Méthodes d'apprentissage automatique pour l'analyse des interactions utilisateurs

Nicolas Labroche

Université Pierre et Marie Curie – Paris VI

HABILITATION À DIRIGER LES RECHERCHES

spécialité **Informatique**

présentée par

Nicolas LABROCHE

Sujet :

Méthodes d'apprentissage automatique pour l'analyse des interactions utilisateurs

soutenue le jeudi 13 décembre 2012

devant le jury composé de

M. Thierry Artières	
Mme Bernadette Bouchon-Meunier	
M. Mohand Boughanem	rapporteur
M. Pierre Gançarski	rapporteur
M. Pascal Poncelet	rapporteur

Table des matières

1 Introduction **5**

2 Passage à l'échelle et traitement de flux de données **7**
2.1 Introduction . . . 7
2.2 Tour d'horizon des méthodes de clustering . . . 7
2.2.1 Les centres mobiles et leurs variantes . . . 7
2.2.2 Les méthodes relationnelles . . . 8
2.2.3 Méthodes basées sur la densité . . . 9
2.2.4 Méthodes hiérarchiques . . . 10
2.2.5 Clustering de flux de données . . . 11
2.2.6 Conclusion sur l'état de l'art . . . 12
2.3 Travaux de thèse : AntClust et Visual AntClust . . . 12
2.4 L'algorithme Leader Ant . . . 13
2.4.1 Description . . . 13
2.4.2 Discussion . . . 14
2.4.3 Étude du paramétrage . . . 15
2.4.4 Résultats comparatifs . . . 16
2.5 Nouveaux algorithmes incrémentaux flous . . . 19
2.5.1 Les c-médoides flous pondérés . . . 20
2.5.2 Online fuzzy c-medoids . . . 21
2.5.3 History-based online fuzzy c-medoids . . . 22
2.5.4 Expérimentations . . . 23
2.6 Conclusions et perspectives . . . 28

3 Analyse de traces utilisateurs sur Internet **31**
3.1 Principales méthodes de web usage mining . . . 31
3.2 Méthode d'analyse proposée . . . 34
3.3 Exemple de résultats obtenus . . . 37
3.4 Conclusions et perspectives . . . 38

4 Clustering semi-supervisé **41**
4.1 Introduction . . . 41
4.2 Sélection active de connaissances . . . 42
4.2.1 Sélection active de contraintes . . . 42
4.2.2 Sélection active de données étiquetées . . . 44
4.3 Algorithmes de clustering semi-supervisé . . . 46
4.3.1 Leader Ant avec des contraintes . . . 46
4.3.2 Semi-Supervised Graph-based Clustering (SSGC) . . . 47
4.4 Conclusion et perspectives . . . 49

5 Extraction automatique de métadonnées **51**
5.1 Introduction . . . 51
5.2 Méthodes d'extraction de métadonnées . . . 52
5.3 Relations entre champs de métadonnées . . . 52
5.3.1 Classification de métadonnées . . . 52
5.3.2 Génération de règles d'association . . . 53
5.4 Extraction de métadonnées à partir du contenu . . . 54
5.4.1 Extraction du titre et de l'auteur . . . 54
5.4.2 Annotation et ordonnancement de documents pédagogiques . . . 55
5.5 Conclusion et perspectives . . . 55

6 Conclusions et perspectives **57**

7 Annexes **61**
7.1 Table des notations . . . 61
7.2 Évaluation des méthodes de clustering . . . 61
7.2.1 Indice de Rand . . . 61
7.2.2 Matrice de confusion . . . 62
7.2.3 F1-score . . . 62
7.3 Jeux de données de test . . . 63
7.4 Description des projets collaboratifs . . . 67
7.4.1 Projet webCSTI . . . 67
7.4.2 Projet Infom@gic . . . 68
7.4.3 Projet DoXa . . . 68
7.4.4 Projet PURPLE : Routage intelligent dans un réseau pair-à-pair . . . 68
7.4.5 Projet Gamelab : analyse des interactions des joueurs dans un jeu vidéo . . . 69
7.4.6 Projet TOPOS . . . 69

8 Publications **71**

9 Références **75**

Chapitre 1

Introduction

Depuis le début de ma thèse, mes travaux de recherche s'intéressent principalement à la définition de méthodes de classification non supervisée et à leur application à l'analyse des interactions utilisateurs sous la forme de traces d'usage. L'idée sous-jacente consiste à construire des groupes d'utilisateurs exhibant le même comportement, de façon à pouvoir condenser les grands volumes d'information d'usage sous la forme d'un petit nombre de profils représentatifs. Ces profils peuvent ensuite être utilisés pour faire de la mesure d'audience ou mettre en place des méthodes de personnalisation du site (adaptation ou recommandation dynamiques).

Ces travaux m'ont amené au fil du temps à aborder des problèmes dont la portée dépasse le simple cadre de l'analyse des usages et qui peuvent être résumés par les questions suivantes :

- Comment traiter efficacement des données complexes qui ne sont pas nécessairement exprimées sous la forme de vecteurs numériques ?
- Comment traiter les grands volumes de données disponibles (problème du passage à l'échelle) ?
- Comment s'adapter à la production continue de nouvelles données (problème du clustering des flux de données) ?
- Comment fournir à un expert du domaine d'application (ici l'analyse des usages sur un site Internet) des outils pour interpréter les résultats du clustering ?

Pour répondre à ces questions, trois algorithmes de clustering principaux sont présentés dans ce mémoire. L'algorithme *Leader Ant*, bio-inspiré, construit la partition des données en une seule passe et remplace la détermination quadratique des médoides par un mécanisme de comparaisons aléatoires entre données. Les algorithmes flous *Online Fuzzy C Medoids* (OFCMd) et *History-based Online Fuzzy C Medoids* (HOFCMd) permettent quant à eux de traiter des flux de données en les divisant sous la forme de lots de données qui sont analysés séquentiellement. Deux modèles sont proposés pour agréger les informations issues de chaque lot de données et produire la partition finale. L'interprétation des résultats est rendue possible par des méthodes de résumé de cluster sous la forme de représentants typiques, ainsi que par des méthodes de visualisation interactive permettant l'exploration des clusters découverts.

Mes recherches se sont également intéressées à deux autres aspects des interactions utilisateurs qui peuvent être résumés par les questions suivantes :

- Comment intégrer de la connaissance experte dans le processus de clustering (problème du clustering semi-supervisé) et comment optimiser dans ce cadre les interactions de l'expert avec le système de classification ?
- Dans le cadre de la recherche d'information, comment produire des moteurs de recherche plus précis, avec lesquels l'utilisateur a des interactions plus riches ? Par exemple, dans le cadre du e-learning, comment trouver des exercices difficiles portant sur une notion ?

Pour ce faire, nous nous sommes tout d'abord intéressés dans le cadre de la thèse de Viet-Vu Vu [43] au développement de méthodes de clustering semi-supervisé. Celles-ci supposent l'interaction d'un expert avec le système de classification pour lui fournir des connaissances sous la fome de données étiquetées ou de contraintes. Dans ce cadre, nous avons notamment proposé des algorithmes actifs pour la sélection de contraintes ou de données étiquetées qui améliorent sensiblement les performances de l'ensemble des méthodes de clustering semi-supervisé tout en minimisant l'effort d'annotation de l'expert.

Nous avons ensuite proposé dans la thèse de Sahar Changuel [42], des solutions au problème de l'extraction automatique de métadonnées à partir de corpus structurés. Les métadonnées revêtent un rôle crucial dans le cadre de la recherche d'information et plus particulièrement pour améliorer les moteurs de recherche en les rendant plus précis, expressifs et efficaces sur les grandes masses de données numériques actuelles. La spécificité des travaux que nous avons conduits dans ce cadre est double : d'une part nous avons introduit des méthodes d'extraction des métadonnées indépendantes du contenu et d'autre part, nous avons proposé des méthodes basées sur le contenu qui combinent apprentissage statistique et descripteurs contextuels, stylistiques et linguistiques.

Enfin, ce mémoire bien que le plus complet possible ne décrit pas l'ensemble des travaux auxquels j'ai pu participer depuis mon arrivée au Laboratoire d'Informatique de Paris 6. Ainsi, les travaux de thèse de Vincent Labbé [41] portant sur l'apprentissage de préférences utilisateurs et leur utilisation dans un système d'impression professionnel ne sont pas présentés car plus en marge de mes travaux de recherche principaux, de même que certains projets collaboratifs qui ne sont mentionnés que dans les annexes (voir la section 7.4).

Ce mémoire est organisé comme suit : le chapitre 2 décrit les travaux réalisés dans le cadre du problème du passage à l'échelle et du traitement de flux de données avec les algorithmes Leader Ant, Online Fuzzy C Medoids et History-based Online Fuzzy C Medoids. Le chapitre 3 présente nos propositions dans le domaine de l'analyse des traces utilisateurs avec des mesures de similarité adaptées aux sessions web ainsi que des outils de visualisation interactive pour l'interprétation des résultats d'analyse. Ensuite, les chapitres 4 et 5 décrivent respectivement les travaux réalisés lors des thèses de Viet-Vu Vu [43] sur le clustering semi-supervisé et de Sahar Changuel [42] sur l'extraction de métadonnées. Enfin, le chapitre 6 présente les conclusions de ce mémoire et en décrit les perspectives de recherche.

Chapitre 2

Passage à l'échelle et traitement de flux de données

2.1 Introduction

La classification non supervisée - ou *clustering* - cherche à construire une partition d'un jeu de données de telle sorte que les données au sein d'un même groupe exhibent des propriétés ou des caractéristiques communes et qui les distinguent des données contenues dans les autres groupes. À ce titre, les méthodes de clustering ont été largement utilisées dans de nombreux domaines d'applications allant de la biologie (classification de protéines ou de séquences de génomes), à l'analyse de documents (textes, images, vidéos) ou encore dans le cadre de l'analyse des traces d'usage qui fait l'objet des travaux de recherche de ce mémoire.

De très nombreuses méthodes de classification non supervisée ont été publiées dans la littérature et il est par conséquent difficile d'en donner une liste exhaustive, malgré les nombreux articles publiés tentant de structurer ce domaine très riche et en constante évolution depuis plus de 40 ans [105, 119, 120, 65, 201, 198, 121].

Du fait du cadre applicatif visé, mes recherches se sont plus particulièrement intéressées aux méthodes de clustering incrémentales permettant le traitement de grandes masses ou de flux de données avec la contrainte de pouvoir manipuler indifféremment tous les types de données. Ce chapitre se décompose comme suit : la section 2.2 décrit les principales méthodes de clustering et leurs développements autorisant le traitement de grands volumes ou des flux de données. Les sections suivantes présentent ensuite mes contributions avec tout d'abord un bref rappel de mes travaux de thèse (section 2.3), puis la description de l'algorithme incrémental Leader Ant (section 2.4) et enfin des méthodes Online Fuzzy C-Medoids et History-Based Online Fuzzy C-Medoids (section 2.5) avant de conclure.

Pour des raisons de concision du mémoire, un certain nombre d'informations ont été placées en annexes : le tableau des notations utilisées pour la description des algorithmes dans la section 7.1, le détail des mesures d'évaluation des méthodes de clustering dans la section 7.2 et le détail des jeux de données utilisés dans la section 7.3.

2.2 Tour d'horizon des méthodes de clustering

2.2.1 Les centres mobiles et leurs variantes

Les algorithmes de partitionnement divisent l'espace des objets en k clusters selon une fonction d'optimisation particulière. La fonction d'optimisation la plus commune est probablement la minimisation de l'inertie intra-classe (voir éq. 2.1) qui cherche à produire des clusters compacts et

bien séparés. Cette fonction se retrouve dans les méthodes des centres mobiles comme l'algorithme *k-means* (ou *k-moyennes*) [148] et ses nombreuses variantes [104, 202, 186, 50], qui comptent parmi les plus populaires en fouille de données [197].

$$\text{Minimiser. } J(\mathcal{V}) = \sum_{j=1}^{k} \sum_{\forall x_i \in C_j} d^2(x_i, v_j) \tag{2.1}$$

où $\mathcal{V} = \{v_j\}_{j\in[1,k]}$ désigne l'ensemble des k centres découverts, $\mathcal{X} = \{x_i\}_{i\in[1,n]}$ le jeu de données , C_j le cluster j et d^2 la norme euclidienne.

L'algorithme fonctionne en deux étapes principales : (1) une phase d'assignation des points aux centres les plus proches, et (2) une phase de calcul des nouveaux centres comme la moyenne des coordonnées des points qui les composent. Ce schéma est similaire à celui de l'algorithme *Expectation Maximisation* [84] qui cherche à déterminer les distributions de probabilités sous-jacentes à un jeu de données.

Cependant, *k-means* possède plusieurs limitations bien connues parmi lesquelles : une sensibilité aux points aberrants, le traitement de données exclusivement numériques, une propension à découvrir des clusters sphériques - du fait de la distance utilisée - ou encore une grande sensibilité à la partition initiale.

Pour palier ces limitations, de nombreux travaux ont été proposés dans la littérature. Par exemple, [106, 114, 62, 161, 163, 110] introduisent des initialisations alternatives à la méthode aléatoire employée dans *k-means* pour améliorer sa qualité et sa vitesse de convergence.

D'autres approches proposent des modèles plus généraux que les *k-means* en ne considérant pas une appartenance stricte des points aux clusters (on parle alors de *soft clustering* [141]). Les c-moyennes floues (*fuzzy c-means*) introduites par Bezdek [67] produisent une partition floue dans laquelle chaque donnée appartient à tous les groupes avec un degré d'appartenance $\mu \in [0, 1]$. Cette appartenance apparaît dans la fonction objectif que l'algorithme cherche à minimiser (voir éq. 2.2) et est pondérée par l'hyper-paramètre $m > 1$ qui permet d'influer sur le degré des sous-ensembles produits (une valeur de m élevée indiquant alors une décroissance rapide des fonctions d'appartenance associées).

$$\text{Minimiser.} J_m(U, \mathcal{V}) = \sum_{\forall x_i \in \mathcal{X}} \sum_{j=1}^{c} \mu_{i,j}^m \times d^2(x_i, v_j) \tag{2.2}$$

où c désigne le nombre de clusters, $m \in [1, \infty)$ est l'exposant flou et $\mu_{i,j}$ désigne l'appartenance du point x_i au cluster C_j. U est la matrice d'appartenance qui indique pour tous les points du jeu de données $\mathcal{X}$ leur appartenance à chacun des c clusters.

L'introduction du flou permet une meilleure résistance de l'algorithme aux points aberrants - qui voient leur influence minimisée par le mécanisme d'appartenance - et la possibilité de mieux traiter les phénomènes de recouvrement entre clusters. Cependant, ces méthodes conservent certaines limitations des *k-means* comme le traitement de données exclusivement numériques et la propension à découvrir des clusters sphériques. Une vue d'ensemble des méthodes floues passant à l'échelle peut par ailleurs être trouvée dans [138].

2.2.2 Les méthodes relationnelles

Dans [108, 109] les auteurs décrivent des méthodes *relationnelles* dérivées des approches classiques de type *c-means* (stricts, flous, possibilistes). Ces dernières sont basées exclusivement sur l'expression d'une relation entre les données sous la forme d'une matrice de (dis)similarité, et peuvent donc s'adapter à tous les types de données. Pour ce faire, l'étape de calcul des centres

est modifiée pour ne dépendre que de l'affectation des points aux clusters et de la matrice de (dis)similarité.

La méthode *nerf* (*Non Euclidian Relational Fuzzy c means*) propose une extension de ces méthodes dans le cas où la matrice de (dis)similarité n'est pas euclidienne. Le principe de cette extension est également employé dans l'algorithme *Robust Fuzzy C Maximal Density Estimator* [156] qui a été appliqué à l'analyse des traces utilisateurs sur Internet tout comme l'algorithme relationnel d'agglomération compétitive *CARD* (*Competitive Agglomeration for Relational Data*) des mêmes auteurs [155]. Dans [66], les auteurs proposent une extension de la méthode *nerf* [107] pour le traitement de très grands jeux de données. Cette méthode repose sur la construction d'un échantillon vérifiant des contraintes de taille (pour pouvoir être manipulé en mémoire) et de qualité. Cet échantillon est ensuite analysé par l'algorithme *nerf* et les résultats sont propagés à l'ensemble des données non classées. Cette méthode ne peut toutefois pas être appliquée pour l'analyse de flux de données dans lesquels les données divergent de leur distribution initiale au cours du temps (concept de dérive ou *drift*), car dans ce cas l'échantillon construit n'est plus représentatif.

D'autres méthodes relationnelles définissent les centres comme des *médoides*, c'est-à-dire des représentants de chaque cluster appartenant au jeu de données. Ces méthodes sont plus résistantes au bruit que celles qui déterminent le centre comme une moyenne, et permettent une meilleure interprétation des résultats. L'algorithme *PAM* [131] construit la partition en échangeant itérativement un médoide avec un point qui n'en est pas un pour en améliorer la qualité. *CLARA* [132] et *CLARANS* [159] réduisent la complexité quadratique de *PAM* en utilisant des méthodes d'échantillonnage : *CLARA* construit aléatoirement de multiples échantillons, exécute *PAM* dessus et fusionne les résultats alors que *CLARANS* ne considère qu'un échantillon des données (le voisinage du médoide que l'on cherche à échanger) à chaque itération pour remplacer un médoide existant. Dans [57] les auteurs définissent une structure nommée *Slim Tree*, similaire à un *R-Tree* ou à un *M-Tree* mais adaptée aux données relationnelles pour réaliser un échantillonnage intelligent du jeu de données. Cependant, comme déjà indiqué, les méthodes reposant sur un échantillonnage des données ne sont pas adaptées au traitement de flux de données.

D'autres travaux proposent des astuces algorithmiques pour rendre la recherche des médoides sous-quadratique [89]. Enfin, [136, 157] décrivent l'algorithme *LFCMdd* (*Linearized Fuzzy C-Medoids*) qui permet la détermination des médoides de manière approchée en temps linéaire en ne considérant qu'un nombre restreint p de candidats médoides pour chaque cluster. Ces candidats sont déterminés à l'aide de la matrice d'appartenance en ne conservant pour chaque cluster que les p points qui maximisent leur appartenance. Cet algorithme obtient de bons résultats et est appliqué à l'analyse des usages sur Internet mais est limité aux données statiques qui peuvent être stockées en mémoire. Nos approches incrémentales floues présentées dans la section 2.5 reposent pour une part sur une extension de ce modèle.

