) initialement, une observation (sur fond jaune) est mal classée. Après avoir mis à jour le centroïde, (b) l'observation est correctement classée. 109

6.5 Distribution des données par classe dans $(X, Y)_A^{[CFEE]}$ 112

6.6 Distribution des données par classe dans $(X, Y)_E^{[CFEE]}$ 112

6.7 Distribution des classes dans CK+ 113

6.8 Pipeline de l'apprentissage personnalisé incrémental supervisé - un modèle NCMF générique est entraîné sur CFEE (1), puis spécialisé sur de nouveaux sujets issus de CK+ (2), enfin, le modèle est évalué sur sa capacité de personnalisation à ces derniers en (3). 118

6.9 Matrice de confusion sur $CFEE_E^{[AU]}$ (normalisée en lignes) - le modèle reconnaît de manière correcte les états neutre et joie, mais a des confusions pour les états de tristesse, colère, dégoût. 119

6.10 Confusions entre tristesse (a), colère (b) et dégoût (c) - les sourcils sont froncés (1ère ligne) et les bouches tirées vers le bas (2ème ligne) pour les différents états expressifs, entraînant des confusions. 120

6.11 Performances des modèles incrémentés en fonction du nombre d'arbres sur $(X, Y)_E^{[CK+]}$ 121

6.12 Comparaison IGTC (à gauche) et IGT (à droite) - l'accuracy est plus élevée avec IGTC et les écarts types plus réduits qu'avec IGT. 121

6.13 Taux de reconnaissance par classe sur $(X, Y)_E^{[CK+]}$ - les lacunes de la baseline pour reconnaître la tristesse et la colère ont été comblées par les stratégies IGT et IGTC. . 122

6.14 Nombre moyen de fois où les stratégies AC (orange) ou UC (bleu) sont activées par slot. 123

6.15 Nombre moyen de fois où les stratégies AC (orange) ou UC (bleu) sont activées par émotion 124

7.1 Diagramme décrivant l'ensemble de la procédure EBSICO proposée - chaque modèle générique s'entraîne sur un premier dataset et sa vue correspondante v_1 ou v_2 (1), puis, les modèles se spécialisent sur quelques nouveaux sujets labélisés (2), enfin, la personnalisation se poursuit sur un grand nombre de nouveaux sujets mais non labélisés via la collaboration entre les deux modèles (3). 137

7.2 Évolution de la variance intra-classe en fonction du nombre de clusters (*courbe en coude*) 139

7.3 Évolution du taux d'erreurs non communes à travers les différentes étapes du framework EBSICO, en fonction du taux de données labélisées - les modèles convergent vers un accord mutuel en fin de procédure EBSICO quel que soit le taux de labélisation utilisé à l'étape (2) du pipeline. 142

8.1 Apprentissage incrémental en classe de la NCMF - a) la forêt est initialisée sur l'émotion peur tandis que dans b) elle commence par apprendre l'émotion Neutre 149

Chapitre 1

Introduction

Contenu

1.1 Genèse du projet 26
1.2 Intérêt de l'informatique affective pour TEEC 27
1.2.1 Défis liés à la modalité visuelle 28
1.2.2 Les contraintes liées à la collecte de données dans le projet TEEC 28
1.3 Biais d'identité et personnalisation 29
1.3.1 Définition 29
1.3.2 Généralisation *vs* personnalisation 31
1.3.3 Apprentissage incrémental : principes et défis 34
1.4 Organisation du mémoire et contributions 34

Les travaux présentés dans ce mémoire s'inscrivent principalement dans le domaine de l'informatique affective. Ce domaine de recherche est assez récent mais a évolué de manière spectaculaire pendant ces deux dernières décennies. Il a l'objectif ambitieux de donner aux machines la capacité de percevoir et d'exprimer les émotions humaines. Une première taxonomie du domaine sépare donc la perception (via des capteurs et des algorithmes d'analyse des signaux issus de ces derniers), de l'expression (basée le plus souvent sur des agents conversationnels contrôlés, eux aussi, par des algorithmes).

Nous nous concentrerons sur la partie perception, qui vise en premier lieu à détecter les émotions humaines afin de les interpréter dans un cas d'usage précis. Ici, nous nous intéressons particulièrement aux émotions exprimées par les apprenants dans le cadre d'interactions pédagogiques et aux liens existants entre états affectifs et états cognitifs.

Nous présenterons d'abord le contexte du projet qui est à l'origine de ces travaux et les différentes contraintes que celui-ci génère en termes d'acquisition de données. Nous verrons ensuite comment ces contraintes impactent les algorithmes d'apprentissage automatique utilisés pour analyser les émotions. Pour finir, nous présenterons les différentes solutions que nous avons proposées pour gérer au mieux ces contraintes.

1.1 Genèse du projet

Les travaux à l'origine de ce projet ont été réalisés par deux équipes de recherche. Le Centre de Recherches et de Ressources de l'Université des Antilles est spécialisé en didactique des sciences et, à ce titre, conçoit des méthodes de scénarisations pédagogiques innovantes. Le LICEF (Laboratoire d'Informatique Cognitive et Environnements de Formation) de la Télé-Université du Québec s'intéresse à l'ingénierie pédagogique et particulièrement aux environnements informatiques pour l'apprentissage humain, en vue d'utiliser au mieux les outils du numérique pour assister l'apprentissage.

Ensemble, ils ont proposé une méthode originale basée sur la confrontation de contexte à distance. Elle s'appuie sur des concepts de la contextualisation didactique comme les effets de contexte et les écarts de contexte [3][4]. Ils ont permis de développer un scénario pédagogique dans lequel sont réunis, à distance en visio-conférence, deux groupes d'étudiants afin qu'ils collaborent ensemble sur une thématique commune. Les deux groupes sont inscrits dans des contextes différents, aussi bien sur le plan géographique que sur le plan des représentations mentales [5]. La confrontation entre

les conceptions des deux groupes provoque un choc cognitif (effet de contexte) chez les étudiants concernés. Le modèle CLASH proposé par [4] décrit ce comportement. Ce "clash" de contexte est un "quiproquo socio-cognitif". Il peut, dans un premier temps, constituer un obstacle à l'apprentissage. Mais, il peut aussi se résoudre, grâce à des interactions répétées et bien scénarisées, et devenir un atout pour l'apprentissage. Il permet, en outre, l'enrichissement mutuel, tant intellectuel que culturel, des deux groupes d'étudiants.

L'analyse qualitative des vidéos recueillies lors des premiers échanges entre les groupes d'élèves du Québec et de Guadeloupe a montré que les différentes étapes du processus (conflit, discussion et résolution) étaient riches en interactions verbales et non verbales. Ce qui suggérait que les aspects cognitif et affectif de l'apprentissage étaient intimement liés et potentiellement corrélés. Il devenait donc nécessaire, pour valider la méthode proposée, d'analyser à la fois les interactions langagières et les émotions apparaissant au fil des échanges. C'est pourquoi, les deux équipes ont fait appel à des équipes de recherches dans ces deux domaines.

Aux deux équipes précédentes se sont jointes l'équipe INTERACT (INteraction, TEchnologie, ACTivité) de l'I3 (Télécom-ParisTech), l'équipe EDA de l'Université Paris-Descartes, l'équipe LDR (Learning, Data, Robotics) de l'ESIEA. La proposition a été soumise à l'appel franco-québécois ANR FRQSC sous le titre TEEC (Technologies Éducatives pour l'Enseignement en Contexte) et finalement, sélectionnée.

1.2 Intérêt de l'informatique affective pour TEEC

L'analyse automatique des émotions est un domaine en constant développement. Les systèmes interactifs, comme par exemple les systèmes tutoriels intelligents, ont tout intérêt à pouvoir percevoir les états émotionnels de leurs utilisateurs et à s'y adapter automatiquement, afin de restituer une expérience utilisateur optimale. En ce qui concerne l'utilisation de l'informatique affective dans l'environnement éducatif, les recherches autour de ce thème ont récemment visé à identifier, analyser et décrire les déclencheurs des émotions et les conséquences de ces dernières sur les performances des apprenants (stimulation, perturbation...).

Les émotions peuvent être analysées à partir de différentes sources telles que l'audio, le visuel ou le texte. Pour le projet TEEC, nous avons à notre disposition des vidéos, mais, les étudiants n'ayant pas

reçu de règles précises concernant l'utilisation du micro lors de leurs interactions par visioconférence, nous nous sommes retrouvés assez souvent avec des étudiants parlant en même temps rendant peu exploitable le canal audio ; nous avons donc décidé de nous concentrer uniquement sur la modalité visuelle dans ce travail.

1.2.1 Défis liés à la modalité visuelle

La détection des expressions faciales ou des émotions dans la modalité visuelle présente son propre ensemble de défis, pouvant être liés au matériel de capture, au recalage, à la luminosité, à la pose de la tête, à l'occultation, à l'émotion forcée ou non, etc. [6]. De plus, bien que bon nombre de solutions aient été présentées pour caractériser de mieux en mieux les données, les performances des modèles pour la reconnaissance d'émotions demeurent souvent dépendantes de la variabilité individuelle chez les individus. Ce **biais d'identité**, que nous détaillons dans la section suivante (cf. Section 1.3) représente un défi majeur pour ce domaine. C'est le défi sur lequel nous avons décidé de nous concentrer, car le modèle que nous proposons doit pouvoir être performant, pour tout élève, quel que soit son âge, son origine ethnique, sa morphologie, ou sa façon d'exprimer une émotion.

1.2.2 Les contraintes liées à la collecte de données dans le projet TEEC

La collecte de données dans le projet TEEC est décrite dans le chapitre 4. Les données recueillies dans le cadre de ce projet sont des vidéos d'étudiants interagissant par visioconférence avec d'autres étudiants vivant dans des régions différentes. Les sessions de travail sont répétées plusieurs fois, de sorte que nous nous retrouvons avec différentes données des mêmes étudiants (voir Figure 1.1). À noter que, les rencontres ne se réalisent ni forcément le même jour, ni dans les mêmes conditions de capture. La question s'est alors posée de savoir comment utiliser l'apprentissage automatique dans ce type de projet, où les données ne sont pas toutes disponibles dès le départ, mais arrivent plutôt au fur et à mesure. C'est pourquoi nous nous sommes tournés vers **l'apprentissage incrémental**. De plus, les données mises à disposition lors du projet n'étant pas labélisées, nous avons poursuivi nos investigations dans le cadre des méthodes incrémentales faiblement supervisées.

FIGURE 1.1 – Plusieurs rencontres avec les mêmes groupes - rencontre 1 en haut, rencontre 2 en bas

1.3 Biais d'identité et personnalisation

Nous tenterons d'abord, modestement, de définir le problème du biais d'identité, ses différentes composantes et son impact sur les performances des systèmes de reconnaissance. Puis, nous présenterons les solutions proposées par différentes communautés de recherche au cours de ces dernières décennies.

1.3.1 Définition

Dès qu'on s'intéresse à la reconnaissance d'une production humaine, qu'il s'agisse de l'écriture, de la voix, d'une activité ou d'une émotion, on se heurte au problème de la variabilité. En effet, chaque individu étant unique, ses productions le sont aussi. Ceci se traduit par une variabilité dans les observations correspondantes qui croît avec le nombre d'individus considérés, mais aussi par une baisse des performances des systèmes de reconnaissance automatique qui constitue ce que l'on appellera, par la suite, le biais d'identité.

Dans le cas émotionnel, et si l'on s'intéresse uniquement à l'émotion faciale, ce biais peut avoir deux sources. La première est morphologique. Certains individus ont des visages plus ou moins ronds, des pommettes plus ou moins hautes, etc. Leurs traits peuvent être plus ou moins marqués ; par exemple, les sourcils peuvent être épais ou fins, courbés ou non, les rides plus ou moins prononcées, etc. La seconde est de nature comportementale. Deux individus sont susceptibles d'exprimer différemment un même état émotionnel ; ainsi, certains exprimeront leur joie par un sourire très marqué alors que

d'autres seront plus réservés (cf. Figure 1.3), en fonction de leur personnalité, extravertie ou introvertie. Notons toutefois qu'un même individu pourra avoir une réaction émotionnelle plus ou moins marquée en fonction de l'intensité du stimulus ayant déclenché cette dernière.

Ces biais sont donc la conjonction de très nombreux facteurs tels que l'âge, le genre, la culture et le milieu social. Ils se traduisent par :

— des données présentant de fortes variabilités, tant intra-individuelles qu'inter-individuelles, dont les distributions sous-jacentes sont complexes et multimodales ;

— des données non-stationnaires présentant des dérives progressives / lentes (cf. Fig.1.2.b) au niveau morphologique, ou brutales (cf. Fig.1.2.a) au niveau comportemental.

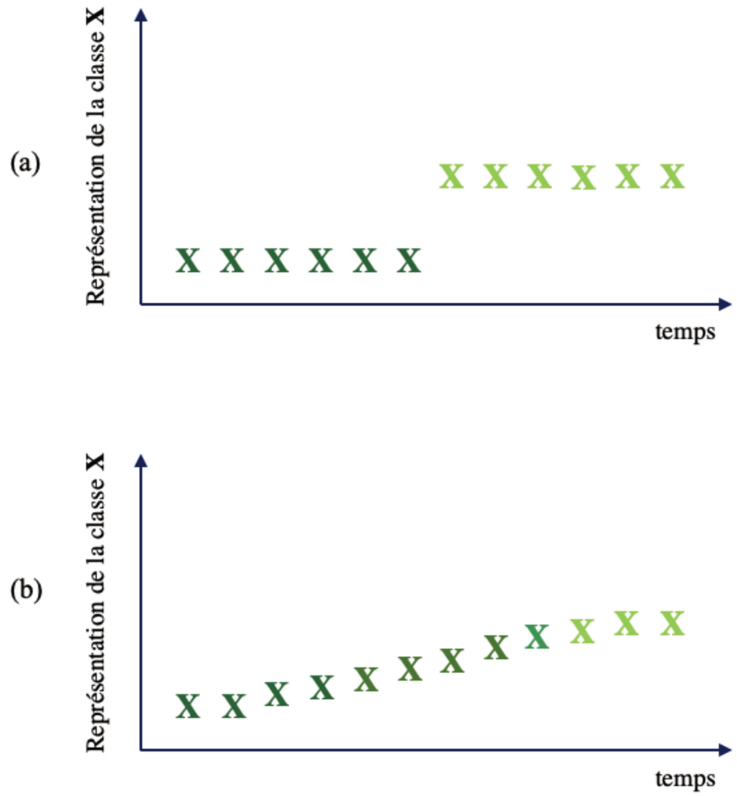

FIGURE 1.2 – Dérives de concept : la représentation de la classe X évolue au cours du temps de manière (a) brutale ou (b) progressive / lente

Ils sont rassemblés sous le terme de **biais d'identité**, qui représente un défi majeur dans le domaine de la reconnaissance des émotions [6].

FIGURE 1.3 – Illustration du biais d'identité : neutre en haut, joie en bas

Le biais d'identité touche d'autres domaines comme celui de la reconnaissance automatique d'activités. On retrouve la notion de biais morphologique (une même activité sportive peut être réalisée de différentes manières selon les individus) et de biais comportemental (un même sujet ne répète pas toujours la même activité en faisant exactement les mêmes gestes). Enfin, ces gestes peuvent effectivement évoluer au cours du temps en raison de l'âge, de l'évolution des performances sportives, des blessures éventuelles, ou bien encore des facteurs environnementaux (météo, terrain ...). On retrouve ici le problème de non-stationnarité des données.

Nous avons évoqué précédemment deux autres productions humaines que sont l'écriture et la voix. Les deux biais constituant le biais d'identité sont aussi présents ici. Mais les informations contenues dans les données ne permettent pas de les identifier. Il n'en reste pas moins que les communautés de recherche ayant travaillé respectivement sur la reconnaissance de l'écriture et la reconnaissance vocale avaient bien observé ce biais, sans le nommer.

1.3.2 Généralisation *vs* personnalisation

1.3.2.1 Taxonomie

"Historiquement", les deux communautés ont d'abord tenté d'optimiser les performances de systèmes de reconnaissance dits "omni-scripteur" (ou omni-locuteur). On parlera ici de classifieurs **génériques**, de "modèle du monde" ou de modèles *subject-independent*. Constatant le plafond de verre dû à la trop forte variabilité intrinsèque des données, ils se sont tournés vers la conception de modèles **personnalisés** à un ensemble d'individu (cas multi-scriteur/locuteur). La suite logique était de concevoir un modèle par individu (cas mono-scripteur/locuteur), appelé aussi modèle *subject-dependent*.

Les deux principales stratégies envisagées pour personnaliser un modèle sont :

— La stratégie de personnalisation "directe". Cette dernière consiste à définir un critère pour séparer un jeu de données en différentes sous-ensembles appelés (*slots*), puis à entraîner un modèle sur chacun d'eux. La taille du sous-ensemble peut se réduire à un individu ce qui conduit à un modèle "subject-dependent" ;

— La stratégie indirecte qui procède en deux étapes. Elle commence par l'apprentissage du modèle générique sur un dataset suffisamment conséquent pour que ce modèle généralise correctement. Puis, ce modèle est mis à jour et spécialisé aux données des différents slots définis précédemment.

Notons que les deux stratégies génèrent autant de modèles différents qu'il y a de slots. Lors de l'évaluation des performances, on obtient ainsi une mesure de performance pour chacun d'eux. Dans le cas indirect, l'écart entre les performances du modèle générique et celles du modèle personnalisé permettra d'évaluer l'impact de la spécialisation.

Notons surtout que ces deux stratégies sortent du cadre "classique" de l'apprentissage automatique et nécessitent de mettre en place des solutions d'apprentissage incrémental. Nous définirons plus en détail ce paradigme dans la section suivante. Mais, nous présentons d'abord ci-dessous quelques travaux intéressants rapportés dans la littérature.

Notons enfin, dans les deux cas d'usage évoqués précédemment (reconnaissance d'écriture et reconnaissance vocale), la présence de contraintes fortes, en particulier lexicales et syntaxiques. De fait, le lexique et les règles syntaxiques permettent de corriger les erreurs commises par le système de reconnaissance et de générer des pseudo-labels utilisés pour mettre à jour le classifieur. Ce type d'apprentissage incrémental peut donc se faire sans supervision de la part de l'utilisateur, ce qui est particulièrement intéressant [7]. Mais malheureusement non applicable dans d'autres cas d'usage [8], dont le nôtre.

1.3.2.2 Solutions de personnalisation

Dans le cadre de la reconnaissance émotionnelle, il n'est pas toujours évident, ni pour une machine, ni même pour un humain, de pouvoir déduire l'émotion d'un individu en se basant sur une seule image, sans connaître son visage lorsque celui-ci est neutre ; sans non plus connaître le contexte dans lequel se trouve l'individu lorsqu'il exprime cette émotion. Ce constat peut-être fait en observant par exemple

l'image de la troisième personne en bas en partant de la gauche (voir Figure 1.3) : si son visage à l'état neutre n'est pas connu, il devient difficile de conclure que cette expression correspond à la joie. Les performances des classifieurs génériques peuvent ainsi être considérablement influencées par un tel biais d'identité.

Différentes solutions ont été envisagées pour aborder le problème du biais d'identité et de son impact significatif sur les performances de classifieurs génériques ; certaines contournent le problème comme [9] où les auteurs proposent une méthode pour atténuer tout indice facial pouvant identifier l'individu ; d'autres comme [6], considèrent la personnalisation comme une approche judicieuse. Certaines études portent sur des méthodes de personnalisation non supervisées, telles que la méthode *Support Vector-based Transductive Parameter Transfer* (SVTPT) [10] ou la méthode *Selective Transfer Machine* (STM) [11] ; celles-ci permettent de personnaliser un modèle soit à un individu, soit à un groupe d'individus partageant des similarités, ce qui atténue par conséquent le biais d'identité. Ces techniques permettent notamment de reconnaître la "forme" des traits caractéristiques chez les individus à partir des distributions de leurs données et peuvent être ainsi appliquées sur des données non labélisées. La personnalisation a été considérée en deep learning ; des techniques comme *Mixture-of-Experts* (MoEs) [12] combinées avec de l'adaptation de domaine supervisée (DA) [13], ou encore la technique classique de *transfer learning* [14], ont été proposées pour personnaliser les réseaux pour le domaine valence / arousal, surpassant ainsi les performances de classifieurs génériques. Toujours dans le domaine valence / arousal, une étude comparative entre différents modèles de machine learning d'apprentissage supervisé a été réalisée par Kollia en 2016 [15] ; ce sont les forêts aléatoires personnalisées qui obtiennent les meilleurs résultats.

Dans le domaine de la reconnaissance d'activité, plusieurs études [16][17][18] ont montré que les performances pourraient être fortement dépendantes des caractéristiques de l'individu, et que la solution des modèles personnalisés aux individus est prometteuse car ils sont plus performants que les modèles génériques. Ils ont donc proposé des méthodes de personnalisation utilisant des modèles de forêt aléatoire, et ont personnalisé leurs modèles en regroupant les individus partageant des traits communs tels que l'âge, le poids, la taille, le sexe, le style de vie, etc.

1.3.3 Apprentissage incrémental : principes et défis

Si nous disposons d'un détecteur d'émotions, entraîné sur quelques données reçues à une itération, nous pouvons nous demander ce que nous faisons de ce modèle lorsque des données supplémentaires, sur lesquelles il pourrait également être entraîné, arrivent. L'apprentissage incrémental se distingue de l'apprentissage classique de la manière suivante : un dataset est donné à un modèle à un instant t, le modèle s'entraîne sur celui-ci et devient le modèle t. Des données supplémentaires, c'est-à-dire un autre dataset, arrivent à un instant $t + 1$. Dans l'apprentissage classique, le modèle est entraîné à partir de zéro, sur les données reçues à l'instant t, mais aussi sur les données reçues à l'instant $t + 1$: il concatène les deux datasets comme s'il les avait reçus en une seule fois et s'entraîne sur eux. En d'autres termes, le modèle précédent est écrasé par le nouveau, de sorte que les connaissances acquises précédemment sont également écrasées. Dans l'apprentissage incrémental, le modèle t est mis à jour à partir des données reçues au temps $t + 1$, sans nécessairement écraser sa structure précédente, ni utiliser l'ancien dataset reçu au temps t.

L'apprentissage incrémental comporte son lot de défis qui sont généralement l'oubli catastrophique, le problème de la baisse des performances, ou encore le problème du stockage des données. L'oubli catastrophique est un phénomène qui peut être résumé par l'oubli des connaissances précédemment acquises lors de l'incrémentation à partir d'un nouveau jeu de données. Il a également été observé que plus un modèle est incrémenté dans le temps avec de nouveaux jeux de données, plus les performances baissent, contrairement à ce qu'aurait pu obtenir un modèle entraîné à partir de zéro sur l'ensemble des données considérées à ce moment-là [19]. Enfin, certaines stratégies incrémentales nécessitent l'utilisation d'anciennes données déjà vues par le modèle, ce qui pose un problème quant à leur stockage : la question se pose de la faisabilité à long terme de devoir les stocker ou, dans le pire des cas, de stocker tous les anciens datasets vus par le modèle.

1.4 Organisation du mémoire et contributions

Les objectifs de cette thèse sont d'adapter automatiquement des modèles de machine learning, entraînés à reconnaître des états émotionnels, aux traits morphologiques et comportementaux de l'individu, afin de réduire le biais d'identité.

Le document est divisé en deux parties principales. La première a pour objectif de présenter le

cadre de travail et l'état de l'existant.

— Le chapitre 2 décrit le cadre général de l'apprentissage supervisé à partir de données. Nous y définissons les concepts de jeu de données labélisées et de modèle (algorithmique) d'apprentissage. Nous présentons ensuite les difficultés rencontrées lors de l'apprentissage et les solutions pour éviter ces dernières. Nous terminons par une revue des modèles mis en œuvre dans le cadre de nos travaux ;

— Le chapitre 3 présente un état de l'art qui se décompose en trois thématiques principales. Nous commençons par décrire les différentes représentations utilisées pour décrire les états affectifs et émotions humaines. Puis, nous passons en revue les différentes familles d'apprentissage que sont l'apprentissage incrémental et enfin l'apprentissage semi-supervisé en données. Ceci nous permet de justifier les choix méthodologiques faits par la suite.

La seconde partie décrit l'ensemble des contributions apportées par nos travaux.

— Le chapitre 4 décrit les données récoltées par les équipes pédagogiques en début de projet. Nous présentons en détail une analyse de ces données, fruit d'une collaboration avec d'autres membres du consortium TEEC. Nous montrons, de façon qualitative, les corrélations existantes entre les interactions verbales (discursives et cognitives) et non verbales (affectives). Nous concluons en présentant les défauts des données récoltées, qui ont orienté le reste de nos travaux ;

— Le chapitre 5 présente deux bases de données grand public, largement citées dans la littérature, qui nous ont permis de réaliser l'apprentissage des modèles génériques et leur personnalisation ;

— Le chapitre 6 décrit le protocole de personnalisation mis en œuvre pour entraîner le modèle générique et le personnaliser à un ensemble d'individus, afin de diminuer le biais d'identité. Nous montrons que les stratégies proposées dans la littérature ne sont pas adaptées aux différents types de dérives apparaissant dans les données au fil du temps. Pour y faire face, nous proposons une stratégie d'incrémentation originale, prenant notamment en compte le caractère (éventuellement) multimodal des données. Cette dernière permet de gérer les dérives lentes et les dérives plus brutales ;

— Le chapitre 7 décrit l'utilisation de l'apprentissage incrémental pour le domaine semi-supervisé, c'est-à-dire lorsqu'un petit nombre seulement de données est labélisé dans un dataset. Nous proposons, à cet égard, une méthode hybride qui combine le co-training et l'apprentissage incrémen-

tal, permettant à deux modèles de collaborer et de partager leurs connaissances. Contrairement à la méthode classique de co-training qui effectue un ré-entraînement à partir de zéro à chaque itération de l'algorithme, notre approche effectue une incrémentation du modèle en continu sur les nouvelles observations. Nous montrons, de plus, que la personnalisation à des individus, basée sur des critères de genre ou morphologique, améliore les performances du système.

— Le chapitre 8 est dédié à la conclusion de ces travaux et présente, en outre, quelques perspectives de recherche.

Chapitre 2

Concepts

Contenu

2.1 Apprentissage supervisé 38
2.1.1 Dataset 38
2.1.2 Classifieur 40
2.1.3 Métriques d'évaluation 43
2.2 Algorithmes usuels 44
2.2.1 Arbre de décision 44
2.2.2 Forêt aléatoire 49
2.2.3 Calcul des probabilités et prédictions 51
2.2.4 Modèle *Nearest Class Mean* 53
2.2.5 Forêt NCM (*NCMF*) 53
2.2.6 Réseaux de neurones 55

2.1 Apprentissage supervisé

Cette section se concentre sur l'apprentissage supervisé pour la classification qui est un des domaines les plus anciens du *machine learning*, mais aussi l'un des plus prolifiques. Il décrit ses principes, ses objectifs et les obstacles qu'il peut rencontrer et doit éviter. De manière générale, la tâche de classification vise à catégoriser une observation, c'est à dire à lui attribuer une classe unique choisie dans l'ensemble, connu et fini, des classes ciblées par la tâche. Pour apprendre à catégoriser, les algorithmes - comme les humains - ont besoin d'un ensemble d'observations pour construire leur base de connaissance. L'apprentissage supervisé nécessite de disposer de la classe (appelée aussi label ou vérité terrain) de chaque observation. On peut le rapprocher de l'apprentissage humain, lorsque ce dernier est supervisé par un professeur qui donne toujours la bonne réponse à l'apprenant. L'apprenant améliore ses connaissances en corrigeant ses erreurs. Il en est de même pour l'algorithme. Enfin, tout comme l'apprenant, l'algorithme doit être évalué pour valider sa maîtrise de la tâche.

Nous décrirons d'abord l'ensemble d'observations labélisées (anglicisé en *dataset*) ainsi que quelques grandeurs statistiques utiles pour le caractériser. Puis, nous développerons le notion de modèle (algorithmique) et celle d'apprentissage de ce dernier sur le dataset pour réaliser la tâche. L'ensemble du processus est loin d'être trivial et peut échouer. C'est pourquoi nous décrirons les obstacles que peut rencontrer l'apprentissage et les moyens mis en oeuvre pour les éviter. Nous présenterons, enfin, différentes métriques permettant d'évaluer les performances du modèle entraîné sur la tâche.

2.1.1 Dataset

2.1.1.1 Définition

En classification, un jeu de données (*dataset*) contient les informations collectées d'un échantillon de taille finie, issu d'une population dont on souhaite apprendre à reconnaître la classe (*label*). On qualifie d'observation chaque élément contenu dans le dataset. Chaque observation est décrite par un ensemble de caractéristiques ou variables explicatives (appelées *features*) qualitatives ou quantitatives. Dans nos travaux, nous n'avons manipulé que des variables quantitatives, à valeurs continues. Enfin, à chaque observation est associée une classe, qui correspond à la variable à prédire.

Par convention, cet ensemble de données sera désigné, dans la suite du document par $(X, Y)^{[NAME]}$. On note :

— $NAME$ le nom du dataset ;

— X l'ensemble des observations, stockées dans une matrice de dimension $n_X \times m$, avec n_X le nombre d'observations, et m le nombre de features. Une observation correspond donc à une ligne de la matrice X ;

— $F = \{f_1, ..., f_m\}$ désigne l'ensemble des m features décrivant les observations ;

— Y est un vecteur de dimension $n_X \times 1$ et contient les labels y asssociés aux observations x ;

— $\mathcal{K} = \{k_1, ..., k_l\}$ désigne l'ensemble de l labels dans lequel y prend ses valeurs.

Par exemple, si le dataset utilisé est $CFEE$ (voir Sec.5.3), la notation $(X, Y)^{[CFEE]}$ pourra être utilisée pour désigner ce dataset. On pourra également utiliser les notations $X^{[CFEE]}$, ou $Y^{[CFEE]}$ pour désigner respectivement les observations, ou les labels du dataset.

Par la suite, ce dataset sera divisé en trois ensembles distincts appelés ensembles d'apprentissage, de validation et de test (ou d'évaluation), indépendants et, de préférence, identiquement distribués. Le rôle de chacun de ces ensembles sera décrit et justifié dans la section suivante.

2.1.1.2 Mesure de qualité des données

Apprendre une tâche n'est possible qui si les données récoltées sont de bonne qualité. Nous ne nous attarderons pas sur les problèmes de complétude du dataset (liée à la fréquence et aux types de données manquantes), que nous n'avons pas eu à gérer pendant nos travaux.

Par contre, la position et la dispersion (dans l'espace des caractéristiques) des densités de probabilité conditionnelles des observations sachant les classes, sont un facteur de qualité décisif que nous détaillons maintenant. On appelle cluster un ensemble de données de la même classe. Dans un problème de classification multi-classe, la classification sera d'autant plus facile que les clusters des différentes classes sont bien définis et éloignés les uns des autres. On dit qu'un cluster est de bonne qualité (ou bien défini) si ses observations sont peu dispersées. Deux clusters sont suffisamment séparés s'il n'y a pas (ou peu) d'intersection entre les enveloppes convexes entourant leurs nuages de points respectifs. On peut mesurer ces deux critères (compacité des clusters et éloignement) en calculant respectivement la variance intra-classe I_W, et la variance inter-classe I_B :

$$I_W = \frac{1}{|X|} \sum_{k \in \mathcal{K}} \sum_{x \in X_k} ||x - c_k||^2 \tag{2.1}$$

$$I_B = \sum_{k \in \mathcal{K}} |X_k| \times ||c_k - c_X||^2 \tag{2.2}$$

avec :

— $|X_k|$ le nombre d'observations de la classe k,

— c_k le centroïde de la classe k,

— c_X le centre de gravité de toutes les observations de X.

Une valeur I_W la plus petite possible et une valeur I_B la plus grande possible sont synonymes de données de qualité, bien adaptées à la tâche visée.

2.1.2 Classifieur

2.1.2.1 Apprentissage et généralisation

Un modèle noté μ est un algorithme réalisant une classification des observations. On l'appelle d'ailleurs, communément, "classifieur". On appelle fonction de classification (ou de prédiction), l'application de $\mathbb{R}^m$ dans $\mathcal{K}$ qui associe à chaque observation x, la prédiction $\hat{y}$ d'une classe par le modèle :

$\eta(x) = \hat{y}$

Intuitivement, l'objectif de l'apprentissage est de faire en sorte que le modèle prédise au mieux les classes d'un ensemble d'observations. Ceci se traduit analytiquement par la minimisation du nombre d'erreurs de prédiction $\sum I(y \neq \eta(x))$ sur cet ensemble (I est la fonction indicatrice).

Cet ensemble doit être choisi judicieusement. Si l'on utilise uniquement l'ensemble d'apprentissage pour minimiser l'erreur de prédiction, l'algorithme apprendra "par coeur" ces données, ce qui n'est pas souhaitable. C'est pourquoi l'on utilise comme critère de qualité l'erreur de prédiction sur la base de validation. Ces deux erreurs décroissent de concert tant que l'algorithme apprend correctement la tâche. L'instant où l'erreur de validation commence à croître alors que l'erreur d'apprentissage continue de diminuer est le point de bascule du modèle en sur-apprentissage et constitue un critère d'arrêt.

En effet, l'objectif d'un classifieur n'est pas d'apprendre par coeur mais de généraliser à partir de son expérience. La **généralisation** est la capacité d'un modèle entraîné à réaliser des prédictions de qualité sur de nouvelles observations (inconnues car absentes du dataset d'apprentissage). Dans le processus décrit précédemment, la base de validation est donc utilisée pour contrôler, pendant l'ap-

prentissage, la capacité du modèle à généraliser. C'est une des techniques classiques de régularisation de l'apprentissage ; on parle de régularisation par *early stopping*.

Dans la suite du document, lorsqu'une formulation comme "le modèle prédit", "le modèle va classer une observation" est employée, cela signifie que le modèle déjà entraîné, va attribuer une classe à l'observation, en fonction des connaissances acquises lors de la phase d'apprentissage. Cette phase de prédiction ou d'évaluation se déroule habituellement sur un dataset dit *d'évaluation ou de test* ; les données le constituant n'ont jamais été utilisées précédemment par le modèle.

2.1.2.2 Dilemme Biais-Variance

Pour obtenir les meilleures performances en généralisation, la complexité du modèle hypothèse doit correspondre à la complexité - inconnue - de la fonction sous-jacente aux données.

Si l'hypothèse est moins complexe que la fonction, alors le modèle ne sera pas capable d'apprendre toute la variabilité présente dans les données. Ce phénomène de sous-apprentissage (*underfitting*) se traduit par des performances médiocres sur l'ensemble d'apprentissage. On parle de modèle à biais élevé.

Plus l'on augmente la complexité du modèle, plus le biais diminue. Mais, si l'hypothèse est trop complexe, le modèle est en sur-apprentissage (*overfitting*). Les performances sur les données d'apprentissage sont excellentes mais la généralisation sera moins bonne.

De fait, plus le biais diminue, plus le modèle devient sensible aux petites variations dans l'ensemble d'apprentissage. Cette sensibilité est appelée variance et augmente avec la complexité du modèle. Cette dernière mesure la capacité du modèle à rester stable dans ses prédictions, lors des changements mineurs dans le dataset d'apprentissage. Une variance élevée est souvent un indicateur d'*overfitting*.

Idéalement, il serait souhaitable qu'un modèle ait un biais faible et une variance faible. Cependant, augmenter la complexité d'un modèle pour réduire son biais, augmentera généralement sa variance, et vice-versa (cf. Figure 2.1). De nombreuses méthodes ont été proposées pour gérer ce dilemme et minimiser conjointement le biais et la variance. Nous en présenterons quelques unes dans la suite de ce chapitre.

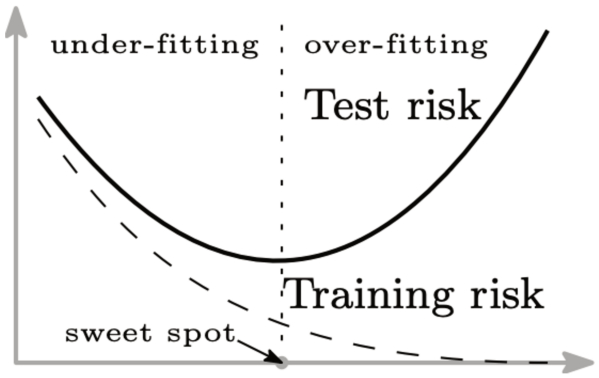

FIGURE 2.1 – Dilemme biais-variance [1]

2.1.2.3 Bruit et Signal

Les humains sont susceptibles de commettre des erreurs lors de la collecte de données. Les instruments recueillant les données peuvent être imprécis et aussi entraîner des erreurs dans les datasets. Ces erreurs peuvent poser problème et être difficiles à identifier et traiter, on parle dans certains cas d'erreurs irreductibles, car quelle que soit la qualité de notre modèle, nos données comporteront une certaine quantité de bruit qui ne pourra être éliminée.

Des informations non pertinentes, généralement une ou plusieurs colonnes dans le dataset, peuvent également poser problème. Elles sont qualifiées de bruit et ne sont pas des erreurs. Cependant, sans apprentissage approprié, le bruit dans les données peut créer des problèmes aux algorithmes d'apprentissage automatique. En effet, ceux-ci peuvent considérer ce bruit comme un motif (*pattern*), et peuvent commencer à généraliser à partir de celui-ci. Le signal s'oppose au bruit, et est considéré comme de l'information pertinente.

2.1.2.4 *k-fold cross validation*

La validation croisée en k itérations est un protocole permettant d'évaluer les performances d'un modèle. Il est également utilisé lorsque l'on recherche la **valeur optimale** d'un hyper-paramètre du modèle : on parle d'optimisation des hyperparamètres ou de *tuning*. On commence par séparer aléatoirement l'ensemble d'apprentissage en k sous-ensembles disjoints. Puis, on entraîne un premier modèle sur les $(k-1)$ premiers sous-ensembles et on l'évalue sur la partie restante, qui sert donc d'ensemble de validation. Le processus est répété k fois, en réservant à chaque fois une partie différente pour l'éva-

luation du modèle correspondant. On peut donc estimer la moyenne et la variance des scores obtenus par les k modèles. Plus le score moyen est élevé (caractéristique d'un biais faible) et plus sa variance est faible, meilleur est le modèle. Le protocole se termine par l'entraînement du modèle sur l'ensemble d'apprentissage, en utilisant la valeur optimale trouvée.