2.2.3 Méthodes basées sur la densité

De nombreux autres travaux considèrent un cluster comme une zone dense de l'espace des données. L'algorithme le plus connu est sans doute *DBSCAN* [88]. Dans cette approche, un cluster est constitué de tous les points connectés au travers de zones denses de l'espace des données. La densité d'une zone est définie par deux paramètres de l'approche qui sont la taille du voisinage et le nombre de points qui doivent y être contenus. Cette définition permet de détecter le bruit et les points aberrants dans les données et autorise l'apprentissage de clusters non sphériques. Cependant, comme les paramètres de densité sont fixés au début de l'algorithme, *DBSCAN* ne peut pas s'adapter à des clusters de densité variable. De plus, l'utilisation d'une structure de données de type *R-Tree* [61] pour déterminer le voisinage de chaque point limite l'approche aux seules données numériques et ne permet de bonnes performances que pour des données de faible dimension (moins de 20 attributs en pratique).

Les auteurs proposent également une extension incrémentale de *DBSCAN* dans [87]. La méthode permet de s'adapter aux modifications de densité locale induites par l'ajout ou la suppression de données et autorise ainsi la mise à jour des clusters existants sans recalculer toute la partition.

L'algorithme *OPTICS* [53] généralise l'approche *DBSCAN* en proposant un ordonnancement des partitions possibles en fonction des densités observées entre les données. Il est ainsi possible, à l'instar d'un dendogramme que l'on peut lire à un certain niveau de dissimilarité pour obtenir une partition, de trouver les partitions à partir de la taille de voisinage spécifiée. Dans la méthode *DENCLUE* [113], la densité est définie analytiquement comme la somme de fonctions d'influence associées à chaque point. Afin de gagner en efficacité, les densités sont évaluées localement grâce à un quadrillage préliminaire de l'espace. Cette approche se révèle beaucoup plus efficace que *DBSCAN* tout en proposant un modèle qui généralise d'autres méthodes de partitionnement ou hiérarchiques. Similairement, d'autres algorithmes reposent sur la définition d'un quadrillage de l'espace afin d'en améliorer les performances [179, 49].

2.2.4 Méthodes hiérarchiques

À la différence des méthodes de partitionnement présentées précédemment, les méthodes hiérarchiques produisent une séquence de partitions imbriquées appelée *dendogramme*. Ce dernier permet tout de même d'obtenir une partition lorsque l'on en considère une coupe transversale à un certain niveau de dissimilarité. Il existe traditionnellement deux types d'approches hiérarchiques :

- les approches *ascendantes* ou *agglomératives* qui partent d'un ensemble de n classes constituées d'un unique objet (appelées *singletons*) et qui vont être itérativement fusionnées de façon à n'obtenir qu'une seule classe à la fin,
- les approches *descendantes* ou *divisives* qui, inversement, partent d'une classe contenant toutes les données et s'arrêtent lorsque n singletons ont été créés.

L'algorithme de classification ascendante hiérarchique le plus connu est probablement le modèle *SAHN* (*Sequential Agglomerative Hierarchical Non-overlapping*) [183] qui possède plusieurs variantes en fonction de son critère d'arrêt (nombre de classes, distance entre les clusters fusionnés) ou de sa stratégie de regroupement des clusters à chaque itération : saut minimal (*single link*) qui détermine la distance entre deux classes comme la distance minimale observée entre deux objets de chacune des classes, saut maximal (*complete link*), ou distance moyenne.

Les méthodes de classification hiérarchiques nécessitent de nombreuses opérations d'entrée/sortie pour manipuler les matrices de distances entre objets et possèdent généralement une complexité temporelle quadratique avec le nombre de données, ce qui les rend difficilement applicables sur de grands jeux de données. De plus, [97] rapporte que les stratégies d'agrégation moyenne ou maximale des méthodes hiérarchiques classiques se montrent incapables de gérer les partitions avec des clusters de forme non sphériques ou de densité et de taille différentes alors que la stratégie de saut minimum est sensible au bruit et aux points aberrants. Pour s'affranchir de ces problèmes, l'algorithme *CURE* [97] (*Clustering Using REpresentatives*) repose sur l'utilisation de plusieurs points de données proche du centre pour représenter chaque cluster. Une variante nommée *ROCK* [98](*RObust Clustering using linKs*), spécialisée dans les données catégorielles, substitue la similarité initiale entre les données par une notion de *liens* qui correspond au nombre de voisins en commun entre chaque paire de données après partitionnement du graphe issu de la matrice de similarité initiale. Similairement, l'algorithme *CHAMELEON* [128] repose sur un graphe des k plus proches voisins au lieu de la matrice de similarité initiale. Le graphe obtenu est ensuite partitionné en petits clusters en utilisant la bibliothèque hMeTiS [129] et ces derniers sont ensuite fusionnés par un algorithme hiérarchique adapté.

Enfin, l'algorithme *BIRCH* (*Balanced Iterative Reducing Clustering Hierarchies*) [203] a été développé pour permettre le traitement de grands volumes de données dans un espace mémoire restreint. Pour ce faire, *BIRCH* réduit la taille du jeu de données initial en le résumant sous la forme d'une structure hiérarchique nommée *cluster features tree* et dans laquelle les points sont agrégés

successivement en fonction de leur similarité et de l'espace mémoire disponible. Chaque *cluster feature* résume un ensemble de points en conservant uniquement les informations suivantes : le nombre de points représentés, la somme des coordonnées des points et la somme des coordonnées des points au carré. Cette représentation permet de connaître à tout instant les coordonnées du centre du groupe et son rayon. Les centres des clusters ainsi formés sont ensuite classés par un algorithme de classification hiérarchique ascendante classique [160]. Un échantillonnage est également proposé dans la méthode *CURE* [97] pour passer à l'échelle. Ces algorithmes n'autorisent toutefois pas le traitement de flux de données.

2.2.5 Clustering de flux de données

L'analyse de flux (*stream mining*) fait référence à des algorithmes qui sont spécifiquement définis pour traiter des flux. Dans ce cas, la difficulté ne réside pas uniquement dans le traitement de très grands jeux de données (possiblement infinis) mais aussi dans la capacité d'adaptation continuelle et automatique aux évolutions du flux de données. Ces évolutions peuvent prendre la forme de déplacements continus d'une classe dans l'espace de définition des objets qui la composent, (on parle alors de dérive ou *drift*), ou peuvent prendre la forme de changements plus abruptes (*shift*). Ce problème revient à identifier des modèles de données similaires au cours du temps et a été traité dans plusieurs communautés.

Par exemple, dans la communauté des réseaux de neurones flous (*fuzzy neural networks*), les méthodes proposées s'intéressent au problème plus général de l'identification d'un système évolutif non linéaire dans lequel la structure ainsi que les paramètres doivent idéalement être appris et mis à jour à partir du flux de données. Dans ce contexte deux types de méthodes ont été décrites : des systèmes à base de règles floues [51, 52, 130, 126] ou des réseaux de neurones évolutifs [174] (*evolving neural networks*).

Par exemple, dans [51] l'auteur décrit l'algorithme *simpl_eTS+* qui construit un ensemble de règles floues et améliore l'algorithme *simpleTS* [52] en utilisant conjointement l'algorithme *Simpl_e_Clustering* pour apprendre la structure du système (le nombre de règles) et un mécanisme de sélection des règles et des attributs des règles grâce à une mesure d'utilité. L'algorithme *Simpl_e_Clustering* s'adapte aux changements (*shift*) dans l'espace grâce à une mesure de densité calculée à partir de la position relative des points au centre (moyenne) de l'ensemble des données.

L'analyse de flux a également été abordé plus spécifiquement par la communauté clustering. Un état de l'art peut être trouvé dans [93]. Dans [99] les auteurs présentent un algorithme de type k-medoides qui décompose le flux de données sous la forme d'un ensemble de lots de données. Chaque lot est partitionné et résumé sous la forme de médoides pondérés. L'algorithme réitère récursivement le clustering à partir des médoides pondérés jusqu'à obtenir la partition finale. L'algorithme de clustering flou *Online Fuzzy c-Means* ($OFCM$) [117] suit le même principe mais ajoute à chaque lot de données un sous-ensemble des centres découverts jusqu'à présent - l'*historique* - pour garantir une meilleure transition entre chaque lot de données. Les résultats expérimentaux montrent que $OFCM$ est plus performant que la méthode introduite dans [99] mais est limitée au traitement de données numériques.

L'algorithme *Clustream* [48] analyse les flux en deux phases. La phase *online* résume le flux de données sous la forme d'un ensemble de micro-clusters et la phase *offline* produit la partition désirée à partir des micro-clusters. Un micro-cluster est une extension temporelle du formalisme *cluster feature* introduit par [203] pour l'algorithme *BIRCH* (voir section 2.2.4). Dans *Clustream*, les micro-clusters sont sauvegardés dans une structure temporelle pyramidale pour rechercher efficacement les clusters à différents moments ou échelles de temps. Similairement, [167] utilise une structure de type *ClusTree* pour maintenir un résumé du flux qui s'adapte efficacement à sa vitesse.

Plus récemment, les auteurs de *Clustream* [48] ont proposé l'algorithme *HPStream* [47] qui améliore *Clustream* en introduisant un mécanisme d'oubli ainsi qu'une méthode de projection des données pour faciliter la prise en compte des données en très grande dimension. L'algorithme *HWStream* [146] améliore [47] grâce à l'utilisation d'une fenêtre temporelle couplée à une représentation efficace de la distribution des points nommée histogramme exponentiel.

Enfin, d'autres travaux [72, 90] décrivent des méthodes de clustering de flux basées sur la densité pour traiter des clusters de formes quelconques et mieux prendre en compte les points aberrants. Similairement aux approches précédentes, l'algorithme *DenStream* [72] repose sur un résumé du flux par des micro-clusters mais différencie les clusters potentiels des points aberrants potentiels. Le partitionnement final est ensuite réalisé par une méthode de type *DBSCAN* (voir la section 2.2.3). L'algorithme *CDenStream* [175] décrit le premier algorithme de clustering semi-supervisé - ie qui intègre de l'information experte au processus de clustering sous la forme de données étiquetées ou de contraintes, voir le chapitre 4 pour plus de détails - pour le traitement de flux, dérivé de *DenStream*. La méthode repose sur l'algorithme *Constraint-DBSCAN* à la place de *DBSCAN* et propose un mécanisme de traduction des contraintes entre données en contraintes entre micro-clusters. Comme pour les autres algorithmes de clustering semi-supervisés, un expert doit spécifier les contraintes entre les données. De plus, du fait de l'utilisation du formalisme des micro-clusters dérivé des *cluster features*, *DenStream* et *C-DenStream* sont limités au traitement de données numériques.

2.2.6 Conclusion sur l'état de l'art

Cet état de l'art vise principalement à montrer les développements récents et les tendances dans le domaine du clustering en se focalisant plus particulièrement sur les problèmes de passage à l'échelle, de traitement de flux ou de la prise en charge de données relationnelles. Faute de place, certaines méthodes (re)connues comme les approches spectrales [181, 158, 147, 184] ou les méthodes à noyau [71, 178] n'ont pas été décrites plus avant. Il ressort de ce rapide tour d'horizon que plusieurs approches cohabitent pour aborder le problème de passage à l'échelle : soit des méthodes d'échantillonnage (par exemple *CLARA* [132], *CLARANS* [159] ou *eNERF* [66]), mais qui sont inadaptées dans le cas du traitement de flux, soit des approches reposant sur une pré-classification des données. Dans ce dernier cas, une approche en deux phases peut souvent être mise en œuvre : une phase en ligne construit ou enrichit le modèle de connaissance en mettant à jour ou en ajoutant des "pré-clusters" (phase de construction du *cluster feature tree* dans BIRCH [203] ou la phase de création des micro-clusters dans DenStream [72]) et la phase hors-ligne, optionnelle, durant laquelle les clusters finaux sont construit sur la base des "pré-clusters". Contrairement à un échantillonnage, cette méthode considère toutes les données et peut être adaptée pour le traitement de flux de données. Enfin, les données peuvent être intégrées incrémentalement ou par lots comme dans l'approche *Online Fuzzy C-Means* [117]. Les contributions présentées dans ce mémoire prennent la forme de trois algorithmes qui, contrairement à la plupart des méthodes de l'état de l'art, permettent le traitement de tous types de données : d'une part l'algorithme incrémental *Leader Ant* qui traite les données en une seule passe et permet de réaliser une estimation du nombre de clusters recherchés, et d'autre part, deux approches floues basées sur les médoides et autorisant le traitement de flux de données.

2.3 Travaux de thèse : AntClust et Visual AntClust

Les algorithmes *AntClust* et *Visual AntClust* décrits dans [39] sont des approches de classification non supervisée qui reposent sur une modélisation du système de reconnaissance chimique des fourmis [28]. Dans le modèle biologique, chaque fourmi possède une odeur propre appelée *label*, partiellement définie par son génome et son environnement. Chaque fourmi possède également un

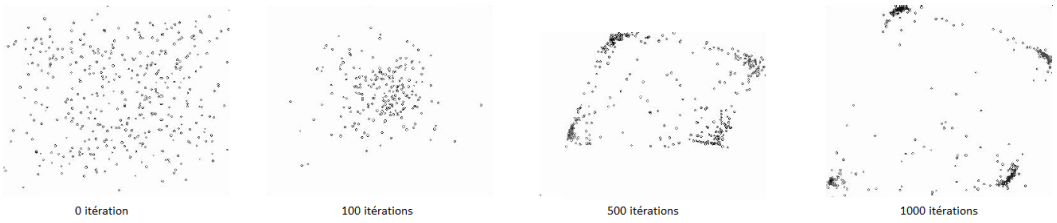

FIGURE 2.1 – Algorithme Visual AntClust : exemple de visualisation de l'évolution des labels en 2D pour le jeu Art_1

modèle neuronal de reconnaissance de ce que doit être l'odeur d'un membre du nid, appelé *template*. Le label et le template permettent à chaque fourmi d'être identifiée et acceptée au sein de son nid (phénomène de *phenotype matching* [115]). Les labels sont échangés continuellement (par toilettage social ou trophallaxie) et évoluent en fonction des rencontres entre fourmis et, ce faisant, permettent l'émergence à l'échelle du nid d'une odeur coloniale (*Gestalt theory*). Dans l'algorithme *AntClust* [5, 26, 27, 33], le génome de chaque fourmi artificielle est associé à un objet du jeu de données. Un seuil d'acceptation (son template) est ensuite appris par chaque fourmi et est défini par les similarités observées entre son génome et celui d'autres fourmis choisies aléatoirement. Chaque fourmi va ensuite réaliser des rencontres aléatoires de façon à déterminer le label (son appartenance à un nid ou cluster) qui correspond le mieux à son génome grâce à un ensemble de règles comportementales individuelles. À la fin, les fourmis ayant des génomes similaires sont réparties dans les mêmes nids, ce qui forme la partition désirée des objets. Dans *Visual AntClust* [24], les labels manipulés ne sont plus discrets comme dans *AntClust*, mais continus et exprimés par un vecteur de réels à deux dimensions. Cette représentation, couplée à des règles de rencontres entre fourmis conditionnées par leur proximité dans le plan des odeurs et des mécanismes de rapprochement et d'éloignements entre labels en fonction des similarités entre les objets représentés par chaque fourmi, autorise le suivi dynamique et visuel de la création de la partition finale comme les méthodes de type nuées dynamiques [34] (voir la figure 2.1).

Ces deux approches ont été appliquées avec succès sur des jeux de données numériques artificiels et réels issus du *UCI Machine Learning Repository* [55].

2.4 L'algorithme Leader Ant

L'algorithme *Leader Ant* a été développé dans le cadre du projet ministériel WebCSTI qui visait à caractériser les usages des internautes sur les sites de plusieurs institutions issues du domaine de la Culture Scientifique, Technique et Industrielle (*CSTI*). L'objectif était alors de disposer d'une méthode simple à mettre en œuvre et capable de traiter rapidement de grands volumes de données d'usage non nécessairement représentées par des vecteurs numériques. La méthode *Leader Ant* est particulièrement adaptée dans ce cadre car elle ne nécessite que peu de paramétrage de la part de l'utilisateur (elle propose automatiquement une estimation du nombre de clusters recherchés) et ne dépend pas de la représentation des données mais d'une mesure de distance (ou (dis)similarité) définie dans l'espace des données.

2.4.1 Description

L'algorithme *Leader Ant* (*LA*) [21, 18] est un algorithme relationnel biomimétique qui s'inspire du système de reconnaissance chimique des fourmis. Cependant, bien que reposant sur le même modèle biologique que les approches *AntClust* et *Visual AntClust* développées durant ma thèse (voir section 2.3), le modèle sous-jacent de *LA* a été simplifié de façon à répondre plus spécifiquement aux

attentes du problème de la classification non supervisée, motivé en cela par le besoin de performance lié au traitement de grands fichiers de sessions web. Ainsi, dans le modèle proposé, le génome de chaque fourmi artificielle est associé à une donnée et l'odeur coloniale est modélisée par un seuil d'acceptation, appelé *template*, qui est utilisé lors de chaque rencontre entre individus, de façon à décider de l'intégration d'une nouvelle fourmi dans un nid existant. Une fourmi artificielle est représentée par trois paramètres :

1. le *génome* est associé à un unique objet du jeu de données ;
2. le *template* T est le même pour toutes les fourmis artificielles et est défini comme la moyenne des distances observées lors de n_r rencontres aléatoires entre individus :

$$T = \frac{\sum_{n_r} D(x_i, x_j)}{n_r} \tag{2.3}$$

où D désigne une mesure de distance (ou dissimilarité) entre les objets x_i et x_j sélectionnés aléatoirement dans $\mathcal{X}$.

▷ le *label* reflète l'appartenance à un nid/cluster de chaque fourmi artificielle. Initialement, les fourmis n'appartiennent à aucun cluster et leur label vaut 0.

LA est un algorithme incrémental qui ne réalise qu'une seule passe sur les données pour construire la partition désirée. Son processus est inspiré de l'algorithme *Leader* qui a été utilisé avec succès dans le cadre de l'analyse de données d'usages [199]. *LA* sélectionne itérativement au hasard une fourmi f qui n'appartient pas déjà à un nid et détermine son label en simulant un nombre fixé de rencontres n_τ avec des fourmis issues de chacun des nids existants $c \in [1, k]$. Durant ces rencontres, la fourmi f estime la distance de son génome avec celui des fourmis issues du nid évalué. À la fin de cette phase de rencontres, la distance $D_f(c)$ entre la fourmi f et un nid c est calculée comme la moyenne des distances observées lors des n_τ rencontres aléatoires (voir éq. 2.4).

$$D_f(c) = \frac{\sum_{j=1}^{n_\tau} D(f, f_j^c)}{n_\tau} \tag{2.4}$$

où f_j^c désigne le génome de la $j^{\text{ème}}$ fourmi sélectionnée aléatoirement dans le cluster c.
Si aucun nid n'existe ou si la distance minimale à l'ensemble des nids est supérieure à la valeur de *template*, la fourmi f construit un nouveau nid. Dans le cas contraire, elle rejoint le nid qui possède l'estimation de distance moyenne la plus faible avec son génome (voir éq. 2.5).

$$Label(f) = argmin_{c \in [1,k]} D_f(c) \tag{2.5}$$

Enfin, lorsque toutes les fourmis sont affectées à un nid, les nids les plus petits dont la taille est inférieure à un seuil t_{min} - exprimé comme un pourcentage du nombre total de données dans $\mathcal{X}$ - peuvent optionnellement être supprimés. Les fourmis qui en sont issues sont alors réaffectées au nid le plus proche selon l'équation (2.5).

2.4.2 Discussion

L'algorithme *Leader Ant* est un algorithme relationnel qui permet de traiter tous types de données et qui ne nécessite pas la détermination préalable du nombre de clusters recherchés, information rarement disponible a priori, et en particulier absente dans le cadre de l'analyse des usages du web. Il enrichit l'approche *Leader* dont il s'inspire, en intégrant d'une part un mécanisme de rencontres aléatoires pour remplacer la détermination exhaustive du médoide de chaque cluster (qui a une complexité quadratique) et, d'autre part, calcule automatiquement une estimation du seuil d'acceptation dans les clusters (le *template*). L'utilisation de plusieurs représentants pour juger de la

distance d'un point à un cluster a déjà été proposée dans d'autres algorithmes comme les approches de classification hiérarchiques de type $single - link$ ou $complete - link$ [183] qui utilisent tous les points du groupe comme représentants, ou l'approche $CURE$ [97] qui ne considère, au contraire, qu'un petit nombre de représentants. Dans ce dernier cas, les représentants sont choisis par une approche de type min-max qui permet une bonne couverture du cluster au prix d'une complexité importante, chaque point du cluster devant calculer sa distance minimale avec les représentants retenus jusqu'alors. Dans LA, le choix aléatoire des représentants permet de couvrir uniformément le cluster tout en conservant une faible complexité. De plus, le mode de sélection des représentants se rapproche d'un échantillonnage aléatoire et, à ce titre, permet d'obtenir une bonne robustesse aux points aberrants. Cependant, la méthode devient également non déterministe et sensible aux points sélectionnés pour évaluer la distance à chaque nid, ce qui peut introduire une certaine variabilité entre deux exécutions successives de l'algorithme sur les mêmes données.

Il est également intéressant de noter que bien que LA utilise plusieurs représentants par cluster, il n'est pas adapté au traitement de clusters non sphériques comme c'est le cas des approches hiérarchiques de type *single-link* par exemple. Cela est dû à la définition du *template* qui est une distance moyenne et non une distance minimale comme dans l'approche *single-link* et qui permet par effet de chaînage de construire des clusters étendus non sphériques. En ce sens, LA est plus similaire d'une approche de type k-means dont la détermination du centre est remplacée par une estimation de distance sur un petit nombre n_τ de comparaisons.
Le mécanisme de rencontres aléatoire permet à l'algorithme LA de s'affranchir de la détermination coûteuse d'un médoide. En conséquence, l'algorithme LA s'exécute en temps linéaire avec le nombre de données n, le nombre de clusters k et dépend également du nombre de rencontres n_τ. La complexité globale de l'algorithme *Leader Ant*, notée C_{LA}, peut donc s'exprimer comme suit :

$$C_{LA} = O(knn_\tau) \approx O(kn) \text{ quand } n_\tau << n \tag{2.6}$$

Enfin, l'algorithme LA est une méthode incrémentale qui peut donc enrichir une partition existante à partir de nouvelles données. Les clusters produits dépendent de l'ordre dans lequel les données sont considérées. Cela pouvant être problématique lors de l'évaluation de la méthode sur des données de tests, LA choisit aléatoirement une nouvelle donnée non traitée à chaque itération. Cela étant, ce problème disparaît lors de traitement de données réelles qui ne sont pas déjà regroupées artificiellement par cluster dans leur fichier, mais dont la production dépend d'un processus complètement indépendant de la méthode de clustering. Enfin, il est intéressant de noter que la méthode Leader Ant ne dispose pas actuellement de mécanisme permettant de résumer l'information des clusters (sous la forme d'un ensemble de représentants) ou de diminuer graduellement l'importance des données les plus anciennes au cours du temps comme les méthodes de traitement de flux plus récentes. Leader Ant, bien que pouvant prendre en charge de très grands jeux de données du fait de sa complexité, est limité au traitement de jeux de données pouvant être stockés en mémoire vive et ne peut pas s'adapter aux flux de données, contrairement à nos approches floues présentées dans la section 2.5.

2.4.3 Étude du paramétrage

L'algorithme *Leader Ant* dépend de trois principaux paramètres : (1) le nombre de rencontres n_r pour le calcul du *template*, (2) le nombre de rencontres n_τ entre une fourmi et les membres de chaque nid pour en évaluer la distance et (3) la taille minimale t_{min} des nids à supprimer à la fin de la procédure. Des expérimentations ont été conduites pour évaluer l'influence de ces paramètres sur les performances. Il en ressort que l'augmentation de la valeur du paramètre n_r permet principalement de diminuer la variance observée sur la valeur du *template*, ce qui peut améliorer les résultats pour certains jeux de données. Similairement, l'augmentation de la valeur du paramètre n_τ permet

d'éviter les erreurs lors de l'estimation de la distance d'une fourmi à un nid. Deux comportements sont observés parmi les jeux de données : soit une invariance à ce paramètre, soit une amélioration des résultats. Ces deux paramètres ayant, d'une part, une influence limitée en terme de performance sur l'ensemble de nos jeux de données, et d'autre part, un impact sur la complexité de la méthode, il est possible de déterminer des valeurs de compromis pour chacun de ces paramètres. Le dernier paramètre t_{min} est optionnel et a une influence beaucoup plus importante sur les performances et également sur le nombre de calculs qui doivent être réalisés : de faibles valeurs de ce paramètre peuvent débruiter la partition en supprimant les clusters les plus petits, alors que de plus grandes valeurs auront tendance à supprimer des clusters statistiquement plus représentatifs et qui auraient mérité d'apparaître dans la partition finale.

2.4.4 Résultats comparatifs

Nous présentons dans cette section des résultats comparatifs entre l'algorithme *Leader Ant* (LA) et les algorithmes k-means (KM) [148], fuzzy c-means (FCM) [67] et linearized fuzzy c-medoids (LFCMdd) [136]. Du fait de la nature non déterministe des algorithmes étudiés, les résultats suivants indiquent pour chaque mesure évaluée une valeur moyenne ainsi que son écart-type. Les algorithmes KM, FCM et LFCMdd ont été initialisés à l'aide de centres générés aléatoirement et avec le nombre attendu de clusters dans la partition, ce qui leur donne un avantage par rapport à *Leader Ant* qui ne dispose pas de cette information.

Erreurs de classification

Nous nous intéressons ici aux erreurs de Rand (figure 2.2-Haut) et de confusion (figure 2.2-Milieu) ainsi qu'au nombre de clusters (figure 2.2-Bas) produits par LA. En effet, il est intéressant de considérer le nombre de clusters conjointement aux erreurs de Rand et de confusion car leurs valeurs en dépendent : la mesure de Rand est sensible aux écarts dans le nombre de clusters entre deux partitions et l'erreur de confusion, si elle ne sanctionne pas la distribution d'une partition vers un plus grand nombre de clusters homogènes, pénalise les résultats de partitions qui possèdent moins de clusters que ce qui est spécifié. Afin de faire le lien avec les travaux de thèse [39] décrits dans la section 2.3, certains résultats concernant l'algorithme AntClust (AC) sont indiqués lorsque l'information est disponible.

La figure 2.2-Haut montre que l'algorithme LA obtient des résultats comparables en terme d'erreur de Rand aux autres approches de clustering pour un certain nombre de jeux de données comme Art_3, Art_4, $Iris$, $Pima$, $Soybean$, $Wine$ et $Waveform$. Les résultats de LA semblent significativement plus mauvais pour 3 jeux de données : Art_2, Art_5 et $Vehicule$. Pour Art_2, LA surestime le nombre de clusters (voir la figure 2.2-Bas) et est donc pénalisé au niveau de l'erreur de Rand. Cela est vérifié par l'étude de l'erreur de confusion associée (voir la figure 2.2-Milieu) qui est au niveau de ce que les autres méthodes proposent. Cela indique que, même si LA a surdivisé le jeu de données, les clusters construits restent homogènes par rapport aux classes initiales. Pour les jeux de données Art_5 et $Vehicule$, LA sous-estime le nombre de clusters et est pénalisé par les deux erreurs. Dans le cas du jeu de données Art_5, des analyses complémentaires, reposant sur la mesure de qualité décrite dans l'équation 7.5, montrent que les clusters produits par LA sont autant séparés que dans les autres méthodes mais moins compacts, ce qui explique les résultats observés. Dans le cas du jeu $Vehicule$, la qualité de la partition, mauvaise, est équivalente pour tous les algorithmes. Similairement, il est intéressant de constater que l'ensemble des méthodes, à l'exception de AC, échoue dans la classification du jeu de données Art_4. Cela s'explique par la propension naturelle des algorithmes visés à découvrir des clusters hyper sphériques qui ne permettent pas de détecter des clusters allongés tels que ceux présents dans Art_4. AC obtient de meilleurs résultats car il inclut une phase de normalisation des jeux de données par attribut qui a pour effet de transformer les clusters allongés en carrés plus à même d'être appris par nos méthodes. Enfin, LA obtient de bons taux

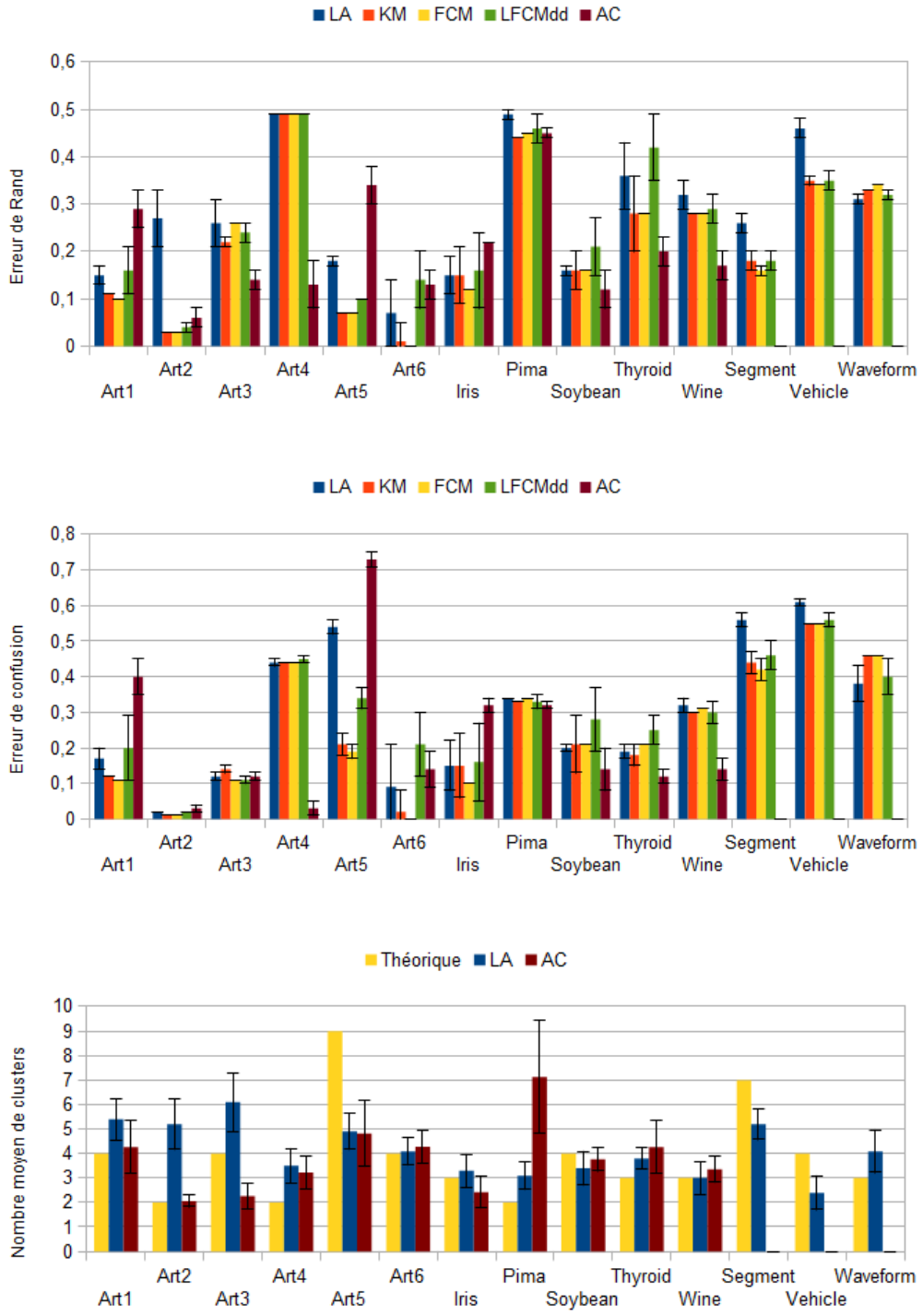

FIGURE 2.2 – Résultats comparatifs. Haut : erreur moyenne de Rand et écart-type ; Milieu : erreur moyenne de confusion et écart-type ; Bas : nombre moyen de clusters découverts par LA. L'ensemble des résultats est calculé pour chaque jeu de données sur 50 tests.

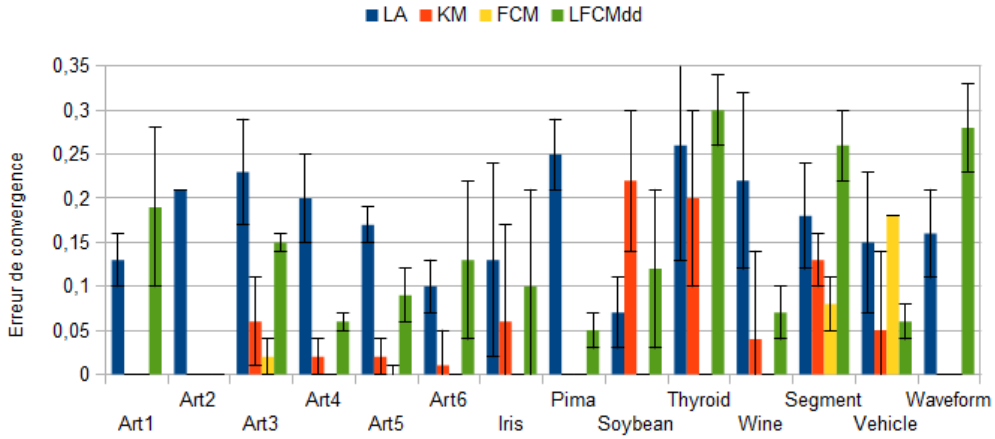

FIGURE 2.3 – Convergence moyenne des algorithmes évaluée sur 50 comparaisons entre les partitions produites pour chaque jeu de données

d'erreur pour les données Art_1, Art_6, $Iris$ et $Soybean$. Si l'on restreint ce comparatif aux approches relationnelles comme LA, LFCMdd ou AC, on observe que les résultats sont comparables, chaque algorithme obtenant de meilleurs résultats que les autres sur certains jeux de données.

Convergence

De façon à évaluer à quel point le choix aléatoire des représentants lors de chaque rencontre avec un nid peut avoir une incidence sur la stabilité de LA, des tests comparatifs ont été conduits avec les algorithmes KM, FCM et LFCMdd (qui sont eux-mêmes non déterministes du fait de leur initialisation). Ainsi pour chaque algorithme et chaque jeu de données, une erreur de convergence est calculée comme l'appariement moyen évalué sur 50 partitions à l'aide de l'erreur de Rand (voir la section 7.2.1). La figure 2.3 montre que les algorithmes KM et FCM obtiennent les erreurs les plus faibles et possèdent donc la meilleure convergence sur l'ensemble des jeux de données. Les deux approches relationnelles basées sur la recherche ou l'estimation de médoides obtiennent des résultats qui sont comparables bien que plus variables sur nos données de tests, comme en attestent les erreurs observées. Cela s'explique par le fait que la recherche de médoides est contrainte à des points du jeu de données alors que la définition d'une moyenne peut évoluer plus graduellement vers un optimum local.

Temps de calcul

La figure 2.4 rapporte les temps de calculs moyens évalués sur 50 tests pour chaque jeu de données et chaque algorithme. L'algorithme KM obtient les temps de calcul les plus courts dans la plupart des cas à l'exception des jeux de données Art_5, $Segment$ et $Vehicule$ pour lesquels LA est plus rapide. Cela peut être expliqué par le fait que KM possède une complexité en $O(nkN)$ où N est le nombre d'itérations avant convergence et que LA possède une complexité en $O(knn_\tau)$, où n_τ est le nombre de rencontres avec chaque nid. Dans le cas de ces 3 jeux de données, une étude plus approfondie du nombre d'itérations N réalisées par KM montre que sa valeur est supérieure à $n_\tau = 10$. Les autres approches FCM et LFCMdd sont globalement plus lentes que LA avec des temps de calcul jusqu'à 32 fois plus long pour le jeu de données Art_5 avec FCM par exemple. On peut également remarquer le temps de calcul moyen de FCM pour les données $Segment$ qui

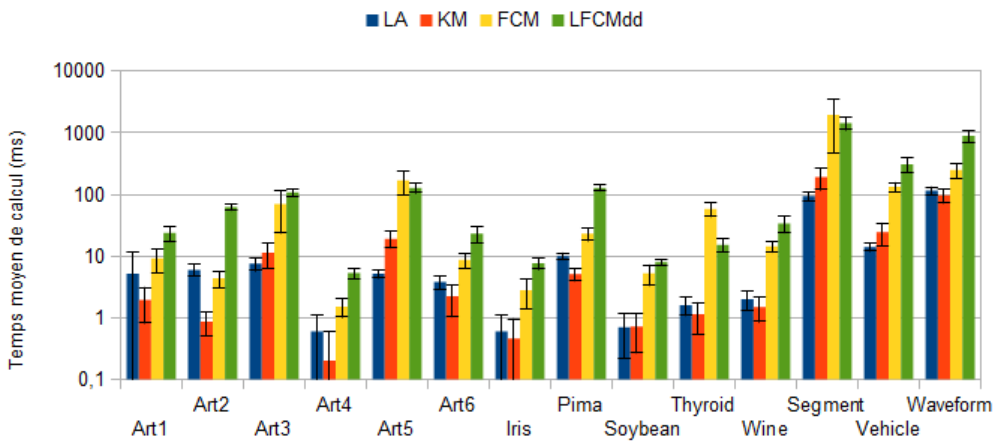

FIGURE 2.4 – Temps moyen de calcul (en ms) pour chaque jeu de données et chaque algorithme

s'explique par le nombre très important d'itérations effectuées par l'algorithme sur ce jeu de données avant convergence.