2.1.3 Métriques d'évaluation

2.1.3.1 Probabilités d'appartenance à une classe

On note $\phi_\mu : \mathbb{R}^m \mapsto \mathbb{R}^l$ la fonction associée au modèle μ et retournant un vecteur contenant l'ensemble des probabilités a posteriori des classes :

$$\forall x \in \mathbb{R}^m, \phi_\mu(x) = [P(k_1|x), ..., P(k_l|x)],$$

où on rappelle que :

- $\mathcal{K} = \{k_1, ..., k_l\}$ est l'ensemble de l labels,
- x est une observation à classer,
- $P(k_i|x)$ la probabilité conditionnelle (*a posteriori*) que x appartienne à la classe k_i, pour $1 \leq i \leq l$.

La classe attribuée à l'observation x, est alors déterminée par la règle du maximum *a posteriori* :

$$\eta(x) = \underset{k_i \in \mathcal{K}}{argmax}[P(k_1|x), ..., P(k_l|x)]. \tag{2.3}$$

2.1.3.2 *Accuracy*

L'*accuracy (acc)*, ou taux de reconnaissance, est le taux d'observations bien classées :

$$acc = \frac{\sum_{x \in X} I(y = \eta(x))}{|Y|} \tag{2.4}$$

où y est la classe (vérité-terrain) de x, l'expression $\sum_{x \in X} I(y = \eta(x))$ correspond au nombre de prédictions correctes et $|Y|$ au nombre total d'observations à labéliser. On utilisera parfois aussi le taux de reconnaissance par classe.

2.1.3.3 Matrice de confusion

Pour évaluer les performances d'un classifieur, on construira sa matrice de confusion qui permet de visualiser les classes que l'algorithme reconnaît correctement ainsi que les classes confusives. C'est une matrice de taille $l \times l$ où la case (i, j) représente le nombre d'observations de la classe i que le classifieur a prédit comme la classe j.

Prenons, par exemple, un dataset d'évaluation contenant 100 émotions : 33 observations de joie, 33 de colère et 34 de surprise. Supposons qu'un classifieur pré-entraîné a prédit qu'il y avait 35 joie, 31 colère et 34 surprise. Parmi les labels prédits comme joie, 28 étaient corrects. Les 31 émotions prédites comme la colère sont toutes correctes. Enfin, parmi les 34 émotions prédites comme la surprise, 28 sont réellement de la surprise, 5 sont en fait de la joie, et 1 observation a été confondue avec de la colère. On construit alors la matrice de confusion comme dans la table 2.1.

labels	joie	colère	surprise	*total vérité terrain*
joie	28	0	5	33
colère	1	31	1	33
surprise	6	0	28	34
total prédictions	35	31	34	100

TABLE 2.1 – Exemple de matrice de confusion d'un modèle faisant des erreurs de classification

2.2 Algorithmes usuels

Cette section n'a pas vocation à présenter de façon exhaustive tous les algorithmes d'apprentissage supervisé. Nous nous concentrons sur les algorithmes qui seront utilisés, ou cités, dans ce mémoire.

2.2.1 Arbre de décision

Cette section présente le classifieur de type **arbre de décision** (*decision tree*). Nous détaillons sa construction récursive sur un ensemble d'apprentissage et son fonctionnement en inférence. Comprendre son principe est essentiel pour saisir celui de la forêt aléatoire (voir Sec.2.2.2).

Un arbre de décision est un ensemble de noeuds n, organisés dans un graphe acyclique, connexe et

orienté. On distingue :

— le premier noeud, appelé racine ;

— les noeuds terminaux appelés feuilles ;

— les noeuds internes situés entre la racine et les feuilles ;

La racine et les noeuds internes contiennent des règles qui permettent d'orienter les observations à classer. Les feuilles sont utilisées pour prédire la classe de celles-ci.

2.2.1.1 Apprentissage

La construction d'un arbre de décision suit une induction descendante (*top-down*). Elle commence au niveau de la racine, qui reçoit le dataset d'apprentissage complet (X, Y). L'arbre de décision se construit récursivement selon la procédure décrite ci-dessous.

Supposons que l'arbre ait été construit jusqu'au noeud n, ce dernier a alors reçu un sous-ensemble du dataset d'apprentissage, noté $(X^{(n)}, Y^{(n)})$. Cet ensemble est séparé en deux sous-parties selon une fonction de séparation (*split*). Un noeud enfant est alors créé pour recevoir la sous-partie correspondante. Ce split est choisi de façon à minimiser l'hétérogénéité H des données (au sens de leur classe) dans les noeuds enfants. Cette hétérogénéité est minimale pour un noeud **pur** (ne contenant que des données d'une même classe) et maximale pour un noeud **uniforme** (qui contient autant de données de chaque classe). Nous décrivons par la suite les différents algorithmes et mesures d'hétérogénéité proposés dans la littérature.

Dans le cas binaire, et pour des variables quantitatives uniquement, l'algorithmique - glouton- va chercher la feature discriminant le mieux les données présentes dans le noeud. Pour ce faire, il explore toutes les paires (f, θ) candidates où f est une feature et θ un seuil qui permet de séparer l'ensemble des observations en deux sous-ensembles selon que la valeur de la feature f est inférieure ou supérieure au seuil θ. On mesure l'hétérogénéité de chaque partition et l'algorithme sélectionne la paire (f, θ) optimale ; les deux sous-ensembles sont affectés respectivement aux fils gauche et droit du noeud.

La construction récursive s'arrête si une condition d'arrêt est atteinte. La liste, non exhaustive, suivante présente quelques conditions d'arrêt généralement utilisées, qui doivent être spécifiées par l'utilisateur :

— toutes les observations de $(X^{(n)}, Y^{(n)})$ appartiennent à la même classe ; le noeud est dit pur et

transformé en feuille ;

— un nombre minimal de données est nécessaire dans n pour réaliser le split n_{stop} ;

— l'arbre ne doit pas dépasser une certaine profondeur.

Dans chaque noeud n où un split est nécessaire, le gain d'information I est calculé pour chaque split candidat ; un split candidat met en jeu le noeud courant n "parent" et ses deux noeuds fils "gauche" et "droit". Ce critère de gain est à maximiser et le split associé au gain d'information le plus grand est choisi. I est calculé en fonction d'une mesure d'hétérogénéité H calculée dans le noeud n "parent", dans le noeud fils gauche et dans le noeud fils droit de la manière suivante :

$$I = H(Y^{(parent)}) - \left(\frac{|Y^{(gauche)}|}{|Y^{(parent)}|} \times H(Y^{(gauche)}) + \frac{|Y^{(droit)}|}{|Y^{(parent)}|} \times H(Y^{(droit)}) \right), \tag{2.5}$$

où $|Y^{(n)}|$ correspond au nombre de données reçues par le noeud n et $H(Y^{(n)})$ à leur hétérogénéité.

Cette mesure d'hétérogénéité H peut se calculer de différentes manières ; Gini (H_G) et l'entropie de Shannon (H_S) sont les plus populaires :

— L'indice de Gini (H_G), introduit dans l'algorithme *Classification And Regression Trees* (CART) par Breiman [20] est mesuré par :

$$H_G(Y^{(n)}) = 1 - \sum_{j=1}^{l} P(k_j|Y^{(n)})^2, \tag{2.6}$$

où $P(k_j|Y^{(n)})$ représente la proportion d'éléments k_j dans le vecteur $Y^{(n)}$ (ou encore la proportion d'observations de classe k_j dans le dataset $(X^{(n)}, Y^{(n)})$).

— L'entropie de Shannon (H_S) est un critère introduit avec les algorithmes *Iterative Dichotomiser 3* (ID3) et C4.5 par Quinlan [21][22] et est mesurée par :

$$H_S(Y^{(n)}) = -\sum_{j=1}^{l} P(k_j|Y^{(n)}) \log_2(P(k_j|Y^{(n)})). \tag{2.7}$$

La figure 2.2 illustre les frontières de décision qui sont des coupes orthogonales pour un arbre de décision classique.

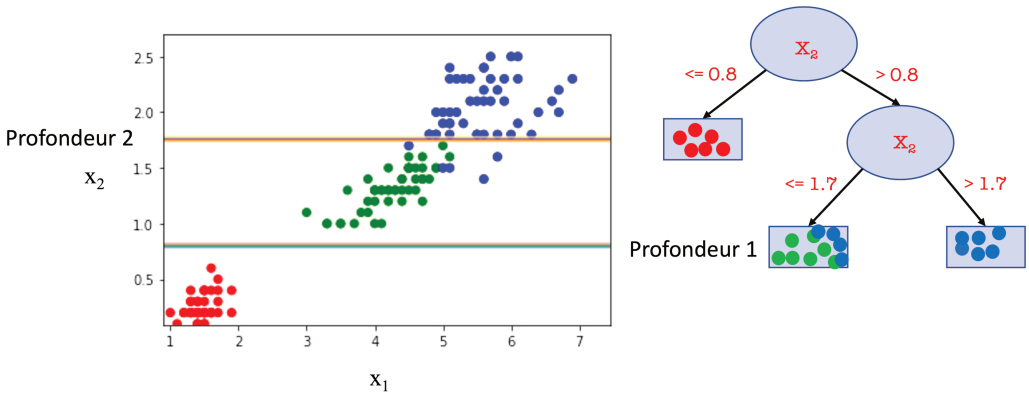

FIGURE 2.2 – Exemple de frontières de décisions pour un arbre de décision classique - les frontières sont des coupes orthogonales.

2.2.1.2 Inférence

Chaque noeud contient une règle qui permet d'orienter l'observation vers le fils gauche, ou le fils droit du noeud. Ainsi, pour classer une nouvelle observation x, on lui fait parcourir l'arbre, depuis la racine jusqu'à une feuille unique ; puis, on lui assigne la classe qui est majoritaire dans cette feuille. Un des principaux avantages de l'arbre de décision face aux algorithmes concurrents est qu'il est facilement interprétable. Le parcours effectué par l'observation dans l'arbre est unique. Il correspond à la conjonction des règles contenues dans les noeuds parcourus.

La figure 2.3 montre un arbre de décision entraîné à reconnaître quatre classes $\mathcal{K} = \{A, B, C, D\}$. Dans chaque feuille, un tableau de taille $\mathcal{K}$ contient le nombre d'occurrences de chaque classe ayant été propagées jusque dans cette feuille lors de la phase d'apprentissage.

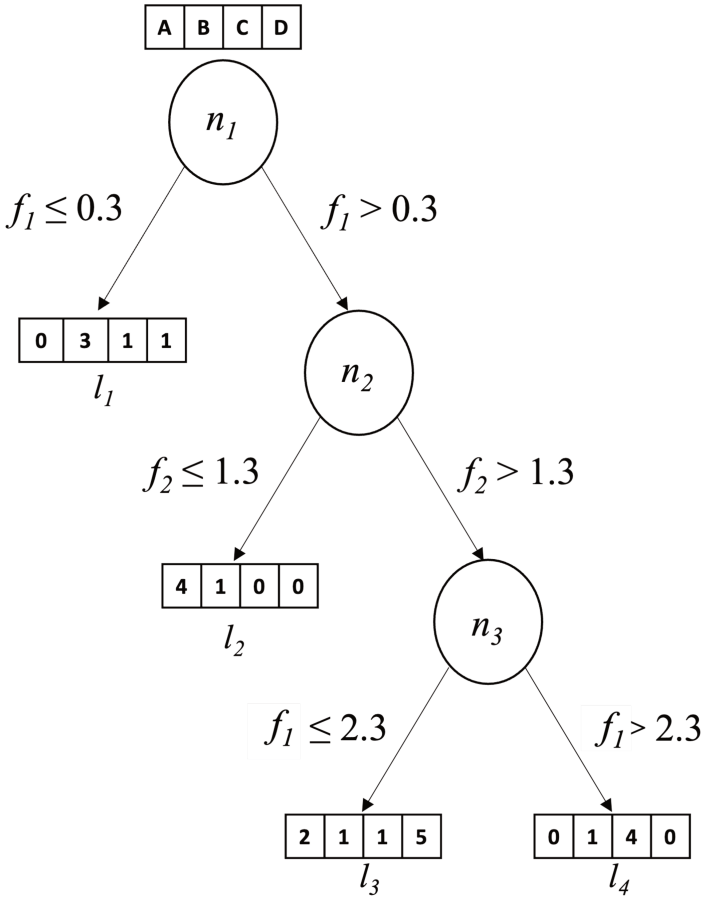

FIGURE 2.3 – Un arbre de décision entraîné à reconnaître les classes A, B, C et D. Les labels majoritaires dans les feuilles l_1, l_2, l_3 et l_4 sont respectivement B, A, D et C.

Soit x une nouvelle observation définie comme $x = [0.6, 1.2]$. Pour classer cette observation, c'est-à-dire lui prédire une classe, l'arbre de l'exemple va la propager depuis la racine n_1. Puisque $f_1 = 0.6$, x est d'abord propagé dans le noeud fils droit n_2 car $0.6 > 0.3$. Comme $f_2 = 1.2 < 1.3$, x est propagé dans la feuille l_2. La classe affectée à l'observation x est alors la classe majoritaire de la feuille, à savoir la classe A.

La prédiction correspond donc à la conjonction de règles suivante :

$$SI\ (f_1 > 0.3)\ ET\ (f_2 \leq 1.3)\ ALORS\ \hat{y} = A.$$

L'apprentissage de l'arbre est très rapide. Plus l'arbre est profond, plus ses performances s'améliorent, se traduisant par un biais de plus en plus faible. Par conséquent, l'arbre aura tendance à apprendre par coeur les données d'apprentissage si on le laisse croître jusqu'à sa profondeur maximale.

Sa variance sera donc élevée et la capacité de généralisation faible. Il est possible d'éviter cette situation en imposant qu'un nombre minimum d'observations soient présentes dans un noeud (n_{stop}) pour pouvoir créer ses deux enfants. Une autre solution est d'entraîner un ensemble d'arbres de décision, appelé, logiquement, forêt.

2.2.2 Forêt aléatoire

La forêt aléatoire (*Random Forest - RF*) est une méthode d'apprentissage ensembliste proposée par Breiman [23]. Elle est composée d'un ensemble d'arbres de décision que Breiman recommande de créer avec une profondeur maximale et de ne pas élaguer, afin de minimiser leur biais. Construire une forêt contenant plusieurs arbres de décision entraînés sur un même jeu de données n'est pas intéressant. L'algorithme d'apprentissage de l'arbre étant déterministe, cela revient à cloner le même arbre. Les clones ayant tous le même comportement, leur combinaison n'a aucun intérêt. Pour qu'un ensemble de classifieurs soit efficace, il faut que ces derniers aient des comportement différents.

C'est pourquoi Breiman introduit deux sources d'aléa dans l'apprentissage de la forêt, qui vont générer des arbres avec des comportements variés : le **b**ootstrap **agg**regat**ing** (bagging) et la **R**andom **F**eature **S**election (RFS).

2.2.2.1 Bagging et Out-Of-Bag

Le bagging consiste à affecter un dataset différent à chaque arbre de décision de la forêt. Chaque dataset est construit de la manière suivante. Un tirage aléatoire, avec remise, d'une observation (et de son label associé) a lieu n_X fois dans le dataset original (X, Y) réservé pour la phase d'apprentissage. Le dataset obtenu a donc les mêmes dimensions que X, mais est différent car il peut contenir des doublons ou, au contraire, ne pas contenir certaines observations. En procédant ainsi pour chaque arbre, on s'assure qu'ils sont entraînés sur un dataset sensiblement différent, et donc finissent par ne pas être construits de manière identique. De ce fait, chaque arbre de décision aura tendance à s'adapter de manière excessive à l'ensemble de données qui lui est fourni, car construit avec une profondeur maximale. Comme il ne s'agit pas des mêmes datasets d'apprentissage, le bruit présent pour un arbre ne sera pas le même que pour un autre. Toutefois, la majorité du signal sera généralement similaire d'un arbre à l'autre. Ainsi, en agrégeant les décisions de tous les arbres, le bruit aura tendance à diminuer. Finalement, la variance de l'ensemble sera faible, tout comme le biais des classifieurs individuels.

Étant donné qu'un arbre ne s'entraîne pas sur toutes les données du dataset original, il est possible d'évaluer les performances de l'arbre sur les données n'ayant pas été sélectionnées. On parle de Out-Of-Bag pour qualifier ces données, qui peuvent servir d'ensemble de validation pour optimiser les hyperparamètres. Le taux de reconnaissance obtenu sur ce dataset est appelé **score oob**. À l'échelle d'une forêt, le score *oob* est la moyenne des scores *oob* de chaque arbre.

2.2.2.2 Random Feature Selection

Certaines features peuvent être plus importantes que d'autres dans l'ensemble de données traité, mais il est possible que ce pattern n'apparaisse pas dans la population générale. Il en résultera alors une plus grande difficulté de généralisation, ainsi qu'une plus grande corrélation entre les arbres, car ces features seront toujours choisies par la fonction de split. Ainsi, pour décorréler davantage les arbres, dans chaque nœud, au lieu de considérer toutes les features, seul un sous-ensemble de F choisi aléatoirement est considéré, on parle de *Random Feature Selection* (RFS). La taille n_F de celui-ci est un hyperparamètre fixé par l'utilisateur.

2.2.2.3 Capacité de généralisation

Un arbre de décision entièrement développé (construit avec une profondeur maximale) est, d'une part, considéré complexe car il prend en compte toutes les features du dataset qu'il a reçues. D'autre part, il considère toutes les informations comme importantes et recherche de nombreux patterns là où il n'y en a parfois aucun. Cela signifie qu'il apprendra à la fois du signal et du bruit. Ce modèle aura un faible biais car il aura appris en profondeur le dataset. Si nous appliquons le bagging sur ce dataset, pour générer un autre modèle, celui-ci finira par apprendre d'une manière complexe également. Mais, comme les jeux de données sont différents, les arbres auront des variances élevées car leurs prédictions seront plus variées en raison de ces changements. Cependant, le biais sera faible car, individuellement, ils ont appris du bruit, mais, surtout, du signal. Ainsi, si nous agrégeons leurs prédictions via un scrutin à vote majoritaire, la variance diminuera car, les arbres ont bien appris le signal de manière générale, et n'ont pas appris le même bruit. Cette façon d'ajouter l'aléatoire dans les deux dimensions de X donne à la RF un meilleur pouvoir de généralisation, capable de minimiser la variance, tout en maintenant le biais faible. Cette propriété a contribué à populariser l'algorithme qui a été appliqué à de nombreux domaines, en particulier, la reconnaissance des expressions faciales.

2.2.3 Calcul des probabilités et prédictions

Cette section décrit différents estimateurs des probabilités *a posteriori* des classes dans une forêt aléatoire.

Soit t un arbre de décision, T une forêt composée de $|T|$ arbres et $x \in \mathbb{R}^m$ une observation. En phase d'inférence, l'observation x, parcourt chaque arbre de la racine à une feuille unique. Cette dernière contient les occurrences des classes des observations du dataset d'apprentissage l'ayant atteintes. Chaque arbre produit donc un vecteur de probabilités $\phi_t(x)$. Il convient de définir un opérateur pour agréger ces $|T|$ vecteurs et attribuer une classe à x.

2.2.3.1 Méthode classique

Les probabilités des classes associées à une observation x sont calculées comme la moyenne des probabilités des classes prédites par les arbres de la forêt :

$$\Phi_A(x) = \frac{1}{|T|} \sum_{t \in T} \phi_t(x). \tag{2.8}$$

2.2.3.2 Méthode proposée

On propose une méthode de calcul différente dans nos expérimentations. On considère, pour une observation x et un arbre de décision t, le vecteur $S^t(x) = [S^t(k_1|x), ..., S^t(k_l|x)]$, où $S^t(k_i|x)$ correspond au nombre d'occurrences de la classe k_i dans chaque feuille de t. Le vecteur Φ_B est alors calculé de la manière suivante :

$$\Phi_B(x) = \frac{1}{R(x)} \sum_{t \in T} S^t(x) \text{ avec } R(x) = \sum_{t \in T} \sum_{i=1}^{l} S^t(k_i|x). \tag{2.9}$$

La méthode classique calcule la probabilité **localement** au niveau de chaque feuille (2.8). Par conséquent, l'information relative à $S^t(k_i|x)$ est perdue. Cela revient à attribuer le même poids à chaque feuille de chaque arbre de décision de la forêt, sans tenir compte des effectifs présents dans la feuille. Or, il a été montré dans la littérature que Φ_A peut être un mauvais estimateur des probabilités de classes [24] [25]. C'est pourquoi nous avons proposé cette nouvelle méthode qui, au contraire, agrège d'abord les occurrences de toutes les feuilles, puis calcule la probabilité **globale** au niveau de la forêt

(2.9). De cette manière, une feuille avec un nombre plus faible d'occurrences aura moins d'impact dans le calcul final des probabilités $\Phi_B(x)$.

L'exemple suivant illustre les différences qui peuvent apparaître lors de l'application des méthodes définies précédemment. Considérons une forêt simple composée de deux arbres pré-entraînés sur les classes $\mathcal{K} = \{A, B, C, D\}$, comme présenté dans la figure 2.4. Supposons maintenant une observation x à classer qui est propagée dans la feuille l_1 du premier arbre, et dans la feuille l_4 du second.

Avec la méthode classique, on calcule pour chaque feuille, la distribution locale des classes :

$$\begin{aligned}\Phi_A(x) &= \frac{\left(\left[\frac{0}{5}, \frac{3}{5}, \frac{1}{5}, \frac{1}{5}\right] + \left[\frac{2}{17}, \frac{8}{17}, \frac{6}{17}, \frac{1}{17}\right]\right)}{2} \\ &= \frac{([0, 0.6, 0.2, 0.2] + [0.12, 0.47, 0.35, 0.06])}{2} \\ &= [0.06, 0.54, 0.28, 0.13].\end{aligned}$$

D'après (2.3), la classe ainsi prédite est B, avec une probabilité de 0.54.

Avec la méthode proposée, nous agrégeons les occurrences de classe des feuilles au niveau de la forêt comme dans 2.9. Nous obtenons ainsi :

$$\begin{aligned}\Phi_B(x) &= \frac{([0, 3, 1, 1] + [2, 8, 6, 1])}{(5 + 17)} \\ &= \frac{([2, 11, 7, 2])}{22} \\ &= [0.09, 0.50, 0.32, 0.09].\end{aligned}$$

D'après (2.3), la classe ainsi prédite est B, avec une probabilité de 0.50.

L'écart entre les probabilités d'appartenir aux classes B et C est légèrement plus faible pour Φ_B que pour Φ_A. Cela est dû au fait que les effectifs présents dans les feuilles ne sont pas les mêmes. Ainsi, les feuilles avec des effectifs élevés auront plus de poids dans le calcul (ici il y a 17 données dans l_4 contre 5 dans l_1).

On remarque que les deux méthodes ne prédisent pas nécessairement la même classe.

Dans la suite de nos expérimentations, cette dernière méthode sera utilisée.

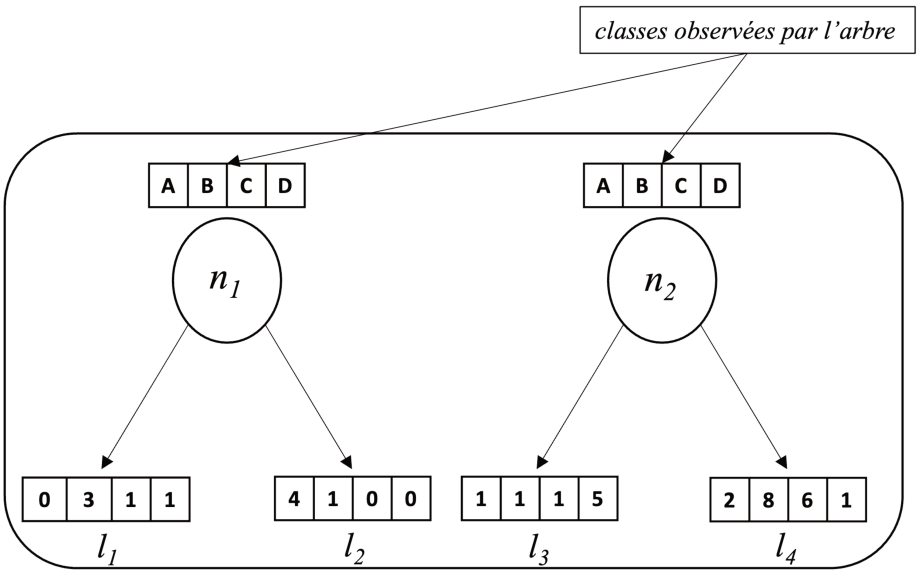

FIGURE 2.4 – Illustration des distributions de classes au sein de feuilles d'une forêt

2.2.4 Modèle *Nearest Class Mean*

On présente ici une autre méthode de classification : le modèle *Nearest Class Mean* (NCM). À partir de notre dataset d'apprentissage, on définit le centroïde c_k de chaque classe $k \in \mathcal{K}$:

$$c_k = \frac{1}{|X_k|} \sum_{i=1}^{|X_k|} x_k^i \tag{2.10}$$

où $X_k = \{x_k^i, 1 \leq i \leq |X_k|\}$ correspond à l'ensemble des observations appartenant à la classe k. $\mathcal{C} = \{c_1, ..., c_l\}$ est l'ensemble des centroïdes par classe, et $|\mathcal{C}| = |\mathcal{K}|$.

La classification d'une observation x par un modèle NCM se fait en recherchant le centroïde le plus proche de x :

$$k^*(x) = \underset{k \in \mathcal{K}}{\operatorname{argmin}}\ d(x, c_k) \tag{2.11}$$

ou d est une mesure de distance.

2.2.5 Forêt NCM (*NCMF*)

La forêt NCM, proposée par Ristin *et al* dans [19], combine les deux classifieurs (RF et NCMF) décrits précédemment. Elle conserve beaucoup de caractéristiques de la forêt "classique" : ensemble

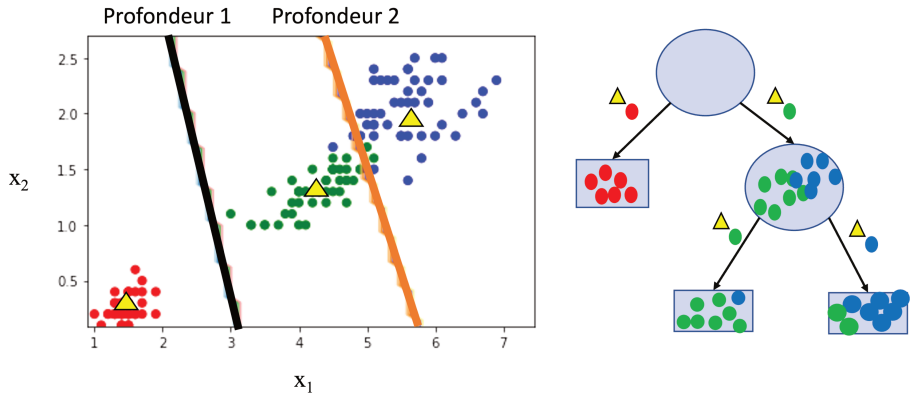

FIGURE 2.5 – Exemple de frontières de décisions pour un arbre de décision NCM.

d'arbres entraînés sur des échantillons boostrap de la base d'apprentissage et tirage aléatoire d'un sous-ensemble de features lors de la construction d'un noeud.

2.2.5.1 Apprentissage

La principale différence est que chaque noeud contient un classifieur NCM binaire. L'apprentissage d'un noeud commence par le tirage aléatoire d'un sous-ensemble de classes (*Random Class Selection* - RCS) qui définit les classes que le noeud va devoir séparer. Pour chaque paire de classes $\{k_i, k_j\}$ du sous-ensemble, on va entraîner un classifieur NCM, autrement dit, estimer les centroïdes des classes k_i et k_j à partir des observations de ces classes présentes dans le noeud. L'ensemble des données du noeud sera séparé, en utilisant la règle du plus proche centroïde, en deux sous-ensembles. Enfin, l'algorithme sélectionne la paire de centroïdes qui maximise le gain d'information I ; la séparation réalisée est alors considérée comme optimale. Les sous-ensembles sont transmis aux fils gauche et droit du noeud et la construction récursive de l'arbre se poursuit. Les critères d'arrêt sont les mêmes que ceux de la forêt aléatoire "classique". La figure 2.5 illustre les frontières de décision pour un arbre de décision NCM.

2.2.5.2 Inférence

L'observation x parcourt chaque arbre, depuis sa racine jusqu'à une feuille unique. Dans chaque noeud, le classifieur NCM oriente x à gauche ou à droite, en fonction de sa distance aux deux centroïdes c_i et c_j comme le montre la figure 2.6 . Arrivée dans la feuille, on assigne à x la classe majoritaire (parmi les classes présentes dans la feuille). Enfin, un vote majoritaire est effectué sur l'ensemble des

prédictions des arbres, pour classer l'observation x.

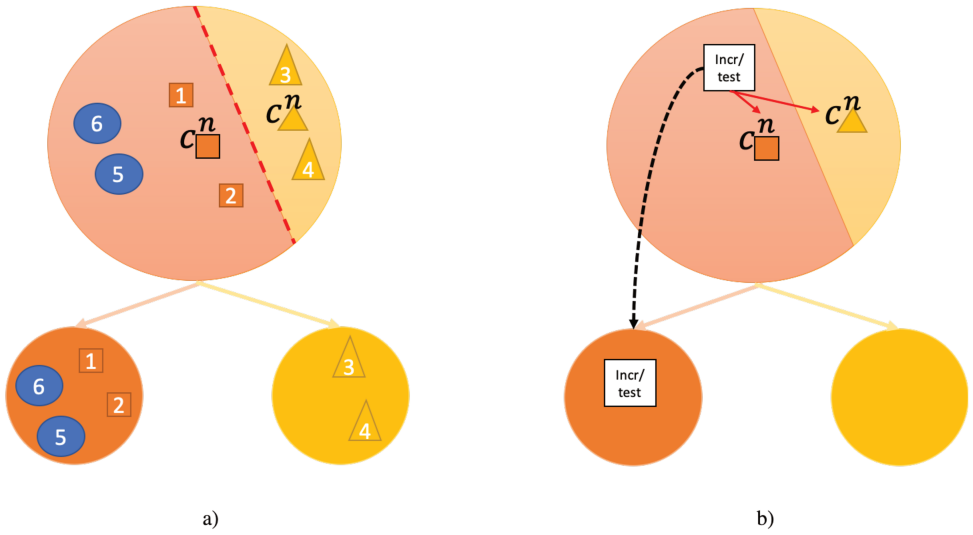

FIGURE 2.6 – Split dans un noeud d'une NCMF - les observations sont dirigées vers le noeud fils associé au centroïde le plus proche, que ce soit lors de la phase d'apprentissage initiale (a), incrémentale ou d'évaluation (b).

2.2.5.3 Intérêt du modèle

On reproche souvent aux arbres (et forêts) classiques le faible pouvoir discriminant de leurs noeuds. De fait, la séparation se faisant sur une dimension unique, il devient nécessaire de multiplier les noeuds pour générer des frontières de décision complexes. Le noeud NCM, au contraire, génère une frontière dans un espace de dimension supérieure qui s'appuie sur les distributions conditionnelles locales des deux classes sélectionnées. Son pouvoir discriminant est théoriquement bien plus élevé. Les NCMF se prêtent aussi beaucoup plus facilement à l'apprentissage incrémental comme nous le verrons dans le chapitre 6.

2.2.6 Réseaux de neurones

Cet algorithme faisait initialement partie des méthodes dites "connexionnistes" car elles avaient vocation à imiter le fonctionnement du cortex, constitué de neurones fortement connectés entre eux. Cette section a pour objectif de décrire l'évolution de cet algorithme, depuis le premier neurone "formel" défini dans les années 1940 jusqu'aux réseaux convolutionnels profonds qui constituent actuellement l'état de l'art en termes de performances dans de nombreux domaines. Nous ne rentrerons pas dans

les détails des algorithmes mis en oeuvre qui nécessitent le réglage de nombreux hyper-paramètres.

2.2.6.1 Apprentissage d'un neurone formel

Le premier neurone "formel" ou artificiel copiait de façon sommaire le fonctionnement du neurone naturel. Il réalisait une pondération des "entrées" - correspondant aux features f_i définies plus haut - par des poids "synaptiques" ω_i. On appelait potentiel ν du neurone, la somme pondérée des entrées. Le neurone était "activé" si ce potentiel dépassait un seuil, et restait inactif, sinon. La sortie du neurone $\hat{y}$ pouvait donc prendre deux états dans $\{0, 1\}$ [26].

Le vecteur des poids $W = \{\omega_0, \omega_1, ..., \omega_m\}$ définissait un hyperplan dans l'espace des caractéristiques, susceptible de séparer deux ensembles de données labélisées 0 ou 1, sous réserve qu'ils soient linéairement séparables. L'apprentissage, itératif, répétait les opérations suivantes tant qu'une erreur minimale n'était pas atteinte :

— propagation d'une observation x ;

— calcul de la sortie $\hat{y}$ correspondante ;

— modification des poids dans la direction opposée au gradient de l'erreur $E = ||y - \hat{y}||$.

On utilise encore aujourd'hui cet algorithme appelé descente de gradient stochastique (*Stochastic Gradient Descent* - SGD). À la fin de l'apprentissage, les poids du vecteur W étaient dits optimaux car l'hyperplan qu'ils définissaient séparait parfaitement les données des deux classes.

2.2.6.2 Réseaux de neurones monocouche

Pour traiter les problèmes de classification multi-classe, on a utilisé $\mathcal{K}$ neurones au lieu d'un seul. Chaque neurone est dédié à une classe k. Il doit donc être actif quand apparaît une observation de cette classe, tandis que tous les autres neurones doivent rester inactifs. Ces réseaux étaient qualifiés de mono-couche puisqu'une seule couche connectait les entrées aux sorties via une matrice de poids W de dimensions $m \times \mathcal{K}$. L'apprentissage des poids par l'algorithme SGD optimisait $\mathcal{K}$ hyperplans dont la combinaison séparait au mieux les différentes classes.

2.2.6.3 Réseaux de neurones multicouche

Pour traiter les problèmes non linéairement séparables, il était assez tentant de s'inspirer encore une fois du cortex et d'ajouter une (ou plusieurs) couche(s) de neurones intermédiaires ("cachées") entre celles d'entrée et de sortie. Mathématiquement, si l'on ajoute une seule couche, celle-ci va réaliser une projection non linéaire de l'espace initial de dimension m dans un espace de dimension C où C est le nombre de neurones de la couche.

L'apprentissage doit maintenant optimiser conjointement deux matrice de poids :

— celle d'entrée, W_1, de dimensions $m \times C$

— celle de sortie, W_2, de dimensions $C \times \mathcal{K}$

L'objectif étant de trouver la matrice W_1 optimale qui projette les données dans un espace où elles seraient linéairement séparables et la matrice W_2 optimale qui sépare parfaitement les données.

Étant donné la complexité de la tâche, il a fallu plus d'une décennie pour trouver l'algorithme capable d'entraîner un tel réseau. L'algorithme de rétro-propagation (du gradient) a permis d'optimiser non seulement la matrice de sortie W_2, mais aussi la matrice d'entrée W_1. Il était même - théoriquement - capable d'entraîner un réseau contenant un nombre quelconque de couches.

Toutefois, augmenter le nombre de couches faisait exploser le nombre de poids à optimiser et le risque de sur-apprentissage, en particulier si l'ensemble d'apprentissage était de taille insuffisante. Cela était particulièrement fréquent lorsque les données à traiter étaient des images, constituées d'un grand nombre de pixels.

2.2.6.4 Réseaux de neurones à convolution

Les architectures complètement connectées précédentes n'étaient pas adaptées au traitement d'images. L'utilisation de neurones à connexions locales et à poids partagés, a permis de réduire drastiquement le nombre de paramètres libres. Ces neurones agissaient comme des filtres de convolution [27], capables, après l'apprentissage, de détecter des *patterns* dont la complexité augmentait avec le nombre de couches de convolution. Afin de réduire encore le nombre de poids, des couches de sous-échantillonnage (*pooling*) ont été insérées entre les couches de convolution.

Ces couches d'extraction de caractéristiques étaient suivies de plusieurs couches denses (complètement connectées) entraînés à discriminer les classes, comme dans les anciens réseaux muticouches. Le

modèle complet permettait, pour la première fois, d'apprendre conjointement les caractéristiques optimales et les hyperplans séparant les différentes classes. On parle donc de *end-to-end learning* puisque la phase d'extraction des caractéristiques, jusqu'ici indispensable en analyse d'images, est maintenant réalisée par le réseau.

Des techniques comme le *DropOut* et, plus récemment, la *Batch Normalization* ont permis de mieux régulariser l'apprentissage. Plusieurs architectures profondes (VGG-X, ResNet, DenseNet, pour ne citer qu'elles) ont démontré de remarquables propriétés sur des données benchmarks et sont depuis largement utilisées par la communauté.

Chapitre 3

État de l'art

Contenu

3.1 Reconnaissance des émotions faciales 60
3.1.1 Informatique affective : modélisation des émotions 61
3.1.2 Intérêt pour l'Éducation 65
3.2 Apprentissage incrémental 66
3.2.1 Réseaux de neurones et oubli catastrophique 68
3.2.2 Méthodes basées sur les forêts aléatoires 70
3.3 Apprentissage semi-supervisé 72
3.3.1 Contexte 72
3.3.2 Self-training 73
3.3.3 Co-training 74
3.4 Discussion 75

3.1 Reconnaissance des émotions faciales

L'expression faciale est considérée comme le principal mode de reconnaissance des émotions humaines. La classification basée sur les informations visuelles n'est toutefois pas le seul indicateur d'émotion. D'autres facteurs contribuent également à la reconnaissance de l'état émotionnel d'une personne, comme la voix, la prosodie, le langage corporel ou la direction du regard. Néanmoins, dans ce travail, nous nous concentrons sur les expressions faciales. Grâce à elles, il est possible, d'une part, d'identifier des indices sur l'état émotionnel d'un sujet et, d'autre part, de fournir d'autres types d'informations, notamment sur le comportement, la personnalité d'un individu, ou encore l'intensité de la douleur qu'il ressent. Les expressions faciales sont donc l'un des canaux les plus informatifs de la communication interpersonnelle : elle constitue un moyen de transmettre aux autres, consciemment ou non, notre état émotionnel.