Pour conclure : les expérimentations conduites sur l'algorithme *Leader Ant* montrent qu'il combine des performances (erreur, convergence) comparables aux autres méthodes basées sur des médoides, tout en conservant un temps de calcul proche de celui des k-means. De plus, *Leader Ant* autorise le traitement de tous types de données, détermine automatiquement une estimation du nombre de clusters et nécessite peu de paramétrage, ce qui en fait un candidat intéressant pour le problème de l'analyse des traces d'usage.

2.5 Nouveaux algorithmes incrémentaux flous

Cette section décrit deux nouveaux algorithmes de clustering flous nommés *Online Fuzzy C-Medoids* ($OFCMd$) et *History-based Online Fuzzy C-Medoids* ($HOFCMd$) dont le but est de traiter soit de très grands jeux de données, soit des flux de données. La spécificité de notre approche réside dans son utilisation de médoides - des points représentatifs des clusters - en lieu et place des moyennes généralement employées dans ce type d'algorithme. L'intérêt d'une telle démarche est de permettre le traitement de tous types de données et de faciliter l'interprétation des clusters, tout en conservant les avantages des méthodes floues dans le cas de recouvrements entre clusters ou la présence de points aberrants.

Similairement à [99, 117, 93], nos algorithmes décomposent les très grands jeux de données (ou les flux) sous la forme d'une séquence de lots de données construits de sorte à être manipulables en mémoire. Chaque lot de données est ensuite résumé à l'aide de notre algorithme des c-médoides flous pondérés (ou *Weighted Fuzzy C-Medoids*, $WFCMd$) qui produit un ensemble de médoides pondérés. L'ordre des lots de données, bien qu'ayant une incidence sur la partition finale produite, est supposé être indépendant de nos algorithmes car imposé par le processus externe qui génère les données.

Comme l'illustre la figure 2.5, deux modèles sont introduits pour agréger les médoides issus des appels successifs à l'algorithme $WFCMd$ sur les lots de données : d'une part l'algorithme *Online Fuzzy C-Medoids* ($OFCMd$) présenté dans la section 2.5.2 et d'autre part l'algorithme *History-based Online Fuzzy C-Medoids* ($HOFCMd$) décrit dans la section 2.5.3.

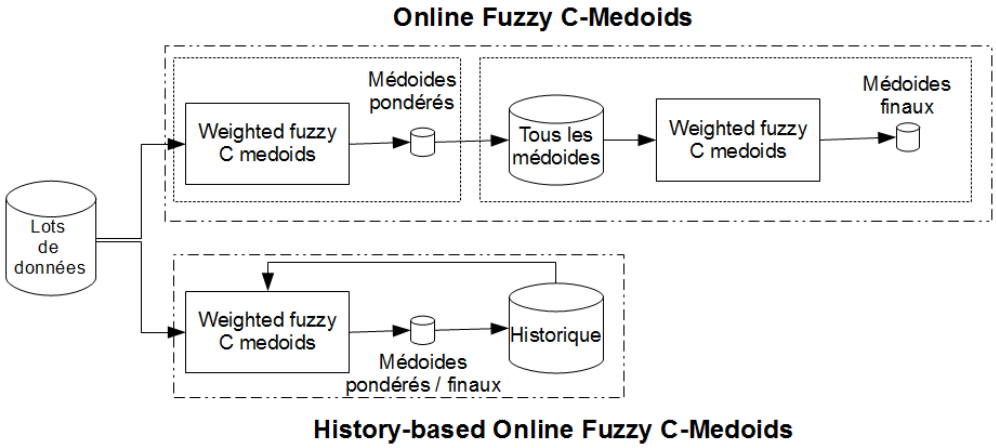

FIGURE 2.5 – Illustration du traitement des lots de données par l'algorithme des c-médoides flous pondérés ($WFCMd$) dans les algorithmes *Online Fuzzy C-Medoids* et *History-based Online Fuzzy C-Medoids*.

2.5.1 Les c-médoides flous pondérés

L'algorithme des c-médoides flous pondérés (ou *Weighted Fuzzy C Medoids*, WFCMd) analyse des jeux de données dans lesquels chaque objet x_i est associé à un poids fixe $\omega_i \in \mathbb{R}^+$ et retourne un ensemble de médoides pondérés. Similairement à [117], le poids ω_i associé à chaque objet x_i peut être interprété comme la présence de plusieurs occurrences du même objet dans le jeu de données. La fonction objectif (J_m) minimisée par le WFCMd est donc définie comme suit :

$$J_m(U, \mathcal{V}) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m w_k r(x_k, v_i) \tag{2.7}$$

où :
- $n = |\mathcal{X}|$ est la taille du jeu de données $\mathcal{X}$ et c est le nombre de clusters,
- U est la matrice d'appartenance dont les valeurs $u_{ik} \in [0, 1]$ indique l'appartenance de l'objet $k \in [1, n]$ au cluster $i \in [1, c]$,
- $\mathcal{V} \subset \mathcal{X}$ est l'ensemble des médoides dans lequel chaque élément correspond au centre d'un cluster,
- m est l'exposant flou ($m \geq 2$), qui permet d'influer sur le degré des sous-ensembles produits (une valeur de m élevée indiquant alors une décroissance rapide des fonctions d'appartenance associées) ;
- r représente la relation définie entre chaque paire d'objets de X. r peut être une norme euclidienne dans le cas de données spatiales $r(x, v) = \|x - v\|^2$, ou, dans le cadre de l'analyse des usages, une mesure de comparaison de séquences d'interactions utilisateurs (distance d'édition par exemple).

Les valeurs de la matrice d'appartenance sont calculées comme suit :

$$u_{ik} = \left[\sum_{l=1}^{c} \left(\frac{r(x_k, v_i)}{r(x_k, v_l)} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{2.8}$$

Cependant, comme la méthode repose sur des médoides et non des moyennes, il est impossible de dériver l'équation 2.7 pour obtenir l'expression des centres de clusters analytiquement. Dans notre modèle, pour chaque cluster c le médoide est défini comme le point qui minimise sa distance

à tous les points du jeu de données X, en fonction de leur appartenance au cluster c et de leur poids. Comme cette détermination exacte a un coût quadratique en $O(n^2)$ avec le nombre n d'objets dans X, nous utilisons l'algorithme de détermination linéaire introduit dans [136]. Cette méthode ne considère, pour chaque cluster c, que le sous ensemble X_c formé des p points ($p << n$) qui maximisent leur appartenance au cluster c comme candidats médoides. Ainsi, le médoide v_c du cluster c est défini comme suit :

$$v_c = argmin_{x_i \in X_c} \sum_{j=1}^{n} u_{cj}^m w_j r(x_j, x_i) \tag{2.9}$$

Grâce à ce mécanisme, la complexité temporelle de l'algorithme $WFCMd$ devient sous-quadratique en $O(cnp)$.

Initialisation des médoides : dans notre approche, les médoides sont initialisés aléatoirement pour conserver une couverture uniforme des données et une complexité modérée. D'autres approches ont été proposées dans la littérature comme les approches de type *min-max* pour favoriser la couverture de l'espace en garantissant une bonne séparation des médoides initiaux [136]. Cependant, cette méthode peut avoir une complexité quadratique (en fonction de la méthode de détermination du premier médoide) et une propension à choisir des points aberrants comme centres initiaux dans le cas de données bruitées.

Critères d'arrêt : l'algorithme $WFCMd$ combine deux critères classiques : un nombre maximal d'itérations ainsi qu'un critère de convergence des médoides des clusters. D'autres méthodes existent et notamment [100] proposent d'utiliser un critère de convergence de la valeur de la fonction objectif. Ce choix n'est pas efficace dans le cas d'algorithmes basés sur des médoides qui peuvent alterner entre plusieurs optima locaux.

De plus, afin de garantir une convergence propre, l'algorithme $WFCMd$ intègre un mécanisme qui empêche que dans de rares cas, deux clusters (ou plus) partagent le même médoide. Le mécanisme de contrôle agit à deux niveaux : un premier mécanisme force les p candidats à être différents lors de la mise à jour du médoide d'un cluster, et le second mécanisme s'assure que le médoide choisi pour un cluster n'a pas déjà été retenu pour un autre. Ces mécanismes pénalisent légèrement la complexité de notre méthode mais assurent, à la différence des autres algorithmes basés sur des médoides, une convergence non dégénérée, qui serait préjudiciable dans notre modèle incrémental où les médoides sont transmis d'un lot de données au suivant.

Discussion : l'algorithme des c-médoides flous pondérés $WFCMd$ produit un ensemble de médoides pondérés. Le poids associé à chaque médoide est défini comme la somme des appartenances des données de $\mathcal{X}$ au cluster associé. L'algorithme $WFCMd$ peut être vu comme une généralisation de l'algorithme $LFCMdd$ [136] : les deux algorithmes sont comparables lorsque les poids des données $\omega_i, i \in [1, n]$ sont fixés à 1. Il y ajoute cependant deux améliorations principales : d'une part la capacité de traiter des données pondérées, et d'autre part l'assurance que tous les médoides sont différents, ce qui est crucial pour ne pas propager d'erreur dans le cas de traitement incrémental de données.

2.5.2 Online fuzzy c-medoids

L'algorithme *Online Fuzzy C-Medoids* ($OFCMd$) analyse le flux de données en deux étapes principales. Tout d'abord, l'algorithme traite itérativement les lots de données à l'aide de l'algorithme $WFCMd$ (voir la section 2.5.1). Afin de mieux capturer l'évolution du flux de données, les médoides découverts pour un lot de données sont utilisés comme médoides initiaux pour le

lot suivant. Similairement à [117], à chaque fois qu'un lot est traité, la mémoire vive est libérée en résumant la partition de sortie en c médoides pondérés. Ensuite, dans une seconde étape, les médoides issus du traitement de l'ensemble des lots sont eux-mêmes partitionnés avec l'algorithme $WFCMd$ pour produire les médoides finaux.

L'algorithme $OFCMd$ intègre un mécanisme d'oubli pour mieux s'adapter à l'évolution des flux de données au cours du temps. Ainsi, une fonction décroissante en fonction du temps est appliquée à l'ensemble des poids des médoides dès qu'un nouveau lot est traité. Le poids ω d'un médoide au temps t_2 peut être exprimé ainsi, en considérant qu'il a été mis à jour précédemment au temps $t_1, t_1 < t_2$:

$$\omega_{t_2} = 2^{-\lambda(t_2 - t_1)} * \omega_{t_1} \tag{2.10}$$

où $\lambda \geq 0$ est un facteur d'oubli qui spécifie la vitesse à laquelle le poids des précédents médoides décroît.

Grâce à ce mécanisme, il devient possible de donner plus de poids aux derniers médoides découverts et ainsi s'adapter aux évolutions du flux (*drift*). Il est intéressant de noter que dans le cas particulier du traitement d'un très grand jeux de données dans lequel les distributions sont stables au cours du temps (puisque la décomposition en lots est purement artificielle), le paramètre λ influence peu les résultats car les médoides issus des derniers lots de données sont supposés être proches de ceux issus des premiers. Enfin, quand $\lambda = 0$, on retrouve l'algorithme $OFCMd$ tel qu'il a été initialement publié dans [11].

Paramètrage du nombre de clusters : comme tout algorithme de la famille des c-means, $OFCMd$ doit être initialisé avec un nombre de clusters c. Dans $OFCMd$, ce nombre de clusters est d'abord utilisé pour traiter chacun des lots de données dans une première étape, puis pour produire la partition finale dans une seconde étape. Dans la première étape, le nombre de clusters doit plus être vu comme un ratio de compression du flux de données que comme un nombre de clusters au sens propre. Dans la seconde étape, le paramètre c détermine le nombre de clusters de la partition finale et le problème de choisir une valeur optimale a été discuté dans de nombreux papiers [75]. Pour des raisons de simplicité et d'efficacité, dans $OFCMd$, la valeur du paramètre c est la même dans les deux étapes.

Complexité : la complexité de $OFCMd$ dépend du nombre de clusters c, de la taille n du jeu de données et du nombre de médoides n_c produit à chaque appel de l'algorithme $WFCMd$ sur les lots de données :

$$C_{OFCMd} = O(p(c \cdot n + c \cdot n_c)) \tag{2.11}$$

2.5.3 History-based online fuzzy c-medoids

Dans le second algorithme nommé *History based Online Fuzzy C-Medoids* ($HOFCMd$), un sous-ensemble des médoides découvert sur les précédents lots, appelé historique, est ajouté à chaque nouveau lot avant son analyse par l'algorithme $WFCMd$. Comme dans l'approche [117], l'objectif est de mieux capturer les évolutions du flux d'entrée. De plus, comme l'information cumulée issue des analyses précédentes est transmise par le biais de l'historique, la partition finale est obtenue à la fin du traitement de chaque lot par la méthode $WFCMd$. Ainsi, dans la méthode $HOFCMd$, contrairement à la méthode $OFCMd$, il n'est pas nécessaire de réaliser une étape de clustering final.

En revanche, similairement à $OFCMd$, $HOFMCd$ intègre un mécanisme d'oubli qui permet de faire décroître le poids des médoides découverts en fonction de leur âge à chaque fois qu'un nouveau lot est traité (voir l'équation 2.10).

Paramètrage du nombre de clusters : tout comme $OFCMd$, $HOFCMd$ doit être initialisé avec un nombre de clusters c qui est ensuite utilisé dans chaque appel à l'algorithme $WFCMd$. Cependant, dans le cas de $HOFCMd$, le traitement de chaque lot peut éventuellement fournir une partition de sortie exploitable. Le paramètre c doit donc de préférence être considéré comme un nombre de clusters même s'il joue aussi le rôle de ratio de compression des données du flux.

Complexité : la complexité de l'algorithme $HOFCMd$ dépend de la taille n des données, du nombre de clusters c, du nombre h de médoides présents dans l'historique qui sont ajoutés à chaque lot (à l'exception du premier lot) et le nombre nb_{iter} de lots qui ont été traités :

$$C_{HOFCMd} = O(cp \cdot (n + h \cdot (nb_{iter} - 1))) \tag{2.12}$$

2.5.4 Expérimentations

Deux expérimentations ont été conduites pour valider notre travail :

1. d'une part la comparaison avec des approches floues ($OFCM$ [117] et $LFCMdd$ [136]) et d'autre par la comparaison avec des méthodes de traitement de flux (*Clustream* [48] et *Clustree* [167] issus de la plateforme de data mining *MOA* (*Massive Online Analysis*) [134, 135]) sur des jeux de données volumineux,
2. une évaluation plus prospective sur les flux artificiels de données $Stream_1$ et $Stream_2$ (voir plus après ainsi que dans la section 7.3).

Jeux de données : dans la première expérimentation, nous utilisons d'une part des données artificielles générées selon des lois normales avec différentes difficultés (attributs inutiles, recouvrement de clusters, déséquilibre de classes) nommées $LArt_{1,2,5,6}$ et d'autre part des données réelles issues du *Machine Learning Repository* [55] nommées $letter - recognition$, une version simplifiée de $statlog - shuttle$ et l'ensemble d'apprentissage du jeu $kddcup99$ limité à ses attributs continus. Dans la seconde expérimentation, nous utilisons deux flux de données artificielles nommés $Stream_1$ et $Stream_2$ qui contiennent respectivement 4000 et 3600 données. $Stream_1$ repose sur les mêmes distributions que le jeu de données $LArt_1$ dans lequel les données sont décrites par deux attributs dont les valeurs suivent des lois normales avec un léger recouvrement entre les 4 clusters. Comme les distributions n'évoluent pas au cours du temps, ce flux sert de base de comparaison pour notre méthode. À l'inverse, les distributions du flux $Stream_2$ évoluent : initialemement similaires à celles du flux $Stream_1$, elles changent au cours du temps de telle sorte que les nouvelles positions de certains clusters recouvrent les anciennes positions d'autres clusters et que d'autres dérivent dans l'espace. Les détails de ces jeux de données peuvent être trouvés en annexes de ce mémoire dans la section 7.3.

Évaluation des méthodes de clustering : deux mesures principales sont utilisées pour comparer les performances de nos algorithmes. Similairement à [72], nous utilisons une mesure de pureté des clusters générés (voir l'éq. 7.3 de la section 7.2). Nous utilisons également la mesure traditionnelle du F1-score telle que présentée dans [196] et utilisée dans l'outil *MOA* [134] (voir l'éq. 7.4 de la section 7.2). L'indice de Rand [171] n'a pas été retenu du fait de sa complexité calculatoire rédhibitoire sans optimisation sur ces jeux de données (comparaison des étiquettes de tous les couples de données).

Enfin, des tests ont été réalisés pour s'assurer de la validité statistique des comparaisons observées avec une confiance de 99% [1].

Analyse de grands jeux de données

On s'intéresse ici à l'évaluation des algorithmes $OFCMd$ et $HOFCMd$ sur des jeux de données artificielles et réelles de grande taille en les comparant aux méthodes floues ou de traitement de flux de l'état de l'art.

Paramètrage des algorithmes : pour les méthodes $OFCMd$ et $HOFCMd$: le nombre de clusters est fixé à la valeur attendue, la taille des lots est fixée à 500 données, l'historique pour les méthodes $HOFCMd$ et $OFCM$ est fixée à 1 (seuls les médoides découverts sur le lot précédent sont transmis au nouveau lot) et le paramètre d'oubli $\lambda = 0$ car les données ne sont pas des flux. La sensibilité des algorithmes $OFCMd$ et $HOFCMd$ à leurs paramètres est discutée plus en détail dans [11] et [1]. Il en ressort que le nombre de clusters c doit être initialisé à la plus grande valeur possible raisonnable en fonction de l'application comme préconisé dans [117]. Le nombre de candidats p doit avoir au moins la valeur de c de façon à garantir que chaque cluster possède un médoide qui lui est propre. L'augmentation de la valeur de p augmentant les temps de calcul, la valuer choisie doit permettre un compromis entre qualité et rapidité. La taille des sous-lots de données ne semble pas influencer les algorithmes bien que, multiplier les sous-lots augmente le nombre de médoides représentants le jeu de données. Enfin, le facteur d'oubli et l'historique sont liés l'un pouvant contrebalancer l'influence de l'autre [1].

Les algorithmes *Clustream* et *Clustree* issus de la plateforme de data mining MOA sont utilisés avec leurs paramètres par défaut avec un facteur d'oubli fixé à 0 et une seule évaluation des performances quand toutes les données sont traitées.

Comparaison avec les approches floues : la figure 2.6 présente les résultats comparatifs en terme de pureté et de temps de calcul avec les algorithmes flous $OFCM$ et $LFCMdd$. D'autres tests, rapportés dans [1], montrent que nos méthodes obtiennent des résultats comparables en terme de F1-score. Les méthodes n'étant pas déterministes (du fait de l'étape d'initialisation des centres) les résultats présentés sont moyennés sur 10 tests et sont présentés avec leur écart-type.

D'après les résultats de la figure 2.6, les algorithmes se comportent de manière comparable, même si l'algorithme $OFCM$ se montre généralement le meilleur, suivi par $HOFCMd$, puis $OFCMd$ et enfin $LFCMdd$. Cela est dû au fait que, étant basé sur des moyennes, $OFCM$ peut raffiner graduellement ces centres jusqu'à ce qu'ils s'adaptent au mieux à la distribution des données, alors que les approches basées sur des médoides sont limitées à certains emplacements de l'espace pour définir les centres. Les différences entre $OFCM$ et $HOFCMd$ ne sont généralement pas significatives [1] exception faite du jeu $LArt_5$ ($OFCM$ est meilleur), du jeu *kddcup* ($HOFCMd$ est meilleur) et du jeu *letter* ($HOFCMd$ partage les meilleures performances avec $LFCMdd$).

Le second résultat est que les méthodes incrémentales basées sur les médoides obtiennent de meilleurs résultats que l'approche $LFCMdd$. En effet, $HOFCMd$ est toujours significativement meilleur que (ou à minima comparable à) $LFCMdd$. $OFCMd$ obtient également des résultats équivalents à $LFCMdd$ (différence non significative pour les jeux $LArt_2$, $LArt_6$ et *kddcup*). Ces résultats montrent expérimentalement que la séparation d'un grand jeu de données, en sous-ensembles traités séparément n'est pas préjudiciable aux performances et valide donc notre approche.

Enfin, les méthodes $OFCM$ et $HOFCMd$, basées sur le mécanisme d'historique, obtiennent les meilleures performances. Le transfert de médoides d'un lot au suivant semble donc bénéfique, car il permet de mettre à jour et de transmettre un résumé des données traitées jusqu'à présent. L'algorithme $OFCMd$ ne transmet de la connaissance que par le biais de l'initialisation des médoides, et ce mécanisme, bien qu'important, ne semble pas à même de représenter l'ensemble des données déjà traitées. Enfin, comme le montre les résultats de l'approche $LFCMdd$, il semble plus judicieux de ne conserver qu'un résumé précis des données sous forme d'un historique, que l'ensemble des données en mémoire, qui peuvent se révéler plus complexes et bruitées.

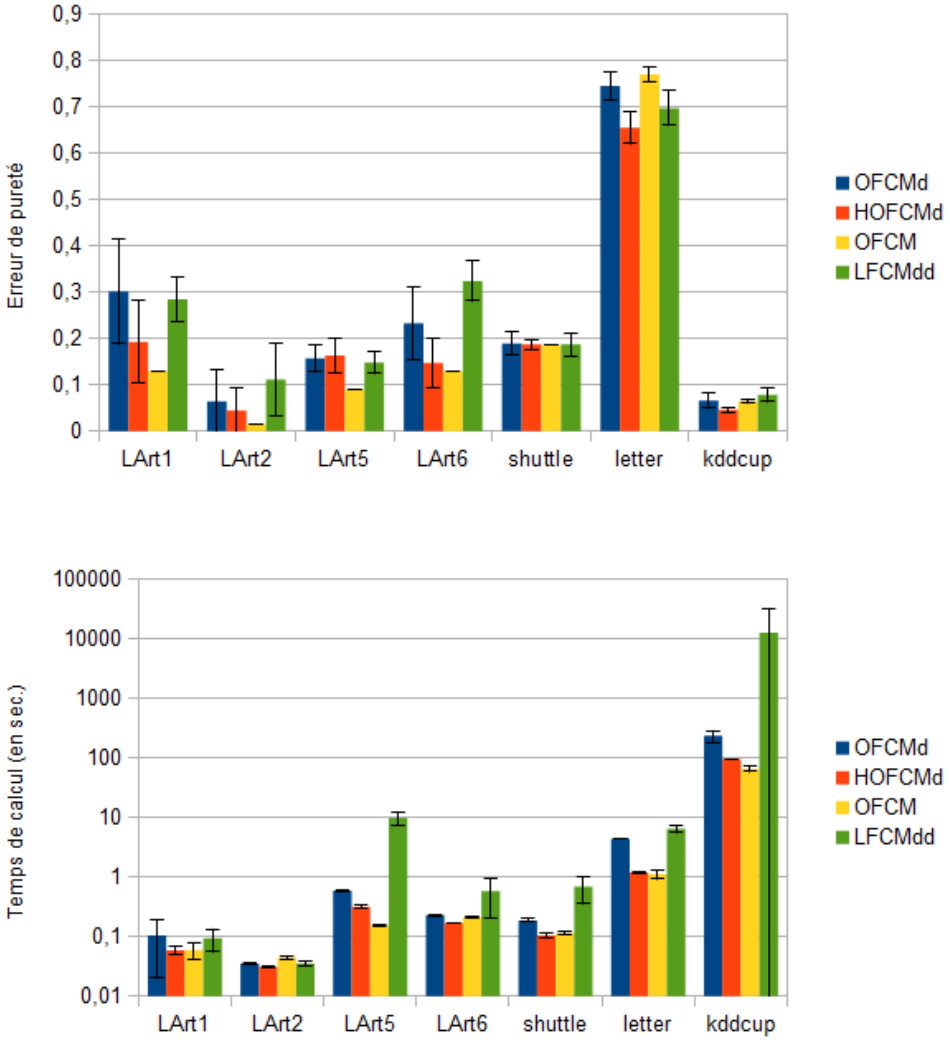

FIGURE 2.6 – *Haut* : comparaison des erreurs de pureté et *Bas* : comparaison des temps de calcul en secondes pour les algorithmes $OFCMd$, $HOFCMd$, $OFCM$ et $LFCMdd$ sur les grands jeux de données artificielles et réelles.

Similairement, des tests comparatifs conduits avec la mesure du F1-score [1] montrent également que nos algorithmes $OFCMd$ et $HOFCMd$ se comportent bien : ils obtiennent le meilleur score pour le jeu $LArt_5$ et l'un des deux partage toujours le meilleur score avec $OFCM$ ou $LFCMdd$ sur les autres jeux de données.

Les temps de calculs rapportés dans la figure 2.6-Bas montrent en premier lieu que les algorithmes incrémentaux ($OFCM$, $OFCMd$ et $HOFCMd$) sont généralement plus rapides que l'algorithme $LFCMdd$. Cela peut s'expliquer par deux raisons principales. D'une part, la convergence d'algorithmes de type k-means semble d'autant plus rapide que le jeu de données est petit (du fait du plus petit nombre d'itérations avant convergence). La décomposition du grand jeu de données en lots semble donc bénéfique pour les approches incrémentales. D'autre part, le langage *Java* utilisé pour l'implémentation voit ses performances décroître d'autant plus que la mémoire vive est sollicitée. Ainsi, là encore, il semble plus intéressant de travailler sur de multiples petits jeux de données qu'un seul volumineux.

En second lieu, les résultats de la table 2.6-Bas montrent que les approches *OFCMd* et *HOFCMd* sont compétitives avec l'algorithme $OFCM$ bien que basées sur des médoides. L'algorithme $OFCMd$ est généralement plus lent que l'algorithme $HOFCMd$ du fait de son étape finale de clustering.

Enfin, le jeu *kddcup* avec environ 490000 données donne une idée de la performance des algorithmes sur de très grands jeux de données : dans ce cas $OFCM$ est clairement plus rapide, suivi par nos deux algorithmes $OFCMd$ et $HOFCMd$. Il est important de souligner que les performances des méthodes basées sur les médoides pourraient être améliorées en supprimant le mécanisme qui évite l'apparition de partitions dégénérées et qui a une complexité en $O(c^2)$ avec le nombre de clusters c.

Comparaison avec les méthodes de traitement de flux : la figure 2.7 montre que nos approches basées sur les médoides obtiennent généralement de meilleurs résultats en terme de F1-score que les algorithmes *Clustree* et *Clustream*. La différence de performances est encore plus marquée pour les jeux de données réelles issus du répertoire *UCI Machine Learning Repository*. Cela peut s'expliquer par le fait que les valeurs par défaut de la plateforme MOA sont suffisantes pour des jeux artificiels simples, mais ne permettent pas de s'adapter à des données plus complexes. Dans le cas des données artificielles, $OFCMd$ et $HOFCMd$ obtiennent les meilleures performances ou les partagent avec les autres méthodes sans différence significative [1]. Cette expérimentation préliminaire doit être approfondie, mais montre que nos méthodes sont compétitives avec des algorithmes récents de traitements de flux quand elles sont utilisées dans un contexte incrémental de traitement de jeux de données volumineux.

Analyse de flux de données

Similairement à [72], nous proposons d'évaluer nos algorithmes sur les flux de données en calculant pour chacun la pureté (voir équation 7.3) de la partition produite sur les derniers lots traités. Le nombre de lots considéré est appelé *horizon*. Ainsi, si l'horizon $H = 3$, on ne calcule la pureté que sur les 3 derniers lots de données. La capacité d'adaptation des clusters aux évolutions dynamiques du flux dépend aussi du facteur d'oubli qui donne plus de poids aux nouveaux centres qu'aux anciens.

La figure 2.8 présente les résultats obtenus par les algorithmes $OFCMd$ et $HOFCMd$ sur les flux de données $Stream_1$ et $Stream_2$ en fonction du paramètre d'oubli et pour des valeurs d'horizon $H \in [1, 5]$. Dans le cas de $HOFCMd$, la valeur d'historique est fixée à 1 mais d'autres études (non reportées ici) montrent que les résultats sont comparables avec des valeurs d'historique comprises

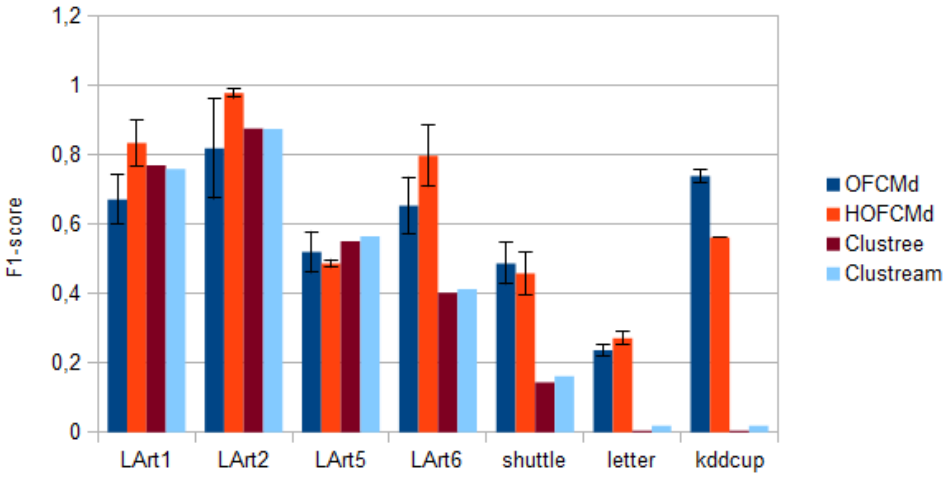

FIGURE 2.7 – Comparaison des F1-scores moyens des algorithmes $OFCMd$ et $HOFCMd$ aux scores des méthodes $Clustream$ et $Clustree$ de la plateforme MOA [134, 135]

dans l'intervalle $[1, 5]$.

Comme attendu, les performances en terme de pureté des algorithmes $OFCMd$ et $HOFCMd$ sont invariantes à la valeur de l'horizon H sur le flux $Stream_1$. Cela peut être expliqué par le fait que la distribution des données pour ce flux est invariante dans le temps, et que les derniers médoides découverts sont donc toujours représentatifs des premières données traitées. On peut également remarquer que les performances s'améliorent avec l'augmentation du facteur d'oubli. Cela est dû au fait que l'augmentation de l'oubli entraîne une baisse de l'influence des premiers médoides découverts, ces médoides étant potentiellement moins représentatifs que les derniers, car déterminés sur moins de données. Enfin, comme les performances sont meilleures quand on donne plus de poids au nouveaux médoides, cela indique que ceux-ci sont de meilleure qualité que les premiers et cela valide l'efficacité de notre approche incrémentale. Enfin, comme pour les grands jeux de données (voir section 2.5.4), l'algorithme $HOFCMd$ obtient de meilleures performances que l'approche $OFCMd$.

Dans le cas du flux $Stream_2$ et similairement à ce qui est observé pour $Stream_1$, les performances s'améliorent quand le facteur d'oubli augmente. Cependant, du fait de l'évolution des données au cours du temps, les résultats sont meilleurs sur un horizon court ($H = 1$) que sur un intervalle de temps plus long ($H = 5$). Ce résultat tend à montrer qu'il y a bien adaptation des médoides aux dernières données traitées et que par conséquence, ceux-ci deviennent moins pertinents pour les données plus anciennes. Par ailleurs, l'algorithme $HOFCMd$ obtient de meilleures performances que $OFCMd$ ce qui tend à montrer que son modèle basé sur un historique est plus apte à s'adapter à la dynamicité des flux.

Enfin, ces expérimentations montre l'intérêt de l'introduction du paramètre d'oubli qui permet soit d'améliorer les performances dans le cas de flux statiques, soit de les stabiliser dans le cas de flux dynamiques, là où un algorithme classique aurait vu son erreur de classification augmenter.

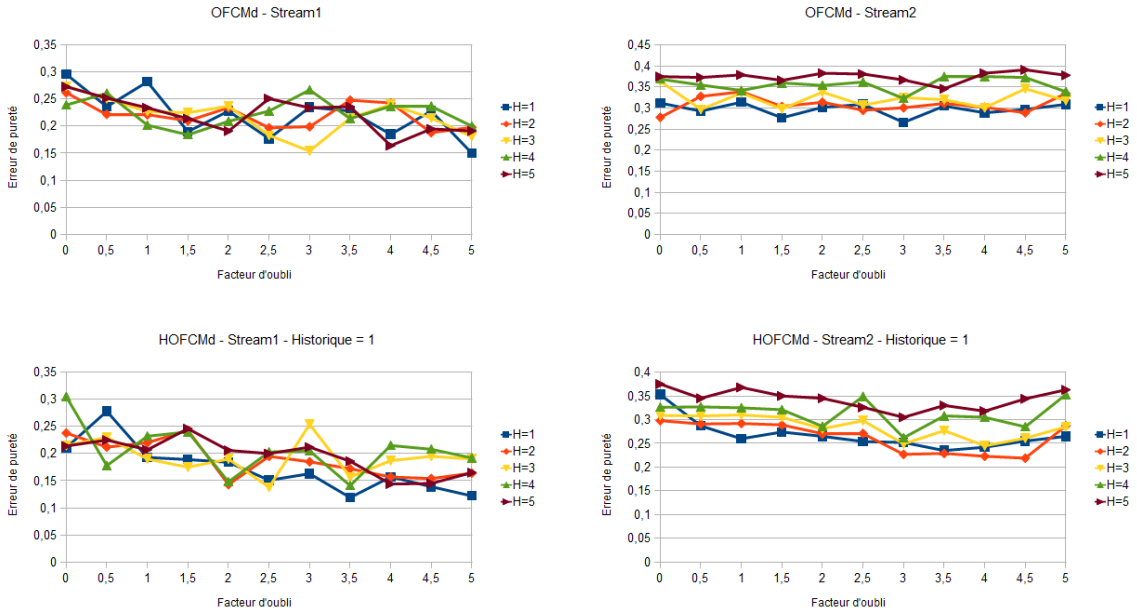

FIGURE 2.8 – Pureté des partitions produites par les algorithmes $OFCMd$ (Haut) et $HOFCMd$ (Bas) sur les flux de données $Stream_1$ et $Stream_2$ en fonction du facteur d'oubli $\in [0, 5]$ et du paramètre d'horizon $H \in [1, 5]$. L'historique est fixée à 1 pour la méthode $HOFCMd$

2.6 Conclusions et perspectives

Ce chapitre résume les différents algorithmes de clustering que j'ai proposés durant ces dernières années de façon à apporter des réponses aux problèmes théoriques de passage à l'échelle ou de traitement de flux de données associés à l'analyse des usages.
Les méthodes développées facilitent le passage à l'échelle en linéarisant la complexité associée à la recherche d'un médoide et en ne réalisant qu'une seule passe sur les données dans l'algorithme *Leader Ant* ou en mettant en place un schéma incrémental dans les approches $OFCMd$ et $HOFCMd$. L'autre objectif de mes travaux est de proposer des approches relationnelles qui sont capables de s'adapter à toutes représentations de données dès lors qu'une mesure de (dis)similarité est définie pour le problème considéré.

Concernant l'algorithme *Leader Ant*, il pourrait être intéressant de définir un mécanisme incrémental de résumé des clusters pour mettre à jour l'information relative à chaque cluster et pour permettre également de libérer de la mémoire au fil du temps. Par ailleurs, plusieurs travaux ont déjà été conduits dans le but d'améliorer LA comme l'algorithme *Leader Ant* avec des contraintes [16] développé dans le cadre de la thèse de Viet-Vu Vu sur le clustering semi-supervisé (voir le chapitre 4), ou encore l'algorithme *Leader Ant Racing*. Cet algorithme vise à accélérer LA et à le rendre plus fiable grâce à un mécanisme de *racing*, issu des méthodes génétiques, qui met en compétition les différents nids lors de l'affectation d'une fourmi.

Concernant nos approches incrémentales $OFCMd$ et $HOFCMd$, les perspectives de travail sont nombreuses. Il faut en premier lieu développer les expérimentations qui ont été réalisées jusqu'alors et étudier des jeux de données relationnelles plus volumineux (centaines de milliers ou millions de données), issus de l'analyse des traces d'usage sur Internet par exemple. En second lieu, il paraît utile de détailler l'étude des différences induites par nos deux modèles incrémentaux quand un

nouveau sous-jeu de données est traité, et notamment la capacité de chacun à suivre l'évolution des tendances exprimées dans un flux de données au cours du temps comme cela a déjà été initié sur les flux $Stream_1$ et $Stream_2$. De nouveaux flux artificiels devront être générés avec de nouvelles difficultés comme l'insertion de déséquilibres entre les clusters en fonction du temps, l'apparition de nouveaux groupes ou encore le rapprochement de groupes existants. Enfin, il serait intéressant d'étudier les possibilités d'adaptation du schéma incrémental proposé en remplaçant l'approche des c-médoides flous pondérés, dont le nombre de clusters doit être fixé, par une approche comme *Leader Ant* par exemple, ou encore d'observer l'impact du remplacement de la méthode finale d'agrégation utilisée dans l'algorithme OFCMd sur les performances.