La conception d'algorithmes fiables de reconnaissance des expressions faciales (*Facial Emotion Recognition* - FER) est donc cruciale pour améliorer les systèmes informatiques interactifs. Ils peuvent être très utiles dans de nombreux domaines tels que la sécurité, la robotique ou le marketing, pour n'en citer que quelques-uns. Des exemples d'applications pourraient être de créer une connexion émotionnelle avec des clients, donner des indicateurs pour des études psychologiques, observer des députés à l'assemblée lors de discussions sur des sujets critiques, intégrer ces indicateurs dans des applications de réalité augmentée [28], analyser l'état de fatigue d'un conducteur de véhicule [29], simuler des entretiens d'embauche [30], aider des enfants souffrant de certaines formes d'autisme à comprendre et exprimer des états émotionnels afin de communiquer avec leur entourage [31], intégrer des indicateurs dans des systèmes de tutorat intelligent pour aider les étudiants durant leur apprentissage [32].

Bien que des progrès importants aient été réalisés, de nombreux défis subsistent toujours dans ce domaine. Différents facteurs comme l'orientation de la tête, l'illumination, les erreurs de recalage, les occultations ou le *biais d'identité* [6] impactent toujours fortement les performances.

Le biais d'identité nous intéresse tout particulièrement dans le cadre de nos travaux. Une émotion n'est pas nécessairement exprimée de manière identique par tous les être humains. Cette variabilité comportementale, mais aussi morphologique, entre les individus, implique que certains systèmes *génériques* ne seront pas capables de reconnaître les émotions de plusieurs individus. Une possibilité de remédier à ce problème, pourrait être de *personnaliser* un modèle générique sur un sujet ou un

ensemble de sujets (voir Sec.1.3).

La reconnaissance automatique des émotions faciales a suscité un grand intérêt dans divers domaines, en particulier pour la reconnaissance des unités d'action (AUs) et des états affectifs. Bien que des progrès notables aient été réalisés, plusieurs questions demeurent quant aux informations déterminantes pour l'interprétation des expressions faciales et à la manière de les encoder. Les systèmes FER visent le plus souvent à reconnaître l'apparence des actions faciales, ou les émotions véhiculées par ces actions [6].

On abordera, tout d'abord, la modélisation des émotions, et enfin l'intérêt de leur analyse pour l'éducation.

3.1.1 Informatique affective : modélisation des émotions

3.1.1.1 Le système catégoriel subjectif

Le modèle catégoriel des émotions prototypiques (dites également basiques), qui inclut la joie, la tristesse, la peur, la colère, le dégoût et la surprise est sans doute l'une des approches les plus répandues pour décrire les expressions faciales. La description des émotions basiques a été spécialement soutenue par les études inter-culturelles conduites par Ekman en 1971 [33]. Elles sont qualifiées de basiques car elles sont simples et reconnues universellement y compris par de très jeunes enfants, et ce, indépendamment de leur culture.

Ce mode de représentation a l'avantage de décrire les états émotionnels observés dans la vie de tous les jours avec des labels faciles à comprendre pour la population car il correspond au vocabulaire et à l'expérience des individus. Étant donné que labéliser des bases de données d'émotions est une tâche fastidieuse et coûteuse, on trouve alors plus facilement dans la littérature des bases de données comportant les labels des expressions prototypiques.

Par conséquent, et favorisé sans doute par la simplicité de cette représentation discrète, un grand nombre d'études en reconnaissance automatique de l'état émotionnel se concentre sur ces émotions basiques incluant en plus la classe neutre [6]. Plus tard dans ses travaux [34], Ekman a ajouté aux expressions prototypiques initiales, les émotions suivantes : mépris, gêne, culpabilité, et honte. Cependant, les données labélisées de ces expressions sont plus rares à trouver dans des datasets. Bien que les émotions prototypiques fournissent une référence émotionnelle, elles sont limitées et ne couvrent

qu'une petite partie de nos manifestations émotionnelles quotidiennes. En effet, nos comportements quotidiens ne peuvent pas être traduits seulement en termes d'émotions prototypiques, dont l'apparition est finalement moins fréquente. Les visages des individus manifestent plutôt une combinaison de ces émotions [35]. La détection d'une unique émotion basique sur un affect non verbal apparaît donc peu réaliste.

Néanmoins, il est impossible de catégoriser tous les affects non-verbaux en tant que combinaisons d'émotions basiques ; en effet, des états comme le stress, la frustration ou l'ennui ne constituent pas un mélange d'émotions basiques [36].

De plus, on notera qu'un des principaux inconvénients de cette approche est la subjectivité de l'annotation et parfois, dans le cas des émotions surjouées, l'incapacité pour un sujet de *produire* l'émotion demandée.

3.1.1.2 Le système dimensionnel subjectif

Une alternative à la description par catégories de l'affect humain est la représentation dimensionnelle [37] ; un état affectif est caractérisé par un petit nombre de dimensions décrites par des variables quantitatives continues plutôt que par un petit nombre d'émotions discrètes. De ce fait, une expression spécifique, par exemple la colère, peut être décrite par sa position dans un espace de faible dimension. Ces dimensions pour n'en citer que quelques unes, incluent la valence, l'éveil, le contrôle, la force ou encore la puissance. Le modèle dimensionnel le plus populaire est la représentation en deux dimensions *"valence/arousal"* proposée par Russel [2] (Figure 3.1) pour décrire les principales émotions. La valence mesure ce qu'un humain ressent, du positif au négatif. L'arousal mesure si l'activité physiologique est plus ou moins intense. Dans un tel diagramme, la joie est alors représentée comme ayant une valence positive et un arousal élevé, à la différence de la tristesse, qui est définie comme ayant une valence négative et un faible arousal. Contrairement à la représentation catégorielle, la représentation dimensionnelle permet aux évaluateurs de labéliser une plus grande gamme d'émotions. Cependant, des émotions projetées sur seulement deux dimensions peuvent devenir difficiles à distinguer (par exemple la peur et la colère sont toutes deux représentées comme valence négative et fort arousal).

Enfin, le processus d'annotation de données est moins intuitif que pour la représentation catégorielle mais est aussi très subjectif. Les codeurs doivent donc être très entraînés pour la tâche d'annotation ce qui limite par conséquent la disponibilité des données. L'accord entre plusieurs annotateurs est un

concept simple lorsqu'il s'agit de labels discrets : nous pouvons dire que deux annotateurs sont d'accord s'ils choisissent le même label. Pour les annotations continues, ce concept devient moins évident et repose alors sur des mesures quantitatives, telle que la corrélation inter-annotateurs [38].

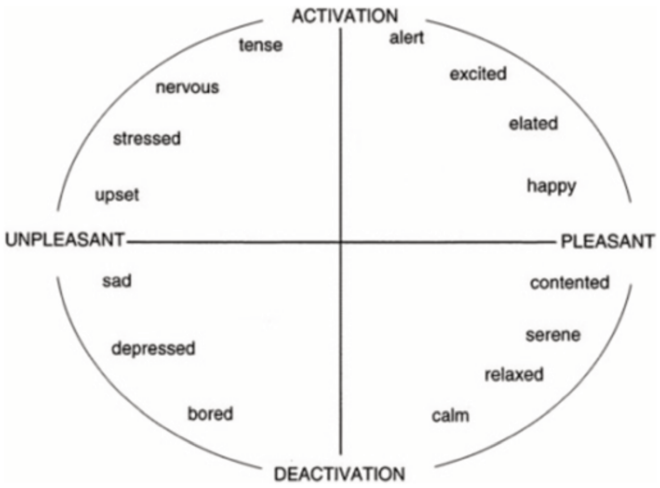

FIGURE 3.1 – Représentation graphique du modèle des émotions dans le circomplexe de Russell [2] Dimension horizontale : valence - Dimension verticale : arousal

3.1.1.3 Le système FACS objectif

Le modèle *Facial Action Coding System* (FACS) est une méthode de description des mouvements du visage développée par les psychologues Ekman et Friesen vers la fin des années 70 [39]. Le système, par opposition à l'approche catégorielle, est complet et objectif. En effet, il se base sur l'activité anatomique pour encoder des expressions. Le FACS décompose les effets visibles de l'activation des muscles faciaux en unités d'action faciales (*Action Units* - AUs). Chaque AU est liée à un ou plusieurs muscles faciaux. Ce système décrit l'activité faciale sur la base de 44 unités d'action uniques (Figure 3.2), mais il y a également d'autres AUs permettant de décrire l'orientation de la tête, la direction du regard, et des actions diverses telles que le déplacement de la mâchoire inférieure vers l'avant par exemple. Au niveau temporel, une unité d'action faciale est typiquement modélisée par quatre segments consécutifs : *neutre*, *onset*, *apex*, et *offset* [40]. Parmi eux, le neutre est la phase sans signe d'activité musculaire ; l'apex est la phase où l'intensité est maximale.

Puisque n'importe quelle expression faciale résulte de l'activation d'un ensemble de muscles faciaux, toute expression faciale peut être ainsi intégralement décrite de manière objective comme une combi-

naison d'AUs. Le FACS ne distingue que les actions faciales et ne donne aucune information sur les émotions. Les combinaisons d'AUs peuvent être utilisées pour la reconnaissance des émotions basiques d'après les règles de Emotional FACS (EMFACS) [41]. Le principal inconvénient de l'approche de codage FACS est que l'annotation prend du temps. Par ailleurs, il faut payer le prix fort et consacrer des mois de formation pour prétendre à la certification de *Facs codeur*[1]. De plus, certaines actions faciales (dans le modèle FACS) nécessitent beaucoup d'entraînement aussi bien pour les codeurs humains que pour les algorithmes [42], ce qui limite possiblement la quantité de données disponibles. Cependant, grâce à un effort de la communauté de la recherche, il existe un certain nombre de bases de données annotées FACS [43].

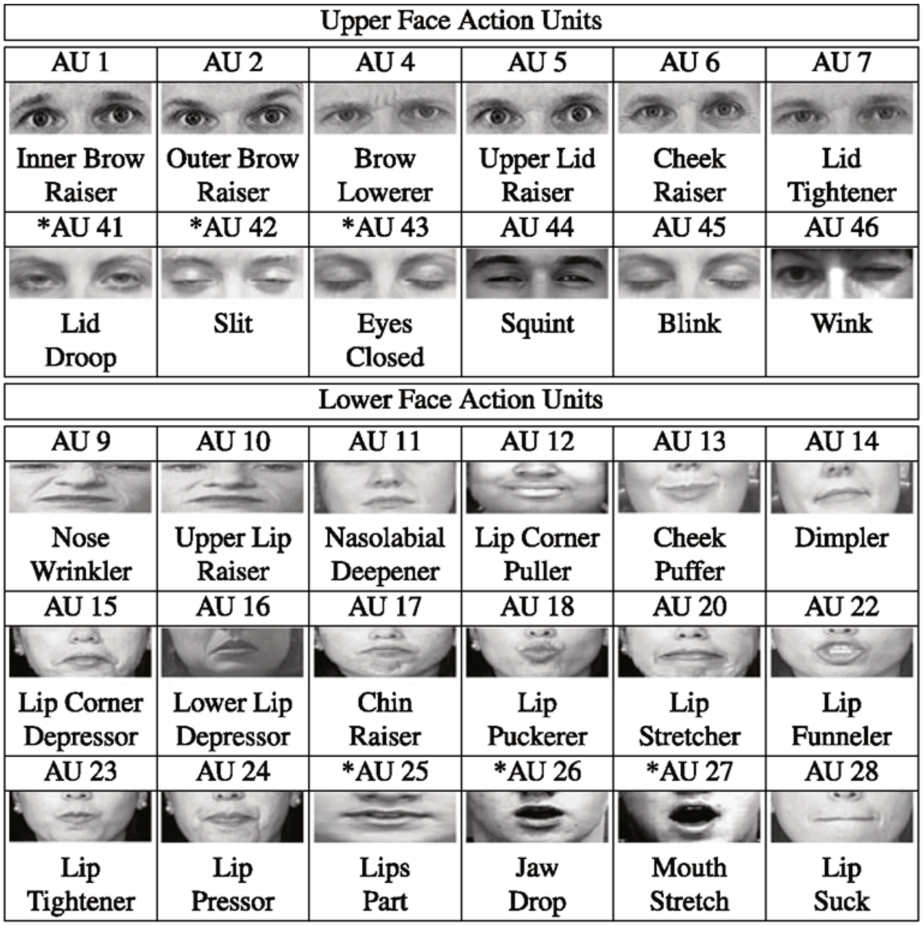

FIGURE 3.2 – Représentation d'une partie des unités d'actions faciales de FACS

1. https ://www.erikarosenberg.com/facs-registration

3.1.2 Intérêt pour l'Éducation

Les émotions sont au cœur du triangle didactique qui relie les enseignants, les élèves et la connaissance. Les stimuli émotionnels peuvent être source de perturbation ou de motivation et affecter l'expérience d'apprentissage. Il est impossible pour un enseignant de se concentrer sur chaque élève et d'adapter son discours afin de maintenir l'élève dans sa zone proximale de développement [44] et ainsi éviter son éventuel décrochage. C'est d'autant plus difficile dans un contexte pandémique où l'enseignement se fait le plus souvent à distance et s'apparente à des cours en ligne ouverts et massifs ($MOOC$). La reconnaissance automatique des émotions pourrait donc contribuer à améliorer ces situations.

L'apprentissage est le résultat d'une interaction complexe de structures et de processus, à la fois internes et externes à l'apprenant, et comportant des aspects cognitifs et affectifs. Une humeur positive aura ainsi tendance à susciter une plus grande créativité et flexibilité dans la résolution de problèmes, ainsi qu'à plus d'efficacité et de rigueur dans la prise de décision [45]. Toutefois, les influences sur la cognition ne se limitent pas qu'à une simple humeur positive, qui n'est pas nécessairement la meilleure pour tous les types de réflexion. En effet, certains états affectifs favorisent certains types de réflexion. Il est de plus en plus reconnu dans l'éducation que l'intérêt et la participation active sont des facteurs importants dans le processus d'un bon apprentissage, et qu'au contraire, des troubles émotionnels comme l'anxiété, la colère ou la dépression peuvent interférer négativement avec celui-ci [46]. La prise de conscience de son propre affect, tel que la frustration par exemple, peut être utile pour aider à gérer cet état de manière productive. Les systèmes qui mesurent les informations affectives peuvent aider, de manière utile et respectueuse, à cette prise de conscience. Par la construction par exemple d'un miroir affectif, l'apprenant est encouragé à réfléchir à la manière dont son état est en train d'influencer son expérience d'apprentissage. La synthèse des expressions émotionnelles par un tuteur intelligent peut permettre d'impliquer l'apprenant dans une expérience plus humaine et naturelle. Si le participant éprouve un état qui a été corrélé négativement avec l'apprentissage, par exemple l'ennui, le tuteur doit alors l'engager dans une activité qui augmente son intérêt et son éveil cognitif. Un tuteur empathique serait efficace pour atténuer la frustration. Le domaine de la reconnaissance d'émotions pour l'éducation commence à saisir ces caractéristiques clés [47] [48].

Par conséquent, on voit grandir l'intérêt de construire des systèmes de tutorat intelligents (tuteurs animés, ordinateurs robotisés, etc.) où il est possible d'adapter l'expérience d'apprentissage en fonction

de tout indice lié à l'affect [49]. Ces technologies pourraient permettre d'identifier les états émotionnels qui sont les plus importants pour l'apprentissage et comment ils changent avec différents types de pédagogie. Picard et al. [45] estiment que de tels modèles peuvent fournir de nouvelles perspectives sur de nombreux mécanismes cognitifs et affectifs et aider à la conception d'outils et d'environnements d'apprentissage.

En d'autres termes, la reconnaissance automatique des émotions peut être utile pour améliorer les résultats de l'apprentissage en fournissant des processus éducatifs personnalisés et adaptatifs en fonction des émotions des étudiants, ainsi que d'autres indicateurs de performance liés à la productivité et aux compétences cognitives.

Cependant, plusieurs études s'accordent à dire que les émotions basiques d'Ekman, bien que très populaires, sont en fin de compte limitées pour représenter le vaste ensemble des émotions que l'on peut rencontrer dans la vie de tous les jours [6] [50]. En particulier, dans un contexte d'apprentissage, elles apparaîtraient bien moins que les expressions non prototypiques telles que l'engagement, l'ennui, la confusion et la frustration [51]. Parmi les émotions que l'on peut identifier chez un étudiant pendant un apprentissage via système tutoriel intelligent (STI), on compte l'ennui, la confusion, le plaisir et l'engagement (profonde concentration), des états affectifs pouvant être bénéfiques ou perturbateurs pour l'apprentissage [52].

3.2 Apprentissage incrémental

Dans de nombreuses applications, l'acquisition d'un dataset complet et représentatif est coûteuse et prend du temps ; par conséquent, il n'est pas rare que les datasets ne soient disponibles que par petits lots, au fil du temps. Dans ce cas, si nous avions entraîné un classifieur sur un premier sous-dataset, la question se pose quant à la manière d'exploiter les données d'un nouveau sous-dataset. Une approche naïve consiste à rejeter le modèle existant et à le ré-entraîner, à partir de zéro, en utilisant toutes les données accumulées jusqu'à présent ; on trouve dans la littérature l'appelation *from scratch* [53][54][55][56]. Cependant, cette méthodologie est très inefficace : d'une part, elle empêche l'apprentissage de nouvelles données en temps réel, d'autre part, cette approche peut ne pas être réalisable si les données originales ne sont plus disponibles ou que leur accumulation est impossible. Il apparaît ainsi nécessaire et plus raisonnable de considérer mettre à jour le modèle existant de

manière incrémentale, plutôt que de faire un apprentissage from scratch, à chaque fois que de nouvelles informations sont disponibles. De plus, le coût en temps de la procédure d'incrémentation, après l'ajout d'une observation supplémentaire à l'ensemble d'apprentissage, devrait être bien moindre que celui pour faire un ré-apprentissage from scratch.

Cette technique d'apprentissage se calque sur la manière qu'ont les humains d'apprendre : nous utilisons nos connaissances existantes lorsque nous en apprenons de nouvelles. C'est ce qui fait que nous sommes plus à l'aise pour reconnaître des objets ayant des formes familières plutôt que des objets n'ayant aucune forme particulière ; d'ailleurs, cela nous permet d'apprendre de nouvelles catégories d'objets à un rythme très rapide et à partir de peu d'exemples [57]. Il faut veiller à ce que l'apprentissage incrémental ne compromette pas pour autant les performances de classification sur des connaissances acquises précédemment [58]. Le dilemme stabilité-plasticité [59] met en évidence le fait qu'un modèle complètement stable préservera les connaissances existantes, mais n'en intégrera aucune nouvelle ; tandis qu'avec une plasticité totale, le modèle apprendra de nouvelles informations mais ne conservera pas les connaissances antérieures [58], on parle alors d'oubli catastrophique [60].

En 2001, Polikar et al. formalisent, pour un modèle, la notion d'apprentissage incrémental selon les critères suivants :

1. le modèle doit pouvoir apprendre de nouvelles informations à partir de nouvelles données ;
2. il ne doit pas nécessiter l'accès aux données originales (utilisées pour construire et entraîner le modèle existant) ;
3. il ne doit pas souffrir d'oubli catastrophique ;
4. il doit pouvoir s'adapter à de nouvelles classes.

Ils proposent ainsi l'algorithme *Learn++* pour l'apprentissage incrémental des réseaux de neurones [58]. Inspiré de l'algorithme AdaBoost, ce dernier fait appel à des méthodes d'ensemble. Les prédictions sont faites selon la stratégie du vote majoritaire pondéré par rapport aux performances de chaque réseau de neurones sur son sous-ensemble de données d'apprentissage. L'idée d'utiliser des méthodes d'ensemble vient du fait que l'utilisation d'apprenants faibles élimine le problème de *fine-tuning* et d'*overfitting*, puisque chaque classifieur ne fait qu'une approximation grossière des frontières de décision. Avec leur méthode, il est possible d'incrémenter sans accéder aux anciennes données d'apprentissage et il est aussi possible d'apprendre de nouvelles classes.

3.2.1 Réseaux de neurones et oubli catastrophique

Un état de l'art détaillé sur l'apprentissage incrémental des réseaux de neurones, a été proposé dans [53] et [61]. Les réseaux de neurones sont très utilisés depuis plus d'une décennie dans des domaines variés, du fait de leurs performances qui sont l'état de l'art pour de nombreux problèmes. Néanmoins, un challenge important pour l'apprentissage incrémental des réseaux de neurones est qu'ils sont confrontés au problème de l'**oubli catastrophique**. Lorsqu'ils apprennent une nouvelle classe, si une quantité d'anciennes données n'est pas présente au moment de l'apprentissage de celle-ci, le réseau ne parviendra plus à les reconnaître et ses performances s'écrouleront. Cela est lié directement à l'utilisation de la méthode de descente de gradient lors de l'apprentissage [62].

3.2.1.1 Les stratégies de répétition (*rehearsal*)

En 1995, Robins reprend les travaux de Ratcliff parus cinq ans plus tôt [63] puis explore et propose différentes stratégies de répétition (*rehearsal*) [64] pour s'attaquer au problème de l'oubli catastrophique. Dans la littérature, on trouve différentes appellations : *Replay*, *Rehearsal* ou encore *Exemplar selection* mais l'idée reste la même. Lorsqu'une nouvelle donnée doit être apprise, leurs études montrent que le réseau est moins sensible à l'oubli catastrophique lorsqu'il est entraîné avec des données déjà vues et plus anciennes que sans. Lors du phénomène d'oubli catastrophique, les informations apprises à l'origine par le réseau sont alors généralement fortement perturbées ou perdues. La méthode de *rehearsal* proposée dans [63] consistait en une file d'attente de quatre emplacements : une pour la nouvelle donnée à intégrer dans le réseau, et les trois autres pour inclure les trois dernières données vues par le réseau. La taille de cette file restant fixe, le dernier item arrivé pousse dehors le plus ancien. Robins, dans [64], qualifie alors cette méthode de *recency rehearsal* (cf. Figure 3.3). Robins relève que le principal inconvénient de cette méthode est qu'au bout de trois incréments (ils incrémentent une unique nouvelle donnée à la fois dans leurs expérimentations), le réseau n'aura déjà plus accès au jeu de données original lors de la prochaine incrémentation.

Il propose alors d'améliorer cette stratégie par ce qu'il qualifiera de *random rehearsal*. Lors de l'intégration de chaque nouvel item, au lieu d'inclure les trois dernières données vues dans la file, ce sont trois items qui sont sélectionnés de manière aléatoire parmi ceux vus jusqu'à présent. Ses résultats montrent que les performances obtenues avec le *random rehearsal* dépassent de loin ceux obtenus avec la stratégie de *recency rehearsal*, ce qui est conforme avec les attentes : en soumettant potentiellement

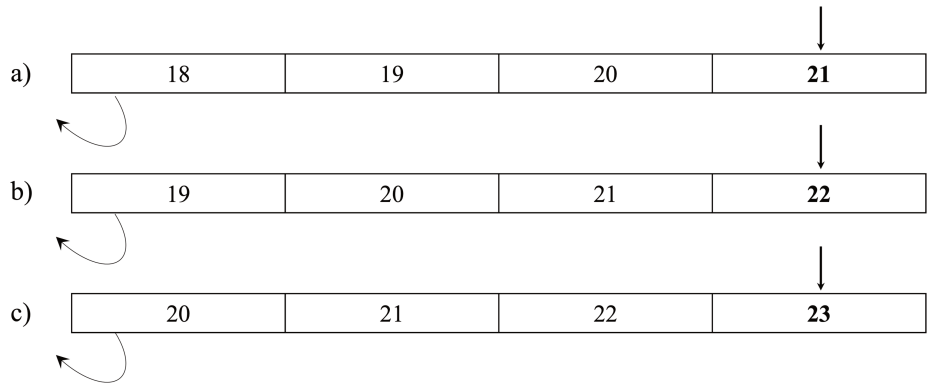

FIGURE 3.3 – Recency rehearsal - Si 20 observations sont présentes lors de l'apprentissage original, alors la file ressemble à a) pour l'incrément de l'observation 21, puis b) pour l'observation 22, et enfin c) pour l'observation 23

le réseau à des données originales lors de l'apprentissage de chaque nouvel item, les poids déjà appris sont "maintenus" et moins sensibles à la dégradation. Robins ne s'arrête pas là et propose une variante à son approche qu'il nomme *sweep rehearsal*. La subtilité de cette méthode réside dans la notion d'*epochs*. Un epoch consiste à entraîner le réseau de neurones avec toutes les données d'entraînement une seule fois pendant un cycle. À l'incrémentation de chaque nouvel item, un certain nombre d'epochs est réalisé par le réseau. La méthode de *sweep rehearsal* consiste en la sélection aléatoire de trois items à chaque epoch, là où dans le *random rehearsal*, cette sélection était fixée pour tous les epochs. De cette manière, le réseau va être exposé à plus de données potentiellement issues du jeu de données original ; c'est avec cette méthode que Robins obtient les meilleurs résultats.

Le rehearsal a été utilisé notamment dans [65] mais est qualifié d'*exemplar replay* ; les auteurs proposent la méthodologie *Incremental Classifier and Representation Learning* (iCaRL) pour un réseau de neurones basé sur le principe de NCM. Ils montrent également que, pour un réseau pré-entraîné sur un lot spécifiques de classes, la technique de finetuning appliquée sur des données provenant de nouvelles classes, a pour effet d'oublier tout simplement l'existence des classes précédentes comme illustré dans la figure 3.4. Généralement avec le rehearsal, une mémoire tampon est allouée pour stocker des observations d'anciennes classes, qui sont rejouées lors de l'apprentissage d'une nouvelle tâche afin de limiter les oublis. De nombreux travaux utilisent l'heuristique du "troupeau" pour la sélection des exemples : cette stratégie consiste à sélectionner et conserver les échantillons les plus proches du centroïde de chaque classe. Dans [66] en revanche, les auteurs n'ont pas recours à cette heuristique et échantillonnent chaque classe de manière aléatoire (*random rehearsal*) pour constituer

l'ensemble d'exemples. Ils parviennent finalement à obtenir de bons résultats sans l'utilisation de cette heuristique. Nous pouvons remarquer que l'inconvénient d'une mémoire fixe est qu'à mesure que l'on incrémente en classe, le système basé *exact replay* se voit stocker une quantité de plus en plus minime d'exemples par classe et peut donc se retrouver à ne plus généraliser correctement lorsqu'un très grand nombre de classes est présent. Une comparaison entre de nombreux protocoles d'expérimentation et d'évaluation pour l'apprentissage incrémental a été réalisée dans [67] ; les auteurs concluent que parmi tous les scénarios envisagés, seules les méthodes basées sur le *replay* sont actuellement capables de produire des résultats acceptables.

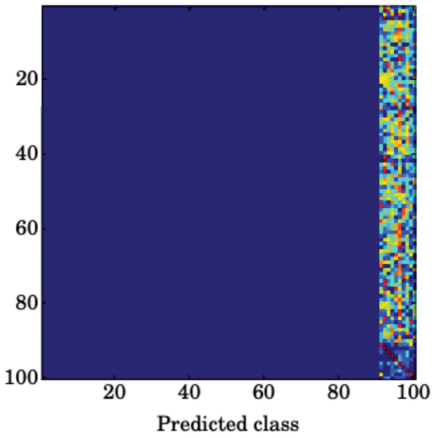

FIGURE 3.4 – Finetuning et oubli catastrophique - le réseau se spécialise sur la dernière classe à partir de laquelle il s'est incrémenté et oublie les anciennes

3.2.2 Méthodes basées sur les forêts aléatoires

Les arbres de Hoeffding ont été proposés dans [68] pour pouvoir entraîner des arbres de manière continue à partir de flux de données, contrairement aux arbres dits hors-ligne, i.e. s'entraînant sur toutes les données disponibles. L'idée des auteurs était de fournir une méthode pour entraîner des arbres sur de très grands ensembles de données (potentiellement de taille "infinie" au sens qu'elles continuent d'arriver sans cesse) tout en prenant un temps stable et court pour traiter chaque observation. L'algorithme fonctionne en maintenant plusieurs splits candidats dans chaque feuille. La qualité de chaque split est estimée au fur et à mesure que les données arrivent dans la feuille, mais comme la totalité de l'ensemble d'entraînement n'est pas disponible, ces mesures de qualité ne sont que des estimations. La limite de Hoeffding [69] est employée dans chaque feuille pour contrôler la quantité

de données qui doivent être collectées pour garantir avec une forte probabilité que le split choisi sur la base de ces estimations est le même qu'avec un arbre hors-ligne. Les auteurs prouvent que, sous des hypothèses raisonnables, l'arbre de Hoeffding incrémental converge vers l'arbre hors-ligne avec une forte probabilité. Néanmoins, les arbres C4.5 semblent plus performants pour les ensembles de données réduits, inférieurs à 25 000 observations. De plus, les auteurs supposent que les observations sont générées à partir d'un processus stochastique stationnaire, dont la distribution ne change pas au fil du temps, i.e. exempt de toute dérive de concept.

Denil et al. décrivent en 2013 des forêts aléatoires dites "en ligne" [70]. De la même manière que dans [68], chaque arbre commence en étant une racine vide et croît de manière incrémentale. Puis, chaque feuille maintient une liste de k *splits* candidats et leurs scores de qualité associés. Lorsqu'une nouvelle observation est ajoutée, les scores du nœud correspondant sont mis à jour. Afin de réduire le risque de choisir un *split* sous-optimal à cause de bruit dans les données, des hyperparamètres supplémentaires, tels que le nombre minimum d'exemples requis pour un split (n_{stop}) et le gain d'information minimum pour autoriser le split, sont utilisés. Une fois que ces critères sont satisfaits dans un nœud, le meilleur split est choisi (faisant de cette feuille un nœud interne) et ses deux enfants sont les nouveaux nœuds feuilles (avec leurs propres splits candidats). Ces méthodes pourraient s'avérer inefficaces pour les arbres profonds en raison du coût élevé associé au maintien des scores de qualité des candidats. Du fait du grand nombre d'informations stockées dans les feuilles, les auteurs ont besoin de 1.6 à 10 Go d'espace de stockage lorsqu'ils font varier les paramètres de leur forêt incrémentale [70].

Lakshminarayanan et al. proposent en 2014 les Mondrian Forests (MF) [71], des RF pouvant faire de l'apprentissage incrémental. Leur méthode permet d'ajouter un noeud ou poursuivre la construction à partir d'une feuille sous certaines conditions. Les MFs obtiennent des résultats compétitifs avec des RFs classiques entraînées sur les mêmes datasets. La même année, Ristin et al. introduisent les Nearest Class Mean Forests (NCMFs), des forêts aléatoires basées sur le principe de NCM [19]. Les auteurs montrent que l'apprentissage de la NCMF sans même avoir recours à une phase d'incrémentation, obtient de meilleures performances que les RFs classiques pour de la classification d'images à grande échelle. Néanmoins, les stratégies proposées pour continuer de faire croître l'arbre pendant la phase incrémentale nécessitent l'accès aux données précédentes.

Hu et al. en 2018 proposent, dans le contexte de la reconnaissance d'activités, une nouvelle méthode incrémentale pour les RFs, permettant de moins recourir aux données initiales d'apprentissage. Elle

se nomme *Class Incremental Random Forests* (CIRF) et s'applique spécifiquement à l'incrémentation en classe [72]. Des boîtes englobantes minimales sont définies pour chaque classe. Lorsqu'une nouvelle classe est introduite dans un arbre pré-entraîné, un noeud est ajouté au dessus du parent lorsqu'il n'y a pas d'intersection entre les boîtes englobantes ; dans ce cas, il n'est pas nécessaire d'avoir accès aux données précédentes. Lorsqu'il y a intersection, un split est réalisé dans la feuille, menant à la construction d'un nouveau sous-arbre ; dans ce cas, il est nécessaire d'avoir à disposition les anciennes données.

3.3 Apprentissage semi-supervisé

3.3.1 Contexte

À l'ère du Big Data, il est parfois compliqué d'obtenir des données labélisées à la main. Dans le domaine de l'informatique affective, le processus de labélisation nécessite d'engager un codeur certifié FACs pour annoter les unités d'action (AUs), ou un psychologue pour annoter le niveau de dépression. En raison des contraintes de temps et de coût, il semble irrationnel d'espérer labéliser manuellement des milliers de données qui seraient nécessaires pour entraîner les modèles actuels, de manière supervisée. Par conséquent, dans un projet d'apprentissage automatique classique, la collecte et la préparation des ensembles de données prennent un temps considérable. C'est une contrainte encore plus forte dans certains secteurs industriels où l'innovation est au centre des préoccupations et où le changement doit être rapidement réalisable. L'apprentissage faiblement supervisé [73] vise à résoudre ce type de problèmes. En effet, il permet de labéliser un grand nombre de données de manière automatique. Un jeu de données est généralement fourni avec un grand nombre de données, dont seule une petite quantité est labélisée. Dans le domaine de l'apprentissage faiblement supervisé, les labels ne sont pas toujours parfaits, on parle alors de "labels faibles", mais il demeure toujours possible de concevoir des modèles à fort pouvoir prédictif à partir de ceux-ci. Les origines d'une mauvaise labélisation sont multiples. Ils peuvent être imprécis ; c'est le cas pour la labélisation de scènes par exemple. Ils peuvent être inexacts ; cela se produit, par exemple, dans le cas où l'on fait appel à des solutions de *crowdsourcing* pour labéliser où la quantité est souvent privilégiée à la qualité de la labélisation. Différentes techniques ont été développées pour traiter les labels faibles. D'une part, on trouve l'"apprentissage actif" qui nécessite l'intervention d'un "oracle" qui peut être un expert humain. D'autre part, l'ap-

prentissage semi-supervisé tente d'éviter l'intervention humaine en exploitant automatiquement les données labélisées et non labélisées. Dans cette section, nous focalisons notre étude sur les méthodes semi-supervisées car nous souhaitons évoluer vers une procédure automatique et donc nous affranchir au maximum de l'intervention humaine. La lecture de [74] décrit de façon très complète le domaine de l'apprentissage semi-supervisé.

3.3.2 Self-training

Yarowsky, souhaitant trouver une méthode pour réduire le coût laborieux de la labélisation manuelle et tirer profit de la grande quantité de données non labélisées, introduit le concept de *self-training* en 1995 dans le contexte de la désambiguïsation lexicale [75]. Il existe beaucoup de mots polysémiques, c'est-à-dire des mots ayant plusieurs significations différentes dans le dictionnaire, et le but de l'algorithme de Yarowsky est de déterminer la signification qui est attendue. Il utilise, pour labéliser les données qui ne le sont pas, l'hypothèse selon laquelle, dans un même document, toutes les instances d'un même mot ont très probablement une signification commune. Il décrit une série d'étapes itératives dans lesquelles un modèle apprend sur quelques exemples labélisés (2 à 15%) pour ensuite faire des prédictions sur de nombreux exemples non labélisés, lesquels sont ajoutés à l'ensemble d'apprentissage si le niveau de confiance est élevé. Le modèle est alors ré-entraîné sur le nouvel ensemble d'apprentissage augmenté. Il parvient à obtenir de bons résultats avec cette procédure.

Ce concept, très utilisé pour l'analyse du langage, apparaît ensuite en 2005 dans le contexte de la détection d'objets à partir d'images avec Rosenberg qui parvient à obtenir des résultats au dessus de l'état de l'art [76]. En 2006, Roli propose d'appliquer la procédure de self-training à la reconnaissance de visages en utilisant l'*Analyse en Composantes Principales* (ACP) pour mettre à jour les systèmes biométriques [77]. En 2010, Cherniavsky propose de l'appliquer à la reconnaissance d'attributs à partir de vidéos centrées sur des visages pour détecter en particulier l'âge et le genre [78]. La méthode qu'ils proposent n'est en revanche pas applicable aux expressions faciales car leur méthodologie est fondée sur l'hypothèse selon laquelle les attributs ciblés restent constants au cours de la vidéo, comme l'âge et le genre. À notre connaissance, le concept de self-training apparaît peu dans la littérature sur la reconnaissance visuelle d'émotions : on trouve davantage d'informations sur la reconnaissance de visages ; concernant les émotions, les travaux sont principalement liés à l'analyse de textes. Le concept apparaît en 2013 pour des données physiologiques comme la pression artérielle dans [79] où la

considération des exemples non labélisés associés à des dictionnaires permettent d'améliorer les taux de classification. Plus récemment, Kumar utilise en deep learning une forme de self-training pour la reconnaissance d'expressions faciales à la différence qu'il rajoute du bruit selon la méthodologie *noisy student training* [80]. Dans cette méthodologie, le modèle de référence est l'"enseignant", entraîné sur la portion de la base contenant seulement les labels. On l'utilise pour obtenir des pseudos-labels sur la portion de la base restante non labélisée. On entraîne ensuite l'"élève" sur la concaténation des labels et pseudo-labels en injectant du bruit aléatoire dans les images ; celui-ci devient à son tour l'enseignant. Celui-ci est ensuite utilisé pour pseudo-labéliser ; ce processus itératif se répète jusqu'à ce que les performances n'évoluent plus. Kumar parvient à obtenir un réseau élève de plus en plus robuste à mesure des itérations et de l'ajout du bruit (méthodologie dite *student-teacher*).

Dans la procédure de self-training, il est question d'injecter dans le dataset les exemples avec leurs pseudo-labels à la condition que la confiance associée soit grande. Cette notion de confiance se calcule à partir des probabilités d'appartenance aux classes dans le cadre des arbres de décision. Malheureusement les arbres de décision sont considérés comme de mauvais estimateurs de ces dites probabilités et différentes solutions ont été proposées pour y remédier, par exemple la correction de Laplace [24][25].