De manière plus générale, d'autres pistes de travail sont envisageables pour permettre le passage à l'échelle des méthodes de clustering, similairement à ce qui est proposé dans l'approche $eNERF$ [66]. Dans cette approche un échantillon de taille réduite est construit à partir d'un très grand jeu de données et est partitionné. Les résultats du partitionnement sont ensuite étendus à l'ensemble des données restantes pour obtenir la partition complète. Dans ce cadre, il est envisageable de combiner dans une approche hybride *Leader Ant* (ou une de ses déclinaisons) à un algorithme relationnel plus précis mais éventuellement plus coûteux en temps de calcul sur un petit échantillon des données et de ne considérer *Leader Ant* que comme un mécanisme d'extension des clusters découverts à l'échelle du jeu de données complet.

Chapitre 3

Analyse de traces utilisateurs sur Internet

L'analyse de traces utilisateurs sur Internet (ou *web usage mining*) est une activité qui a pour objectif de mettre en évidence les différents comportements de navigation des internautes ainsi que leurs besoins d'information à l'aide de méthodes de fouille de données [76]. Cette information extraite prend la forme de profils utilisateurs ou de motifs de navigation sur Internet. Les deux principales applications sont d'une part la mesure d'audience, qui s'intéresse à la caractérisation des usages, et d'autre part la personnalisation dynamique [157, 153], qui découle de la première, pour modifier la structure [151] ou le contenu d'un site dynamiquement.

Comme déjà indiqué dans ce mémoire, le développement des algorithmes Leader Ant (LA), OFCMd et HOFCMd a été motivé par la problématique du traitement de grandes quantités de données d'usages qui ne sont pas nécessairement représentées par des vecteurs numériques. Ces méthodes ont donc été intégrées à un système d'analyse de traces utilisateur, présenté dans ce chapitre, qui regroupe des méthodes de récupération, prétraitement et analyse des données d'usages ainsi que d'interprétation des résultats.

Ce chapitre est organisé comme suit : la section 3.1 présente un rapide tour d'horizon des travaux du domaine de l'analyse de traces utilisateurs sur Internet. La section 3.2 décrit ensuite la spécificité de la méthode proposée, en terme de comparaison de parcours utilisateurs (ou *sessions*) et d'interprétation des résultats, avec la mise en place d'un mécanisme de visualisation qui étend celui initié par le système WebViz [168] en intégrant notamment un mécanisme de détermination de profils représentatifs pour chaque groupe d'utilisateurs. La section 3.3 propose des illustrations de résultats de ce système lors de projets collaboratifs. Enfin la section 3.4 présente les conclusions et perspectives de ces travaux.

3.1 Principales méthodes de web usage mining

Un très grand nombre de travaux ont été conduits durant les quinze dernières années pour extraire, analyser, modéliser ou prédire les besoins d'information des utilisateurs d'un site Internet. Pour ce faire, plusieurs approches ont été décrites dans la littérature comme rappelé dans [153]. Certaines approches cherchent à extraire des motifs de navigation ou des ensembles de règles comportementales. Ces méthodes sont basées sur la recherche de règles d'association comme dans le système WebMiner [77], de règles séquentielles comme dans l'approche WebTool [150, 151] ou sur la détermination de motifs de navigations à partir de la représentation des parcours dans un arbre agrégé dans le système WebWUM [185].

D'autres, comme [199, 26, 56, 110, 91, 136, 156, 157, 92], appliquent des algorithmes de classification non supervisée pour découvrir des groupes de sessions homogènes, c'est-à-dire qui rassemblent les internautes qui exhibent le même comportement de navigation. L'hypothèse sous-jacente de ces méthodes est que deux internautes avec le même comportement possèdent le même besoin d'information et les mêmes préférences. L'objectif des méthodes de clustering peut alors être de découvrir les utilisateurs qui accèdent aux mêmes ressources dans le site, ou de trouver des groupes de pages qui apparaissent fréquemment dans les mêmes sessions.

La mise en œuvre de méthodes de clustering doit permettre de faire apparaître des motifs de navigation que les méthodes à base de règles, qui reposent sur une exploration systématique des co-occurrences entre pages visitées, ne sont pas à même de découvrir. En revanche, les méthodes de clustering doivent posséder une complexité faible pour traiter les volumes de données d'usage et pouvoir manipuler des sessions sous d'autres formes que de simples vecteurs [26, 136, 156, 157]. Ensuite la capacité des méthodes de clustering à trouver des clusters pertinents dépend de la métrique utilisée pour comparer les sessions.

Dans [156], les auteurs proposent une mesure de similarité S_{kl} entre des sessions $s^{(k)}$ et $s^{(l)}$ représentées par des vecteurs de transactions tels que :

$$s_j^{(i)} = \begin{cases} 1 & \text{si l'url } j \text{ a été accédée durant la session } i \\ 0 & \text{sinon.} \end{cases}$$

La mesure de similarité S_{kl} repose sur la combinaison de deux autres mesures de similarité entre sessions $S_{1,kl}$ et $S_{2,kl}$:

$$S_{kl} = \max(S_{1,kl}, S_{2,kl}) \tag{3.1}$$

La mesure de similarité entre sessions $S_{1,kl}$ est un produit scalaire normalisé entre les vecteurs représentatifs des sessions $s^{(k)}$ et $s^{(l)}$ et permet de capturer les visites co-occurrentes de pages entre les deux sessions.

$$S_{1,kl} = \frac{\sum_{i=1}^{Nu} s_i^{(k)} s_i^{(l)}}{\sqrt{\sum_{i=1}^{Nu} s_i^{(k)}} \sqrt{\sum_{i=1}^{Nu} s_i^{(l)}}} \tag{3.2}$$

où Nu représente le nombre d'urls dans le site et détermine la dimension des vecteurs de sessions. Cette similarité permet de favoriser la création de groupes s'intéressant aux mêmes pages exactement mais ne prend pas en compte l'arborescence du site ou la proximité qu'il peut exister entre deux pages et donc entre sessions. Pour cela les auteurs décrivent la seconde mesure de similarité entre sessions $S_{2,kl}$ qui repose sur une similarité entre pages visitées. Cette similarité entre pages est calculée à partir de l'url des pages et est définie en fonction de la portion des chemins p_i et p_j commune aux deux urls i et j en partant de la racine du site.

$$S_u(i,j) = \min(1, \frac{|p_i \cap p_j|}{\max(1, \max(|p_i|, |p_j|) - 1)}) \tag{3.3}$$

La similarité S_2 reprend l'expression de S_1 en y intégrant la similarité entre urls S_u comme le montre l'équation suivante.

$$S_{2,kl} = \frac{\sum_{i=1}^{Nu} s_i^{(k)} s_i^{(l)} S_u(i,j)}{\sqrt{\sum_{i=1}^{Nu} s_i^{(k)}} \sqrt{\sum_{i=1}^{Nu} s_i^{(l)}}} \tag{3.4}$$

La mesure de similarité S_2 peut prendre de petites valeurs quand les deux sessions sont identiques et voit ses valeurs diminuer lorsque les pages qui n'ont pas été visitées en commun possèdent des urls très différentes. Dans ces cas, la mesure S_1 est plus performante, ce qui explique l'agrégation opérée pour obtenir la similarité globale S.

Dans [194] les auteurs proposent une autre mesure de similarité entre sessions qui repose également sur la définition d'une mesure de similarité entre les urls basée sur la comparaison de leurs chemins. Soit p_i et p_j les chemins associés aux urls i et j. La méthode proposée affecte à chacune des portions des chemins comparés un poids qui dépend de la longueur du plus long chemin, les premières portions ayant un poids maximal et la dernière portion du chemin le plus grand un poids de 1. Soit $s_{max} = \max(|p_i|, |p_j|)$ et s_{eq} la longueur du chemin commun entre les urls i et j jusqu'à la première portion différente. La similarité $S_u^2(i,j)$ entre les urls i et j peut être définie comme suit :

$$S_u^2(i,j) = \frac{\sum_{k=1}^{s_{eq}}(s_{max} - k + 1)}{\sum_{k=1}^{s_{max}} k} \tag{3.5}$$

Cette mesure de similarité entre urls donne plus d'importance aux premières portions dans les urls et pénalise moins les différences de chemins qui apparaissent plus profondément dans l'arborescence du site. Une mesure de similarité entre sessions dérivée d'une distance d'édition (ou distance de Levenshtein) est ensuite décrite. Dans ce modèle les sessions sont assimilées à des mots écrits dans l'alphabet des pages du site. La mesure S_u^2 est alors utilisée pour calculer les coûts de substitution compris entre -10 - quand les pages sont complètement différentes - et 20 - quand les pages sont identiques. Comparablement, dans [78] les *string kernels* sont utilisées pour caractériser, à partir de traces d'usage, le rapprochement au cours du temps du comportement des utilisateurs d'un système pédagogique à celui d'un individu référence.

D'autres mesures de similarités ont été proposées de façon à s'adapter à des représentations plus ou moins riches des sessions. Par exemple, dans [110, 111, 74], les auteurs introduisent une description multi-modale des sessions dans laquelle chaque session est représentée plusieurs vecteurs correspondant au contenu (mot-clés), aux liens entrants, sortants ainsi qu'aux urls des principales pages visitées durant la session. Dans [91, 92], les auteurs s'appuient sur un mécanisme de généralisation des urls qui consiste à remplacer des requêtes sur des pages par leur ancêtre dans l'arborescence du site. Ce faisant, le nombre de page diminue drastiquement et évite les problèmes de vecteurs creux, tout en améliorant l'interprétabilité des profils extraits. Plus récemment, des travaux ont été conduits pour améliorer les modèles utilisateurs à l'aide d'interactions plus riches (comme imprimer ou éditer des documents) [133] ou à partir du contenu. Dans [154], les navigations utilisateurs sont stockées dans une matrice qui représente à la fois les usages et les occurrences des termes issus des documents visités. [182] propose des profils utilisateurs basés sur des ontologies, et dans [124], les auteurs utilisent une analyse probabiliste latente (PLSA) pour étudier à la fois les usages et le contenu dans le but améliorer la pertinence de recommandation.

D'après Cooley [76, 77], les méthodes d'analyse de traces ne sont utiles que si elles sont couplées à des mécanismes d'interrogation pour aider à l'interprétation des résultats. Dans le système *WebViz* [168], les auteurs introduisent le paradigme *web-path* dans lequel un site Internet est représenté par un graphe orienté dont les sommets sont les ressources et les arcs, les liens hypertextes entre les pages. Dans ce cadre, un parcours utilisateur est vu comme un chemin dans ce graphe. Ces systèmes de visualisation, du fait de la taille des sites étudiés, se heurtent cependant au problème de la représentation de très grands graphes [112]. Plusieurs outils [73, 116, 200] ont été proposés depuis pour enrichir la visualisation introduite dans [168] : [73] prend en compte la topologie, le contenu et les usages d'un site Internet pour aider à comprendre les relations entre production d'informations (nouvelles pages ajoutées ou mises à jour) et consommation lors des visites des internautes (notion d'"écologie"). *Webquilt* [116] propose un canevas complet pour l'analyse et la visualisation des sessions des utilisateurs avec des méthodes de filtrage des sessions affichées (par exemple les chemins optimaux au sens de l'utilisabilité de l'interface). Enfin, dans [200], les auteurs définissent le "Visual Web Mining" comme l'application de méthodes de visualisation sur des résultats issus de fouille de données du Web ("web mining").

3.2 Méthode d'analyse proposée

Nous présentons ci-après les spécificités de notre approche qui repose sur la combinaison de nos algorithmes de clustering utilisés conjointement avec des mesures de similarités entre sessions et des mécanismes de visualisation et de résumé de l'activité sous forme de profils représentatifs.

En premier lieu, et bien qu'il existe de nombreuses méthodes pour récupérer les traces d'activité, notre approche repose principalement sur la méthode de référence introduite dans [77] qui construit les sessions à partir des fichiers log disponibles sur les serveurs Web. Une seconde méthode développée par Lionel Yaffi de la société ILObjects est parfois utilisée pour obtenir des traces de meilleure qualité. Cette approche, plus précise, tire profit des nouvelles technologies offerte par le web 2.0 pour affecter un identifiant à chaque internaute et enregistrer et transmettre ensuite chacune de ses actions.

En second lieu, deux mesures de comparaison entre sessions ont été proposées pour notre méthode. Elles s'inspirent des mesures décrites dans [193] d'une part et [156] d'autre part. La première mesure de comparaison entre sessions proposée est une distance d'édition modifiée dans laquelle les poids de substitution entre deux pages visitées sont définis par la similarité présentée dans [193] (voir équation 3.5) et les poids d'insertion et de substitution sont maximums (égaux à 1). Si les deux urls possèdent un chemin très similaire, le coût de substitution est très faible et inversement, si les deux urls sont complètement différentes, le coût de substitution est égal aux coûts de suppression ou d'insertion. Cette mesure est très proche de celle proposée par [193] mais ne dépend pas des valeurs de pénalité -10 et de bonus 20 introduites dans ces travaux pour favoriser les sous-séquences communes. Notre mesure est donc plus stricte lors de la comparaison entre deux urls et aura tendance à produire plus de clusters mais contenant des séquences plus similaires.

La seconde mesure de comparaison entre sessions repose sur le calcul de la similarité entre toutes les paires d'urls visitées dans chaque session comme le montre l'équation 3.6 suivante. Soit $S_u^2(i,j)$ la similarité entre les urls i et j définie dans l'équation 3.5. La similarité $S_{3,kl}$ entre deux sessions $s^{(k)}$ et $s^{(l)}$ est définie comme suit :

$$S_{3,kl} = \sum_{\forall i \in s^{(k)}} \sum_{\forall j \in s^{(l)}} S_u^2(i,j) \tag{3.6}$$

Cette valeur de similarité peut être normalisée dans l'intervalle $[0,1]$ de manière classique en la divisant par $\sqrt{S(s^{(k)},s^{(k)}) \times S(s^{(l)},s^{(l)})}$. Elle tient compte du nombre de visites de chaque url, ainsi que de la similarité entre urls. De ce fait, elle porte plus d'information que les mesures basées sur les représentations plus simples de types vecteurs d'impacts ou de transaction associées à des distances euclidiennes ou de Jaccard. De plus, elle peut aisément être améliorée en modifiant la mesure de similarité entre urls, par exemple en considérant le contenu des pages ou une description sémantique liée à une ontologie, comme ce qui a été fait dans le projet DoXa (voir section 3.3). Des expérimentations rapportées dans [18] montrent que la seconde distance proposée produit des clusters plus compacts et plus facilement interprétables en un temps plus court que la première, pénalisée par le calcul de la distance d'édition.

Enfin, notre approche intègre des fonctionnalités de visualisation des parcours qui étendent le modèle *web-path* introduit dans le système *WebViz* [168]. De façon à contourner le problème de l'affichage de grands graphes, notre approche représente chaque cluster de sessions par son propre graphe et résume chaque cluster par un petit nombre de profils représentatifs. Deux types de représentation principales sont proposés : soit un *graphe des usages* comparables à *WebViz* [168], soit un arbre agrégé inspiré de [185].

Le *graphe des usages* est interactif : les nœuds et les arcs peuvent être filtrés selon leur fréquentation, et le graphe peut être simplifié soit en ne considérant que le voisinage d'un nœud sélectionné soit par le mécanisme de généralisation des urls [92] décrit dans la section 3.1. Ce passage d'un graphe de pages à un graphe de répertoires, correspondant à des parties du site, réalise un *zoom structurel*. Nous proposons également un mécanisme de *zoom sémantique* qui consiste à projeter des étiquettes sémantiques (des métadonnées) sur des sous-ensembles de pages, ce qui est matérialisé par des zones de couleurs englobant les nœuds concernés du graphe. Ces métadonnées sont supposées intégrées à chacune des pages du site, mais notre outil autorise également d'extraire les mot-clés les plus fréquents de chacune des ressources pour les utiliser dans ce cadre. Enfin, si les métadonnées sont elles-mêmes organisées de manière hiérarchique (dans une taxonomie), il est également possible d'afficher ces informations selon un niveau de profondeur désiré pour réaliser le zoom sémantique dans l'espace de représentation des parcours utilisateurs.

Dans la représentation sous forme de *graphe des usages*, seules les transitions entre deux pages peuvent être étudiées, car la projection réalisée ne retranscrit pas visuellement l'ordre complet dans lequel les pages sont accédées durant les sessions. Pour palier cette limitation, nous proposons également une visualisation sous forme d'arbre agrégé inspirée de [185]. Celle-ci a pour vocation de représenter les sessions des utilisateurs sous la forme d'un arbre dans lequel chacune des branches correspond à un parcours observé sur le site. Tant que deux sessions suivent le même parcours (les mêmes pages), elles empruntent la même branche de l'arbre et dès lors qu'elles divergent, la branche se subdivise pour représenter les deux nouveaux flux de navigation. L'intérêt de cette approche réside dans le fait qu'elle permet de conserver l'ordre dans lequel les différentes pages ont été visitées, et autorise également la visualisation des retours en arrière dans le site, ce qui peut être utile dans le cadre d'études ergonomiques. En contrepartie, et contrairement à l'approche précédente basée sur les graphes, chaque nœud peut être représenté plusieurs fois : au sein d'une même session s'il est revisité ou bien dans plusieurs sessions avec des parcours différents qui auraient accédées ce nœud.

Enfin, notre outil permet de résumer l'activité d'usage de chaque cluster par le biais de représentants. Ceux-ci peuvent être soit déterminés par les médoides issus des méthodes OFCMd ou HOFCMd (il est également possible de considérer les points qui maximisent leur appartenance aux clusters), ou soit calculés à partir de *degrés de typicalité* [172, 143]. La typicalité est une notion issue des études cognitives sur les représentants de catégories [173]. Elle indique que la représentativité d'un point dans un groupe est déterminée à la fois par sa ressemblance avec les autres points de son groupe (comme dans le cas des moyennes classiques ou de ses dérivées comme la médiane) mais également par sa dissimilarité avec les autres groupes. Un représentant typique est donc un point qui assure un compromis entre ressemblance interne et dissimilarité externe et qui souligne également la spécificité d'un cluster par rapport aux autres, ce qui le rend plus représentatif que d'autres approches (voir [142, 143] pour une comparaison plus complète entre les différentes méthodes). De nombreux opérateurs d'agrégation peuvent être utilisés pour moduler le comportement désiré : conjonctif (par exemple min), disjonctif (par exemple max) ou de compromis selon les valeurs à agréger (par exemple la moyenne)). De plus amples détails peuvent être trouvés dans [19, 85].