3.3.3 Co-training

Le *co-training* est considéré par [81] comme un paradigme représentatif des méthodes semi-supervisées basées sur le désaccord. On peut le considérer comme une extension du concept de self-learning, à la différence que deux classifieurs, ou plus, entraînés sur les données labélisées (disponibles en petite quantité), vont itérativement faire des prédictions sur des données non labélisées (pseudo-labels), et ajouter les plus fiables à l'ensemble de données labelisées des autres classifieurs. Ces derniers se ré-entraînent sur l'ensemble de données augmenté au cours de chaque itération. Pour espérer de bons résultats de cette méthode, il est important que les prédictions des classifieurs initiaux soient le plus décorrélées possible.

Théoriquement, les ensembles de *features* sur lesquels sont entraînés les différents modèles doivent être indépendants. Autrement, leur potentiel à se fournir mutuellement des informations pertinentes est limité. Blum et Mitchell définissent et formalisent le co-training en 1998 [82] dans le contexte de la classification de pages web. Ils parviennent à montrer avec leur méthode que deux classifieurs

entraînés conjointement sur deux ensembles de features différents (vues), et tirant partie des données non labélisées, parviennent à obtenir de bons taux de classification, comparé à un classifieur qui aurait seulement appris sur le petit ensemble de données labélisées.

Cohen et al. ont été les premiers à proposer, en 2003, un apprentissage semi-supervisé pour utiliser efficacement les données non labélisées pour la reconnaissance des expressions faciales avec le jeu de données de Cohn-Kanade [83].

Zhang et al. proposent en 2016 [84] la méthode *enhanced Semi-Supervised Learning* (eSSL) pour corriger les mauvaises labélisations qui peuvent avoir lieu au cours des itérations du co-training. Effectivement, un des inconvénients du co-training est l'accumulation de bruit. En accordant trop de confiance au modèle, on court le risque qu'il labélise de manière incorrecte des observations. Ces derniers étant rajoutés dans le dataset d'apprentissage à chaque itération, on peut potentiellement se diriger vers un cercle vicieux d'erreurs d'apprentissage. Ils obtiennent de cette manière de meilleures performances en co-training multimodal que unimodal, sur les données audio-visuelles du dataset RECOLA [85].

Alyuz et al. proposent en 2016 une approche multimodale dans le contexte des systèmes de tutorat intelligents pour détecter l'engagement des étudiants [86]. Avec la forme de co-training unilatéral qu'ils proposent, ils montrent dans leur étude qu'ils ont pu améliorer les performances et personnaliser un modèle d'apparence en utilisant les labels fournis par le second modèle entraîné à partir d'informations de session et de formulaires d'auto-évaluation. Ils utilisent une régression isotonique pour corriger le biais des distributions de probabilités issues des random forests comme expliqué dans [87].

3.4 Discussion

On a pu voir que l'apprentissage incrémental offrait la possibilité de mettre à jour des modèles existants sans avoir à tout ré-apprendre depuis le début, c'est pourquoi dans nos travaux nous nous intéressons à celui-ci ; il pourrait, d'une part, aider à gérer la contrainte d'une collecte progressive de données, d'autre part, il pourrait aider à personnaliser des modèles génériques à certains sujets pour tenter d'atténuer le biais d'identité.

Le réseau de neurones convolutif (profond) (CNN) est l'un des modèles les plus populaires et a obtenu des résultats faisant office d'état de l'art dans divers domaines, notamment la reconnaissance des

expressions faciales [6][88][89]. Cependant, malgré des avancées significatives en apprentissage incrémental, les modèles actuels de réseaux de neurones sont encore loin d'offrir la flexibilité, la robustesse et la même évolutivité que les systèmes biologiques et sont confrontés au problème de l'oubli catastrophique. Ils sont en général limités au domaine supervisé et reposent sur de grandes quantités de données annotées recueillies dans des environnements contrôlés. Nous sommes encore loin des agents autonomes pouvant opérer dans des environnements hautement dynamiques et non structurés [53]. À cet égard, nous avons alors décidé d'utiliser des forêts aléatoires de type *Nearest class mean* (NCMF) en raison de leur nature multi-classes et de leur capacité de généralisation. Par ailleurs, elles présentent d'autres avantages : un apprentissage très rapide, une incrémentation possible en données mais aussi en classes et une meilleure interprétabilité. Elles ont également démontré qu'elles pouvaient surpasser les RF et permettre une incrémentation facile [19]. En outre, les modèles de forêt aléatoire ont été utilisés avec succès pour la personnalisation [9][11][15].

Dans la littérature, on trouve très souvent des propositions fonctionnant pour l'apprentissage supervisé uniquement [53]. L'exploration du domaine semi-supervisé a pour but de pouvoir s'affranchir d'une grande partie de labélisation, qui serait coûteuse dans tout projet ayant de nombreuses vidéos. Nous avons vu que le co-training offrait des perspectives intéressantes pour le domaine semi-supervisé ; une limitation observable, cependant, est que le classifieur conserve plusieurs sessions d'entraînement sur le même ensemble de données. Nous pourrions profiter des progrès de l'apprentissage incrémental pour ne faire qu'un seul passage et mettre à jour l'arbre tout en pseudo-labélisant les données sans ré-entraîner à partir de zéro à chaque itération.

Dans la suite de ce mémoire, nous proposons de combiner ces deux paradigmes que sont l'apprentissage incrémental et les NCMF afin de créer une méthode qui permette au modèle de s'adapter à différentes bases de données. Nous explorons également la possibilité de pouvoir intégrer un pipeline permettant de réaliser la personnalisation de manière semi-supervisée avec le co-training afin de faire face au biais d'identité qui intervient dans le contexte de la reconnaissance d'expressions faciales.

Chapitre 4

Travaux préliminaires réalisés sur les données TEEC

Contenu

4.1 Travaux connexes 78
4.2 Méthodologie DBR et collecte de données 79
4.3 Effet Eurêka 80
4.4 Analyse de l'itération "Langues" 83
4.5 Méthodologie 84
4.5.1 Interactions verbales et élaboration des connaissances 84
4.5.2 Analyse des interactions non verbales (affectives) 85
4.6 Résultats : Analyse conjointe 86
4.7 Conclusion 89

Dans ce chapitre, nous présentons une analyse réalisée sur les données du projet TEEC au début de la thèse. Les données à disposition sont des interactions par visioconférence impliquant deux groupes d'apprenants effectuant des tâches d'apprentissage collaboratif à distance. Du fait qu'ils vivent dans deux pays différents, leurs représentations mentales sur divers thèmes sont différentes et produisent ce que nous appelons un effet de contexte ou conflit socio-cognitif.

Nous nous intéressons ici à la fois aux aspects cognitifs et affectifs des effets de contexte. Des expériences ont montré que les étudiants manifestent un large spectre d'émotions lorsqu'ils interagissent entre eux, au sein d'un groupe, et également avec l'autre groupe. Dans cette section, nous étudions ces interactions et essayons de trouver une corrélation entre les états affectifs non verbaux et les échanges verbaux, reflétant le degré de (mauvaise) compréhension. Nous n'essayons pas d'étudier comment les émotions influencent le processus d'apprentissage, mais plutôt quelles sont les émotions que les apprenants éprouvent lors de la construction collaborative des connaissances.

On présente d'abord le contexte de la collaboration entre les différents membres du consortium, le principe de l'itération du projet TEEC et de sa collecte de données, l'effet Eurêka, puis, l'analyse conjointe verbale et non-verbale que nous avons réalisée.

4.1 Travaux connexes

Les recherches sur la contextualisation didactique [90] peuvent être classées en plusieurs "écoles" en fonction de leurs objectifs. Parmi elles, une tendance, issue de la psycho-didactique, étudie l'impact des représentations mentales (comme les contextes internes) sur le processus d'apprentissage. Dans ce cas, le contexte existe en termes de représentations mentales des élèves et est décrit comme un processus de décontextualisation-recontextualisation [5]. Une autre préconise une approche basée sur le contexte environnemental [91][92][93]. Dans ce cas, le contexte est considéré comme externe à l'apprenant et les élèves mènent leurs investigations dans des environnements naturels, à partir de situations authentiques. Dans ce projet, nous considérons les spécificités contextuelles comme une richesse et nous les utilisons pour construire une approche pédagogique, intégrant les technologies numériques non seulement comme des outils pour les apprenants (par exemple, les services Web, la visioconférence, les espaces de travail numériques, le matériel ultra-mobile pour l'investigation), mais aussi comme un élément de modélisation des enseignements basé sur l'émergence des effets de contexte.

Ces derniers génèrent ce que nous avons appelé des quiproquos socio-cognitifs, qui sont en quelque sorte des conflits socio-cognitifs, et qui génèrent, lors de leur résolution, ce que l'on nomme par la suite un effet Eurêka (défini dans la section 4.3). Les confrontations entre élèves pendant l'apprentissage par les pairs pourraient être bénéfiques lorsqu'on travaille sur des informations complémentaires et nouvelles [94]. Les interactions collaboratives concernent les idées, les représentations, la compréhension et l'ancrage [95][96]. À partir de cette définition, les recherches sur l'apprentissage collaboratif ont étudié le rôle des interactions communicatives pour réaliser la tâche et co-élaborer les connaissances [97]. Les études se concentrent spécifiquement sur le débat dans lequel l'argumentation est engagée dans l'interaction afin de répondre à une question spécifique par des moyens purement verbaux [98]. Deux processus principaux de création de connaissances sont associés au débat : la production d'arguments ou de contre-arguments et la négociation du sens. Ces deux interactions présentent des potentiels d'apprentissage tels que le changement conceptuel et l'apprentissage réflexif [95].

Une autre question clé dans l'apprentissage, et particulièrement dans l'apprentissage avec des outils électroniques est l'affect. Dans l'apprentissage "traditionnel" en face à face, l'enseignant, en fonction de son empathie, est capable de gérer à la fois les charges cognitives et affectives [99]. Il peut utiliser ces dernières pour déterminer comment ajuster le rythme ou le contenu de l'apprentissage. Quel est alors l'impact de l'absence de celui-ci, lorsque l'apprenant est confronté seul, en face à face avec une machine ?

4.2 Méthodologie DBR et collecte de données

Dans le projet TEEC, le travail est réalisé sous forme de cycles itératifs, en suivant la méthodologie *Design-Based Research* (DBR) [100], au cours desquels, nous avons obtenu différentes captations vidéo. Cette méthodologie a été initialement proposée pour réduire l'écart entre la recherche en laboratoire et la recherche in situ ; elle peut améliorer les pratiques éducatives par le biais d'une analyse itérative, de la conception, du développement et de la mise en oeuvre, basée sur la collaboration entre les chercheurs et les praticiens dans le monde réel [101].

Cinq itérations ont été réalisées sur différents sujets : l'énergie géothermique (une itération), les sciences de l'environnemeent (deux itérations), la socio-économie (une itération) et "les Langues" (une itération). Pour chacune d'elles, le processus DBR a été instancié de la manière suivante :

1. modélisation du contexte,
2. scénario pédagogique,
3. expérimentation avec collecte de données,
4. résultats et leçons apprises.

Le scénario pédagogique a été conçu par des enseignants et des chercheurs pour produire une confrontation entre deux groupes d'étudiants immergés dans deux contextes contrastés Guadeloupe et Québec. Les populations d'étudiants ont des âges différents selon le sujet, allant de la primaire à l'université.

Pour une itération donnée, la collecte de données a combiné trois visioconférences avec les deux classes entières des deux régions différentes, en interaction les unes avec les autres. Par la suite, les élèves ont été réunis en groupes (3-4 élèves par groupe). Chaque groupe d'une région a interagi avec un autre groupe de l'autre région, au cours de trois visioconférences ayant toutes lieu dans des environnements non contrôlés que sont les salles de classe ; toutes ces sessions de travail entre étudiants ont été enregistrées.

Les enregistrements issus des premières itérations étaient peu exploitables et nous ont conduit à rédiger un guide des bonnes pratiques pour rendre possible l'analyse automatique des séquences par des algorithmes. Parmi ces bonnes pratiques, notons :

— utiliser un matériel d'enregistrement (caméra et micro) de qualité ;
— faire en sorte que les apprenants soient toujours "face caméra" ;
— veiller à ce que le micro n'occulte pas le visage des apprenants ;
— éviter des situations de contre-jour ;
— veiller à ce que les apprenants ne changent pas sans arrêt de place ;
— veiller à ce que les groupes ne comptent pas plus de quatre apprenants.
— etc.

4.3 Effet Eurêka

L'effet Eurêka est un phénomène de compréhension soudaine qui est au coeur de cette collaboration TEEC. En terme d'interactions verbales, on peut caractériser cet effet comme la succession d'états

cognitifs de désaccord, d'argumentation, d'accord et enfin de compréhension. En terme d'interactions non verbales, nos observations, sur les séquences vidéos du projet, nous ont permis de caractériser cet effet sous la forme d'une succession de changements d'états émotionnels : frustration, surprise, puis, satisfaction. La figure 4.1 illustre cet effet :

1. au début, nous assistons à une phase d'écoute active, pendant laquelle des états possibles tels que la frustration, due en partie à une forme d'incompréhension du sujet, peuvent être perçus sur certains visages ;
2. s'en suit ensuite une phase de compréhension en deux temps : d'abord, une compréhension soudaine, souvent marquée par une réaction de surprise sur le visage de la personne, puis généralement des dyades intra-groupe, au cours desquelles, les élèves échangent entre eux sur ce qu'ils viennent de comprendre, permettant en général à ceux pour qui ce n'était pas encore le cas, de comprendre à leur tour ;
3. enfin, une phase de contagion émotionnelle et de propagation des connaissances, dite intercompréhension, se termine, selon nos observations, sur des réactions de plaisir et de satisfaction.

En se focalisant sur les émotions basiques d'Ekman, nous avons émit l'hypothèse que les interactions verbales caractérisant l'effet Eurêka pouvaient, par association, être représentées par la succession d'états de tristesse, surprise puis de joie. Ces états émotionnels sont plus ou moins subtils selon l'individu qui les exprime, en partie à cause du biais d'identité.

FIGURE 4.1 – Illustration de l'effet Eurêka avec des images issues de TEEC

4.4 Analyse de l'itération "Langues"

L'objectif était de décrire et caractériser des effets de contexte dans un scénario d'apprentissage collaboratif. Pour ce faire, nous avons combiné les expertises de deux équipes, membres du consortium. La première s'est intéressée à cette problématique du point de vue des interactions langagières. La deuxième s'est focalisée sur l'analyse non verbale, notamment l'analyse émotionnelle des apprenants :

Interactions verbales : Chloé Le Bail, chercheuse postdoctorale en Psychologie Ergonomique du laboratoire EDA (Éducation Discours Apprentissages), sous la direction de François-Xavier Bernard (Maître de Conférences HDR) à l'université Paris Descartes, Françoise Détienne et Michael Baker, Directeurs de Recherche à l'I3 (l'Institut Interdisciplinaire de l'Innovation) et au département SES (Sciences Économiques et Sociales) de Télécom Paris,

Interactions non verbales : Mélanie Piot et Thybault Alabarbe durant leur stage de 4ème année, Jordan Gonzalez, doctorant au laboratoire LDR, sous la direction de Lionel Prevost, Directeur de Recherche au LDR à l'ESIEA.

L'itération "Langues" est celle sur laquelle nous nous sommes focalisés pour réaliser une étude conjointe avec l'autre équipe. Cette itération implique des élèves de l'école primaire à qui il a été demandé de réaliser une étude sur les contes populaires des Antilles et du Québec, qui sont de nature et de structure différentes [102]. Le groupe de la Guadeloupe a travaillé sur un conte populaire créole ; le groupe de Québec a travaillé sur un conte traditionnel québécois. Chaque groupe travaille sur une composante des contes populaires parmi : la structure narrative, les personnages, les lieux, le lexique et vocabulaire utilisés et les références culturelles.

La séquence avec effet de contexte que nous avons analysé conjointement concerne le groupe qui a travaillé sur les personnages des contes populaires, voir Figure 4.2. La séquence dure environ 15 minutes. Les élèves discutent de la manière dont ils vont rechercher les informations concernant les personnages de leurs textes respectifs. L'effet de contexte apparaît car les personnages ne sont pas populaires de la même manière dans les deux régions. Au Québec, les personnages sont décrits uniquement dans le texte original. Les contes populaires antillais, quant à eux, ne sont pas seulement écrits dans les livres, ils peuvent être chantés lors de festivités, ou les conteurs peuvent les présenter et les enrichir de traditions et de coutumes. Ainsi, la description des personnages peut être différente, plus ou moins complète, selon la source d'information.

FIGURE 4.2 – Itération Langues - étudiants de Québec et de Guadeloupe

4.5 Méthodologie

Dans cette section, nous décrivons la méthodologie adoptée par chacune des deux équipes, concernant respectivement, les interactions verbales et les interactions non-verbales.

4.5.1 Interactions verbales et élaboration des connaissances

L'équipe, travaillant sur les interactions verbales entre les apprenants (inter et intra-groupe) de l'itération, a tout d'abord réalisé une transcription de celle-ci. Les tours de parole, les intervenants et le discours ont été ainsi annotés. L'analyse de chaque tour de parole a consisté à identifier, en particulier pour cette analyse conjointe, deux dimensions liées à l'alignement cognitif (compréhension mutuelle) entre les apprenants de chaque groupe. Ces dimensions sont les suivantes :

Dimension énonciative : cette dimension décrit par exemple le ou les apprenants qui expriment leur point de vue ; en général, l'énonciateur est le locuteur mais cette dimension décrit également des cas particuliers ou un porte-parole désigné parle au nom de son groupe, ou bien lorsque l'enseignant parle pour tout le groupe ;

Dimension interactive : cette dimension décrit la manière dont les apprenants interagissent entre eux ; la construction du sens et la co-élaboration des connaissances peuvent être identifiées à travers les indicateurs verbaux suivants :

C+ (+5) : compréhension totale du concept,

A (+4) : être en accord,

Arg (+3) : argumenter pour défendre une idée sur le concept,

Exp (+2) : clarifier un concept,

? (+1) : poser une question,

D (-1) : être en désaccord,

C- (-2) : petite compréhension du concept,

IC (-3) : exprimer un malentendu.

4.5.2 Analyse des interactions non verbales (affectives)

Un extracteur de données a été développé à cet effet pour rassembler toutes les données non verbales pouvant être utilisées pour identifier l'effet Eurêka dans chacune des vidéos. Cet extracteur est composé des modules suivants :

1. Identification et suivi : dans les vidéos disponibles, plusieurs étudiants sont présents en même temps dans l'image. L'identification des étudiants dans les vidéos est effectuée en deux étapes. Tout d'abord, la détection des visages dans l'image est réalisée par la librairie OpenFace. Ensuite, nous avons utilisé la librairie face_recognition [1] pour obtenir une représentation des caractéristiques du visage de chaque individu qui a été détecté. Ces informations, ainsi que la position de ces visages dans chaque image de la vidéo, sont ensuite utilisées pour suivre les étudiants à l'aide d'un algorithme du plus proche voisin.

   Plus précisément, nous procédons de la manière suivante. Dans la première image de la séquence vidéo, un identifiant numérique arbitraire est attribué à chacun des visages détectés par OpenFace. Le suivi de visage d'une frame à une autre est ensuite réalisé selon deux méthodes. La première méthode consiste à utiliser le modèle de reconnaissance faciale. Il convertit l'image du visage de la frame courante en un vecteur de caractéristiques. Puis, il calcule la distance entre celui-ci et les vecteurs de référence collectés dans la première image, afin de donner au visage l'identifiant de son plus proche voisin. La seconde méthode, moins complexe et donc moins gourmande en temps, consiste à utiliser un algorithme du plus proche voisin "spatial" sur la boîte englobante calculée à partir des landmarks. Il va permettre de relier les boîtes englobantes de deux frames et de les associer au sein d'une même piste. Il arrive qu'un individu se déplace au cours de la vidéo, et l'algorithme du plus proche voisin n'est plus suffisant pour faire le suivi de visage de manière robuste. Au-delà d'un certain seuil de variabilité de la position du visage, la première méthode est à nouveau utilisée.

1. https://github.com/ageitgey/face_recognition#face-recognition

2. Détection des émotions : nous avons utilisé le détecteur d'émotions disponible avec la librairie *Emotion-recognition*[2] pour obtenir des valeurs sur 7 émotions selon le modèle d'Ekman [33]. Nous avons émis une hypothèse faible, non confirmée par le codage des unités d'action des deux émotions, que la confusion pouvait, par association avec les émotions basiques, être considérée comme une tristesse de faible intensité. La librairie utilise un *Convolutional Neural Network* (CNN) qui a été entraîné à reconnaître les émotions basiques d'Ekman, sur les données du dataset Facial Expression Recognition (FER2013) utilisé dans le cadre d'un challenge organisé lors de la conférence ICML (*International Conference On Machine Learning*) [103].

4.6 Résultats : Analyse conjointe

La figure 4.3 montre les résultats obtenus avec l'analyse des émotions. La vidéo défile à gauche avec, pour chaque visage détecté, la boîte englobant les landmarks. Les sept graphes (six émotions basiques en plus du neutre) à droite représentent le nombre d'étudiants exprimant en même temps une émotion donnée au cours de la vidéo. Les émotions ont été détectées pendant la vidéo avec les fréquences suivantes : neutre (59%), tristesse (16%), surprise (9%), bonheur (8%), peur (6%), colère (2%), dégoût (0.02%).

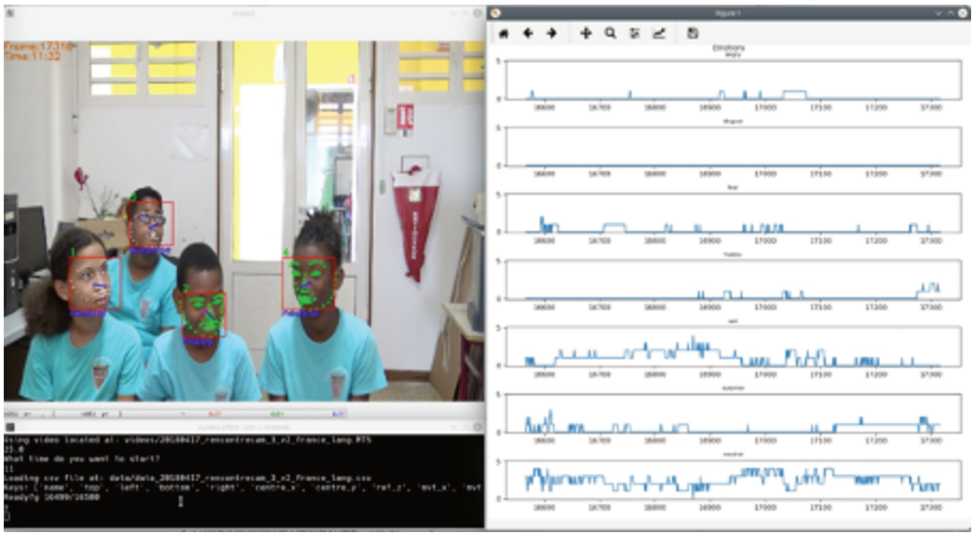

FIGURE 4.3 – Itération Langues (Guadeloupe), résultats de l'analyse émotionnelle - chaque graphe est associé à une émotion basique, et décrit le nombre d'élèves exprimant l'émotion au même instant de la séquence vidéo

2. https://github.com/omar178/Emotion-recognition

La figure 4.4 détaille l'analyse conjointe des canaux affectif (non verbal) et cognitif (verbal) pour l'itération langue. L'analyse affective (premier et deuxième graphe) ne concerne que les élèves Antillais pour simplifier la lecture. Le premier graphe indique le cumul de tristesse chez les quatre élèves du groupe, au fil du temps. À 3 min, on observe un pic : cela signifie que 4 étudiants sur 4 ont exprimé la tristesse à ce moment là de la vidéo. Le deuxième graphe indique le cumul de joie chez les quatre élèves du groupe, au fil du temps. Les flèches entre ces deux graphes montrent bien que les pics de tristesse (4 élèves sur 4), c'est-à-dire de frustration/confusion ou peut-être d'écoute active, sont suivis dans le temps par des pics de joie (2 ou 3 élèves sur 4), ou autrement dit de satisfaction. Pour autant, bien que ces successions d'états aient lieu comme décrits dans la section 4.3, ces moments ne correspondent pas tout le temps à l'effet Eurêka. L'analyse des interactions verbales (troisième graphe) utilise les valeurs entre l'incompréhension (-3) et la compréhension totale (+5), la co-élaboration de connaissances est très positive tandis que l'incompréhension est négative ; cette analyse concerne les élèves Antillais et Québécois, les couleurs rouge et orange sur le graphe désignent les interactions faites par le groupe d'étudiants Antillais, et la couleur verte désigne celles des étudiants Québécois. Les flèches entre le deuxième et troisième graphe ne sont pas verticales car l'échelle de temps du dernier graphe n'est pas linéaire. Nous avons recherché le motif "tristesse suivi de joie" pour savoir à quels instants de la vidéo on pouvait les identifier par rapport aux résultats de l'autre équipe. Sur ce principe, si l'on compare ce troisième graphe avec les deux premiers (flèches et crochets), on observe deux phénomènes :

— Micro-moments d'intercompréhension intra-groupe ou effet Eurêka intra-groupe : les crochets 1 et 2 montrent des moments d'explication entre différents locuteurs du côté de la Guadeloupe ; l'énonciateur répond et les autres élèves du même groupe comprennent. On observe alors une contagion cognitive et émotionnelle ;

— Micro-moments d'intercompréhension (ou d'accord mutuel) inter-groupes ou effet Eurêka inter-groupe : les crochets 3, 5 et 8 montrent des moments d'Explication-Accord/Compréhension entre les locuteurs de la Guadeloupe et du Québec ; le groupe du Québec explique un concept et le groupe de Guadeloupe comprend ce dernier.

Nous remarquons un autre résultat intéressant : l'équipe de l'analyse des interactions verbales a identifié un effet Eurêka (en vert sur le schéma) ; le plus grand pic de joie de toute la séquence vidéo que nous avons obtenu est présent pendant cet effet Eurêka.

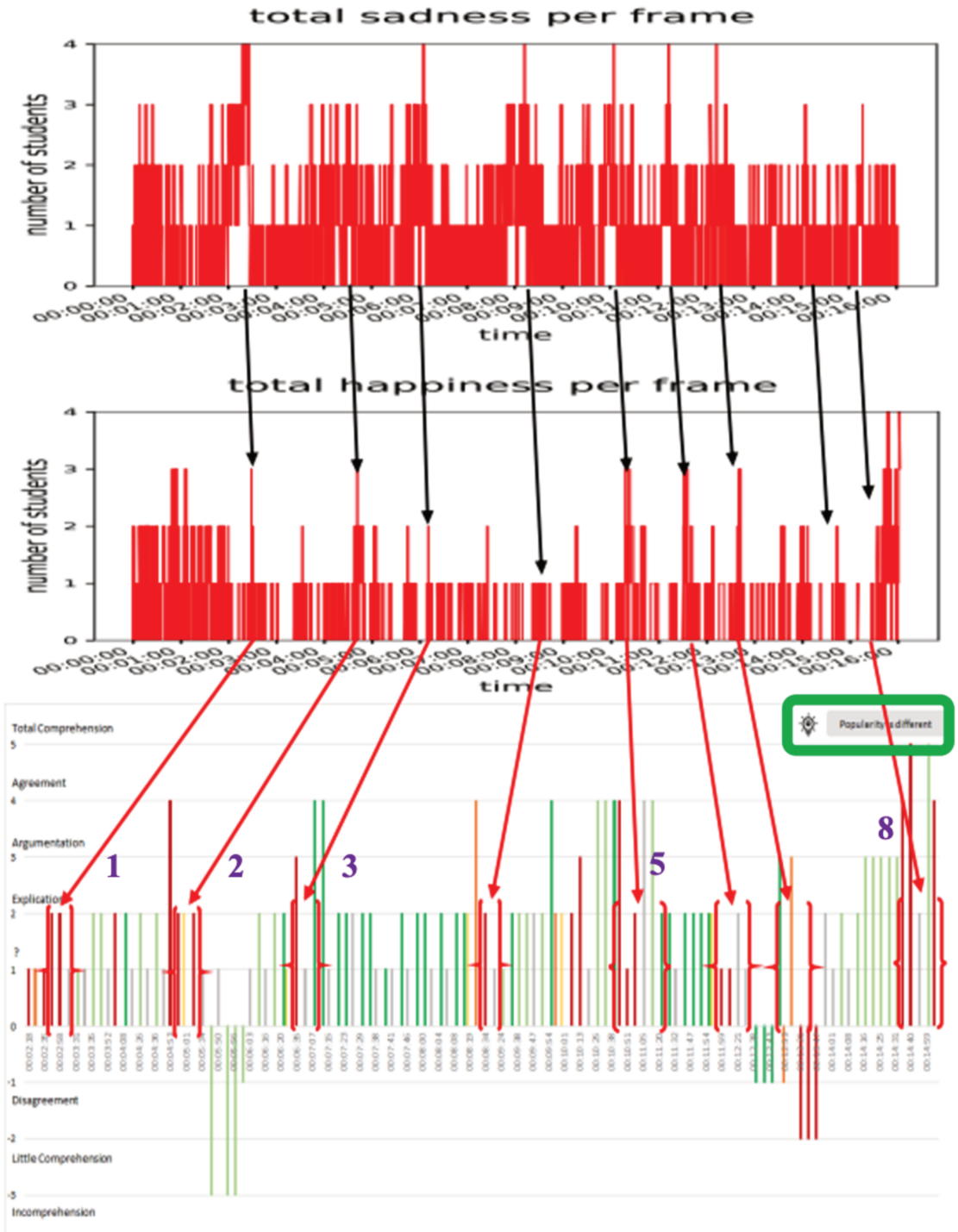

FIGURE 4.4 – Analyse conjointe : verbal et non verbal - des successions de tristesse puis joie correspondent à des phases d'explications et d'accord mutuel dans la vidéo (1,2,3,5). Effet eurêka observé en (8) correspondant à une compréhension totale

4.7 Conclusion

Nous avons présenté dans cette section une analyse conjointe des informations verbales (cognitives) et non verbales (affectives). Celle-ci nous a permis une confrontation intéressante de nos démarches respectives ainsi qu'un enrichissement mutuel de nos deux approches. Ces résultats préliminaires montrent une corrélation intéressante, en particulier, un effet Eurêka final, laissant supposer que les interactions intra- et inter-groupes aident à construire la connaissance. Nous noterons, toutefois, que l'analyse affective peut parfois être biaisée lorsque le tuteur interagit. Ce dernier joue un rôle prépondérant dans l'émergence et surtout dans la résolution du conflit socio-cognitif en orientant les discussions pour favoriser l'apparition du moment Eurêka. Parfois, nous avons observé que les jeunes élèves de l'école primaire d'un groupe laissent percevoir sur leur visage un malaise, symbolisé par une certaine forme de sourire forcé, lorsque le tuteur les fait participer avec l'autre groupe. En conséquence, notre analyse non verbale semble identifier, à plusieurs reprises, des "mini-moments" Eurêka tandis que l'analyse verbale considère cet effet comme la résolution finale du quiproquo, et donc, n'en détecte qu'un seul au cours de la séquence vidéo. Nous ne pouvons donc pas nous accorder sur la signification de ces mini-moments d'intercompréhension, qui reste pour le moment, une question ouverte.

Dans ce qui suit, nous revenons sur les limites concernant les émotions ; ces limites vont orienter la suite des travaux de ce mémoire vers une approche de personnalisation. Les bases de données d'émotions les plus populaires dont dispose la communauté scientifique sont généralement constituées uniquement d'émotions basiques. Ce type d'émotions est, en revanche, assez limité lorsqu'il s'agit d'analyser les émotions susceptibles de se produire dans les salles de classe ; ces dernières sont appelées émotions académiques [104]. À notre connaissance, des bases de données disponibles pour la communauté et constituées de telles émotions est rare. Le modèle de reconnaissance faciale utilisé ici a donc été entraîné, d'une part, sur des émotions basiques, et d'autre part, sur des visages d'adultes et peut donc ne pas bien fonctionner sur les enfants [105]. Néanmoins, les algorithmes doivent pouvoir fonctionner pour tous les élèves, quel que soit l'âge, et quelle que soit l'origine ethnique. Or, entraîner un modèle générique unique à tous les élèves, impliquerait que celui-ci puisse rencontrer des difficultés lorsqu'il serait confronté à des cas particuliers en raison de biais morphologiques et comportementaux. Notre but est donc, de spécialiser (personnaliser) le modèle à l'individu grâce à l'apprentissage incrémental en données. Celui-ci nous permettrait, à partir d'un modèle de référence commun à tous (*baseline*), de

le personnaliser à un ou plusieurs élèves, en fonction de données répétitives de ces étudiants que l'on recevrait au fil de l'eau. Cela permettrait de s'attaquer au biais d'identité mentionné précédemment grâce à un pipeline conçu pour la personnalisation.

Chapitre 5

Datasets

Contenu

5.1 Extraction des features 92
5.1.1 Prétraitements réalisés par la librairie OpenFace 92
5.1.2 Unités d'actions (AUs) 93
5.1.3 Textures (TX) 94
5.2 Extended Cohn-Kanade Dataset CK+ 94
5.2.1 Description des sujets 94
5.2.2 Description des données 94
5.2.3 Description des datasets utilisés 95
5.3 Compound Facial Expressions of Emotion (CFEE) 96
5.3.1 Description des sujets 96
5.3.2 Description des données 96
5.3.3 Description des datasets utilisés 96

Dans ce chapitre, on décrit les features et les datasets utilisés pour nos expérimentations.

5.1 Extraction des features

5.1.1 Prétraitements réalisés par la librairie OpenFace

OpenFace est un logiciel open source, développé par [106], que nous avons utilisé pour détecter les visages, les points caractéristiques, ainsi que 17 Action Units et leurs intensités (codées par des valeurs continues entre 0 et 5).

Détection du visage : la première étape de toute méthode d'analyse de visage est de détecter le visage. Le détecteur de visage de Viola et Jones est de loin le plus utilisé [107]. En raison de sa fiabilité pour les visages en vue frontale et sa rapidité de calcul, ainsi que la mise à disposition au public de modèles pré-entraînés (comme par exemple dans OpenCV [1]), il a longtemps été considéré comme l'algorithme de détection de visage de référence. La librairie OpenFace [2] que nous avons utilisée intègre celui-ci. Cette librairie propose deux autres alternatives pour détecter le visage : celle de la librairie traditionnelle dlib [3] et le *Multi-task Convolutional Neural Network* (MTCNN). Ce dernier s'est révélé plus performant que dlib et que les cascades de Viola et Jones. De plus, les auteurs d'OpenFace [106] confirment que le MTCNN est plus précis lorsqu'il est question de détecter des visages ayant une inclinaison ou une rotation importante, où la librairie dlib échoue plus souvent.

Détection des points caractéristiques du visage (*landmarks*) : les repères faciaux sont des points caractéristiques (*landmarks*) dans les régions faciales (cf. Figure 5.1) telles que les extrémités des sourcils, du nez, des yeux, de la bouche, etc. Les coordonnées de chaque point de repère, ou la texture locale d'un point de repère, sont utilisées comme vecteur de caractéristique en FER. OpenFace nous a permis de faire la détection de 68 points de repère faciaux sur chaque image contenant un visage.

1. https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html
2. https://github.com/TadasBaltrusaitis/OpenFace
3. http://dlib.net/face_landmark_detection.py.html

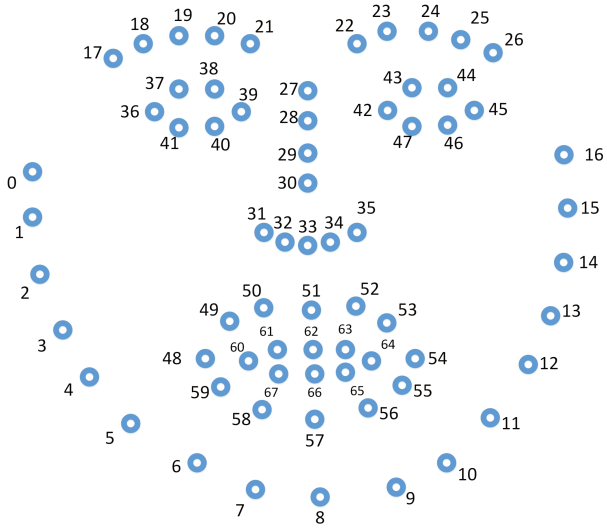

FIGURE 5.1 – Modèle de 68 points caractéristiques (*landmarks*) utilisés par OpenFace

5.1.2 Unités d'actions (AUs)

La bibliothèque OpenFace nous a permis d'extraire les activations de muscles spécifiques du visage appelés unités d'action (AU) [40]. La plupart des systèmes reposent généralement sur le système FACS [40] qui encode ces AUs. La production d'une AU a une évolution temporelle, typiquement modélisée par quatre segments temporels [6]. La figure 5.2, tirée de[4], illustre l'activation de muscles faciaux. Nous nous concentrerons ici uniquement sur les segments neutre et apex en raison des bases de données que nous utilisons dans cette étude (par exemple, la base de données CFEE, qui ne comprend que des images statiques). Neutre signifie qu'il n'y a aucun signe d'activité musculaire. Apex est le terme pour spécifier le cas où l'intensité de l'expression faciale atteint un niveau maximal. Chaque image de visage est donc encodée par 17 features à valeurs dans $[0; 5]$

FIGURE 5.2 – Activation de muscles faciaux lors d'expressions faciales - *tiré de imotions.com*

4. https://imotions.com/blog/facial-action-coding-system/

5.1.3 Textures (TX)

Enfin, nous avons extrait des caractéristiques bas niveau, à savoir des motifs binaires locaux (LBP) [108] et des histogrammes de gradients orientés (HoG) [109] sur les images de mêmes dimensions grâce au traitement réalisé par OpenFace (visages rognés et recalés) ; ces features sont disponibles avec la librairie *skimage.feature*[5]. Après concaténation, on obtient 224 features que nous utiliserons plus tard, dans le chapitre 7.

5.2 Extended Cohn-Kanade Dataset CK+

Le dataset CK+ est la base de données la plus populaire dans le domaine de la reconnaissance des émotions [43].

5.2.1 Description des sujets

On compte 123 participants âgés entre 18 et 50 ans, parmi lesquels 85 sont des femmes et 38 des hommes. Les origines ethniques sont variées avec principalement des européens-américains, des afro-américains ; une grande majorité des individus du dataset ont un profil caucasien.

5.2.2 Description des données

Ces participants ont été filmés en plan frontal dans des conditions dites de laboratoire : la même caméra prend la photo de tous les sujets pour l'émotion demandée, les captures ont lieu dans la même pièce, l'éclairage est contrôlé. Il y a également un minimum d'occultations faciales : les participants qui ont des lunettes les retirent et les hommes sont tenus de se raser. Il leur a été demandé par les expérimentateurs de produire une émotion parmi les six expressions basiques (colère, surprise, joie, peur, dégoût et tristesse), ainsi que l'expression du mépris. Au total, ce sont 593 séquences vidéo qui ont été collectées pour le dataset CK+. Pour chacune d'elles, l'émotion de départ est le visage neutre ; l'émotion finale est nommée l'apex et correspond à l'intensité maximale de l'émotion ; on observe ainsi, pour chaque séquence, une transition en images du visage neutre jusqu'à l'émotion souhaitée, voir Figure 5.3. Chaque séquence vidéo a une durée pouvant varier : en conséquence, le nombre d'images par séquence varie de 4 à environ 60 images. Toutes les séquences ne sont pas labélisées : au total, il

5. https://scikit-image.org/docs/stable/api/skimage.feature.html

y a 327 séquences labélisées ; nous avons décidé de retirer 18 images ayant l'émotion du mépris car elle n'apparaît pas dans tous les datasets. Il reste ainsi 309 séquences labélisées parmi les 6 émotions basiques. En général, nous prenons la dernière image de la séquence pour l'émotion car elle correspond à l'apex (intensité maximale). Les auteurs de [110] parviennent ainsi à extraire 309 images labélisées.

FIGURE 5.3 – Exemple d'une séquence vidéo de CK+ pour la surprise - la séquence commence avec le sujet à l'état neutre (0) et se termine sur l'état de surprise exprimé avec une intensité maximale (n)

5.2.3 Description des datasets utilisés

Nous avons également considéré l'expression Neutre (1ère image de chaque séquence) et arrivons de cette manière à rassembler 902 images (593 neutres + 309 apex labélisés).

On peut remarquer que, dans CK+, on ne possède pas la séquence de chaque émotion pour chaque individu : certains n'en n'ont qu'une, d'autres deux ou trois, quelques-uns les ont toutes.

Pour chaque séquence vidéo de n images, on note i_0 la première image correspondant à l'émotion neutre et i_n la dernière image correspondant à l'émotion d'intensité maximale (apex). On distingue grossièrement les expressions surjouées des expressions subtiles (intensité plus faible qu'apex) en assignant ces images aux sous-ensembles E et I comme suit :

— E contient l'ensemble des images i_0 et i_n ;

— I contient l'ensemble des images i_1 et i_{n-1}.

Nos datasets $X_E^{[CK+]}$ et $X_I^{[CK+]}$ ont les mêmes dimensions, et selon les features que l'on utilise, leurs dimensions sont :

AU : 902×17,

TX : 902×224.

5.3 Compound Facial Expressions of Emotion (CFEE)

5.3.1 Description des sujets

Le dataset *Compound Facial Expressions of Emotion* (CFEE) a été collecté à partir de 230 participants. Un peu plus de la moitié de ceux-ci sont des femmes, l'âge moyen est d'environ 23 ans. Différentes origines ethniques sont représentées telles que des profils caucasiens, asiatiques, afro-américains et hispaniques.

5.3.2 Description des données

Pour ce dataset, à la différence de CK+ vu précédemment, il s'agit d'une collection d'images statiques et non des séquences vidéos. Les visages sont capturés dans un plan frontal dans des conditions de laboratoire.

Le dataset comporte 22 émotions : 6 émotions basiques, 15 émotions composées (c'est-à-dire une combinaison de deux émotions prototypiques), et l'expression neutre [35].

5.3.3 Description des datasets utilisés

À partir des 5060 images contenues dans la base de données, nous avons pu extraire 1607 émotions basiques, dont les neutres.

On peut remarquer que dans cette base, contrairement à la précédente, on dispose de l'image de toutes les émotions pour tout individu la constituant.

Selon les features que l'on utilise, les dimensions du dataset que l'on dispose pour $X^{[CFEE]}$ sont :

AU : 1607×17,

TX : 1607×224.

Chapitre 6

Apprentissage incrémental supervisé en données

Contenu

6.1 Avantages et limites du classifieur NCM 98
6.1.1 Évaluation 98
6.1.2 Personnalisation 99
6.2 Stratégies d'apprentissage incrémental dans la NCMF 100
6.2.1 Stratégie mettant à jour les feuilles 100
6.2.2 Stratégie créant de nouveaux noeuds 101
6.3 Stratégie proposée de création conditionnelle d'un nouveau mode 102
6.3.1 Stratégie UpdateCentroïd (UC) 103
6.3.2 Stratégie AddCentroïd (AC) 104
6.3.3 Critère de choix de la solution optimale 104
6.4 Protocole expérimental 110
6.4.1 Préparation des datasets 110
6.4.2 Choix des hyperparamètres du modèle générique 114
6.4.3 Architecture du pipeline de personnalisation 117
6.5 Résultats 119
6.5.1 Qualité de la baseline 119
6.5.2 Comparaison des stratégies incrémentales sur CK+ 120
6.5.3 Performances par classe 122
6.5.4 Statistiques sur les stratégies d'IGTC 122
6.5.5 Performances inter-groupes des modèles personnalisés 124
6.5.6 Performances sur des expressions plus subtiles 125
6.6 Conclusions 127

Dans ce chapitre, nous nous intéressons à la personnalisation d'un modèle générique (*subject independent*) à un ensemble fini d'individus. Nous n'avons pu appliquer les algorithmes proposés à un unique individu en raison du très petit nombre de données disponibles pour chaque d'entre eux. dans ce chapitre, cette personnalisation "multi-sujets" est faite à l'aide de données labélisées, de façon supervisée.

Dans un premier temps, nous présentons, de façon intuitive, les différentes solutions d'incrémentation applicables à un noeud classifieur NCM. Puis, nous mettons en exergue un défaut de ce dernier. Comme il met en compétition les centroïdes de deux classes, il est parfaitement adapté aux distributions uni-gaussiennes. Par contre, il ne peut qu'échouer sur des données complexes dont les distributions conditionnellement aux classes comportent plusieurs modes.

Nous présentons ensuite les approches incrémentales proposées dans [19], incapables de gérer ce problème. Nous proposons une solution, s'appuyant sur un critère statistique de séparabilité des modes de la distribution locale, qui va nous permettre de mettre à jour le centroïde du noeud ou d'en créer un nouveau, si cela s'avère nécessaire.

La section suivante se concentre sur le protocole expérimental. Nous commençons par détailler les données dédiées à l'apprentissage et à l'évaluation du modèle générique, puis celles utilisées pour le personnaliser. Nous étudions en détail l'impact des différents hyperparamètres sur les performances du modèle générique. Enfin, nous présentons l'organisation de la phase de personnalisation.

La dernière section est dédiée à la présentation et l'analyse détaillée des performances du modèle générique (*baseline*) et des différentes stratégies d'incrémentation décrites et proposées.

6.1 Avantages et limites du classifieur NCM

6.1.1 Évaluation

Le noeud classifieur NCM, avec ses deux centroïdes $\{c_i, c_j\}, 1 \leq i, j \leq l, i \neq j$, peut orienter une observation x de la classe k_i dans la mauvaise direction. Autrement dit, x est orienté vers le nœud enfant associé au centroïde de l'autre classe c_j. Ceci peut se produire quand les données d'une classe sont très variables et qu'un seul centroïde ne suffit plus pour les représenter. Ce phénomène apparaît dès que la distribution conditionnelle des données de la classe k_i devient multimodale.

La figure 6.1 illustre ce phénomène. Des observations de deux classes sont présentes dans ce noeud $\mathcal{K} = \{\triangle, \square\}$. Le centroïde de chaque classe est le barycentre des observations présentes dans le noeud et est symbolisé par la lettre c_k sur la figure. La frontière, définie ici par la distance euclidienne aux deux centroïdes, illustre les décisions prises dans le noeud pour toute observation x. Si x est plus proche de $c_\square$ que de $c_\triangle$, il sera dirigé vers le fils gauche (associé au centroïde $c_\square$), sinon il sera orienté vers le fils droit (associé à $c_\triangle$).

L'observation x_5 est située du mauvais côté de la frontière (Fig.6.1.a et Fig.6.1.b) : elle sera orientée vers le noeud fils droit, associé au label $\triangle$, bien qu'elle soit de la classe $\square$. À ce stade, si l'un des critères d'arrêt de l'algorithme d'apprentissage (décrits dans la section Sec.2.2.1.1) est atteint, il ne sera plus possible de corriger cette erreur, car les noeuds enfants deviendront des feuilles. La donnée x_5 se retrouvera donc dans une feuille dont la classe majoritaire n'est pas la sienne. De même, toutes les observations proches de x_5 seront mal classées.

6.1.2 Personnalisation

Si maintenant des données comme x_5 apparaissent lors de la phase de personnalisation, on voit assez intuitivement que :

— les effectifs de la classe $\square$ peuvent augmenter et changer la classe majoritaire dans la feuille droite, changeant aussi la prédiction ;

— mettre à jour le centroïde $c_\square$ en utilisant x_5 permet de déplacer la frontière, de manière à bien classer cette observation. Cette stratégie pourrait fonctionner pour le cas Fig.6.1.(a).

Dans le cas Fig.6.1.(b), on voit clairement la dérive de la classe $\square$ qui se traduit par l'apparition d'un nouveau mode dans la distribution et la stratégie de mise à jour du centroïde décrite précédemment ne peut qu'échouer ; il devient indispensable d'instancier un nouveau centroïde avec l'observation x_5. Notons qu'au fur et à mesure de la phase de personnalisation et de l'arrivée de nouvelles observations, ce dernier pourrait être amené à se déplacer dans l'espace des caractéristiques propre au noeud.

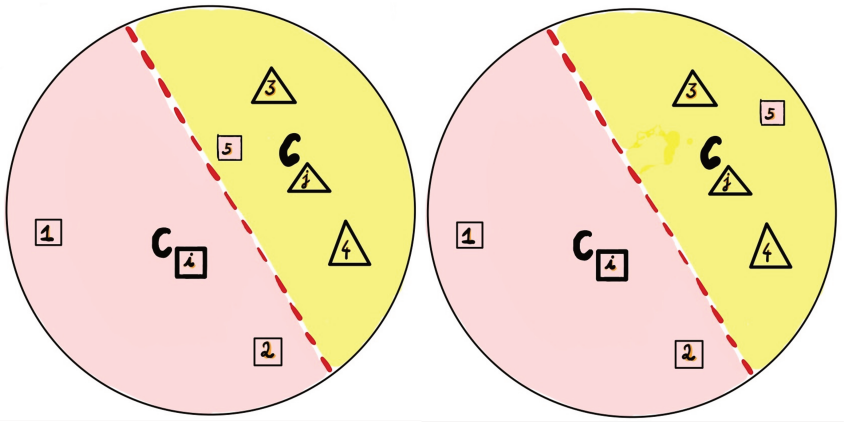

FIGURE 6.1 – Illustration de la mauvaise orientation de certains exemples lors de l'incrémentation - l'observation (5) sera dirigée vers le mauvais centroïde (j), le centroïde (i) peut se déplacer avec sa frontière pour le faire passer du bon côté en (a), mais ce n'est plus possible en (b) car le noeud devient multimodal.

6.2 Stratégies d'apprentissage incrémental dans la NCMF

Les stratégies présentées dans ce qui suit ont été proposées par les auteurs de la NCMF pour permettre d'étendre les capacités d'une forêt, pré-entraînée sur un nombre limité de classes, à classer de nouvelles classes qui n'étaient pas présentes dans les données d'apprentissage [19]. Elles ont été proposées dans un contexte de big data, où mettre à jour la forêt plutôt que de réapprendre tout depuis le début, se justifie en termes de temps de calcul par rapport à la phase d'apprentissage. On considère donc une première phase d'apprentissage classique par le modèle NCMF puis une deuxième phase incrémentale de mise à jour de la forêt.

6.2.1 Stratégie mettant à jour les feuilles

Dans la stratégie *Update Leaf Statistics (ULS)*, les fonctions de split calculées pendant la première phase d'apprentissage, le nombre de nœuds ainsi que la profondeur des arbres restent inchangés. Seules les occurrences des classes dans les feuilles de chaque arbre sont mises à jour comme suit. Soit $x^{[INCR]}$ une observation à apprendre de manière incrémentale, $y^{[INCR]}$ son label. L'observation est propagée dans l'arbre (pré-entraîné avec le dataset reçu jusqu'alors) comme lors d'une phase de prédiction, jusqu'à atteindre une feuille prédictrice l_i. L'algorithme ULS consiste alors à incrémenter de 1 l'occurrence du label de $x^{[INCR]}$ de chaque feuille prédictrice. Par exemple, disons que cette

observation se propage aux feuilles l_1 et l_3 des deux arbres de la figure 2.4 ; cela signifie que si le label $y^{[INCR]} = A$, les compteurs mis à jours de l_1 et l_3 deviennent $[1, 3, 1, 1]$ et $[2, 1, 1, 5]$ respectivement.

Si le label $y^{[INCR]}$ n'était pas présent dans $\mathcal{K}$ lors de la phase initiale d'apprentissage de la NCMF, on parle alors d'apprentissage incrémental en classe. Le principe de ULS reste similaire : les compteurs des feuilles prédictrices sont agrandis d'une cellule supplémentaire contenant la valeur 1, correspondant au nouveau label.

Au fur et à mesure que nous incrémentons les modèles avec de nouvelles données, les distributions évoluent et les prédictions sont susceptibles de changer si une nouvelle majorité apparaît au sein d'une feuille.

6.2.2 Stratégie créant de nouveaux noeuds

La stratégie IGT est au départ similaire à la stratégie ULS : les données incrémentales sont propagées dans chaque arbre de la forêt et les occurrences au niveau des feuilles prédictrices sont mises à jour, que ce soit de l'incrémentation en données ou en classe. Cependant, la différence par rapport à ULS est qu'on vérifie si l'incrémentation satisfait une condition pouvant mener à la construction locale d'un sous-arbre. La liste non exhaustive suivante peut donner des exemples de la condition à définir :

— un changement de label majoritaire a lieu ;

— une nouvelle classe égalise les occurrences du label majoritaire original de la feuille prédictrice ;

— les occurrences de certaines classes sont trop proches (en fonction d'un paramètre à définir) ;

— une classe différente du label majoritaire a un nombre d'occurrences dépassant un certain seuil (à définir).

Si c'est le cas, la feuille se transforme en un noeud, ce qui déclenche la construction récursive du sous-arbre à partir de cette position. Les données considérées par le sous-arbre sont toutes celles qui ont été présentes dans cette feuille, soit pendant l'apprentissage, soit pendant l'incrémentation.

On peut remarquer que, pour splitter les feuilles prédictrices, la stratégie IGT nécessite d'avoir conservé les données issues de la première phase d'apprentissage, ainsi que celles des phases incrémentales ayant potentiellement eu lieu.

Enfin, on peut noter que les stratégies incrémentales proposées dans le cadre de la NCMF [19] ne s'appliquent pas nécessairement qu'à ce modèle ; elles peuvent être également utilisées pour d'autres

types de modèles de forêt, comme les RF [111].

6.3 Stratégie proposée de création conditionnelle d'un nouveau mode

Nous supposons qu'une phase initiale d'apprentissage a permis de construire la NCMF. Ce modèle générique, entraîné sur un premier dataset, sera appelé par la suite *baseline*. Les performances de ce modèle (accuracy, accuracy par classe, confusions) sont mesurées à l'aide de sa matrice de confusion, estimée sur les données de test du premier dataset.

Lors de la personnalisation, les arbres seront incrémentés sur des données d'un second dataset, composés d'images de sujets différents de ceux du premier. Les stratégies ULS et IGT pourront être utilisées.

Toutefois, la variabilité entre les données de sujets vus pendant l'apprentissage du modèle et celles utilisées lors de la personnalisation à un groupe d'individus donné, peut être élevée tant au niveau inter-individuel (morphologies plus variées) qu'intra-individuel (émotion plus ou moins subtile ou marquée). Ceci peut se traduire par l'apparition d'un nouveau mode - voire de plusieurs modes - dans la distribution conditionnelle des données d'une classe comme décrit précédemment 6.1.2. Il est donc nécessaire, de (1) détecter cette dérive et de la quantifier afin de (2) créer un centroïde représentant un nouveau mode potentiel.

À cette fin, nous proposons la stratégie *Incremental Growing Tree Correction* (**IGTC**). Il s'agit d'une évolution de la stratégie IGT, s'appuyant sur la qualité des clusters présents dans un noeud (voir Sec.2.1.1.2). Cette nouvelle méthode consiste à appliquer une des deux solutions suivantes si une observation x de classe k_i a comme plus proche voisin le centroïde c_j : mettre à jour le centroïde c_i existant (appelée par la suite *UpdateCentroïd* - UC) ou ajouter un nouveau centroïde pour "mieux" modéliser la classe k_i (*AddCentroïd* - AC). Ces solutions, détaillées ci-dessous, permettent de prendre en compte les dérives potentielles, qu'elles soient progressives ou brutales.

Considérons une feuille transformée en noeud par la stratégie IGT et observons le classifieur NCM entraîné sur les données de ce noeud. Soit $\{c_i, c_j\}$ la paire de centroïdes associés aux classes $\{k_i, k_j\}$ et x une nouvelle observation de classe $k \in \{k_i, k_j\}$ *a priori* mal orientée par le classifieur. On note $n_k = |X_k^{(n)}|$ le nombre d'observations de la classe k du noeud n (contenant les données de la phase d'apprentissage et de personnalisation).

6.3.1 Stratégie UpdateCentroïd (UC)

La densité conditionnelle de la classe reste unimodale après l'incrémentation.

La mise à jour d'un centroïde en présence d'une nouvelle observation se fait classiquement comme suit :

$$\begin{cases} c_k \leftarrow & \dfrac{n_k \times c_k + x_k}{n_k + 1} \\ n_k \leftarrow & n_k + 1 \end{cases} \tag{6.1}$$

L'inconvénient majeur de l'équation 6.1 est que la modification appliquée au centroïde sera d'autant plus faible que n_k est grand. Nous choisissons donc de pondérer la nouvelle observation avec un poids $w_{i,j}$ proportionnel à la probabilité de confusion entre les classes k_i et k_j du modèle générique.

$$\begin{cases} c_k \leftarrow & \dfrac{n_k \times c_k + w_{i,j} \times x_k}{n_k + w_{i,j}} \\ n_k \leftarrow & n_k + w_{i,j} \end{cases} \tag{6.2}$$

où :

$$w_{i,j} = \begin{cases} \text{confusion}(i,j) & \text{si confusion}(i,j) > 0 \\ 1 & \text{sinon.} \end{cases} \tag{6.3}$$

Le poids $w_{i,j}$ est un entier calculé à partir de la probabilité de confusion de la *baseline* pour les classes correspondantes k_i et k_j, avec k_i le label de l'observation x, et k_j le label du centroïde vers lequel elle s'est par erreur dirigée. L'idée sous-jacente est de modifier d'autant plus le centroïde c_k que les performances du modèle générique sur la paire de classes (k_i, k_j) sont faibles. Si, au contraire, les performances sont élevées et les confusions rares, le poids $w_{i,j}$ sera, lui aussi, faible. En l'absence de confusion entre les deux classes, il ne sera que de 1 ; on retrouve l'équation de mise à jour classique, et malgré tout aura un impact sur le centroïde. Ce dernier se rapprochant de x, la frontière de décision se déplace dans la même direction, augmentant la probabilité de bien classer x (voir Fig. 6.2.a). La figure 6.4 illustre l'influence que peut avoir la mise à jour du centroïde. Cette solution est très bien adaptée aux dérives lentes, caractérisées par des changements morphologiques et émotionnels limités.

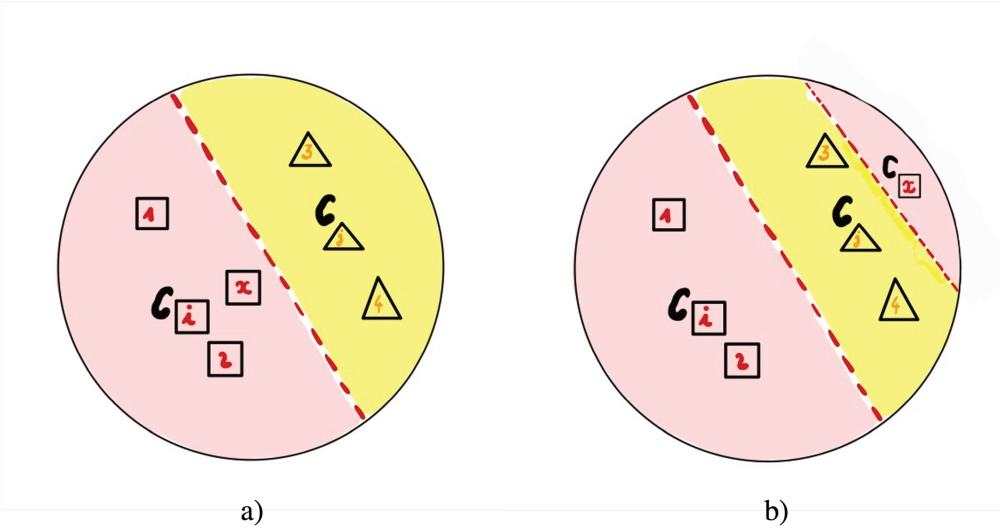

FIGURE 6.2 – Stratégies UC (a) et AC (b) utilisées lors de IGTC - le centroïde s'est déplacé vers l'observation (x) avec la stratégie UC (a), un nouveau centroïde est créé avec la stratégie AC (b) pour bien diriger l'observation (x) quand UC n'est plus applicable.

6.3.2 Stratégie AddCentroïd (AC)

Lors d'une dérive brutale, la solution précédente n'est plus efficace car elle risque de détériorer la frontière de décision. Il semble pertinent de considérer l'observation responsable de la dérive comme un nouveau mode de la distribution conditionnelle des données. Ici, l'observation x devient alors un nouveau centroïde de la classe k (dans le noeud n), voir Fig. 6.2.b. L'idée sous jacente de cette stratégie AC est de construire une frontière supplémentaire, via l'ajout d'un autre centroïde. Si, ultérieurement, un noeud enfant est créé, les deux centroïdes de classe k lui seront affectés.

6.3.3 Critère de choix de la solution optimale

Nous proposons d'utiliser l'indice de Calinsky-Harabasz (noté, par la suite, CH) [112] pour décider laquelle des deux solutions est la plus pertinente.

$$CH = \frac{I_B}{I_W} \times |X^{(n)}| - 2 \tag{6.4}$$

avec :

— I_B la variance inter-classe de la paire k_i, k_j (voir Eq.(2.2)),

— I_W la variance intra-classe k_i, k_j (voir Eq.(2.1)),

— $|X^{(n)}|$ l'effectif du noeud n pour l'ensemble des deux classes k_i, k_j,

— où 2 correspond au nombre de classes considérées dans le clustering du noeud n.

L'indice CH est une bonne mesure de la qualité de la modélisation. Il est utilisé en clustering où l'on cherche à maximiser le ratio de la variance inter-classe divisée par la variance intra-classe. Si les clusters sont de bonne qualité dans le noeud alors la classification du modèle NCM sera performante. Il a été montré récemment qu'il était robuste quand la dimension des données augmentait raisonnablement. Dans l'étude [1] , sa valeur reste stable jusqu'à 10^3 features. Mais l'étude n'a, malheureusement, pas testé des espaces de dimensions supérieures.

Plus I_B est grand, meilleure est la discrimination entre les classes. Plus I_W est petit, plus la vraisemblance du modèle augmente : les données sont plus proches de leur centroïde et le cluster correspondant, plus compact. Plus ce ratio sera élevé, meilleures seront les performances du noeud en discrimination. On peut logiquement supposer que cela impactera favorablement les performances des arbres de la forêt en diminuant progressivement leur biais et donc à une amélioration des performances de cette dernière.

Nous calculons deux indices pour décider de la stratégie à appliquer. Le premier est noté CH_1 et se calcule en considérant les clusters formés par les données $X_i^{(n)} \cup X_j^{(n)} \cup \{x\}$ sans modifier le label de l'observation x, autrement dit, en supposant que x est correctement classé. Le deuxième, noté CH_2, se calcule en considérant les clusters formés par les données où l'on a permuté le label de x avec celui de l'autre classe. Nous expliquons ci-dessous, à l'aide d'un exemple jouet, pourquoi nous décidons d'appliquer la stratégie AC si CH_1 est inférieur ou égal à CH_2 et la stratégie UC, sinon.

La figure 6.3 illustre le processus automatique proposé pour décider de la stratégie à appliquer :

Scénario (a) : Deux clusters sont clairement visibles. Une observation (à droite) de la classe □, indiqué sur fond jaune, se trouve du mauvais côté de la frontière et ira donc vers le nœud enfant associé à la classe △. Ici, le premier indice de Calinski-Harabasz notée CH_1 est calculé. Ensuite, un second indice CH_2 est calculé, faisant passer l'observation pour l'autre classe (on permute artificiellement son label avec celui du triangle vert). Comme on le remarque ici, la position des observations de la classe △ est plus éloignée de $c_\triangle$ que celles de la classe □ à $c_\square$. Conclusion : la variance intra-

1. https://imada.sdu.dk/Research/EDML/2019/Radovanovic-ClusteringEvaluation.pdf

classe sans permutation est plus petite qu'avec et donc $CH_1 > CH_2$; dans ce cas, la stratégie UC est appliquée. En effet, cela signifie qu'en mettant à jour le centroïde $c_\square$, et ce, en accordant plus de poids à l'observation proche de la frontière, il se déplacera vers cette observation, et donc la frontière également.

Scénario (b) : Dans le deuxième scénario (b), la donnée de classe bleue se trouve du mauvais côté de la frontière. À nouveau, deux mesures CH_1 et CH_2 sont calculées, d'abord en conservant le label de l'observation, puis en le permutant avec celui de l'autre classe. Les observations de classe $\triangle$ sont plus proche de leur centre que ne le sont celles de classe $\square$ avec leur centre respectif, il en résulte une variance intra-classe plus petite et donc $CH_1 < CH_2$. Dans ce cas, on calcule deux distances euclidiennes ; la première, d_1, correspond à l'écart entre l'observation mal placée et son centroïde ; la deuxième, d_2, correspond à l'écart entre les deux centroïdes. Ici, $d_1 < d_2$, on en conclut que la stratégie à appliquer est UC. En effet, cela signifie qu'en mettant à jour le centroïde $c_\square$, et ce, en accordant plus de poids à l'observation proche de la frontière, il se déplacera vers cette observation, et donc la frontière également.

Scénario (c) : Dans le dernier scénario (c), la configuration est très proche du scénario (b), à la différence que dans ce cas, $d_1 > d_2$, on en conclut que la stratégie à appliquer est AC. Dans une telle situation, il n'est, *a priori*, pas possible de mettre à jour le centroïde $c_\square$ avec la stratégie UC de manière à pouvoir tracer une frontière linéaire entre les observations des deux classes. Cela signifie que le centroïde $c_\square$ initial ne suffit pas à capturer toute la variabilité des données de sa classe : un second centroïde est nécessaire initialisé avec cette observation.

L'algorithme 1 résume les étapes précédentes pour gérer le cas d'une observation qui se dirige vers le centroïde de l'autre classe pendant la procédure IGTC. Par ailleurs, nous incrémentons seulement les arbres qui ne reconnaissent pas correctement la classe d'une observation ($\hat{y} \neq y$).

Algorithm 1 Handling misdirection in IGTC

Require: x is an observation misdirected in node n, best centroids in n are already calculated : 'centroids = $\{c_i, c_j\}$', and centroids' labels are 'clabels = $\{k_i, k_j\}$'. The label of x is either k_i or k_j ; to simplify the reading, let ' s say that its label is k such that $k \in \{i, j\}$, and that it has been oriented to the centroid of the other label $k' \in \{k_i, k_j\}$, $k' \neq k$; when we refer in the algorithm to c_k, it is either c_i or c_j depending on the value of k. CH_1 and CH_2 are already calculated. Confusion matrix is normalized, and available to each tree of the forest and to each node.

Choose the strategy to apply :
if $CH_1 \geq CH_2$ **then**
  strategy $\leftarrow UC$
else
  $d_1 \leftarrow dist(x, c_k)$
  $d_2 \leftarrow dist(c_i, c_j)$
  if $d_1 \leq d_2$ **then**
    strategy $\leftarrow UC$
  else
    strategy $\leftarrow AC$
  end if
end if
Apply the strategy :
if strategy is AC **then**
  add x to centroids {centroids = $\{c_i, c_j, x\}$}
  add k to centroids' labels {clabels = $\{k_i, k_j, k_i\}$ or clabels = $\{k_i, k_j, k_j\}$ depending on the label of x}
else
  if confusion$(k, k') > 0$ **then**
    $w = round(100 \times \text{confusion}(k, k'))$
  else
    $w \leftarrow 1$
    $c_k \leftarrow (n_k \times c_i + w \times x)/(n_k + w)$
    $n_k \leftarrow n_k + w$
  end if
end if

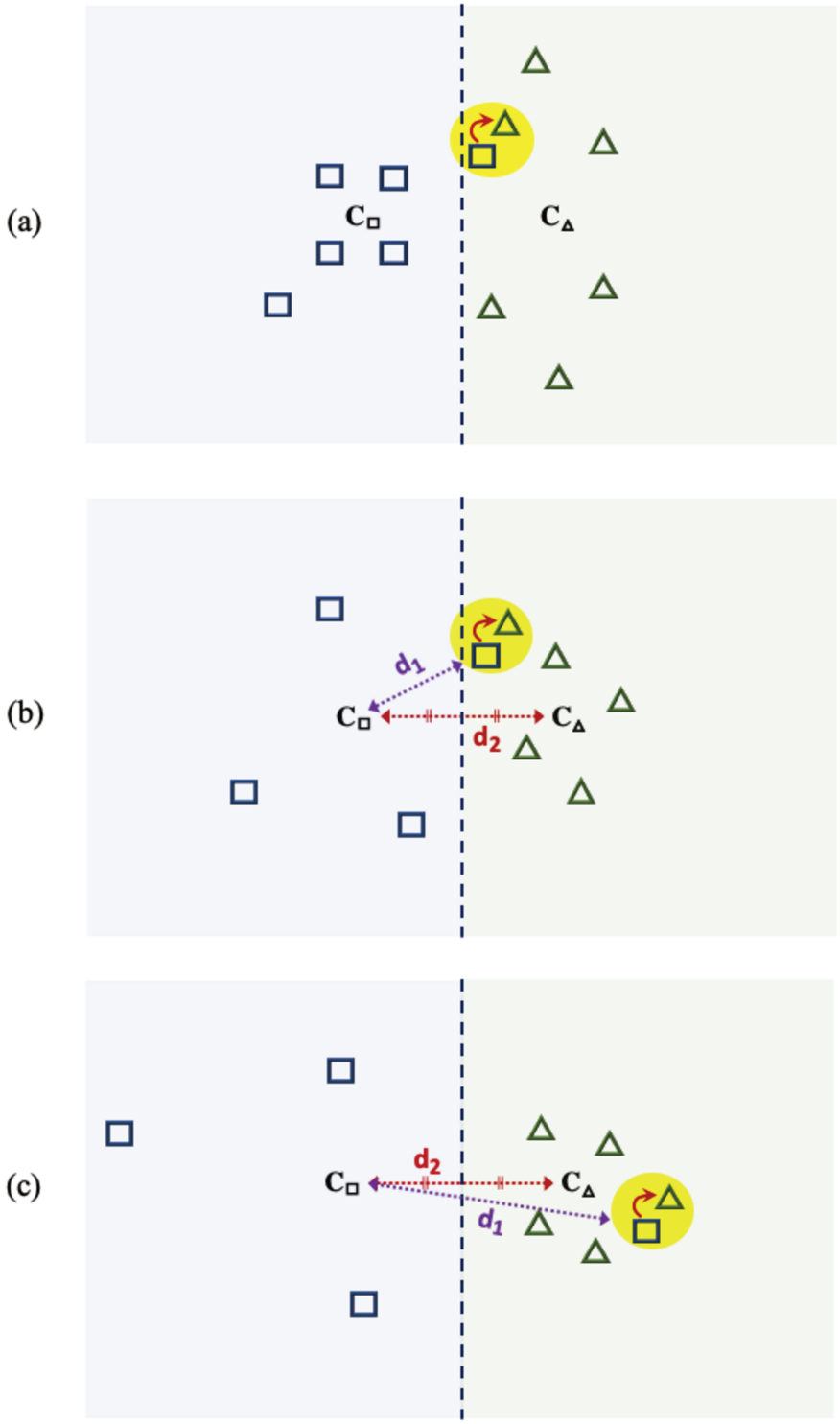

FIGURE 6.3 – Choix de la stratégie basé sur la comparaison des deux indices de Calinski-Harabasz - quand $CH_1 > CH_2$, UC est appliqué (a), sinon, UC est encore appliqué si $d_1 \leq d_2$ (b), dans le cas contraire, AC est appliqué (c).

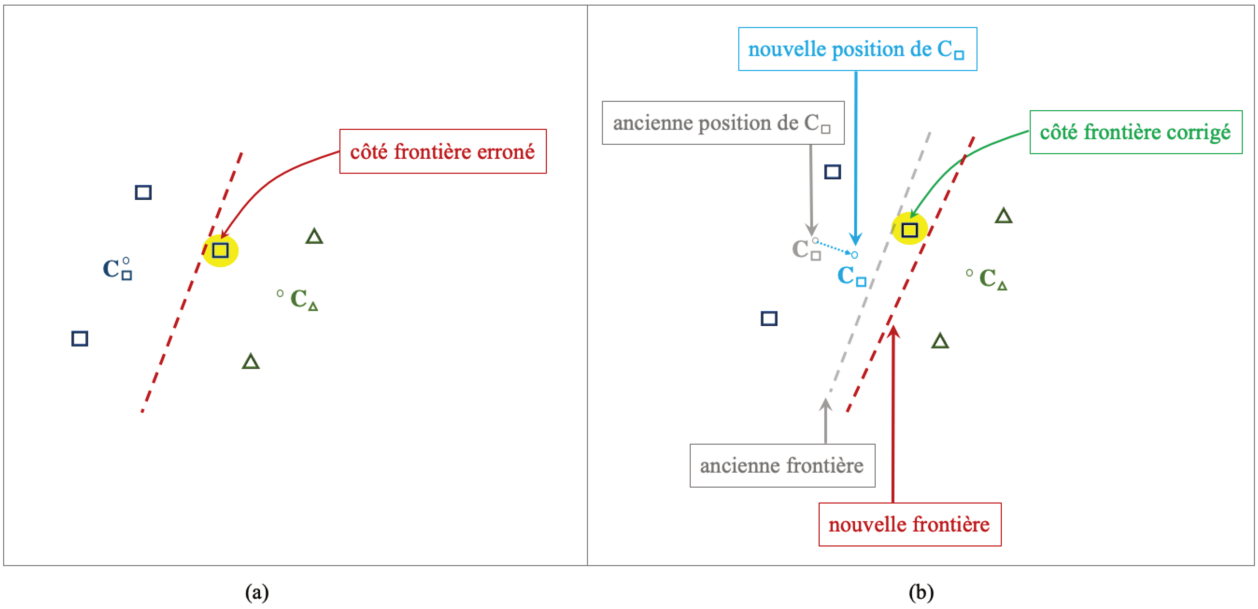

FIGURE 6.4 – Illustration de la stratégie UC : (a) initialement, une observation (sur fond jaune) est mal classée. Après avoir mis à jour le centroïde, (b) l'observation est correctement classée.

6.4 Protocole expérimental

Cette section décrit le protocole mis en place pour entraîner un modèle générique avant de le personnaliser à des groupes d'individus via une procédure d'apprentissage incrémental supervisé en données. On précise, tout d'abord, la manière dont on a utilisé les datasets CFEE et CK+ (décrits dans le Chapitre 5). Puis, on réalise une analyse détaillée du modèle générique en précisant comment les différents hyper-paramètres de la forêt NCMF ont été optimisés et leur impact sur ses performances et son architecture. Enfin, on présente la procédure complète de personnalisation.

6.4.1 Préparation des datasets

Par la suite, le dataset CFEE est utilisé pour la phase d'apprentissage du modèle générique. Le dataset de référence CK+ est utilisé pour la phase de personnalisation incrémentale et son évaluation.

Entraîner un modèle sur CFEE et l'évaluer sur CK+ permet tout d'abord de tester les performances de généralisation du modèle sur de nouveaux sujets. Par ailleurs, la proportion des données dans chaque classe de CFEE est quasiment uniforme car on dispose pour chaque sujet de son image pour toutes les émotions, ce qui n'est pas le cas de CK+. Le modèle générique sera ainsi moins biaisé puisqu'il est entraîné sur la base équilibrée CFEE. De plus, il y a moins de données dans CK+ ; lorsque les données d'une tâche ne sont pas forcément disponibles en grande quantité, il est alors possible de tirer partie des connaissances passées sur des tâches similaires afin de pouvoir mieux généraliser sur la nouvelle tâche [113].

Une autre raison qui a porté notre choix sur CFEE comme jeu d'entraînement est qu'il comporte des images statiques. Dans CK+, les données sont des séquences vidéos. Puisqu'ici nous nous intéressons à l'incrémentation en données, il est intéressant de pouvoir mettre à jour le modèle avec des images de CK+ potentiellement différentes de la première image (correspondant au visage neutre) et de la dernière (correspondant à l'apex) de la séquence vidéo. Cela permettra de mesurer la capacité de la phase de personnalisation à gérer des émotions plus ou moins subtiles.