En conclusion, les deux visualisations sont complémentaires : le graphe des usages privilégie une vue plus globale des flux de navigation et autorise une analyse et une localisation plus simple des pages les plus fréquentées alors que l'arbre agrégé permet d'estimer la longueur des parcours et la dispersion au fil du temps des utilisateurs vers certaines portions du site ainsi que les cycles dans les navigations. La vue par cluster ainsi que la recherche de sessions représentatives permet de résumer l'information d'usage. La figure 3.1 présente des exemples des principales visualisation.

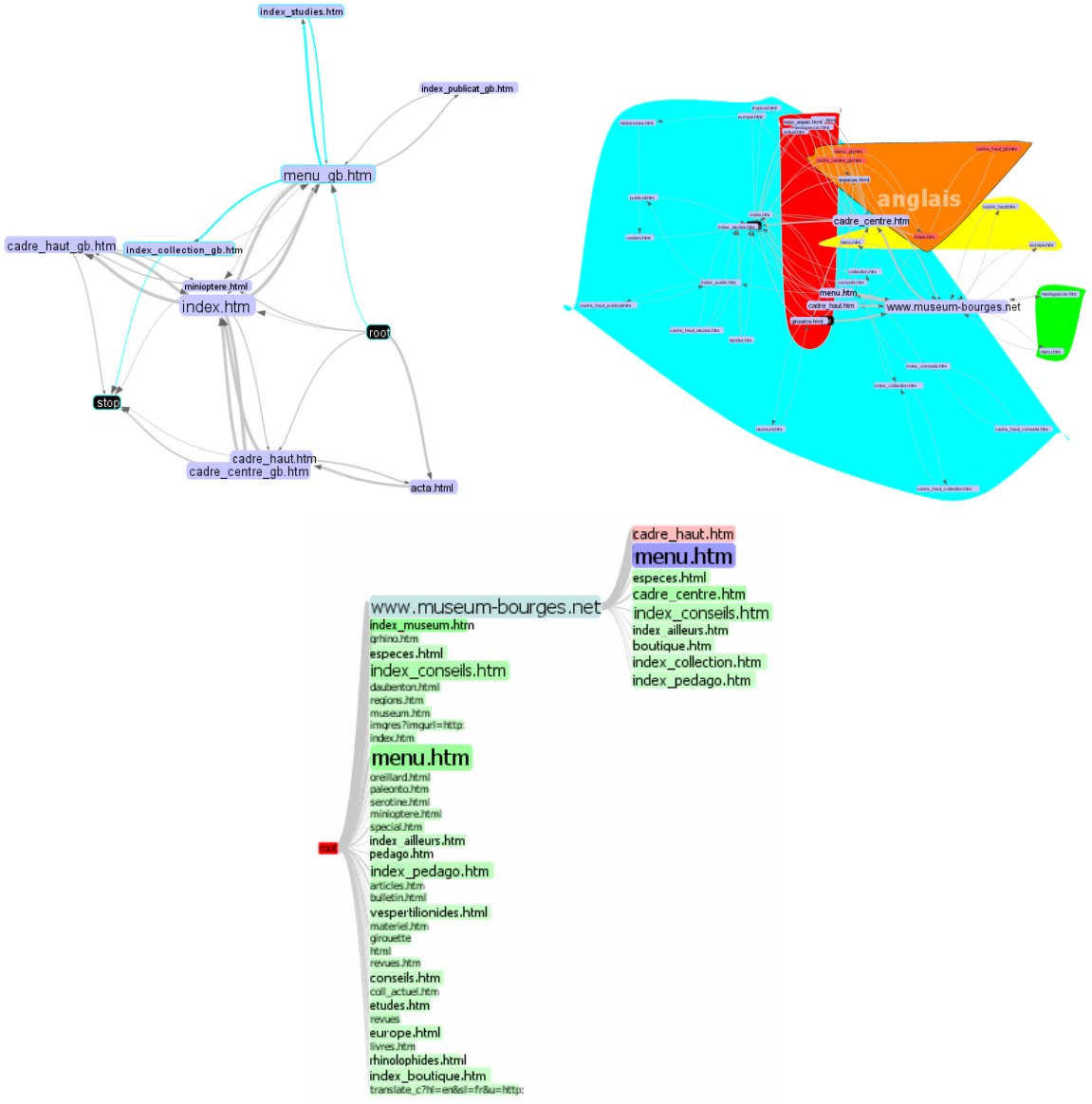

FIGURE 3.1 – Illustrations des différents modes de visualisation proposés : haut-gauche : graphe des usages avec représentation des profils typiques (en *cyan*), haut-droite : zoom structurel, bas : arbre agrégé.

3.3 Exemple de résultats obtenus

Les outils d'analyse et de visualisation de traces utilisateurs ont été mis en œuvre et validés dans le cadre de projets collaboratifs dont nous proposons un rapide aperçu ci-après.

Les premiers travaux ont été conduits dans le cadre du projet WebCSTI (voir la section 7.4.1). Notre sous-projet s'intéressait à l'analyse de données d'usage issues d'une dizaine de sites Internet, parmi lesquels des sites de grandes institutions comme la Cité des Sciences et de l'Industrie de la Villette à Paris, le site de l'Onera ou encore le site des Arts et Métiers. Les quantités de données (jusqu'à 98000 sessions pour le site de la Cité des Sciences et de l'Industrie) et l'impossibilité de déterminer a priori pour tous ces sites le nombre de profils utilisateurs à rechercher ont motivé le développement de l'algorithme *Leader Ant* qui répond à ces deux besoins. Ce projet a fourni un cadre propice à l'évaluation de la méthode *Leader Ant* (LA) sur des données d'usage réelles. Les expérimentations conduites dont les résultats sont présentés dans le tableau 3.1 pour 3 sites étudiés lors du projet *webCSTI* ont montré que : (1) LA produit des partitions de meilleure qualité que l'approche LFCMdd sur les données de test, et (2) que LA converge lorsque l'on observe l'erreur de Rand estimé sur la comparaison de 50 partitions pour chaque jeu de données d'usage réelles, malgré son caractère non déterministe. Enfin, le projet *webCSTI* a également permis de révéler

Sites web	**Muséum de Bourges**	**Cap Sciences**	**CCSTI Grenoble**
Convergence (LA)	0.06	0.17	0.03
Qualité partition (LA)	0.61 ± 0.02	0.88 ± 0.03	0.38 ± 0.03
Qualité partition (LFCMdd)	0.71 ± 0.00	0.94 ± 0.00	0.64 ± 0.00

TABLE 3.1 – Évaluation de l'algorithme *Leader Ant* sur des données d'usage réelles lors du projet *webCSTI*

l'importance de disposer d'outils pour l'aide à l'interprétation des profils utilisateurs découverts par la méthode *Leader Ant*.

Le projet Infom@gic (voir la section 7.4.2) a permis le développement du logiciel d'analyse et de visualisation de traces grâce à la collaboration initiée avec la société Intelligent Learning Objects (start-up du LIP6). Plusieurs applications de l'outil proposé ont été réalisées lors de collaborations. La première collaboration a été mise en place avec l'association *Fil Santé Jeunes* dont la mission est d'informer et de mettre en place un dialogue avec les jeunes adolescents à propos des différents problèmes auxquels ils sont confrontés. Notre étude a permis de mettre en lumière des groupes d'usagers bien définis avec des problèmes relatifs à la sexualité masculine ou féminine, ou bien encore le rôle des assistantes sociales ou du juge pour enfant. Plusieurs clusters relatifs à la partie forum faisaient également ressortir le rôle important du dialogue avec les adolescents, mais mettaient en lumière le peu de navigations transverses entre le forum où les questions sont posées et la partie éditoriale du site qui contient de nombreuses réponses. La seconde application a été mise en œuvre avec le site Maxicours, spécialisé dans le soutien scolaire sur Internet. Un mois de données d'usage correspondant à la partie publique du site a été traité par notre outil soit 1.246.185 requêtes pour 102.548 sessions différentes. La particularité de cette étude repose sur l'expression des sessions non pas dans l'ensemble des pages du site mais dans l'espace des métadonnées associées à chacune des pages. Cette transformation a permis d'augmenter significativement la sémantique des groupes découverts et l'étude des profils utilisateurs a permis de valider un changement de structure du site engagée par ses concepteurs. De manière générale, les expérimentations conduites ont montré la capacité de l'outil proposé à répondre à différentes problématiques applicatives ainsi que l'intérêt d'enrichir les profils utilisateurs avec la projection des métadonnées sur les parcours

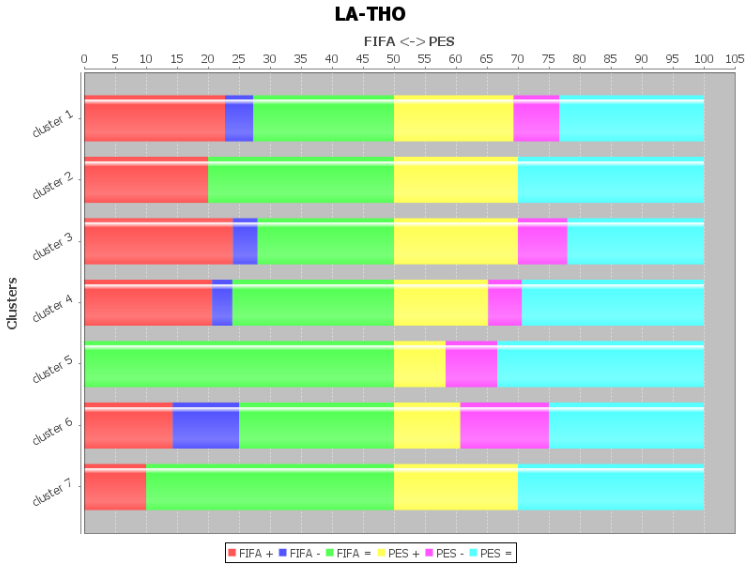

FIGURE 3.2 – Description des clusters découverts par Leader Ant avec un modèle de session basé sur les thèmes et les opinions. Chaque barre représente un cluster et indique dans sa moitié gauche les opinions relatives au jeu FIFA et dans sa moitié droite les opinions relatives au jeu PES pour les internautes dont les parcours sont dans le cluster. Notations : '+' représente les opinions positives, '=' les opinions neutres et '-' les opinions négatives.

pour aider dans la tâche d'interprétation des résultats d'analyse.

Enfin le projet DoXa (voir la section 7.4.3) du pôle de compétitivité Cap Digital s'est intéressé à l'hypothèse selon laquelle les parcours des internautes pouvaient être liés à leur opinion ou bien, a minima, que certains parcours pouvaient influencer leur opinion. Deux études ont été conduites : la première sur le site *TGV Rézo* mis en place par la SNCF pour créer une communauté d'utilisateurs des TGV, et la seconde sur un site artificiel regroupant des pages (de tests ou des comparatifs) relatives à deux simulations sportives FIFA et PES et véhiculant des opinions plus ou moins favorables à l'un ou l'autre des jeux. De façon à mesurer le changement d'opinion, un panel d'utilisateurs volontaires a été interrogé avant et après leur navigation. Au final, l'étude conduite sur 189 parcours utilisateurs et basée sur une mesure de comparaison de sessions prenant en compte les thèmes de chaque pages et les opinions ont permis la réalisation de groupes utilisateurs interprétables reflétant des opinions marquées (voir la figure 3.2).

3.4 Conclusions et perspectives

Ce chapitre a présenté nos méthodes d'analyse des usages qui concilient le traitement de grandes masses de données à l'aide des algorithmes de partitionnement décrits dans le chapitre 2 et l'interprétation des clusters de sessions par le biais d'un outil de visualisation interactive et de recherche de représentants. Celui-ci autorise un expert du domaine à adapter la visualisation des parcours utilisateurs à ses besoins et permet de donner du sens aux profils découverts grâce aux mécanismes de zooms structurels (qui rattachent les usages à une portion du site) et de zooms sémantiques (qui rattachent les parcours à des métadonnées extraites des pages visitées).

Ces outils ont été évalués dans le cadre de plusieurs projets collaboratifs sur des données d'usage réelles et ont permis dans tous les cas de fournir des informations pertinentes concernant l'analyse

des usages sur le site étudié. De nombreux développements sont possibles dans le futur, comme par exemple l'introduction de mécanismes de zooms plus flexibles au niveau de la visualisation, qui permettraient d'observer les flux entre ressources à différentes échelles en agrégeant les sommets et les arcs du graphe des usages selon différents critères structurels ou sémantiques. Il pourrait être également intéressant de travailler sur la génération automatique de commentaires en langue naturelle pour expliquer chaque cluster.

Ces travaux ont été réalisés en collaboration avec Marie-Jeanne Lesot (MCF, UPMC LIP6) pour les mesures de typicalité et avec Lionel Yaffi (Ingénieur, ILObjects) pour le développement du logiciel d'analyse et de visualisation de traces.

Chapitre 4

Clustering semi-supervisé

4.1 Introduction

Comme le notent [121, 169], une des pistes possibles d'évolution pour les méthodes de clustering consiste à s'orienter vers des modèles dits *semi-supervisés* tirant partie d'indications données par un expert du domaine. Ces informations prennent soit la forme de *contraintes* qui spécifient sous quelle condition deux points doivent appartenir au même cluster, soit la forme de points étiquetés ou "graines" (*seeds* en anglais) qui indiquent directement le groupe d'appartenance de la donnée. On distingue deux principaux types de contraintes : les contraintes *must-link* (ML) qui indiquent que deux points de l'ensemble de données doivent être dans le même groupe et les contraintes *cannot-link* (CL), qui inversement imposent que deux points appartiennent à deux clusters différents [192].

De nombreuses méthodes de clustering semi-supervisé ont été proposées dans la littérature afin d'améliorer la qualité des partitions produites ou la vitesse de convergence des méthodes. Par exemple, les algorithmes Seed K-Means [58], Seed Fuzzy C-Means [162, 63, 164] ou encore les MPCK-Means [69] améliorent les approches de type c-means en rendant le processus d'initialisation déterministe et/ou en utilisant les connaissances expertes dans la phase d'affectation des points aux clusters de façon à accélérer la convergence. Les algorithmes C-DBSCAN [176, 177], SSDBSCAN [140] et HISSCLUS [68] autorisent le traitement de clusters avec des densités différentes, ce que ne peut pas faire l'algorithme de référence DBSCAN [88]. L'introduction de connaissances expertes permet également d'améliorer la séparation entre clusters dans les approches hiérarchiques [80] et dans les approches spectrales [127, 195, 152].

Bien que les travaux actuels s'intéressent plus particulièrement à l'adaptation de méthodes de clustering existantes pour la prise en charge de contraintes ou de données étiquetées, ils conservent les mêmes limitations que les méthodes dont ils s'inspirent (traitement de données exclusivement numériques, paramétrage complexe) et reposent sur une sélection aléatoire des connaissances qui peut conduire à de mauvaises performances comme le note [190].

Les travaux conduits durant la thèse de Viet-Vu Vu [43] apportent une réponse à ces problèmes et s'articulent autour de deux contributions principales. La première concerne des algorithmes d'apprentissage actif pour la sélection des contraintes ou de données étiquetées, qui permettent de choisir les connaissances les plus profitables aux méthodes de clustering tout en minimisant l'effort d'annotation. La seconde propose de nouveaux algorithmes de clustering semi-supervisé, basés sur les contraintes et les données étiquetées, qui visent à améliorer les méthodes décrites dans la littérature. Ce chapitre est donc organisé comme suit : la section 4.2 présente nos contributions pour la sélection de contraintes et de données étiquetées. La section 4.3 présente les algorithmes *Leader Ant avec des Contraintes* (LAC) et Semi-Supervised Graph-Based Clustering (SSGC). Enfin la section 4.4 propose les conclusions et perspectives de ces travaux.

4.2 Sélection active de connaissances

Nous présentons dans cette section nos travaux pour la sélection de connaissances sous la forme de contraintes ou de données étiquetées. Comme déjà indiqué, la plupart des travaux du domaine s'intéresse en premier lieu à proposer de nouvelles approches de clustering semi-supervisé [191, 69, 79, 81, 80, 176, 140, 58, 54] et font l'hypothèse que des connaissances expertes de qualité sont fournies aux algorithmes de manière passive. Pour cela, les méthodes sont généralement mise en œuvre à l'aide de connaissances générées aléatoirement à partir de jeux de données pour lesquels l'information de classe est connue, ou bien les connaissances sont supposées fournies manuellement. Si la première approche s'avère inutilisable en pratique (puisque les labels de classes sont inconnus), la seconde nécessite un effort d'annotation parfois important et, dans les deux cas, si les questions posées à l'expert ne sont pas ciblées, peuvent induire des erreurs de classification [190, 82, 149]. Les travaux présentés dans cette section reposent sur un modèle d'apprentissage actif dans lequel les algorithmes guident l'expert vers les connaissances les plus pertinentes pour les algorithmes de clustering semi-supervisé, tout en minimisant l'effort d'annotation.

4.2.1 Sélection active de contraintes

Peu de méthodes ont été décrites dans la littérature pour collecter les contraintes dans le cadre d'un processus actif dans lequel l'utilisateur est sollicité pour étiqueter des contraintes candidates supposées pertinentes. Dans [59] les auteurs proposent la méthode *Farthest First Query Selection* (FFQS), adaptée aux algorithmes de type K-Means, qui fonctionne en deux phases :

- ▷ l'*exploration* vise à construire un ensemble de contraintes CL appelé "squelette". Celui-ci est défini de telle sorte qu'à chaque itération on considère le point le plus éloigné du squelette comme contrainte CL. Les auteurs font l'hypothèse forte qu'à la fin de la phase un point de chaque cluster soit représenté dans le squelette ;
- ▷ la *consolidation* vise à sélectionner des points du jeu de données au hasard (et qui n'appartiennent pas au squelette) et de solliciter l'utilisateur jusqu'à ce que celui-ci souhaite arrêter. Cette phase, coûteuse en interactions avec l'utilisateur a été améliorée dans l'approche MMFFQS [149] dans laquelle les points dont la distance minimale avec le squelette est maximale, sont considérés en premier pour générer les questions à destination de l'utilisateur.

Nous proposons une approche de sélection active de contraintes adaptée à la recherche de clusters non nécessairement sphériques et qui repose sur un graphe des k plus proches voisins (k-NNG).