6.4.1.1 Données d'apprentissage CFEE

Lorsqu'on se réfère désormais à CFEE, on se réfère au sous-ensemble de données de CFEE ne contenant que les émotions basiques. Seules les émotions basiques sont utilisées afin que les labels traités

soient communs aux deux datasets : $|\mathcal{K}^{[CFEE]}| = |\mathcal{K}^{[CK+]}| = 7$. À partir du dataset initial contenant 1607 observations décrites par 17 features (seules les AUs ont été utilisées ici), nous avons décidé de conserver 80% des données pour l'apprentissage, soient 1286 données (voir Fig. 6.5). Le nombre de données varie de 173 à 191 par classe. On appelle par la suite ce sous-ensemble d'apprentissage $(X, Y)_A^{[CFEE]}$ ou, plus simplement, $CFEE_A^{[AU]}$.

On pourra noter ici que la séparation ne tient donc pas compte des individus, ni des émotions et qu'elle est faite de manière aléatoire sans contraintes particulières hormis celle d'avoir des classes équilibrées.

6.4.1.2 Données d'évaluation intermédiaires CFEE

Les 20% du dataset CFEE, soient 321 données constituent un ensemble d'évaluation intermédiaire noté $(X, Y)_E^{[CFEE]}$ ou $CFEE_E^{[AU]}$ (cf.Fig. 6.6). Le nombre de données par classe varie de 38 à 56. On précise "intermédiaire" car ce jeu de données ne servira pas dans l'évaluation finale. Il va nous permettre d'évaluer les performances du modèle générique mais aussi de calculer la matrice de confusion et vérifier si un oubli catastrophique a lieu après incrémentation.

La matrice de confusion (cf. Fig. 6.9) nous permet d'avoir un aperçu des éventuelles confusions de la *baseline* après l'apprentissage initial. De plus, les valeurs de confusions vont nous servir lors de l'incrémentation avec la méthode IGTC afin d'attribuer des poids aux observations en fonction de leur classe d'appartenance. Par la suite, on pourra observer l'évolution de ces confusions lorsque l'on réalisera de l'incrémentation en données via CK+.

Concernant l'oubli catastrophique, nous souhaiterions mesurer l'impact de la personnalisation aux sujets provenant de CK+, sur les performances initiales de notre modèle *baseline*. Cet indicateur nous permettra donc de suivre si le modèle souffre d'un éventuel oubli catastrophique au cours de la phase d'incrémentation. Il est donc fixé à l'avance et n'est pas modifié dans la suite de l'expérimentation (aucune donnée de CK+ ne sera incluse dedans).

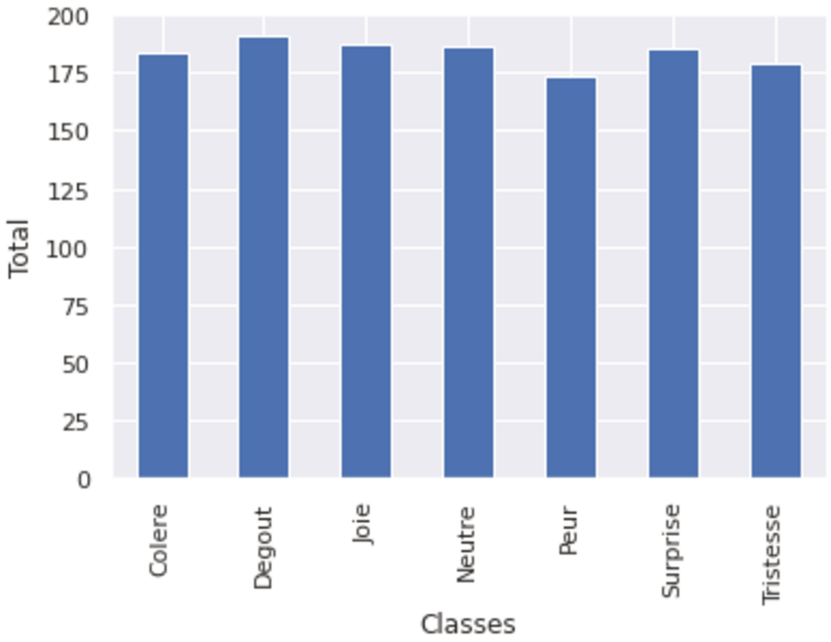

FIGURE 6.5 – Distribution des données par classe dans $(X, Y)_A^{[CFEE]}$

FIGURE 6.6 – Distribution des données par classe dans $(X, Y)_E^{[CFEE]}$

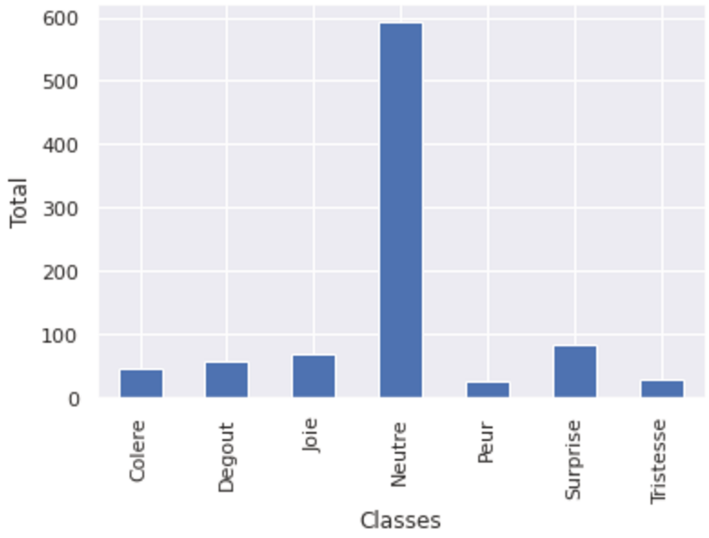

FIGURE 6.7 – Distribution des classes dans CK+

6.4.1.3 Données d'incrémentation CK+

Cette base est constituée pour réaliser l'incrémentation en données et est notée $(X, Y)_I^{[CK+]}$. Le classifieur baseline a donc déjà appris les classes constituant CK+ sur $(X, Y)_A^{[CFEE]}$. L'incrémentation se fera donc en données avec celles provenant de CK+. Cette dernière comprend 902 données. Pour chaque séquence vidéo labélisée, les images 1 et $n-1$ ont été sélectionnées pour constituer la base d'incrémentation, voir Fig. 5.3 illustrant une séquence reprenant les indices précédents.

Puisque les images 1 et $n-1$ sont respectivement proches de 0 et n, on leur attribue les mêmes labels, à savoir neutre et apex, en supposant que l'erreur de labélisation engendrée est minime.

6.4.1.4 Données d'évaluation CK+

Cette base contient le même nombre de données que $(X, Y)_I^{[CK+]}$ puisqu'elle a été constituée à partir des mêmes séquences vidéos. La différence est qu'on sélectionne ici les images 0 et n (neutre et apex). Ce dataset comprend donc également 902 données et est noté $(X, Y)_E^{[CK+]}$.

Dans le dataset CK+, les effectifs des classes varient varient de 25 à 593. La classe "neutre" est très majoritaire, étant donné que la première *frame* de chaque séquence vidéo est considérée comme neutre. La distribution de classe est identique pour les bases d'incrémentation et d'évaluation (Fig 6.7).

6.4.2 Choix des hyperparamètres du modèle générique

On entraîne un modèle NCMF sur $(X, Y)_A^{[CFEE]}$ où chaque observation x est constituée d'un vecteur F contenant les intensités des 17 AUs.

Le meilleur compromis entre un score *oob* élevé et un temps d'apprentissage faible est une forêt de 50 arbres (voir Tableau 6.1).

Afin de dé-corréler les arbres de la NCMF, un bootstrap a premièrement été réalisé pour chacun d'eux : tirage aléatoire avec remise pour constituer pour chaque arbre un dataset de mêmes dimensions que celles de $(X, Y)_A^{[CFEE]}$. Ensuite, dans chaque noeud, une sélection aléatoire de n_F features a lieu (RFS), ainsi qu'une sélection aléatoire de classes $n_{\mathcal{K}}$ (RCS). Plusieurs critères d'arrêt sont possibles lors de la construction d'un arbre. Ici, nous avons laissé les arbres se construire sans définir un critère de profondeur maximale. La construction d'une sous-branche se termine donc lorsqu'un noeud est

Nombre d'arbres $\|T\|$	**score oob**
10	0.68 ± 0.02
20	0.68 ± 0.02
50	0.69 ± 0.02
100	0.70 ± 0.02

TABLE 6.1 – Sélection du nombre d'arbres en fonction du score oob

pur, ou lorsque le nombre minimal d'exemples n_{stop} est atteint dans le noeud. Nous avons réalisé une *5-fold cross validation sur* $(X, Y)_A^{[CFEE]}$ pour trouver les valeurs optimales des hyperparamètres qui maximisent l'accuracy de la 5-fold cross validation, ce qui revient à un *split* d'environ 80% à chaque itération ; les résultats sont réunis dans les tableaux 6.2, 6.3 et 6.4.

Le tableau 6.5 résume les paramètres optimaux, utilisés pour entraîner la baseline qui a servi dans les expérimentations suivantes.

n_F	**5-fold acc**
1	0.763 ± 0.01
2	0.788 ± 0.01
3	0.785 ± 0.01
4	0.791 ± 0.01
5	0.794 ± 0.01
6	**0.826 ±0.01**
7	0.819 ± 0.01
8	0.819 ± 0.01
9	0.801 ± 0.01
10	0.813 ± 0.01
11	0.813 ± 0.01
12	0.810 ± 0.01
13	0.804 ± 0.01
14	0.794 ± 0.01
15	0.801 ± 0.01
16	0.791 ± 0.01
17	0.794 ± 0.01

TABLE 6.2 – 5-fold cross validation pour trouver le nombre optimal de features n_F

n_{stop}	**5-fold acc**
2	0.802 ± 0.010
3	0.801 ± 0.014
4	0.799 ± 0.016
5	**0.809 ± 0.009**
6	0.794 ± 0.017
7	0.800 ± 0.010
8	0.796 ± 0.008
9	0.797 ± 0.009
10	0.807 ± 0.010

TABLE 6.3 – 5-fold cross validation pour trouver le nombre optimal de n_{stop}

$n_{\mathcal{K}}$	**5-fold acc**
2	**0.804 ± 0.015**
3	0.798 ± 0.004
4	0.794 ± 0.014
5	0.799 ± 0.014
6	0.793 ± 0.013
7	0.801 ± 0.016

TABLE 6.4 – 5-fold cross validation pour trouver le nombre optimal pour $n_{\mathcal{K}}$

Paramètre	**Valeur**
$\|T\|$	50
n_F	6
$n_{\mathcal{K}}$	2
n_{stop}	5

TABLE 6.5 – Récapitulatif des paramètres utilisés pour entraîner la baseline NCMF

6.4.3 Architecture du pipeline de personnalisation

La figure 6.8 illustre les différentes étapes du pipeline de personnalisation proposé. Cette architecture a été réalisée pour que le modèle puisse classifier des observations pour des individus vus lors de la phase d'incrémentation. Si le modèle a été correctement personnalisé à l'individu, il est attendu une performance plus grande dans la reconnaissance des expressions subtiles (plus éloignées des extrêmes) de celui-ci. Les étapes sont les suivantes :

Étape 1 : apprentissage du modèle générique

Étape 2 : phase de personnalisation. Pour évaluer la capacité du modèle à s'adapter à un groupe de sujets, la base de données $CK+$ est divisée en $n_G = 8$ sous-groupes dits *subject independant* qui n'ont pas d'individus en commun (un groupe peut tout de même contenir les données de plusieurs sujets). L'affectation des sujets aux différents groupes est aléatoire. Le modèle initial utilisé pour personnaliser aux individus de chaque sous-groupe est une copie du modèle générique entraîné à l'étape 1. Ainsi, les n_G copies du modèle générique, disposant d'une base de connaissances initiales communes, vont progressivement se personnaliser aux individus de chaque sous-groupe.

Les modèles personnalisés sont par la suite notés $\mu_I^{(i)}$ (où i correspond à l'identifiant du groupe) et la stratégie incrémentale utilisée est IGTC (voir Sec. 6.3).

Étape 3 : évaluation de la capacité de personnalisation de la NCMF incrémentée. Chaque modèle $\mu_I^{(i)}$ est évalué sur le groupe de test qui lui est associé, contenant les mêmes sujets que dans la phase incrémentale. Il y a donc n_G modèles incrémentés et autant de mesures d'accuracy collectées. L'accuracy moyenne permet d'évaluer la capacité du modèle à se personnaliser.

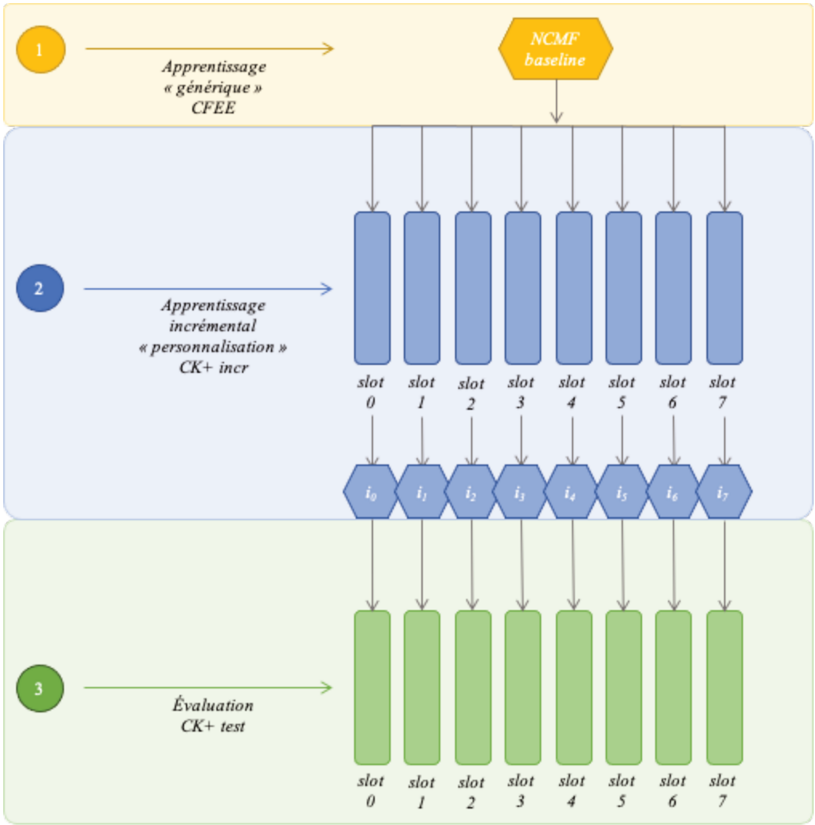

FIGURE 6.8 – Pipeline de l'apprentissage personnalisé incrémental supervisé - un modèle NCMF générique est entraîné sur CFEE (1), puis spécialisé sur de nouveaux sujets issus de CK+ (2), enfin, le modèle est évalué sur sa capacité de personnalisation à ces derniers en (3).

À noter que l'on peut sauter l'étape 2 de personnalisation pour immédiatement évaluer la NCMF baseline sur $(X, Y)_E^{[CK+]}$ et ainsi obtenir le score du modèle générique sans personnalisation car il n'aura pas déjà rencontré les sujets de $CK+$.

6.5 Résultats

6.5.1 Qualité de la baseline

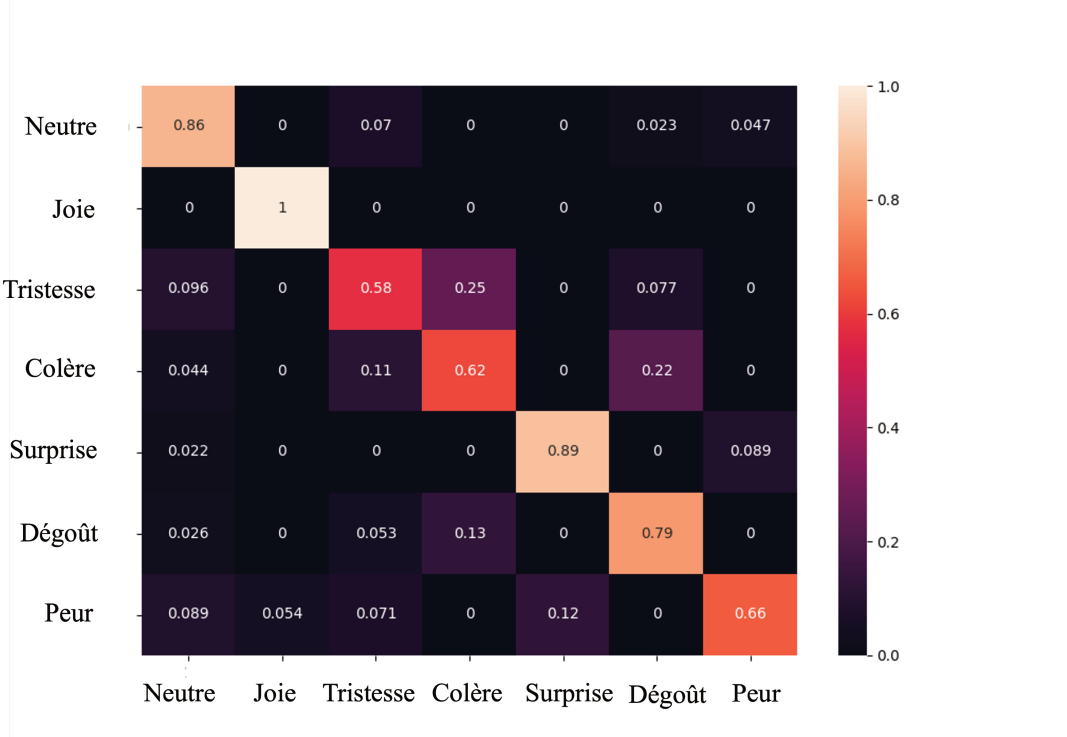

FIGURE 6.9 – Matrice de confusion sur $CFEE_E^{[AU]}$ (normalisée en lignes) - le modèle reconnaît de manière correcte les états neutre et joie, mais a des confusions pour les états de tristesse, colère, dégoût.

La Figure 6.9 présente la matrice de confusion calculée sur $(X, Y)_E^{[CFEE]}$. Les valeurs sur la diagonale de la matrice correspondent au taux de reconnaissance par classe. On peut observer que les classes Neutre, Joie et Surprise sont plutôt bien reconnues (plus de 80% de prédictions correctes pour ces classes). On remarque cependant que des confusions sont présentes entre certaines classes. On observe des confusions entre les classes Tristesse et Colère : confusions déjà évoquées dans la littérature [114] [115] [116]. On observe également des confusions entre la colère et le dégoût.

Comme l'illustre bien la Figure 6.10 sur des images du dataset CFEE, il est même difficile pour un humain de distinguer ces émotions. On remarque que l'AU4, correspondant au froncement des sourcils, est activée pour les trois émotions. D'autres AUs sont partagées entre les différentes émotions. Il n'est donc pas surprenant qu'un modèle entraîné uniquement sur les intensités des AUs fasse lui aussi des confusions entre ces trois émotions.

FIGURE 6.10 – Confusions entre tristesse (a), colère (b) et dégoût (c) - les sourcils sont froncés (1ère ligne) et les bouches tirées vers le bas (2ème ligne) pour les différents états expressifs, entraînant des confusions.

Le score obtenu par la baseline sur le dataset intermédiaire $(X, Y)_E^{[CFEE]}$ est de 0.76. On le mesurera de nouveau après la phase d'incrémentation pour déterminer si un oubli catastrophique a eu lieu.

6.5.2 Comparaison des stratégies incrémentales sur CK+

La figure 6.11 montre l'accuracy moyenne calculée sur les n_G groupes, obtenue par le modèle générique et le modèle personnalisé en fonction de la stratégie utilisée et du nombre d'arbres dans la forêt. L'incrémentation basée sur la stratégie ULS améliore les performances mais nous pouvons remarquer que la stratégie IGT est plus performante. Enfin, la stratégie IGTC a le plus grand impact, dès lors que le nombre d'arbres est supérieur à 10. La forêt incrémentale avec 50 arbres obtient les meilleures performances.

La figure 6.12 montre plus en détail les différences de performances entre IGTC et IGT en faisant évoluer le nombre d'arbres : à partir de 20 arbres, la stratégie IGTC surpasse IGT. Les intervalles de confiance correspondent aux variances des performances sur l'ensemble des 8 slots. On peut remarquer que ces dernières sont légèrement plus faibles avec IGTC.

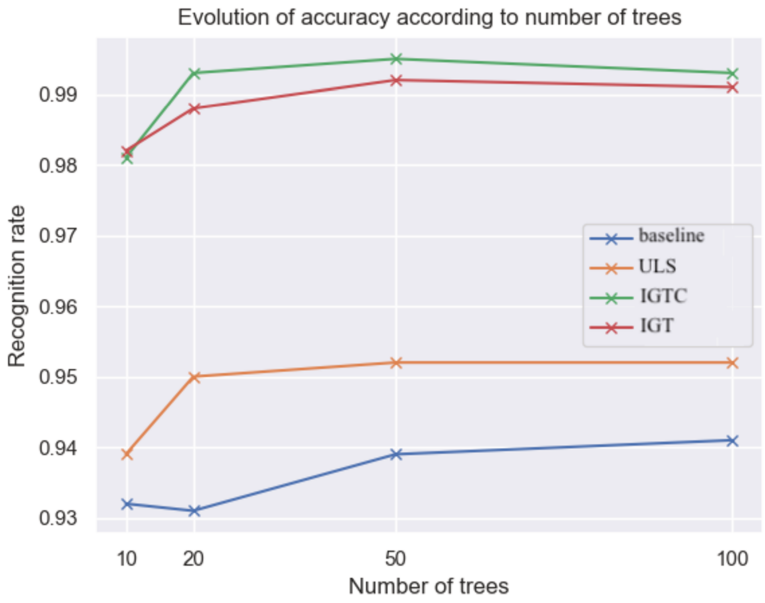

FIGURE 6.11 – Performances des modèles incrémentés en fonction du nombre d'arbres sur $(X, Y)_E^{[CK+]}$

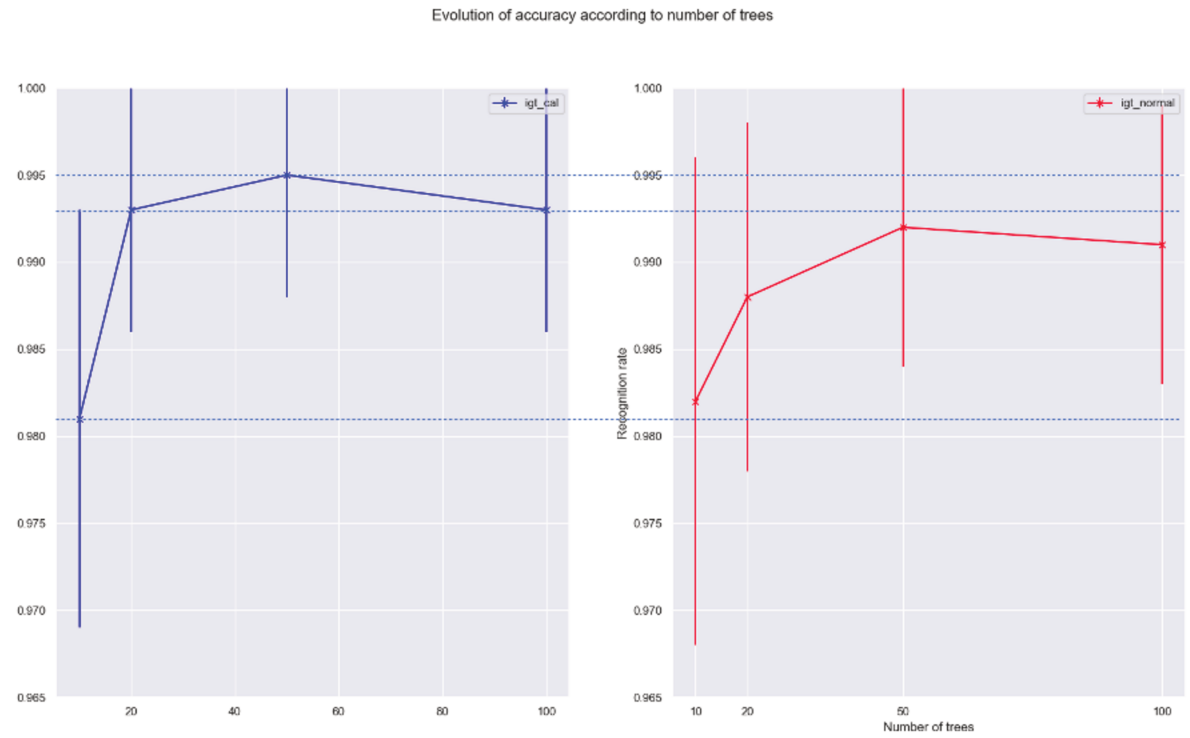

FIGURE 6.12 – Comparaison IGTC (à gauche) et IGT (à droite) - l'accuracy est plus élevée avec IGTC et les écarts types plus réduits qu'avec IGT.

6.5.3 Performances par classe

La Fig. 6.13 indique l'accuracy moyenne par classe obtenue sur les différents sous-groupes de $(X, Y)_E^{[CK+]}$, avant et après apprentissage incrémental. Dans tous les cas où ils ne sont pas égaux, c'est IGTC qui surpasse la reconnaissance des différentes labels. Si on regarde les classes "Tristesse" et "Colère", on peut observer que l'apprentissage incrémental utilisant la méthode IGTC améliore considérablement le taux de reconnaissance (accuracy) de la classe. La raison en est que la méthode IGTC utilise les confusions de la NCMF sur l'ensemble d'apprentissage. Les confusions les plus frappantes concernent principalement les classes Tristesse, Colère et Peur (voir Fig. 6.9). Le poids étant plus important pour ces classes (cf. section 6.3), on peut observer une nette amélioration des performances du modèle NCMF qui a porté son attention sur la bonne séparation de ces classes lors de la phase incrémentale. Pour un cas extrême comme la tristesse, le taux de reconnaissance de la baseline a subi une augmentation de presque 80% avec la stratégie IGTC.

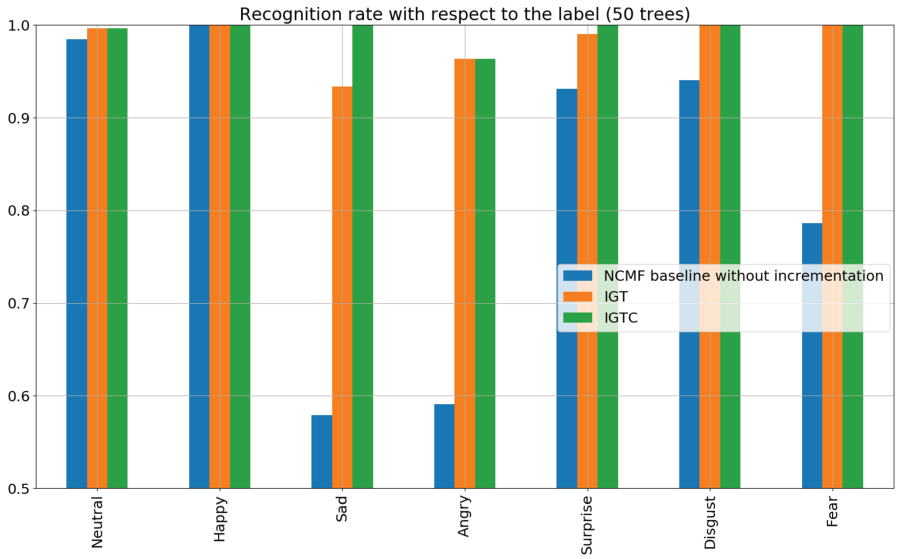

FIGURE 6.13 – Taux de reconnaissance par classe sur $(X, Y)_E^{[CK+]}$ - les lacunes de la baseline pour reconnaître la tristesse et la colère ont été comblées par les stratégies IGT et IGTC.

6.5.4 Statistiques sur les stratégies d'IGTC

On rappelle que la méthode IGTC active conditionnellement deux stratégies : AC (ajout de centroïde) et UC (mise à jour de centroïde).

La figure 6.14 montre le nombre d'activation moyen des deux stratégies lors de la phase d'incrémentation selon le groupe associé au modèle. On peut observer que la stratégie AC est plus employée que UC sauf dans le premier et avant dernier groupe. Cependant, les comptes sont en général proches entre les deux stratégies, quel que soit le groupe.

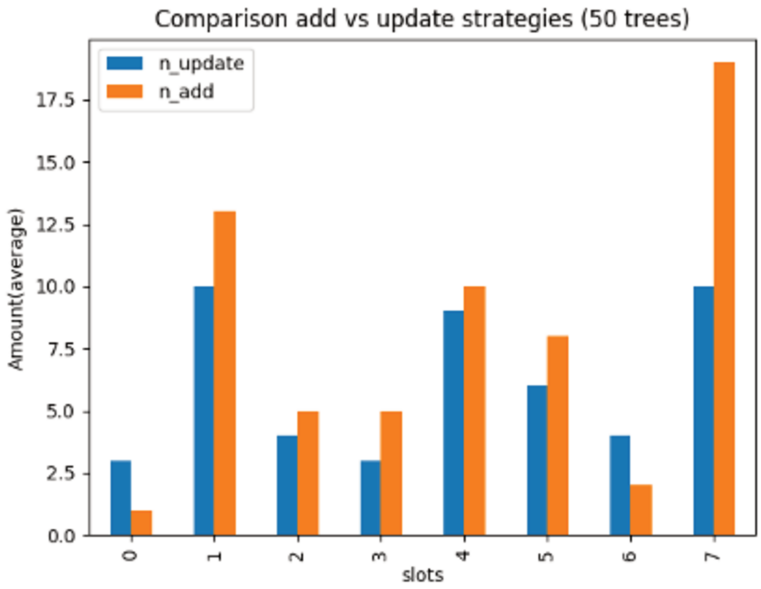

FIGURE 6.14 – Nombre moyen de fois où les stratégies AC (orange) ou UC (bleu) sont activées par slot.

La figure 6.15 présente une perspective différente de la figure précédente : elle montre le nombre moyen d'activations des deux stratégies en fonction de l'émotion. On peut ainsi observer qu'il y a plus de corrections de type UC pour l'émotion neutre que pour les autres, laissant suggérer qu'une dérive est donc présente, mais légère. La joie ne semble pas avoir de problèmes concernant une éventuelle multimodalité, les unités d'action semblent donc plutôt bien la caractériser. Un résultat intéressant dans cette figure est que la colère d'abord, puis la tristesse, sont les deux émotions nécessitant le plus de corrections de type AC. Une dérive plus forte, et donc une multimodalité, semble être présente pour ces émotions lorsqu'il y a quelques observations mal orientées au niveau des noeuds ; ce résultat est conforté par la matrice de confusion de la baseline (voir Fig.6.9) qui suggère que la colère et la tristesse sont justement les émotions les plus susceptibles d'être confondues.

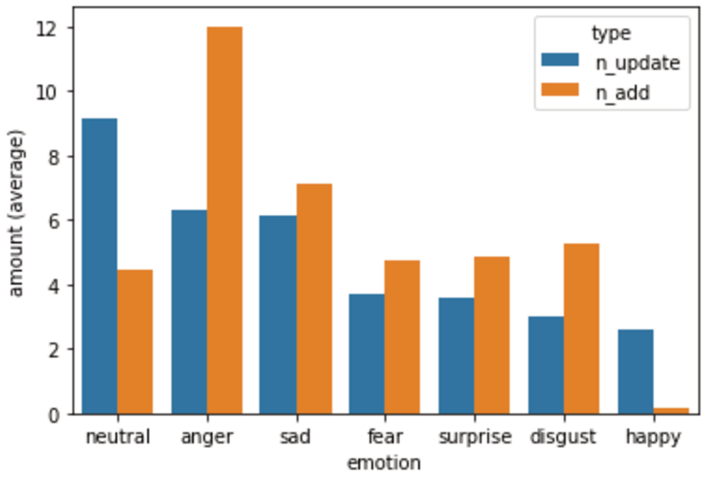

FIGURE 6.15 – Nombre moyen de fois où les stratégies AC (orange) ou UC (bleu) sont activées par émotion

6.5.5 Performances inter-groupes des modèles personnalisés

Ces modèles, incrémentés sur chaque groupe, ont également été évalués sur les autres groupes (voir Tableau 6.6). On peut observer que les modèles incrémentés donnent les meilleures performances sur leur groupe respectif (valeurs de la diagonale en gras). Ce résultat était attendu puisque le protocole a été réalisé dans le but de personnaliser chaque modèle à ses sujets. Dans l'ensemble, nous notons que les modèles personnalisés améliorent ou égalent dans la plupart des cas la performance de la baseline sur les autres groupes, comme l'indiquent les scores soulignés.

Les modèles baseline et incrémentés ont la même accuracy sur la base d'évaluation intermédiaire $CFEE_E^{[AU]}$ (0.76 ± 0.01). Ceci montre qu'il n'y a pas eu d'oubli catastrophique.

	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
i_0	**0.99**	0.97	0.93	0.97	0.94	0.90	0.97	0.91
i_1	0.94	**1.00**	0.93	0.96	0.94	0.87	0.97	0.88
i_2	0.95	0.97	**1.00**	0.96	0.94	0.90	0.97	0.91
i_3	0.94	0.97	0.93	**1.00**	0.95	0.90	0.98	0.91
i_4	0.95	0.97	0.93	0.98	**1.00**	0.88	0.98	0.89
i_5	0.95	0.97	0.92	0.95	0.94	**0.98**	0.96	0.91
i_6	0.95	0.97	0.93	0.95	0.94	0.89	**1.00**	0.90
i_7	0.97	0.97	0.92	0.97	0.95	0.91	0.97	**0.99**
b	0.96	0.95	0.92	0.96	0.94	0.89	0.97	0.91

i se réfère au modèle incrémenté, s au slot et b à la baseline

TABLE 6.6 – Performances inter-groupes avec les modèles IGTC - chaque modèle (i), spécialisé sur son slot correspondant, a été évalué sur les autres slots (s). La dernière ligne correspond à l'évaluation du modèle générique (baseline) non spécialisé sur les différents slots. Le score le plus élevé de chaque ligne apparaît en gras. Pour chaque ligne, un score est souligné lorsque celui-ci est supérieur ou égal à celui de la baseline.

6.5.6 Performances sur des expressions plus subtiles

Jusqu'à présent, en ce qui concerne l'utilisation des séquences vidéo CK+, nous incrémentons sur les images $(1, n-1)$ et évaluons ensuite le modèle sur les images $(0, n)$. Nous avons également évalué le modèle personnalisé sur les images $(2, n-2)$ et $(3, n-3)$. Les résultats présentés dans le tableau 6.7 sont la moyenne des scores obtenus par les 8 modèles sur leur slot respectif, avant et après incrémentation avec la méthode IGTC sur les images $(1, n-1)$. On peut observer que la performance diminue quand les expressions deviennent plus subtiles, en s'éloignant du neutre et de l'apex.

Toutefois, on constatera, en mesurant l'écart de performances relatif entre le modèle générique et le modèle personnalisé qu'ici, ce dernier tient toutes ses promesses. Ainsi, l'erreur de reconnaissance est diminuée en moyenne de 55% sur les images montrant des expressions subtiles. Ceci démontre bien l'intérêt de la phase de personnalisation, qui permet, en apprenant une image montrant une expression proche d'être surjouée d'un sujet, de mieux reconnaître ses expressions subtiles.

Nous avons également réalisé une analyse afin d'évaluer les performances d'un réseau de neurones dans des conditions expérimentales similaires à celles du modèle NCMF et afin d'évaluer la robustesse du réseau, avant et après la technique de finetuning, pour ainsi vérifier la présence ou non d'un

	$(2, n-2)$	$(3, n-3)$
Modèle générique	0.901 ± 0.035	0.869 ± 0.040
Modèle incrémenté	$\mathbf{0.974 \pm 0.013}$	$\mathbf{0.919 \pm 0.035}$
écart relatif	**73% ±**	**38%**

TABLE 6.7 – Performances moyennes sur d'autres images des séquences de CK+ - évaluation sur les états neutre de la frame numéro 2 et émotions de la frame numéro $(n-2)$, puis, sur les états neutre de la frame numéro 3 et émotions de la frame numéro $(n-3)$. Plus on travaille avec des images vers le milieu de la séquence vidéo, et plus les états correspondants sont subtils et résultent de transitions d'un état neutre avec une émotion.

oubli catastrophique après personnalisation. Le modèle utilisé est un réseau de neurones de type Convolutional Neural Network (CNN) et le protocole expérimental a été le suivant :

— Le CNN a été d'abord été entraîné directement sur les images correspondant au dataset $(X, Y)_A^{[CFEE]}$;

— La technique de finetuning a été ensuite utilisée sur les images correspondant au dataset $(X, Y)_I^{[CK+]}$ séparés selon les mêmes slots qu'avec la NCMF, afin d'incrémenter le modèle pour qu'il puisse se personnaliser aux sujets appartenant à chacun des slots.

Ce réseau de neurones reçoit en entrée des images 2D de visages avec une taille fixée à 112×112, normalisées avec la méthode CLAHE [2]. L'architecture consiste en d'abord, 3 couches de convolution, respectivement de tailles 64, 128, et 256 et utilisant des filtres de taille 7×7. Chaque couche inclut une *batch normalization* et est suivie d'une couche de *max pooling*. Ces couches d'extraction de caractéristiques sont, suivies d'une couche dense de taille 256 neurones à fonction d'activation RELU incluant un Dropout (avec un taux de 0.2). Enfin, la couche de décision correspond à une couche Dense de taille 7. Enfin, une couche SoftMax calcule la probabilité *a posteriori* de chacune des 7 émotions.

Le CNN a obtenu le score moyenné sur tous les slots de 0.993 ± 0.013, surpassant ainsi les performances de la NCMF. Le tableau 6.8 montre également les performances du CNN sur les images $(2, n-2)$ et $(3, n-3)$, ainsi que ses performances sur la base test CFEE avant et après fine-tuning. Nous pouvons observer que le CNN obtient de meilleures valeurs d'accuracy que la NCMF malgré des écart-types légèrement plus élevés. Par contre, il souffre d'un sérieux oubli catastrophique après s'être personnalisé sur les données de CK+, contrairement à la NCMF qui maintient une accuracy stable.

2. *Contrast Limited Adaptive Histogram Equalization* (CLAHE)

	$(n+2, a-2)$	$(n+3, a-3)$	CFEE (avant)	CFEE (après)
NCMF	0.974 ± 0.013	0.919 ± 0.035	0.76	**0.76 ± 0.01**
CNN	**0.984 ± 0.019**	**0.933 ± 0.048**	**0.822**	0.653 ± 0.035

TABLE 6.8 – Comparaison des performances moyennes des deux modèles NCMF et CNN sur les émotions subtiles et sur les données initiales avant et après incrémentation

6.6 Conclusions

Nous avons présenté ici une adaptation du modèle NCMF pour la reconnaissance des émotions, qui est bien adapté à l'apprentissage incrémental. C'est pourquoi nous l'avons utilisé pour améliorer un modèle de référence *baseline*, entraîné sur le jeu de données CFEE, sur les données du benchmark CK+. L'objectif de l'incrémentation est de personnaliser le modèle sur un (ou un ensemble d') individu(s), afin de réduire le biais d'identité. Nous avons évalué deux stratégies, ULS et IGT. Nous avons amélioré cette dernière en utilisant le clustering multimodal dans les nœuds internes de l'arbre (IGTC) et montré que la précision s'améliore. Un résultat important que nous avons obtenu est que ces modèles incrémentés n'ont pas subi d'oubli catastrophique sur la base de données initiale $CFEE$ et respecte le dilemme stabilité-plasticité.

Ces résultats nous confortent, pour la suite de nos travaux, dans l'idée de conserver ce pipeline de personnalisation pour s'attaquer au biais d'identité causant des chutes de performances des modèles génériques. Nous allons, dans le prochain chapitre, l'appliquer à des flux de données partiellement supervisées.

Chapitre 7

Apprentissage incrémental semi-supervisé en données

Contenu

7.1 Travaux préliminaires sur l'apprentissage incrémental semi-supervisé 130
7.1.1 One Pass Incrementation 130
7.1.2 Two Pass Incrementation 131
7.1.3 Continuous Incrementation 132
7.1.4 Résultats expérimentaux 132
7.2 Algorithme de co-incrementation 133
7.2.1 Algorithme original de co-training 133
7.2.2 Algorithme de co-training incrémental (EBSICO) 134
7.3 Pipeline expérimental et résultats 137
7.3.1 Préparation des datasets 137
7.3.2 Apprentissage des modèles de référence 137
7.3.3 Personnalisation par cluster 139
7.3.4 Évaluation du modèle selon la stratégie de personnalisation choisie 139
7.3.5 Résultats expérimentaux 140
7.4 Conclusion 146

La création de vérité terrain (labélisation) représente un défi dans le domaine de l'apprentissage automatique, celle-ci peut être coûteuse en temps et en effort. Nous nous retrouvons donc de plus en plus dans des contextes de supervision incomplète où les datasets contiennent des données non labélisées en abondance, et où les données labélisées sont en quantité insuffisante pour former un modèle performant. Différentes techniques d'apprentissage ont été proposées pour y remédier : avec intervention humaine comme l'apprentissage actif (*active learning*) ou sans intervention humaine comme l'apprentissage semi-supervisé. Pour réaliser ce dernier, on rappelle que le co-training multi-vue est l'un des algorithmes les plus populaires. Il nécessite que les classifieurs sur les différentes "vues" soient performants et aient un comportement suffisamment différent, se traduisant par divers désaccords [117].

7.1 Travaux préliminaires sur l'apprentissage incrémental semi-supervisé

Pour traiter le cas de l'apprentissage incrémental semi-supervisé en données, nous avons initialement proposé trois approches pour mettre en application la NCMF avec la stratégie incrémentale IGTC. Comme précédemment, le modèle générique est entraîné sur un premier dataset G. Il est ensuite incrémenté sur un second dataset I partiellement labélisé seulement. Ce dataset peut être subdivisé en deux sous-ensembles : L, l'ensemble des données labélisées et U, celui des données non labélisées. Le protocole de séparation en slots *subject-dependent*, présenté dans la section 6.4, est également utilisé ici avec le même nombre de slots (8) et notre objectif est de montrer que le modèle arrive à se spécialiser sur chacun des slots.

7.1.1 One Pass Incrementation

La première approche que l'on propose ici *One Pass Incrementation* (OPI) consiste, comme son nom l'indique, en une seule phase d'incrémentation, suivant les étapes suivantes :

Étape 0 : La baseline est entraînée sur le premier dataset G.

Étape 1 : La baseline est utilisée pour pseudo-labéliser les données non labélisées U du deuxième dataset I.

Étape 2 : Le modèle est mis à jour de façon incrémentale avec les données labélisées et pseudo-labélisées $L \cup U$ du deuxième dataset I.

Ici le modèle NCMF entraîné sur G, baseline, est d'abord utilisé pour assigner un pseudo-label à chaque exemple non labélisé de U du deuxième dataset I. Ce sont donc uniquement les connaissances antérieures apprises via G qui sont mises à profit. On parle de $pseudo-labélisation$ pour différencier les labels prédits par les modèles de ceux des observations véritablement labélisées L de I (vérité terrain) ; les labels prédits par la baseline, pour les observations U, ne sont donc pas des labels fournis par un annotateur humain expert. Une fois cette étape réalisée, on se retrouve avec le dataset I complètement labélisé, et nous pouvons alors procéder à la phase d'incrémentation de la même manière qu'on le fait en mode supervisé avec la stratégie IGTC.

7.1.2 Two Pass Incrementation

L'approche *Two Pass Incrementation* (TPI) est directement inspirée de la première. Contrairement à OPI, nous souhaitons avec TPI utiliser à la fois les connaissances de G, mais aussi celles de L. Le principe de TPI est alors de procéder à une phase d'incrémentation en deux temps :

Étape 0 : La baseline est entraînée sur le premier dataset G.

Étape 1 : La baseline est incrémentée avec les données labélisées L du deuxième dataset I (première phase d'incrémentation).

Étape 2 : Le modèle précédemment incrémenté est utilisé pour pseudo-labéliser les données non labélisées U de I.

Étape 3 : Le modèle est incrémenté avec les données pseudo-labélisées U (deuxième phase d'incrémentation).

Les phases d'incrémentation se font de la même manière qu'en mode supervisé, avec la stratégie IGTC. La motivation derrière cette approche est d'utiliser la nouvelle vérité terrain (observations labélisées du nouveau dataset) pour mettre à jour le modèle qui servira à l'étape de pseudo-labélisation. Le processus de pseudo-labélisation pouvant être sujet à des erreurs, on limite ainsi avec TPI le risque de mettre à jour la structure des arbres d'une manière non attendue, ce qui pourrait arriver dans le cas où des erreurs de pseudo-labélisation auraient eu lieu ; en effet, on attribue moins de confiance à la baseline pour pseudo-labéliser les données U qu'au modèle intermédiaire, puisque ce dernier est formé sur un nombre de données plus grand (expérience donc plus grande que la baseline).

7.1.3 Continuous Incrementation

Continuous Incrementation (CI) est une évolution des deux approches précédentes. Dans cette configuration on suppose que le dataset complet n'est pas donné à l'avance ; avec CI, nous traitons donc les données au fur et à mesure que nous les recevons. Cette approche correspond davantage au scénario incrémental, c'est-à-dire appliquer des mises à jour à un modèle existant à partir de données arrivant au fil de l'eau, sans avoir à faire d'entraînement depuis zéro (*from scratch*). Dans CI, le modèle suit donc ce schéma de pensée et se met ainsi à jour en fonction des données arrivant en continu, et ce, de la manière suivante :

Étape 0 : La baseline est entraînée sur le premier dataset G. Un deuxième dataset I est ensuite fourni au fur et à mesure.

Étape 1 : Tant que les données qui nous parviennent de I sont labélisées, le modèle se met à jour en s'incrémentant à partir de ces données.

Étape 2 : Lorsque l'on rencontre une observation non labélisée, c'est le dernier modèle à jour qui est utilisé dans un premier temps pour assigner un pseudo-label à l'observation.

Étape 3 : Puis, dans un second temps, on le met à jour à partir de cette même observation pseudo-labélisée.

Les étapes 1 à 3 sont répétées tant que l'on reçoit des données de I.

7.1.4 Résultats expérimentaux

Afin d'expérimenter ces trois méthodes, on utilise le dataset $CFEE$ pour entraîner le modèle générique et le dataset $CK+$ pour le personnaliser. Pour évaluer chaque approche, on calcule le taux de reconnaissance moyen obtenu par chaque baseline sur chaque slot, après incrémentation. Nous avons évalué l'impact du nombre de données non labélisées (U) sur les performances en choisissant successivement : 5, 10, 20, 50, et 75%.

Les résultats expérimentaux obtenus par les trois méthodes décrites précédemment sont résumés dans le Tableau 7.1 ; les performances présentées sont celles obtenues par le modèle à l'issue du processus complet d'incrémentation.

Il est intéressant de constater que, même lorsque le ratio de données non labélisées est élevé, l'incré-

mentation conduit à une amélioration du taux de reconnaissance moyen sur les 8 slots, la performance de la $NCMF$ baseline sur $(X, Y)_E^{[CK+]}$ étant de 0,939±0, 026.

Méthode / %	OPI	TPI	CI
5%	0.991 ± 0.008	0.991 ± 0.010	0.990 ± 0.012
10%	0.990 ± 0.009	0.991 ± 0.010	0.990 ± 0.012
20%	0.984 ± 0.017	0.984 ± 0.016	0.986 ± 0.012
50%	0.971 ± 0.011	0.969 ± 0.024	0.972 ± 0.023
75%	0.957 ± 0.019	0.960 ± 0.019	0.957 ± 0.019

TABLE 7.1 – Taux de reconnaissance moyen sur 8 slots en fonction du pourcentage de données non labélisées (incrémentation avec la méthode IGTC)

Les trois stratégies que nous avons proposées (OPI, TPI et CI) obtiennent des résultats assez similaires. On peut noter que nous utilisons un seul modèle NCMF entraîné sur une seule famille de *features* que sont les AUs. Nous les avons proposé comme travail préliminaire pour traiter les données non labélisées, et il est possible que se limiter à ces features ne nous permette pas d'obtenir d'améliorations supplémentaires. L'état de l'art nous a ensuite mené à utiliser d'autres méthodes pour traiter les données semi-supervisées, comme le co-training, faisant appel à la collaboration de deux modèles, chacun entraîné sur une famille de features différente.

7.2 Algorithme de co-incrementation

Une limitation observable dans la procédure classique de co-training est que le classifieur se ré-entraîne à chaque itération sur des données déjà vues aux itérations précédentes. Nous pourrions profiter des progrès de l'apprentissage incrémental pour ne faire qu'une seule passe sur chaque observation et mettre à jour le modèle tout en pseudo-labélisant les données, sans ré-entraîner à partir de zéro à chaque itération.

7.2.1 Algorithme original de co-training

Le co-training [82] est une technique d'apprentissage semi-supervisé qui peut être utilisée lorsqu'un dataset est partiellement labélisé. Il consiste à faire collaborer deux modèles d'apprentissage automatique. Soient V_1 et V_2 deux familles de features, aussi appelées "vues", décrivant entièrement chaque

observation du dataset $x = (V_1(x), V_2(x))$. Les datasets correspondants sont notés $L^{[V_1]}, L^{[V_2]}$ pour les données labélisées et $U^{[V_1]}, U^{[V_2]}$ pour les données non labélisées. Chaque modèle est entraîné sur une vue et les deux modèles doivent satisfaire l'hypothèse d'indépendance (voir Sec.3.3.3).

0. ***Pré-training*** : chaque modèle s'entraîne initialement sur son propre ensemble labélisé $L^{[V_1]}$ ou $L^{[V_2]}$.

Le co-training est un processus itératif. Pour chaque observation de U, les 2 étapes suivantes sont réalisées :

1. ***Labeled set extension*** : chaque modèle prédit un pseudo-label pour l'observation ; la prédiction la plus fiable (compte tenu d'un critère de confiance) est utilisée pour ajouter l'observation et son pseudo-label le plus fiable à l'ensemble labélisé de l'autre vue, $L^{[V_2]}$ si le modèle 1 a été le plus fiable et vice et versa ;
2. ***Self-training*** : le modèle dont le pseudo-label s'est avéré le moins fiable est ré-entraîné sur le nouvel ensemble labélisé.

Nous pouvons remarquer que les classifieurs à vue unique nécessitent un ré-apprentissage complet à chaque itération. Ce processus peut devenir coûteux en termes de temps de calcul, surtout si la taille de U est importante.

7.2.2 Algorithme de co-training incrémental (EBSICO)

Nous proposons une méthode hybride de co-training qui diffère de la méthode classique et qui combine les techniques d'apprentissage semi-supervisé et d'apprentissage incrémental. Notre objectif n'est pas de construire des classifieurs génériques mais de personnaliser des classifieurs génériques à un sous-ensemble de sujets. Ainsi, nous utilisons un premier dataset G pour construire ces classifieurs génériques à vue unique. Ensuite, un deuxième dataset I, partiellement labélisé seulement, est utilisé pour personnaliser ces modèles de manière incrémentale en utilisant un algorithme basé sur le co-training. Le principal avantage de ce processus est d'éviter de ré-entraîner à partir de zéro les modèles lors des itérations de co-training.

La figure 7.1 montre les différentes étapes de notre méthode qui sont décrites ci-après.

1. ***Pre-training*** : à l'étape 1, deux modèles génériques sont entraînés respectivement sur leurs

vues $G^{[V_1]}$ et $G^{[V_2]}$. Ceux-ci sont les modèles de référence et seront désignés par le nom de leur vue ;

2. ***Error-Based Self-Incrementation (EBSI)*** : à l'étape 2, le modèle associé à la vue V_i prédit une classe pour chacune des observations de $L^{[V_i]}$. Si cette classe prédite est différente de la vérité terrain (le dataset $L^{[V_i]}$ est labélisé), le modèle s'incrémente sur cet exemple. Cette stratégie incrémentale basée sur les erreurs est possible puisque nous travaillons sur des modèles génériques déjà entraînés. Cette étape est décrite dans l'algorithme 2 ;
3. ***Error-Based CO-incrementation (EBCO)*** : Pour chaque observation x non labélisée de $U^{[V_i]}$,
   (A) Nous déterminons le modèle le plus fiable, en utilisant les probabilités a posteriori $p^{1+}(x)$ et $p^{2+}(x)$:

$$p^{1+}(x) = \max \phi_{\mu_1}(x). \tag{7.1}$$

$$p^{2+}(x) = \max \phi_{\mu_2}(x). \tag{7.2}$$

   La probabilité a posteriori la plus grande nous renseigne ainsi sur le modèle le plus fiable pour la prédiction du pseudo-label de x. Nous utilisons les prédictions des modèles $\eta^{1+}(x)$ et $\eta^{2+}(x)$ (voir Eq.2.3) comme *pseudo-label* (l^{1+} et l^{2+} sur la figure).
   (B) Ensuite, le modèle le moins fiable est incrémenté de manière supervisée à partir de l'observation non labélisée en utilisant le pseudo-label fourni par le modèle le plus fiable. L'algorithme 3 décrit plus précisément la procédure de co-incrémentation.

Contrairement à l'algorithme de co-training classique, avec notre approche $EBCO$, pour chaque itération, lorsque la probabilité maximale d'appartenir à une classe dépasse le seuil de confiance θ, le modèle le moins fiable ne recommence pas son apprentissage depuis le début. Il s'incrémente seulement sur l'observation de l'itération considérée. De plus, il ne s'incrémente que si son pseudo-label diffère de celui délivré par le modèle le plus fiable.

Algorithm 2 Error-based Self-incrementation (EBSI)

Require: generic model m_i pretrained on $G^{[V_i]}$
  for all $(x, y) \in L^{[V_i]}$ **do**
    The model uses the function $\phi(x)$ to obtain the probability vector of class membership :
    $p^{(i)} \leftarrow \phi(x^{[V_i]})$
    The predicted label is thereby determined as :
    $\hat{p} \leftarrow argmax(p^{(i)})$
    if $\hat{p} \neq y$ **then**
      $m_i \leftarrow INCR(m_i, x, y)$
    end if
  end for

Algorithm 3 Error-Based Co-Incrementation (EBCO)

Require: models m_1 and m_2 incremented following **EBSI** method, $\theta >= 0$
  for all $u \in U$ **do**
    Each model uses the function $\phi(u)$ to obtain the probability vectors of class membership :
    $p^{(1)} \leftarrow \phi(u^{[V1]})$
    $p^{(2)} \leftarrow \phi(u^{[V2]})$
    The pseudo-labels are then :
    $l^{1+} \leftarrow argmax(p^{(1)})$
    $l^{2+} \leftarrow argmax(p^{(2)})$
    The probabilities associated are then :
    $p^{1+} \leftarrow max(p^{(1)})$
    $p^{2+} \leftarrow max(p^{(2)})$
    if $p^{1+} > p^{2+}$ **and** $p^{1+} >= \theta$ **then**
      if $l^{2+} \neq l^{1+}$ **then**
        $m_2 \leftarrow INCR(m_2, u^{[V2]}, l^{1+})$
      end if
    else if $p^{2+} > p^{1+}$ **and** $p^{2+} >= \theta$ **then**
      if $l^{1+} \neq l^{2+}$ **then**
        $m_1 \leftarrow INCR(m_1, u^{[V1]}, l^{2+})$
      end if
    end if
  end for

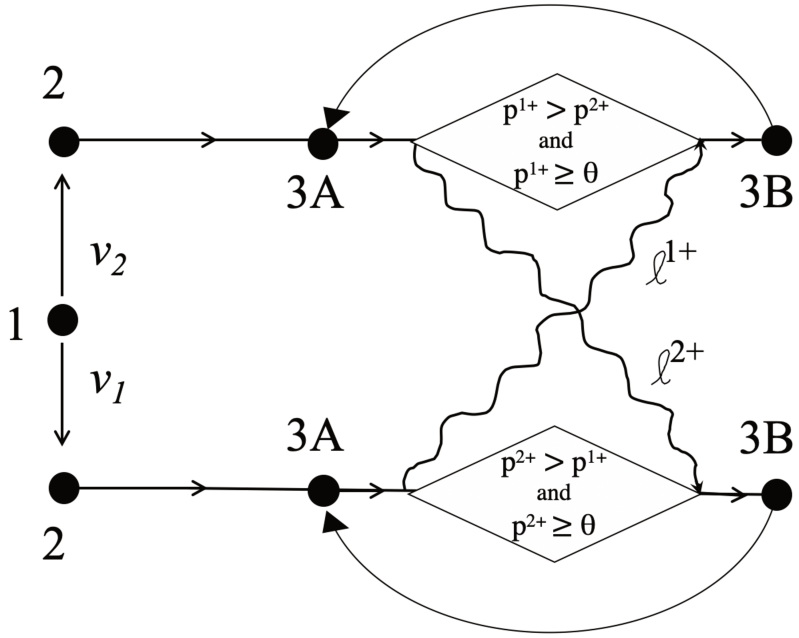

FIGURE 7.1 – Diagramme décrivant l'ensemble de la procédure EBSICO proposée - chaque modèle générique s'entraîne sur un premier dataset et sa vue correspondante v_1 ou v_2 (1), puis, les modèles se spécialisent sur quelques nouveaux sujets labélisés (2), enfin, la personnalisation se poursuit sur un grand nombre de nouveaux sujets mais non labélisés via la collaboration entre les deux modèles (3).

7.3 Pipeline expérimental et résultats

7.3.1 Préparation des datasets

Le dataset CFEE est utilisé pour entraîner les classifieurs génériques à vue unique ; il correspond donc au dataset G. Le dataset CK+ a été divisé en deux sous-ensembles nommés I et E, utilisés respectivement pour l'apprentissage incrémental et l'évaluation. La section 5.2 indique précisément quelles images de CK+ sont considérées dans les datasets I et E. Ensuite, I est divisé en sous-ensembles L et U correspondant aux ensembles labélisés et non labélisés (i.e. dont on va ignorer les labels), avec $L \cup U = I$.

7.3.2 Apprentissage des modèles de référence

Chaque observation x est décrite par deux vues : $(AU(x), TX(x))$ où AU correspond à la famille des unités d'action et TX (pour textures) à la concaténation des features LBP et HoG.

Deux modèles NCMF ont d'abord été entraînés de manière supervisée respectivement sur les vues AU et TX du dataset CFEE. Ces modèles, correspondant à l'étape 1 de la procédure EBSICO, sont des modèles génériques. Ils seront utilisés comme modèles de référence quelle que soit la stratégie de personnalisation.

On va réaliser une séparation des données en slots en regroupant celles-ci, non pas de manière aléatoire comme dans le chapitre précédent, mais selon un critère dit de *personnalisation*. Pour chacun de ces critères où Θ est l'acronyme de personnalisation, nous nommons ces modèles respectivement par $AU^{[\Theta]}$ et $TX^{[\Theta]}$. Les personnalisations proposées sont les suivantes :

ALL L'absence de personnalisation est une approche qui sert de référence et qui consiste à appliquer l'algorithme de co-incrémentation directement sur toutes les données de I, comme on le ferait pour un modèle générique. Elle n'utilise qu'un seul groupe contenant toutes les observations de I. Dans la suite, nous désignons cette stratégie par le terme ALL.

GENDER L'approche de la personnalisation selon le genre effectue un regroupement des données en fonction du genre perçu. Nous obtenons deux groupes de données nommés M (*man*) et W (*woman*) correspondant respectivement aux données des hommes et des femmes. Les informations sur le genre ont été affectées manuellement. Dans la suite, nous désignons cette stratégie par le terme $GENDER$.

MORPHO L'approche de la personnalisation selon un critère morphologique effectue un regroupement des données en fonction de la morphologie du visage. Pour ce faire, nous avons utilisé les landmarks (voir Sec.5.1.1) détectées sur le visage neutre de chaque sujet. Par la suite, l'algorithme K-means a été utilisé sur ces données afin d'identifier plusieurs clusters basés sur des caractéristiques morphologiques.

Nous avons décidé de répartir les sujets du dataset en 8 slots comme dans le chapitre précédent ; la figure 7.2 illustre la variation de la variance intra-classe totale pour un nombre k de clusters et nous conforte dans ce choix. Dans la suite, nous désignons cette stratégie par le terme $MORPHO$.

Notons que chaque groupe (*cluster*) est dit *subject-independent*. D'une part, ces données n'ont pas été vues lors de l'apprentissage des modèles génériques. D'autre part, toutes les images d'un même sujet appartiennent à un même groupe et un même groupe peut contenir les données de plusieurs

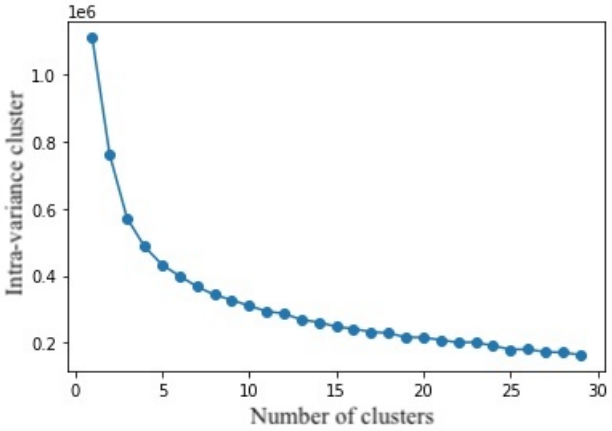

FIGURE 7.2 – Évolution de la variance intra-classe en fonction du nombre de clusters (*courbe en coude*)

sujets. Pour un cluster donné, les ensembles associés I et E contiennent les mêmes individus mais des images différentes (en termes d'intensité émotionnelle), afin d'évaluer l'impact de la personnalisation sur les modèles de référence après co-incrémentation.

7.3.3 Personnalisation par cluster

Dans la troisième étape du pipeline proposé, le processus EBSICO est exécuté dans chaque slot comme suit. Dans nos expériences, nous avons considéré une valeur θ de 0.8 et utilisé des NCMF de 50 arbres avec la stratégie IGT pour l'incrémentation (voir Sec.6.2).

Nous décidons d'utiliser un ratio de 5% de données labélisées, donc, dans la suite, $|L| = 0,05 \times |I|$ et $|U| = 0,95 \times |I|$.

Ensuite, nous exécutons séquentiellement et pour chaque classifieur à vue unique :

— $EBSI$ sur L

— $EBCO$ sur U

7.3.4 Évaluation du modèle selon la stratégie de personnalisation choisie

La dernière étape du pipeline consiste à évaluer les deux modèles lors des différentes étapes de la procédure EBSICO afin de suivre l'évolution des performances en termes de personnalisation. Ces scores sont mesurés grâce à l'accuracy du modèle *baseline* (étape 1), à la fin de la procédure EBSI (étape 2), et à la fin de la procédure EBCO (étape 3).

De même, pour évaluer la contribution d'une stratégie de personnalisation spécifique Θ qui peut conduire à plusieurs clusters, nous avons mesuré l'accuracy :

$$acc^{[\Theta]} = \frac{\sum_{c \in C} acc(c) \times |c|}{|C|} \tag{7.3}$$

où C représente l'ensemble des clusters (slots) et c correspond aux observations appartenant à un cluster (slot).

7.3.5 Résultats expérimentaux

7.3.5.1 Contribution de la co-incrémentation sur les modèles individuels

Le but de cette analyse est d'évaluer la contribution fournie par le processus de co-incrémentation EBCO. Les tableaux 7.2 et 7.3 décrivent les résultats obtenus pour les modèles baseline, EBSI, EBCO, entraînés respectivement sur les vues AU et TX.

Model	*baseline*	*EBSI*	*EBCO*
$AU^{[ALL]}$	0.946	0.947	0.948
$AU^{[M]}$	0.946	0.949	**0.953**
$AU^{[W]}$	0.946	0.939	0.947
$AU^{[C1]}$	0.950	0.950	**0.962**
$AU^{[C2]}$	0.931	**0.954**	0.931
$AU^{[C3]}$	**0.974**	**0.974**	0.962
$AU^{[C4]}$	0.875	0.917	**0.958**
$AU^{[C5]}$	0.950	0.950	**0.966**
$AU^{[C6]}$	0.933	0.933	**0.958**
$AU^{[C7]}$	**0.967**	0.963	0.949
$AU^{[C8]}$	0.911	0.931	**0.960**

TABLE 7.2 – Vue AU. Mesures de l'accuracy à différentes étapes du pipeline EBSICO avec **5%** d'observations labélisées - chaque ligne correspond à un slot selon le critère de personnalisation choisi pour séparer les sujets : un seul slot avec le critère ALL, deux slots Man et Woman avec le critère $GENDER$, et huit slots avec le critère $MORPHO$ allant de $C1$ à $C8$.

Nous pouvons observer que pour de faibles taux de labélisation, le processus $EBSI$ n'a pas eu

Model	*baseline*	*EBSI*	*EBCO*
$TX^{[ALL]}$	0.811	0.866	**0.943**
$TX^{[M]}$	0.766	0.847	**0.939**
$TX^{[W]}$	0.833	0.894	**0.936**
$TX^{[C1]}$	0.843	0.824	**0.931**
$TX^{[C2]}$	0.713	0.701	**0.885**
$TX^{[C3]}$	0.821	0.833	**0.974**
$TX^{[C4]}$	0.625	0.625	**0.958**
$TX^{[C5]}$	0.891	0.950	0.950
$TX^{[C6]}$	0.849	0.840	**0.941**
$TX^{[C7]}$	0.808	0.869	**0.944**
$TX^{[C8]}$	0.752	0.832	**0.921**

TABLE 7.3 – Vue TX. Mesures de l'accuracy à différentes étapes du pipeline EBSICO avec **5%** d'observations labélisées

d'influence et dans certains cas a diminué la performance de prédiction. Ainsi, la phase de *self-training* incrémental avec seulement 5% de données labélisées n'était pas suffisant pour améliorer les performances du modèle.

En revanche, on remarque surtout que le processus $EBCO$ améliore les performances des modèles de référence et d'$EBSI$ quelle que soit la méthode de personnalisation choisie. Ainsi, la co-incrémentation s'est avérée suffisamment robuste, même pour une quantité de données labélisées faible.

Pour une analyse plus approfondie, nous avons également étudié l'évolution de la mesure de désaccord (*disagreement measure*) proposée par [118] : il s'agit du taux d'erreurs que les modèles ne font pas en commun lors de leurs prédictions. Les résultats obtenus par les modèles $(AU^{[ALL]}, TX^{[ALL]})$, en fonction de différents taux de données labélisées, sont présentés dans la Fig. 7.3. Tout d'abord, nous pouvons observer que la mesure de désaccord pour les modèles de base est de 0.2. En d'autres termes, avant la co-incrémentation, nos modèles satisfont suffisamment l'hypothèse d'indépendance, condition nécessaire pour que la co-training fonctionne. Le graphique montre également que la procédure $EBCO$ a considérablement réduit le taux d'erreurs non communes, convergeant presque vers

0. Par conséquent, les deux modèles ont pu améliorer leurs performances et le processus a permis au meilleur modèle d'augmenter sa performance initiale.

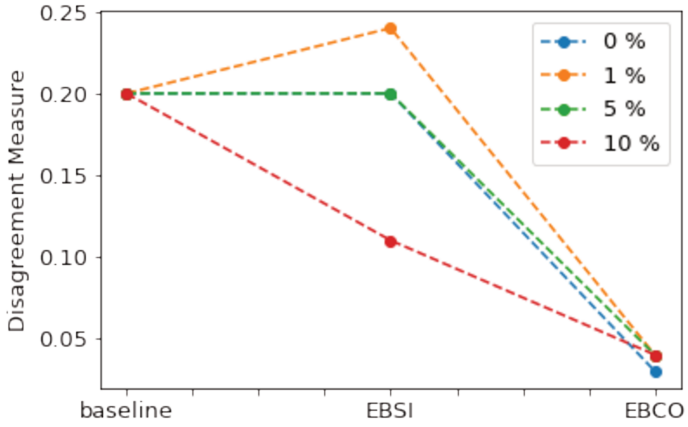

FIGURE 7.3 – Évolution du taux d'erreurs non communes à travers les différentes étapes du framework EBSICO, en fonction du taux de données labélisées - les modèles convergent vers un accord mutuel en fin de procédure EBSICO quel que soit le taux de labélisation utilisé à l'étape (2) du pipeline.

7.3.5.2 Apport de la personnalisation

L'objectif de cette analyse est d'évaluer la contribution apportée par la personnalisation.

Les résultats ont montré que $AU^{[MORPHO]}$ a obtenu l'*accuracy* la plus élevée par rapport à $AU^{[GENDER]}$ et $AU^{[ALL]}$ (voir tableau 7.4). Cela peut être dû au fait que le regroupement des images selon le critère morphologique a permis de mieux isoler les sujets ayant des caractéristiques communes dans des groupes. Cela a conduit chaque modèle à se spécialiser sur un groupe de sujets ayant des traits communs. La position des points de repère faciaux s'est ici révélée être un critère plus robuste pour la séparation des données que le critère de genre.

Par ailleurs, $AU^{[ALL]}$ a présenté le taux de performance le plus faible, pour 5% et 10%, et le deuxième plus faible pour 0% et 1%. Cela permet de conclure que le processus de personnalisation a augmenté la robustesse du modèle de manière générale.

Pour une analyse plus approfondie, nous avons également calculé le nombre moyen d'observations traitées par modèle. $AU^{[ALL]}$ contenait la plus grande quantité de données labélisées puisqu'il ne recourrait pas au clustering de données. Avec un taux de labélisation de 1%, 5% et 10%, nous avons calculé une moyenne de 4, 22 et 45 observations labélisées par cluster pour $AU^{[GENDER]}$ et une moyenne de

%	$AU^{[ALL]}$	$AU^{[GENDER]}$	$AU^{[MORPHO]}$
0	0.947	0.946 ± 0.0	**0.952** ± 0.013
1	0.948	0.946 ± 0.0	**0.952** ± 0.013
5	0.948	0.949 ± 0.003	**0.956** ± 0.01
10	0.951	0.954 ± 0.001	**0.956** ± 0.01

TABLE 7.4 – Accuracy moyenne par slot en fonction du pourcentage de données labélisées - EBSICO sur la vue (AU)

1, 5 et 11 d'observations labélisées par cluster pour $AU^{[MORPHO]}$ (voir tableau 7.5). En dépit de la faible quantité de données labélisées par cluster, $AU^{[MORPHO]}$ a été capable d'obtenir de meilleures performances que $AU^{[GENDER]}$ et $AU^{[ALL]}$. Par conséquent, nous pouvons déduire qu'une stratégie de clustering rationalisée offre une certaine robustesse pendant le processus de co-incrémentation, quel que soit le nombre d'observations labélisées ; celle-ci est cruciale pour l'amélioration des modèles.

%	$\lvert L^{[ALL]}\rvert$	$\lvert L^{[GENDER]}\rvert$	$\lvert L^{[MORPHO]}\rvert$
0	0	0	0
1	9	4 ± 2	**1.0** ±1
5	45	22 ± 8	**5** ±3
10	90	45±15	**11** ± 6

TABLE 7.5 – Nombre moyen d'images labélisées selon le critère de personnalisation

Le tableau montre également que grâce à la personnalisation, nous parvenons à labéliser moins d'images en moyenne que le modèle généralisé $AU^{[ALL]}$.

De plus, dans le tableau 7.2, on observe que AU et TX ont tous deux bénéficié de la co-incrémentation. Ceci confirme notre hypothèse selon laquelle le partage des connaissances entre différents modèles conduit à une amélioration du taux de prédiction, indépendamment de la vue et du critère de clustering choisi.

Pour s'attaquer au biais d'identité, nous avons choisi de personnaliser les modèles à des groupes d'individus. Lorsque plusieurs sujets produisent une même émotion différemment, conséquence directe

du biais d'identité, cela se traduit par une variance intra-classe plus grande. Nous avons souhaité observer l'évolution de celle-ci, en fonction de la séparation des slots selon les critères de personnalisation que nous avons présenté dans cette section. Le nombre de slots créé dépend du critère de personnalisation choisi. Pour que la comparaison soit juste, nous avons donc réalisé pour chaque critère, excepté pour *ALL*, une séparation aléatoire avec le même nombre de slots ; celle-ci a été réalisée sur 100 itérations, et le résultat final est la moyenne des moyennes des variances intra-classes sur les slots, ainsi que la moyenne des écarts-types associés. On distingue de cette manière, une séparation faite sur un critère aléatoire, d'une séparation faite sur un critère rationnel. De plus, en raison du grand nombre de neutre, celui-ci a été *downsamplé* à 50 pour correspondre aux distributions des autres labels. Les critères de séparations avec leur nombre de slots associés sont les suivants :

ALL : 1 slot (tout CK+incr, AUs seulement),

RANDOM 2 : 2 slots, séparation aléatoire, (moyenne sur 100 itérations),

GENDER 2 : 2 slots,

RANDOM 5 : 5 slots, séparation aléatoire, (moyenne sur 100 itérations),

MORPHO 5 : 5 slots,

RANDOM 8 : 8 slots, séparation aléatoire, (moyenne sur 100 itérations),

MORPHO 8 : 8 slots.

Les résultats sont disponibles dans le tableau 7.6. Nous pouvons remarquer que, plus il y a de slots, plus la variance intra-classe diminue. Dans un cas extrême comme mettre un seul sujet par slot, la variance intra-classe tendra vers 0. Nous pouvons ainsi observer dans le tableau que la variance intra-classe moyenne diminue lorsque nous constituons plus de slots avec un critère différent. Ces résultats suggèrent que la variance intra-classe diminuant, alors le biais d'identité s'en retrouve atténué. De plus, en comparant à nombre de slots égal, nous observons que le critère rationnel de séparation offre une variance intra-classe légèrement plus faible que le critère aléatoire. Ce résultat conforte, d'une part, sur la qualité de ces critères de personnalisation, et d'autre part, motive la recherche de critères rationnels de séparation encore plus fins pour de futures expérimentations.

ALL	4.719
RANDOM 2	4.623 ± 0.167
GENDER 2	**4.466** ± 0.624
RANDOM 5	4.327 ± 0.353
MORPHO 5	**4.211** ± 0.304
RANDOM 8	4.035 ± 0.488
MORPHO 8	**3.935** ± 0.483

TABLE 7.6 – Évolution de la variance intra-classe selon la séparation des sujets

7.3.5.3 Comparaison des performances avec la procédure de co-training classique

Nous avons, à titre d'information, comparé notre procédure EBSICO avec la procédure de co-training classique dans la configuration qui suit :

1. entraînement de μ_1 et μ_2 respectivement sur $CFEE_A^{[AU]}$ et $CFEE_A^{[TX]}$,
2. l'incrémentation des modèles se fait à partir de $CK+_I$ qui a été divisé à 10% de labels, afin que les effectifs soient : 90 données pour L et 811 données pour U, sans slots,
3. première évaluation des modèles après incrémentation sur L,
4. incrémentation ensuite de type co-train, avec la méthode classique (ré-entraînement depuis zéro à chaque itération) et notre méthode co-incrémentation,
5. deuxième évaluation sur $CK+_E$ après incrémentation sur U.

Les accuracies sont données par paire : la première correspond au modèle AU, et la seconde au modèle TX. Le seuil de confiance a été fixé à 0.8. Enfin, on calcule le temps d'exécution total de la séquence, afin de comparer la méthode classique et la méthode incrémentale.

Les résultats sont présentés dans le tableau 7.7. On peut observer que lorsqu'un nombre conséquent de données dans U est présent, le modèle classique voit ses performances décroître drastiquement, là où le modèle incrémental stagne, voire améliore le modèle μ_2. Une explication possible est que les données sur lesquelles le modèle doit se spécialiser sont présentes en plus petite quantité que celles sur lequelles il s'est entraîné initialement et se retrouvent "noyées" dans l'ensemble de données complet

à chaque ré-apprentissage from scratch, là où ce n'est pas le cas avec la stratégie d'incrémentation IGT. Enfin, la procédure EBSICO que l'on propose présente un intérêt majeur en terme de rapidité d'exécution, près de 80 fois plus rapide dans cette expérimentation. La procédure de co-training a tout intérêt dans le futur à bénéficier de l'aspect incrémental si elle se veut plus efficace.

Mesures / Méthodes	acc après L (90)	acc après U (811)	temps exécution
co-training classique	$(0.957, 0.888)$	$(0.91, 0.909)$	245 min
co-incrémentation	**(0.96,0.897)**	**(0.954, 0.933)**	**3 min**

TABLE 7.7 – Comparaison de la méthode de co-training classique avec notre méthode de co-incrémentation

7.4 Conclusion

Dans cette section, nous avons proposé une méthode hybride, qui combine deux techniques d'apprentissage automatique, à savoir le co-training et l'apprentissage incrémental. Contrairement à la méthode classique de co-training qui effectue un ré-entraînement à partir de zéro à chaque itération de l'algorithme, notre approche effectue une incrémentation du modèle en continu sur de nouvelles observations. Le principal avantage de l'utilisation de l'apprentissage incrémental par rapport au ré-entraînement d'un modèle à partir de zéro, outre le temps d'exécution, est d'éviter "l'oubli catastrophique". Les NCMFs offrent une résistance robuste aux modèles entraînés plusieurs étapes auparavant.

Deuxièmement, nous fournissons une analyse approfondie de la personnalisation des modèles pour la reconnaissance des émotions. Les modèles prenant en compte les caractéristiques morphologiques ont présenté de meilleures performances que le regroupement par sexe. En effet, une technique rationalisée de regroupement des features est cruciale pour la performance des modèles de co-training.

Enfin, notre troisième contribution concerne le domaine de l'apprentissage semi-supervisé, plus précisément, sur la capacité des modèles à augmenter leurs performances avec seulement 5% d'observations labélisées, comme le démontrent nos expérimentations. Nos expériences ont été menées avec de petits datasets, mais nous pourrions imaginer dans de futurs travaux de recherche utiliser cette technique avec des bases de données plus importantes.

Chapitre 8

Conclusions et perspectives

8.1 Conclusions

Parmi les nombreux défis entourant le domaine de la reconnaissance des émotions, le travail présenté dans cette thèse s'est concentré sur la problématique du biais d'identité.

Tout d'abord, nous avons présenté les résultats obtenus conjointement avec les autres membres du consortium TEEC sur une analyse préliminaire des données du projet en adoptant deux points de vue différents : l'étude des interactions verbales (cognitives) et non verbales (affectives). L'analyse qualitative de ces résultats a renforcé notre idée que des corrélations existent entre ces deux types d'interaction et qu'elles peuvent conduire à un effet Eurêka, validant ainsi le scénario pédagogique mis en place par les didacticiens du consortium. Par la suite, nous avons porté notre attention sur le biais d'identité qui peut causer des problèmes de performance dans des situations comme ici dans TEEC où nous devons faire face à une grande diversité dans les données.

Dans ce contexte, les objectifs de cette thèse étaient d'adapter automatiquement des modèles d'apprentissage automatique aux traits morphologiques et comportementaux de l'individu, afin de réduire le biais d'identité ; celui-ci constitue un défi majeur car, entraînant une augmentation de la variabilité intra-classe, les performances d'un classifieur générique s'en retrouvent altérées. Nous avons réalisé à cet effet :

— une analyse approfondie de l'approche visant à personnaliser un modèle générique NCMF sur un groupe particulier de sujets pour améliorer sa reconnaissance de leurs états affectifs. L'approche consiste, tout d'abord, à entraîner le modèle sur un premier dataset, afin qu'il généralise correctement, puis, après avoir séparé les sujets d'un second dataset en slots et assigné une copie

du modèle à chacun d'eux, à incrémenter chaque modèle avec des données du slot auquel il est assigné, afin de le spécialiser aux sujets le constituant.

Dans nos expérimentations, nous avons observé que l'apprentissage incrémental pouvait être sensible aux dérives conceptuelles et avons proposé un clustering multimodal dans les nœuds au moment de la séparation des données lors de l'incrémentation (IGTC). Nous montrons que notre modèle atteint des performances compétitives avec un réseau CNN. De plus, le réseau déroge au principe de stabilité du dilemme plasticité-stabilité en ayant souffert d'un sérieux oubli catastrophique, contrairement à la NCMF.

— une combinaison de deux approches, l'apprentissage semi-supervisé et l'apprentissage incrémental via le pipeline de co-incrémentation EBSICO. Nous montrons que notre modèle est stable et est capable de fournir de très bonnes performances avec seulement 5 % de données labélisées (25 données en moyenne). Cette analyse a été réalisée en utilisant différents critères de clustering : nous montrons que l'utilisation d'un critère comme la morphologie, montre de meilleurs résultats qu'un critère de séparation tel que le genre ou qu'un modèle non spécialisé. De plus, en utilisant les avantages de l'apprentissage incrémental, la procédure proposée dans ce mémoire est 80 fois plus rapide que la procédure classique de co-training.

Un modèle effectuant un apprentissage incrémental présente l'avantage de pouvoir être utilisé dans des dispositifs qui doivent apprendre au fil du temps à partir de données variables, de manière rapide et fiable. Les résultats obtenus nous montrent que les modèles d'apprentissage automatique tels que les forêts aléatoires NCM ne doivent pas être considérés comme moins fiables que les réseaux de neurones dans un contexte d'apprentissage incrémental et que les AU peuvent être considérées comme des features de qualité pour le domaine de la reconnaissance des expressions faciales.

En ce qui concerne l'apprentissage semi-supervisé, la combinaison de l'apprentissage incrémental et de l'apprentissage semi-supervisé nous a permis d'obtenir un résultat pertinent : l'algorithme que nous proposons a réduit le temps de calcul inutilement perdu lors de la phase classique de co-training. Les pistes significatives qui pourraient résulter d'une consommation moins gourmande en énergie, devraient faire l'objet d'une attention particulière. À cet égard, nous pensons que les techniques d'apprentissage incrémental sont un moyen possible d'améliorer notre utilisation des algorithmes d'apprentissage automatique de manière plus intelligente.

Au fur et à mesure de l'avancement de nos travaux, nous avons constaté que plusieurs pistes

intéressantes devaient être explorées.

8.2 Perspectives

Apprentissage incrémental supervisé en classe : L'intérêt de l'apprentissage incrémental en classe est d'éviter de devoir ré-entraîner un modèle déjà spécialisé lorsqu'une nouvelle classe à intégrer est disponible, par exemple une émotion académique [104], plus appropriée à notre contexte. Le modèle proposé dans ce travail est capable d'apprendre classe par classe, selon une méthodologie proche de [19] [65] : le modèle est incrémenté à partir de batchs qui arrivent successivement ; un batch contient toutes les données d'une classe particulière. Les résultats préliminaires que nous avons obtenu (cf. Figure 8.1) avec le jeu de données CK+ suggèrent qu'il pourrait y avoir un impact sur les performances du modèle selon l'ordre des classes à apprendre de manière incrémentale. Il serait intéressant de mener des expérimentations afin d'étudier cela plus en profondeur ; certaines stratégies dans la littérature ont déjà été proposées, comme par exemple, commencer par l'apprentissage avec le batch de la classe neutre [119], considéré comme une tâche plus simple, on parle de « curriculum learning » [120] ; certains modèles seraient performants en recevant les classes en suivant un ordre de difficulté, du plus simple au plus complexe, alors que les performances chutent lorsque cet ordre est aléatoire.

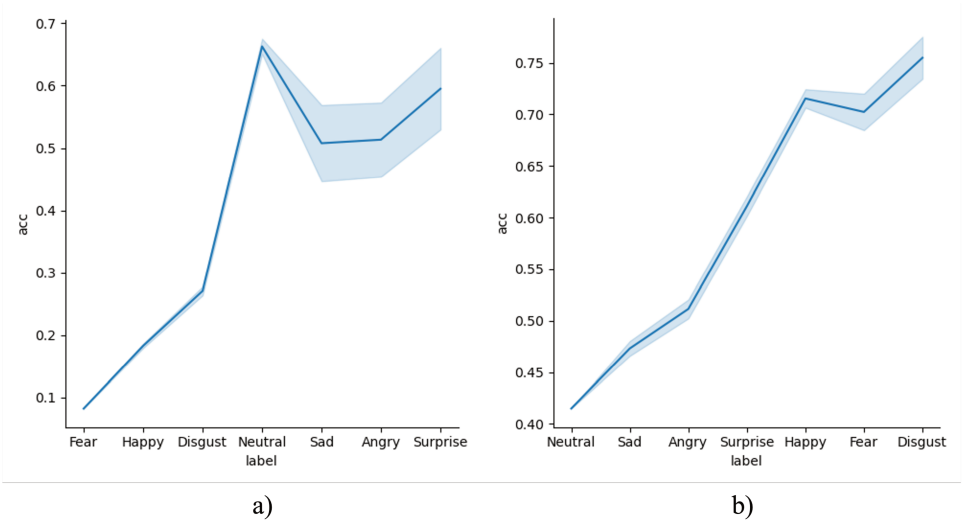

FIGURE 8.1 – Apprentissage incrémental en classe de la NCMF - a) la forêt est initialisée sur l'émotion peur tandis que dans b) elle commence par apprendre l'émotion Neutre

Modèle de mélange gaussien : Nous avons considéré et étudié dans ce mémoire la séparation des sujets selon un critère rationnel pour diminuer la variabilité intra-classe introduite par le biais d'identité. Une solution plus élégante à ce problème serait de modéliser directement les densités de probabilité conditionnelles des classes et d'identifier leurs modes. Pour ce faire, les modèles de mélange gaussiens (*Gaussian Mixture Models* - GMM) offrent des perspectives intéressantes.

Un GMM est un modèle probabiliste basé sur une distribution gaussienne multimodale qui peut être utilisé pour estimer la présence de sous-populations au sein d'une classe sans nécessiter l'identification de la sous-population d'intérêt (non supervisée) [121]. Il a déjà fait ses preuves dans de différents domaines lorsqu'il s'agit de traiter la variabilité intra-classe ou intra-sujet [122][123][124]. Il a également été utilisé dans le contexte de l'apprentissage incrémental des dérives de concept. Les auteurs de [125] proposent à cet égard une version incrémentale des GMM utilisant l'algorithme *Incremental Gaussian Mixture Model with Concept Drift* (IGMM-CD). Enfin, cet algorithme permet de mettre en place des techniques de répétition de type *pseudo-rehearsal*. En tant que modèles génératif, il permet la régénération des données. Dans [126], les auteurs les utilisent pour régénérer des données synthétiques afin d'éviter l'oubli catastrophique.

C'est pourquoi nous avons proposé une nouvelle architecture de forêt NCMF hybride. Le noeud racine de chaque arbre est désormais un noeud génératif qui contient un mode de la distribution conditionnelle (instancié par sa moyenne et sa matrice de covariance). Il va permettre de séparer les données de la classe de celles des autres classes et de repérer les *outliers*. Les noeuds suivants sont des noeuds NCM "classiques", discriminants. Les performances obtenues par cette forêt hybride sur des données *benchmarks* (dont MNIST) sont très encourageantes et le modèle fera l'objet d'une prochaine publication, en cours de rédaction.

Annotation automatique des effets Eurêka : L'analyse conjointe réalisée dans le projet TEEC a été réalisée manuellement (voir Sec.4) ; il serait intéressant d'étudier la possibilité de pouvoir automatiser l'analyse verbale. En outre, nous pourrions obtenir davantage d'informations concernant les tours de paroles à partir des canaux visuel et audio.

La détection automatique du locuteur (*speaker diarization*) permet de savoir ce qui est dit, à quel moment et par qui [127] ; ceci permettrait d'enrichir le contexte autour des moments de la vidéo susceptibles d'être marqués par des changements d'états émotionnels. Il est courant de fusionner les informations collectées via les canaux audio et visuel pour améliorer la détection

[128]. *TalkNet* est un pipeline *end-to-end* proposé récemment par [129] ; il prend la vidéo du visage recadré et l'audio correspondant en entrée, puis, il décide si la personne parle dans chaque image de la vidéo. Dans [130], les auteurs utilisent les landmarks autour des lèvres pour détecter les mouvements de bouche lors de meetings vidéo, avec une seule personne par webcam. À l'inverse, dans TEEC, plusieurs élèves apparaissent en même temps par webcam. OpenFace intègre un modèle de détection du regard que nous pourrions exploiter pour (éventuellement), d'une part, identifier les dyades lors de conversations intra-groupe, typiquement lorsque deux élèvent se regardent par exemple ; d'autre part, pour identifier des instants où tous les regards convergent vers un point fixe devant eux, i.e. la webcam, (potentiellement) pendant des échanges inter-groupe. Ainsi, l'agrégation de toutes ces informations serait une piste intéressante à étudier pour savoir quand et qui parle, et à qui.

Enfin, l'interprétation en langage naturel (*Natural Language Inference* - NLI), une branche de plus en plus importante pour la recherche sur la compréhension du langage naturel (*Natural Language Understanding* - NLU), consiste à déterminer si un segment de texte est impliqué, contredit ou non, par ou sans rapport, avec un autre texte [131]. Ceci permettrait de qualifier l'information énoncée, collectée via le module de transcription en général intégré au pipeline de speaker diarization, selon qu'il s'agit d'un accord, d'un désaccord, etc. C'est une technique prometteuse pour automatiser le processus d'annotations verbales vu à la Section 4.

Les informations précédemment décrites, collectées via deux canaux audio et visuel, couplées à l'analyse émotionnelle visuelle nous permettraient ainsi de détecter, de manière automatique, les contagions cognitives ainsi que les conflits socio-cognitifs, autrement dit les effets Eurêka.

Publications

Lionel Prevost, Chloé Le Bail, François-Xavier Bernard, Michael Baker, Françoise Detienne, **Jordan Gonzalez**, Context effects in collaborative learning : verbal and non-verbal indicators, **Conference of the European Science Education Research Association** (ESERA). Bologna, Italia, **2019**.

Mélanie Piot, Thybault. Alabarbe, **Jordan Gonzalez**, Chloé Le Bail, Lionel Prevost, Jacqueline Bourdeau, François-Xavier Bernard, Michael Baker, Françoise Detienne, Joint analysis of verbal and nonverbal interactions in collaborative e-learning, 8th International **Conference on Affective Computing and Intelligent Interaction**, Workshops and Demos (ACIIW), Cambridge, UK, **2019**.

Mélanie Piot, Bérangère Bourdoulous, **Jordan Gonzalez**, Aurelia Deshayes, Lionel Prevost, Dual-Memory Model for Incremental Learning : The Handwriting Recognition Use Case, **International Conference on Pattern Recognition** (ICPR), 5527-5534, Milano, Italia, **2020**.

Jordan Gonzalez, Lionel Prevost, Personalizing emotion recognition using incremental random forests, **European Signal Processing Conference** (EUSIPCO), 781-785, virtual only, **2021**.

Jordan Gonzalez, Lionel Prevost, Mélanie Piot, Thybault. Alabarbe, Analyse des interactions émotionnelles dans les apprentissages collaboratifs, **Congrès de l'ACFAS**, virtual only, **2021**.

Jordan Gonzalez, Thibault Geoffroy, Aurelia Deshayes, Lionel Prevost, Co-incrementation : combining co-training and incremental learning for subject-specific facial expression recognition, soumis à **International Conference on Pattern Recognition Applications and Methods** (ICPRAM), Lisbon, Portugal, **2022**.

Bibliographie

[1] M. Belkin, D. Hsu, S. Ma et S. Mandal, "Reconciling modern machine learning practice and the bias-variance trade-off," *arXiv preprint arXiv :1812.11118*, 2018.

[2] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, n°. 6, p. 1161, 1980.

[3] J. Bourdeau, T. Forissier, Y. Mazabraud et R. Nkambou, "Web-based context-aware science learning," dans *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, p. 1415–1418.

[4] T. Forissier, J. Bourdeau, Y. Mazabraud et R. Nkambou, "Modeling context effects in science learning : the clash model," dans *International and Interdisciplinary Conference on Modeling and Using Context*. Springer, 2013, p. 330–335.

[5] B. Van Oers, "From context to contextualizing," *Learning and instruction*, vol. 8, n°. 6, p. 473–488, 1998.

[6] E. Sariyanidi, H. Gunes et A. Cavallaro, "Automatic analysis of facial affect : A survey of registration, representation, and recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, n°. 6, p. 1113–1133, 2014.

[7] L. Oudot, L. Prevost et M. Milgram, "Self-supervised adaptation for on-line script text recognition," *ELCVIA : electronic letters on computer vision and image analysis*, p. 87–97, 2005.

[8] C. Leroy, E. Anquetil et N. Girard, "Drift anticipation with forgetting to improve evolving fuzzy system," dans *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, p. 5836–5843.

[9] S. Yang et B. Bhanu, "Facial expression recognition using emotion avatar image," dans *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE, 2011,

p. 866–871.

[10] G. Zen, E. Sangineto, E. Ricci et N. Sebe, "Unsupervised domain adaptation for personalized facial emotion recognition," dans *Proceedings of the 16th international conference on multimodal interaction*, 2014, p. 128–135.

[11] W.-S. Chu, F. De la Torre et J. F. Cohn, "Selective transfer machine for personalized facial action unit detection," dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, p. 3515–3522.

[12] M. Feffer, O. O. Rudovic et R. W. Picard, "A mixture of personalized experts for human affect estimation," dans *International conference on machine learning and data mining in pattern recognition*. Springer, 2018, p. 316–330.

[13] J. Jiang, "A literature survey on domain adaptation of statistical classifiers," *URL : http ://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, vol. 3, n°. 1-12, p. 3, 2008.

[14] M. Rescigno, M. Spezialetti et S. Rossi, "Personalized models for facial emotion recognition through transfer learning," *Multimedia Tools and Applications*, vol. 79, n°. 47, p. 35 811–35 828, 2020.

[15] V. Kollia, "Personalization effect on emotion recognition from physiological data : An investigation of performance on different setups and classifiers," *arXiv preprint arXiv :1607.05832*, 2016.

[16] G. M. Weiss et J. Lockhart, "The impact of personalization on smartphone-based activity recognition," dans *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[17] T. Sztyler et H. Stuckenschmidt, "Online personalization of cross-subjects based activity recognition models on wearable devices," dans *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2017, p. 180–189.

[18] A. Ferrari, D. Micucci, M. Mobilio et P. Napoletano, "On the personalization of classification models for human activity recognition," *IEEE Access*, vol. 8, p. 32 066–32 079, 2020.

[19] M. Ristin, M. Guillaumin, J. Gall et L. Van Gool, "Incremental learning of ncm forests for large-scale image classification," dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, p. 3654–3661.

[20] L. Breiman, J. H. Friedman, R. A. Olshen et C. J. Stone, *Classification and regression trees*. The Wadsworth Statistics/Probability Series, Wadsworth, Belmont, 1984.

[21] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, n°. 1, p. 81–106, 1986.

[22] ——, *C4. 5 : programs for machine learning*. Elsevier, 2014.

[23] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, n°. 1, p. 5–32, oct. 2001. [En ligne]. Disponible : https://link.springer.com/article/10.1023/A:1010933404324

[24] F. Provost et P. Domingos, "Tree induction for probability-based ranking," *Machine learning*, vol. 52, n°. 3, p. 199–215, 2003.

[25] J. Tanha, M. van Someren et H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *International Journal of Machine Learning and Cybernetics*, vol. 8, n°. 1, p. 355–370, 2017.

[26] W. S. McCulloch et W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of mathematical biology*, vol. 52, n°. 1, p. 99–115, 1990.

[27] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, n°. 10, p. 1995, 1995.

[28] D. Mehta, M. F. H. Siddiqui et A. Y. Javaid, "Facial emotion recognition : A survey and real-world user experiences in mixed reality," *Sensors*, vol. 18, n°. 2, p. 416, 2018.

[29] Z. Liu, Y. Peng et W. Hu, "Driver fatigue detection based on deeply-learned facial expression representation," *Journal of Visual Communication and Image Representation*, vol. 71, p. 102723, 2020.

[30] I. Naim, M. I. Tanveer, D. Gildea et M. E. Hoque, "Automated prediction and analysis of job interview performance : The role of what you say and how you say it," dans *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, vol. 1. IEEE, 2015, p. 1–6.

[31] C. Grossard, S. Hun, A. Dapogny, E. Juillet, F. Hamel, H. Jean-Marie, J. Bourgeois, H. Pellerin, P. Foulon, S. Serret *et al.*, "Teaching facial expression production in autism : The serious game jemime," *Creative Education*, vol. 10, n°. 11, p. 2347, 2019.

[32] S. D. Craig, S. D'Mello, A. Witherspoon et A. Graesser, "Emote aloud during learning with autotutor : Applying the facial action coding system to cognitive–affective states during learning," *Cognition and Emotion*, vol. 22, n°. 5, p. 777–788, 2008.

[33] P. Ekman et W. V. Friesen, "Constants across cultures in the face and emotion." *Journal of personality and social psychology*, vol. 17, n°. 2, p. 124, 1971.

[34] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, n°. 3-4, p. 169–200, 1992.

[35] S. Du, Y. Tao et A. M. Martinez, "Compound facial expressions of emotion," *Proceedings of the National Academy of Sciences*, vol. 111, n°. 15, p. E1454–E1462, 2014.

[36] M. Pantic et L. Rothkrantz, "Toward an affect-sensitive multimodal human-computer interaction," *Proceedings of the IEEE*, vol. 91, n°. 9, p. 1370–1390, sept. 2003. [En ligne]. Disponible : http://ieeexplore.ieee.org/document/1230215/

[37] M. K. Greenwald, E. W. Cook, P. J. Lang et others, "Affective judgment and psychophysiological response : Dimensional covariation in the evaluation of pictorial stimuli," *Journal of psychophysiology*, vol. 3, n°. 1, p. 51–64, 1989.

[38] A. Metallinou et S. Narayanan, "Annotation and processing of continuous emotional attributes : Challenges and opportunities," dans *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*. IEEE, 2013, p. 1–8.

[39] P. Ekman et W. V. Friesen, *Manual for the facial action coding system*. Consulting Psychologists Press, 1978.

[40] R. Ekman, *What the face reveals : Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[41] W. V. Friesen et P. Ekman, "EMFACS-7 : Emotional facial action coding system," *Unpublished manuscript, University of California at San Francisco*, vol. 2, n°. 36, p. 1, 1983.

[42] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic et K. Scherer, "Meta-Analysis of the First Facial Expression Recognition Challenge," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, n°. 4, p. 966–979, août 2012.

[43] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar et I. Matthews, "The extended cohn-kanade dataset (ck+) : A complete dataset for action unit and emotion-specified expression," dans

2010 ieee computer society conference on computer vision and pattern recognition-workshops. IEEE, 2010, p. 94–101.

[44] M. Venet, E. Correa Molina et F. Saussez, "Pédagogie universitaire et accompagnement dans la zone proximale de développement des enseignants et enseignantes en formation initiale et continue," *Nouveaux cahiers de la recherche en éducation*, vol. 19, n°. 1, p. 1–10, 2016.

[45] R. W. Picard, S. Papert, W. Bender, B. Blumberg, C. Breazeal, D. Cavallo, T. Machover, M. Resnick, D. Roy et C. Strohecker, "Affective learning—a manifesto," *BT technology journal*, vol. 22, n°. 4, p. 253–269, 2004.

[46] K. Loderer, R. Pekrun et J. C. Lester, "Beyond cold technology : A systematic review and meta-analysis on emotions in technology-based learning environments," *Learning and instruction*, vol. 70, p. 101162, 2020.

[47] S. D'mello et A. Graesser, "Autotutor and affective autotutor : Learning by talking with cognitively and emotionally intelligent computers that talk back," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 2, n°. 4, p. 1–39, 2013.

[48] S. D'Mello, R. W. Picard et A. Graesser, "Toward an affect-sensitive autotutor," *IEEE Intelligent Systems*, vol. 22, n°. 4, p. 53–61, 2007.

[49] C. Cherniss, M. Extein, D. Goleman et R. P. Weissberg, "Emotional intelligence : what does the research really indicate ?" *Educational psychologist*, vol. 41, n°. 4, p. 239–245, 2006.

[50] B. Martinez, M. F. Valstar, B. Jiang et M. Pantic, "Automatic analysis of facial actions : A survey," *IEEE Transactions on Affective Computing*, 2017.

[51] S. D'Mello et R. A. Calvo, "Beyond the basic emotions : what should affective computing compute ?" dans *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, p. 2287–2294.

[52] N. Bosch, S. K. D'mello, J. Ocumpaugh, R. S. Baker et V. Shute, "Using video to automatically detect learner affect in computer-enabled classrooms," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 6, n°. 2, p. 17, 2016.

[53] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan et S. Wermter, "Continual lifelong learning with neural networks : A review," *Neural Networks*, vol. 113, p. 54–71, 2019.

[54] J. He, R. Mao, Z. Shao et F. Zhu, "Incremental learning in online scenario," dans *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, p. 13 926–13 935.

[55] C. Zhang, Y. Zhang, X. Shi, G. Almpanidis, G. Fan et X. Shen, "On incremental learning for gradient boosting decision trees," *Neural Processing Letters*, vol. 50, n°. 1, p. 957–987, 2019.

[56] A. Rosenfeld et J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, n°. 3, p. 651–663, 2018.

[57] I. Biederman, "Recognition-by-components : a theory of human image understanding." *Psychological review*, vol. 94, n°. 2, p. 115, 1987.

[58] R. Polikar, L. Upda, S. S. Upda et V. Honavar, "Learn++ : An incremental learning algorithm for supervised neural networks," *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, vol. 31, n°. 4, p. 497–508, 2001.

[59] G. Jaber, "An approach for online learning in the presence of concept changes," Thèse de doctorat, Université Paris Sud-Paris XI, 2013.

[60] M. McCloskey et N. J. Cohen, "Catastrophic interference in connectionist networks : The sequential learning problem," dans *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, p. 109–165.

[61] K. Shmelkov, "Approaches for incremental learning and image generation," Thèse de doctorat, Université Grenoble Alpes, 2019.

[62] K. Javed et F. Shafait, "Revisiting distillation and incremental classifier learning," dans *Asian conference on computer vision*. Springer, 2018, p. 3–17.

[63] R. Ratcliff, "Connectionist models of recognition memory : constraints imposed by learning and forgetting functions." *Psychological review*, vol. 97, n°. 2, p. 285, 1990.

[64] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, n°. 2, p. 123–146, 1995.

[65] S.-A. Rebuffi, A. Kolesnikov, G. Sperl et C. H. Lampert, "icarl : Incremental classifier and representation learning," dans *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, p. 2001–2010.

[66] S. Mittal, S. Galesso et T. Brox, "Essentials for class incremental learning," dans *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, p. 3513–3522.

[67] G. M. Van de Ven et A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv :1904.07734*, 2019.

[68] P. Domingos et G. Hulten, "Mining high-speed data streams," dans *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, p. 71–80.

[69] W. Hoeffding, "Probability inequalities for sums of bounded random variables," dans *The collected works of Wassily Hoeffding*. Springer, 1994, p. 409–426.

[70] M. Denil, D. Matheson et N. Freitas, "Consistency of online random forests," dans *International conference on machine learning*. PMLR, 2013, p. 1256–1264.

[71] B. Lakshminarayanan, D. M. Roy et Y. W. Teh, "Mondrian forests : Efficient online random forests," *Advances in neural information processing systems*, vol. 27, p. 3140–3148, 2014.

[72] C. Hu, Y. Chen, L. Hu et X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognition*, vol. 78, p. 277–290, 2018.

[73] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, n°. 1, p. 44–53, 2018.

[74] J. E. Van Engelen et H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, n°. 2, p. 373–440, 2020.

[75] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," dans *33rd annual meeting of the association for computational linguistics*, 1995, p. 189–196.

[76] C. Rosenberg, M. Hebert et H. Schneiderman, "Semi-supervised self-training of object detection models," 2005.

[77] F. Roli et G. L. Marcialis, "Semi-supervised pca-based face recognition using self-training," dans *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2006, p. 560–568.

[78] N. Cherniavsky, I. Laptev, J. Sivic et A. Zisserman, "Semi-supervised learning of facial attributes in video," dans *European Conference on Computer Vision*. Springer, 2010, p. 43–56.

[79] M. Kächele et F. Schwenker, "Semi-supervised dictionary learning of sparse representations for emotion recognition," dans *IAPR International Workshop on Partially Supervised Learning*. Springer, 2013, p. 21–35.

[80] V. Kumar, S. Rao et L. Yu, "Noisy student training using body language dataset improves facial expression recognition," dans *European Conference on Computer Vision*. Springer, 2020, p. 756–773.

[81] W. Wang et Z.-H. Zhou, "A new analysis of co-training," dans *ICML*, 2010.

[82] A. Blum et T. Mitchell, "Combining labeled and unlabeled data with co-training," dans *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, p. 92–100.

[83] I. Cohen, N. Sebe, F. G. Cozman et T. S. Huang, "Semi-supervised learning for facial expression recognition," dans *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, 2003, p. 17–22.

[84] Z. Zhang, F. Ringeval, B. Dong, E. Coutinho, E. Marchi et B. Schüller, "Enhanced semi-supervised learning for multimodal emotion recognition," dans *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, p. 5185–5189.

[85] F. Ringeval, A. Sonderegger, J. Sauer et D. Lalanne, "Introducing the recola multimodal corpus of remote collaborative and affective interactions," dans *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*. IEEE, 2013, p. 1–8.

[86] N. Alyuz, E. Okur, E. Oktay, U. Genc, S. Aslan, S. E. Mete, B. Arnrich et A. A. Esme, "Semi-supervised model personalization for improved detection of learner's emotional engagement," dans *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, p. 100–107.

[87] A. Niculescu-Mizil et R. Caruana, "Predicting good probabilities with supervised learning," dans *Proceedings of the 22nd international conference on Machine learning*, 2005, p. 625–632.

[88] B. C. Ko, "A brief review of facial emotion recognition based on visual information," *sensors*, vol. 18, n°. 2, p. 401, 2018.

[89] S. Li et W. Deng, "Deep facial expression recognition : A survey," *IEEE transactions on affective computing*, 2020.

[90] B. Goldberg et M. Boyce, "Experiential intelligent tutoring : Using the environment to contextualize the didactic," dans *International Conference on Augmented Cognition*. Springer, 2018, p. 192–204.

[91] R. S. Schwartz, N. G. Lederman et B. A. Crawford, "Developing views of nature of science in an authentic context : An explicit approach to bridging the gap between nature of science and scientific inquiry," *Science education*, vol. 88, n°. 4, p. 610–645, 2004.

[92] D. King, E. Winner et I. Ginns, "Outcomes and implications of one teacher's approach to context-based science in the middle years," *Teaching science*, vol. 57, n°. 2, p. 26–34, 2011.

[93] M. Van Eijck et W.-M. Roth, "Towards a chronotopic theory of "place" in place-based education," *Cultural Studies of Science Education*, vol. 5, n°. 4, p. 869–898, 2010.

[94] C. Buchs et F. Butera, "Socio-cognitive conflict and the role of student interaction in learning," *New Review of Social Psychology*, vol. 3, p. 80–87, 2004.

[95] H. Clark et E. Schaefer, "'contributing to discourse'cognitive science 13," 1989.

[96] B. B. Schwarz et M. J. Baker, *Dialogue, argumentation and education : History, theory and practice*. Cambridge University Press, 2016.

[97] P. Dillenbourg, M. Baker, A. Blaye et C. O'malley, "The evolution of research on collaborative learning in h. spada and p. reimann (eds) learning in humans and machines," *Elsevier*, vol. 1, n°. 1, p. 58–94, 1996.

[98] M. Baker, J. Andriessen, K. Lund, M. van Amelsvoort et M. Quignard, "Rainbow : A framework for analysing computer-mediated pedagogical debates," *International Journal of Computer-Supported Collaborative Learning*, vol. 2, n°. 2, p. 315–357, 2007.

[99] M. R. Lepper, M. Woolverton, D. L. Mumme et J. Gurtner, "Motivational techniques of expert human tutors : Lessons for the design of computer-based tutors," *Computers as cognitive tools*, vol. 1993, p. 75–105, 1993.

[100] J. Bourdeau, "The dbr methodology for the study of context in learning," dans *International and Interdisciplinary Conference on Modeling and Using Context*. Springer, 2017, p. 541–553.

[101] F. Wang et M. J. Hannafin, "Design-based research and technology-enhanced learning environments," *Educational technology research and development*, vol. 53, n°. 4, p. 5–23, 2005.

[102] L. Chartofylaka et T. Forissier, "Underpinning intercultural competence within a context-effect didactic method," dans *21st General Meeting and Conference (CAS-2018), 27th–30th November 2018*, 2018.

[103] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee *et al.*, "Challenges in representation learning : A report on three machine learning contests," dans *International conference on neural information processing*. Springer, 2013, p. 117–124.

[104] R. Pekrun et E. J. Stephens, "Academic emotions." dans *APA educational psychology handbook, Vol 2 : Individual differences and cultural and contextual factors.* American Psychological Association, 2012, p. 3–31.

[105] Z. Zheng, X. Li, J. Barnes, C.-H. Park et M. Jeon, "Facial expression recognition for children : Can existing methods tuned for adults be adopted for children ?" dans *International Conference on Human-Computer Interaction*. Springer, 2019, p. 201–211.

[106] T. Baltrusaitis, A. Zadeh, Y. C. Lim et L.-P. Morency, "Openface 2.0 : Facial behavior analysis toolkit," dans *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018, p. 59–66.

[107] P. Viola et M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, n°. 2, p. 137–154, 2004.

[108] T. Ahonen, A. Hadid et M. Pietikainen, "Face description with local binary patterns : Application to face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, n°. 12, p. 2037–2041, 2006.

[109] N. Dalal et B. Triggs, "Histograms of oriented gradients for human detection," dans *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, p. 886–893.

[110] A. Mollahosseini, D. Chan et M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," dans *2016 IEEE Winter conference on applications of computer vision (WACV)*. IEEE, 2016, p. 1–10.

[111] M. Piot, B. Bourdoulous, J. Gonzalez, A. Deshayes et L. Prevost, "Dual-memory model for incremental learning : The handwriting recognition use case," dans *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, p. 5527–5534.

[112] T. Caliński et J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, n°. 1, p. 1–27, 1974.

[113] S. Thrun, "Is learning the -th thing any easier than learning the first ?" 1996.

[114] D. Goren et H. R. Wilson, "Quantifying facial expression recognition across viewing conditions," *Vision research*, vol. 46, n°. 8-9, p. 1253–1262, 2006.

[115] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann et S. Narayanan, "Analysis of emotion recognition using facial expressions, speech and multimodal information," dans *Proceedings of the 6th international conference on Multimodal interfaces*, 2004, p. 205–211.

[116] L. S. Chen, T. S. Huang, T. Miyasato et R. Nakatsu, "Multimodal human emotion/expression recognition," dans *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 1998, p. 366–371.

[117] Z.-H. Zhou et M. Li, "Semi-supervised learning by disagreement," *Knowledge and Information Systems*, vol. 24, n°. 3, p. 415–439, 2010.

[118] C. A. Shipp et L. I. Kuncheva, "Relationships between combination methods and measures of diversity in combining classifiers," *Information fusion*, vol. 3, n°. 2, p. 135–148, 2002.

[119] N. Churamani et H. Gunes, "Clifer : Continual learning with imagination for facial expression recognition," dans *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*. IEEE, 2020, p. 322–328.

[120] L. Gui, T. Baltrušaitis et L.-P. Morency, "Curriculum learning for facial expression recognition," dans *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, p. 505–511.

[121] G. J. McLachlan et D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[122] N. Kim, M. Heo, R. Fleysher, C. A. Branch et M. L. Lipton, "A gaussian mixture model approach for estimating and comparing the shapes of distributions of neuroimaging data : diffusion-measured aging effects in brain white matter," *Frontiers in Public Health*, vol. 2, p. 32, 2014.

[123] L. Moraru, S. Moldovanu, L. T. Dimitrievici, N. Dey, A. S. Ashour, F. Shi, S. J. Fong, S. Khan et A. Biswas, "Gaussian mixture model for texture characterization with application to brain dti images," *Journal of advanced research*, vol. 16, p. 15–23, 2019.

[124] C. Migliorelli, S. Romero, A. Bachiller, J. Aparicio, J. F. Alonso, M. A. Mañanas et V. San Antonio-Arce, "Improving the ripple classification in focal pediatric epilepsy : identifying pa-

thological high-frequency oscillations by gaussian mixture model clustering," *Journal of neural engineering*, vol. 18, n°. 4, p. 0460f2, 2021.

[125] L. S. Oliveira et G. E. Batista, "Igmm-cd : a gaussian mixture classification algorithm for data streams with concept drifts," dans *2015 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2015, p. 55–61.

[126] B. Pfülb et A. Gepperth, "Overcoming catastrophic forgetting with gaussian mixture replay," dans *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, p. 1–9.

[127] J. Roth, S. Chaudhuri, O. Klejch, R. Marvin, A. Gallagher, L. Kaver, S. Ramaswamy, A. Stopczynski, C. Schmid, Z. Xi *et al.*, "Ava active speaker : An audio-visual dataset for active speaker detection," dans *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, p. 4492–4496.

[128] J. S. Chung, B.-J. Lee et I. Han, "Who said that ? : Audio-visual speaker diarisation of real-world meetings," *arXiv preprint arXiv :1906.10042*, 2019.

[129] R. Tao, Z. Pan, R. K. Das, X. Qian, M. Z. Shou et H. Li, "Is someone speaking ? exploring long-term temporal features for audio-visual active speaker detection," dans *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, p. 3927–3935.

[130] P. Cabañas-Molero, M. Lucena, J. M. Fuertes, P. Vera-Candeas et N. Ruiz-Reyes, "Multimodal speaker diarization for meetings using volume-evaluated srp-phat and video analysis," *Multimedia Tools and Applications*, vol. 77, n°. 20, p. 27 685–27 707, 2018.

[131] X. F. Zhang et M.-C. de Marneffe, "Identifying inherent disagreement in natural language inference," dans *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, 2021.