Construction du k-NNG et mesure d'utilité d'une contrainte : le k-NNG est défini comme un graphe non orienté pondéré dans lequel chaque sommet représente un point du jeu de données $\mathcal{X}$ et possède au maximum k voisins. Une arête est créée entre les points x_i et x_j si et seulement si x_i et x_j appartiennent respectivement à leur ensemble de k plus proches voisins. Le poids ω associé à chaque arête est défini comme le nombre de voisins communs que partagent x_i et x_j [122] :

$$\omega(x_i, x_j) = |NN_k(x_i) \cap NN_k(x_j)| \tag{4.1}$$

où $NN_k(x)$ désigne l'ensemble des k plus proches voisins de x. À partir du k-NNG, il est possible d'estimer une mesure de densité locale LDS [139] autour d'un point x comme la moyenne du nombre de voisins en commun avec ses voisins les plus proches :

$$LDS(x) = \frac{1}{k} \sum_{x' \in NN_k(x)} \omega(x, x') \tag{4.2}$$

Une valeur de LDS élevée indique une association forte de x avec ses voisins et donc que x est dans une région dense (un cluster). Par opposition, lorsque la valeur de LDS est basse, x est soit dans une région de transition entre deux clusters à densité plus faible, soit un point aberrant.

Cette mesure peut donc être utilisée pour caractériser l'utilité d'une contrainte. En effet il semble plus intéressant de poser des questions à l'utilisateur dans les cas où l'incertitude d'affectation au cluster est maximale, c'est-à-dire à la frontière entre plusieurs groupes. Nous proposons à cet effet une première mesure ASC pour évaluer l'utilité de formuler une contrainte entre deux points x_i et x_j :

$$ASC(x_i, x_j) = k - \omega(x_i, x_j) + \frac{1}{1 + \min(LDS(x_i), LDS(x_j))} \tag{4.3}$$

Cette mesure dépend en premier lieu du poids $\omega(x_i, x_j) \in [0, k]$ pour s'intéresser préférentiellement au points qui ne sont pas dans le même voisinage et en second lieu des densités locales de chacun des points $LDS(.) \in [0, 1]$ de telle sorte qu'au moins un des points soit dans une zone peu dense. D'autres mesures possédant les mêmes propriétés mais reposant sur des schémas de pondération différents auraient pu être envisagées.

Identifier les contraintes candidates : la détermination des contraintes candidates est à la base du mécanisme de génération des questions posées à l'utilisateur. Dans nos travaux, nous proposons de définir l'ensemble des contraintes candidates C_c à partir de l'ensemble A_k des arêtes du k-NNG comme suit :

$$C_c = \{(x_i, x_j) \in A_k | \omega(x_i, x_j) < \theta\} \tag{4.4}$$

À partir de cet ensemble C_c deux méthodes ont été proposées pour sélectionner les contraintes qui servent de base aux questions : soit les contraintes sont choisies aléatoirement dans l'ensemble C_c, soit les contraintes candidates sont classées par ordre décroissant de leur valeur ASC et dans ce cas, l'ordre des questions suit celui des contraintes candidates.

Algorithme d'apprentissage actif pour la collecte des contraintes : l'algorithme d'apprentissage actif construit initialement l'ensemble des contraintes candidates et applique ensuite itérativement l'une des deux alternatives décrites précédemment (choix aléatoire ou en fonction du score ASC). Pour chaque question proposée, l'utilisateur peut répondre ML, CL ou "*Je ne sais pas*". Les réponses sont conservées dans l'ensemble $\mathcal{Y}$ indexé par des couples de données et qui indique pour chacun le type de contrainte retenu par l'utilisateur. La particularité de notre approche réside dans le fait qu'après chaque réponse de l'utilisateur, notre algorithme cherche à utiliser la nouvelle information pour découvrir automatiquement de nouvelles contraintes dans l'ensemble C_c. Ce mécanisme de propagation repose sur la définition d'un chemin fort dans un k-NNG.

Définition 4.2.1.1 *Un chemin entre deux sommets x_i et x_j du k-NNG est dit fort ($CF(x_i, x_j)$) si et seulement si $\exists$ une séquence de sommets $(z_1, z_2, \ldots, z_t)$ — $x_i = z_1$ et $x_j = z_t$ et $\forall k \in [1, t-1]$: $\omega(z_k, z_{k+1}) \geq \theta$ ou qu'il existe une contrainte $ML(z_k, z_{k+1})$.*

Quatre règles sont ensuite utilisées pour réaliser la propagation :

1. $ML(u, v) \wedge ML(v, w) \Rightarrow ML(u, v)$
2. $ML(u, v) \wedge CL(v, w) \Rightarrow CL(u, v)$
3. $CL(u, v) \wedge CF(u, t) \wedge CF(v, l) \Rightarrow CL(t, l)$
4. $ML(u, v) \wedge CF(u, t) \wedge CF(v, l) \Rightarrow ML(t, l)$

Enfin une dernière procédure est utilisée pour raffiner l'ensemble des contraintes candidates C_c en supprimant tous les couples de sommets qui sont reliés par un chemin fort. Cette dernière procédure permet de réduire sensiblement la taille de l'ensemble des candidats et permet donc de ne pas trop solliciter l'utilisateur.

La figure 4.1 rapporte certains résultats expérimentaux obtenus par comparaison de notre approche ASC avec d'une part les contraintes candidates issues du GkPPV sélectionnées aléatoirement

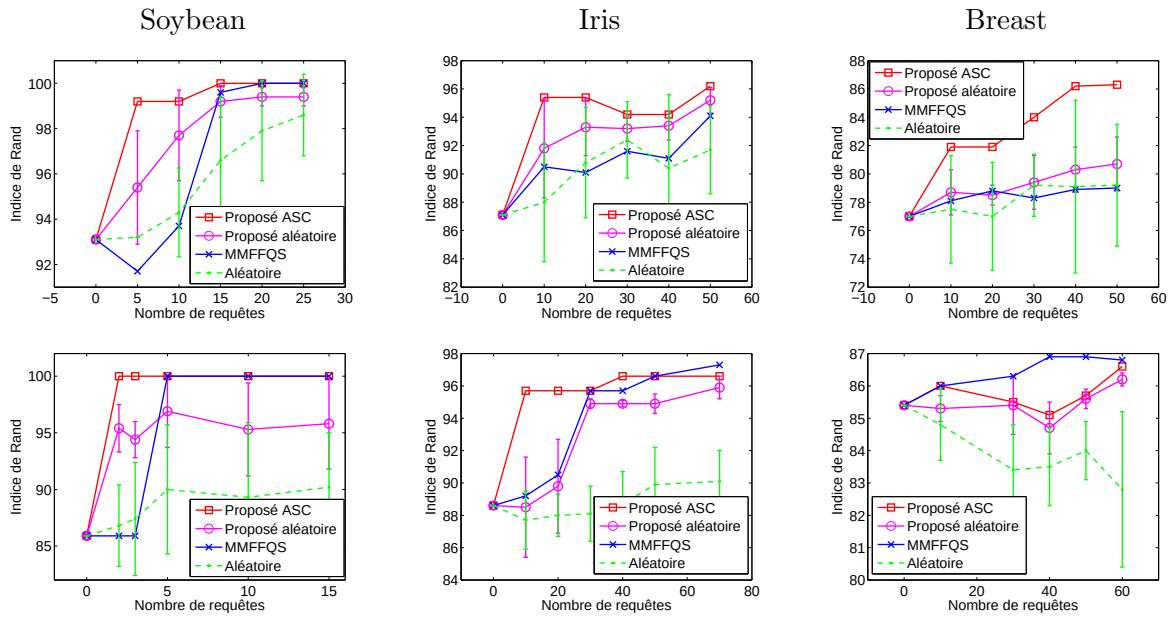

FIGURE 4.1 – Comparaison des approches de sélection de contraintes en terme d'erreur de Rand [171] pour les algorithmes AHCC [80] (Haut) et MPC-KMeans [69] (Bas) sur 3 jeux de données réelles.

("Proposé aléatoire") et d'autre part la méthode MMFFQS [149] et une sélection purement aléatoire des contraintes ("Aléatoire"). Les tests ont été conduits avec les algorithmes AHCC [80] et MPC-KMeans [69] sur des jeux de données issus du répertoire UCI Machine Learning Repository [55] et décrits dans la section 7.3 de ce mémoire.

Nos approches obtiennent de meilleurs résultats en général que la méthode MMFFQS lorsqu'elle est couplée avec l'algorithme hiérarchique. Dans le cas de l'algorithme MPC-KMeans, notre approche obtient de meilleures performances que la méthode MMFFQS qui lui est destinée lorsque le nombre de requêtes utilisateurs est faible. Ces études ont donné lieu à plusieurs publications [9, 8, 2] qui montrent également la pertinence du mécanisme de propagation pour diminuer le nombre de questions posées à l'utilisateur.

4.2.2 Sélection active de données étiquetées

Nous nous sommes également intéressés au problème de la sélection efficace de données étiquetées (*seeds*) pour des algorithmes de clustering semi-supervisés comme la variante des k-means nommée Seed k-means décrite dans [60]. Nous proposons à cet effet 3 approches qui, comme précédemment, s'intègrent dans le cadre d'un algorithme actif dont le but est de solliciter l'utilisateur le moins possible tout en récupérant des graines qui couvrent l'ensemble des clusters.

La première approche, nommée *Min-Max*, repose sur la sélection de points racines à partir d'une méthode de type min-max qui sélectionne initialement un point au hasard comme première racine et définit ensuite chaque nouveau point racine y^* comme le point dont la distance minimale par rapport aux points racines déjà sélectionnés $\mathcal{Y}$ est maximale :

$$y^* = arg \max_{x \in \mathcal{X}} (\min_{y \in \mathcal{Y}} d(x, y)) \tag{4.5}$$

De telles méthodes ont déjà été utilisées pour initialiser les algorithmes classiques de type k-means. Notre proposition est légèrement différente car notre objectif est ici de générer des questions

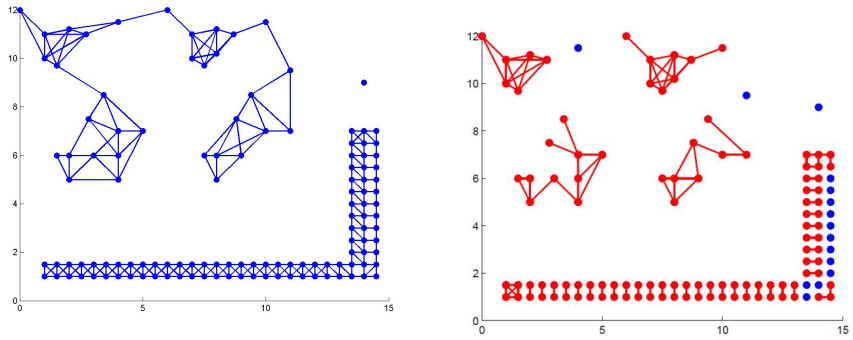

FIGURE 4.2 – Illustration du principe de l'approche de sélection de graines S-GkPPV. Gauche : exemple de graphe des k plus proches voisins. Droite : exemple des composantes connexes (en rouge) obtenues par filtrage des arcs du GkPPV (voir l'éq. 4.7).

à destination de l'utilisateur qui aura à charge de donner les étiquettes de clusters, ce qui nous permet de nous affranchir de l'hypothèse selon laquelle l'approche min-max sélectionne obligatoirement des points issus de clusters différents.

L'approche min-max précédente est sensible aux points aberrants qui généralement maximisent leur distance minimale avec les autres points plus "centraux" dans le jeu de données. Nous proposons dans notre seconde approche nommée *Min-Max-D*, d'utiliser un graphe des k plus proches voisins et de construire un ensemble de racines candidates. Celui-ci doit être formé de points éloignés comme dans l'approche min-max mais situés à l'intérieur des clusters et non pas en bordure ou complètement à l'extérieur. Pour cela nous utilisons la notion de densité locale LDS (voir éq. 4.2) de façon à filtrer l'ensemble initial de points $\mathcal{X}$. L'ensemble de points $\mathcal{X}_\epsilon$ résultant contient des points qui appartiennent exclusivement à des zones dont la densité est supérieure ou égale à un seuil ϵ. La relation de sélection des points de l'approche *Min-Max-D* est donc similaire à celle de la méthode *Min-Max* en remplaçant l'ensemble $\mathcal{X}$ par l'ensemble $\mathcal{X}_\epsilon$ dans l'équation 4.5.

$$\mathcal{X}_\epsilon = \{p \in \mathcal{X} | LDS(p) \geq \epsilon\} \tag{4.6}$$

où ϵ est un seuil de densité.

La troisième méthode nommée $S-GkPPV$ repose sur le graphe des k plus proches voisins pour déterminer des régions denses dans les données. L'ensemble des régions denses $\mathcal{X}_\delta$ est défini comme l'ensemble des composantes connexes obtenues après filtrage du GkPPV issu de $\mathcal{X}$ en fonction du poids ω de ses arcs selon un seuil δ de connectivité (voir l'éq. 4.7 et la figure 4.2).

$$\mathcal{X}_\delta = \{u \in \mathcal{X}, \exists v | \omega(u,v) \geq \omega\} \tag{4.7}$$

Les composantes connexes de $\mathcal{X}_\delta$ sont ensuite ordonnées par ordre décroissant de leur cardinalité. Dans chaque composante connexe en partant de la plus grande, une graine est aléatoirement sélectionnée pour pour formuler une requête à l'expert. Enfin, l'étiquette de classe obtenue est propagée à l'ensemble de la composante connexe.

Des expérimentations [10] ont été conduites pour comparer nos 3 méthodes de sélection de graines avec une méthode aléatoire en prenant comme référentiel 2 algorithmes de clustering différents : Seed k-means [60] adapté aux clusters sphériques et SSDBSCAN [140] basé sur la densité et adapté à toutes formes de clusters. Les résultats présentés dans la figure 4.3-Haut montrent que l'approche S-GkPPV donne de meilleures performances en terme d'indice de Rand que les autres méthodes sur nos données de test. On remarque également que l'approche S-GkPPV donne de

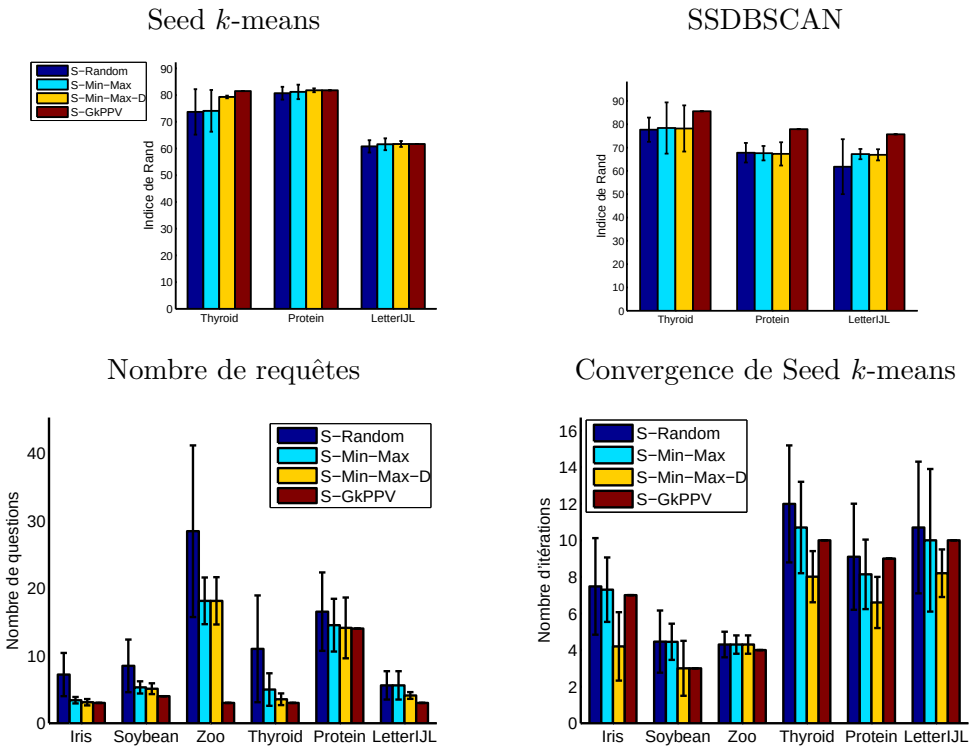

FIGURE 4.3 – Haut : résultats comparatifs en terme d'indice de Rand des approches de sélection de graines combinées aux algorithmes Seed k-means et SSDBSCAN. Bas : comparaison du nombre de questions posées à l'expert de façon à ce qu'il y ait au moins une graine par cluster et vitesse de convergence de Seed k-means en fonction de la méthode de sélection des graines.

meilleurs résultats que les autres approches lorsqu'elle est combinée avec SSDBSCAN, ce qui est cohérent car les deux méthodes reposent sur la notion de densité. D'autres tests rapportés dans la figure 4.3-Bas montrent également que l'approche S-GkPPV minimise le nombre de questions posées à l'expert de façon à ce qu'il y ait au moins une graine par cluster. En revanche, l'approche *Min-Max-D* permet de mieux améliorer la vitesse de convergence de l'approche Seed k-means que les autres, du fait de sa sélection des graines près des centres de clusters potentiels.

Les résultats expérimentaux montrent que nos méthodes permettent d'améliorer sensiblement les performances de l'algorithme Seed k-means par rapport à une sélection aléatoire des points racines. La méthode *Min-Max-D* donne de meilleurs résultats que l'approche *Min-Max* du fait du filtrage des points racines qu'elle réalise, mais est limitée par la complexité liée aux requêtes dans le graphe des k plus proches voisins qui est en $O(n^2)$ quand le nombre d'attributs est important. Le choix de l'une ou l'autre de nos méthodes pour déterminer les points racines dépend donc de la dimension des données et du temps dont on dispose.

4.3 Algorithmes de clustering semi-supervisé

4.3.1 Leader Ant avec des contraintes

Comme déjà indiqué, de nombreux algorithmes de clustering semi-supervisé ont déjà été proposés dans la littérature comme par exemple des variantes des k-means avec les algorithmes KMC [80] et $COP-Kmeans$ [192], des algorithmes hiérarchiques [81], des méthodes basées sur la densité

[176, 177], des modèles incrémentaux [83], des méthodes de type SVM [118] et des méthodes de co-clustering [165, 166]. La plupart de ces méthodes sont limitées aux contraintes de type ML et CL. Pourtant, selon [192] et [80], quatre principaux types de contraintes peuvent être définis :

- ▷ les contraintes de type **Must-Link** (ML) qui indiquent pour deux points x_i et $x_j \in \mathcal{X}$ s'ils doivent appartenir au même cluster,
- ▷ les contraintes de type **Cannot Link** (CL) qui indiquent que deux points x_i et $x_j \in \mathcal{X}$ ne doivent pas appartenir au même cluster,
- ▷ les contraintes de type **delta** (δ) qui indiquent, $\forall \delta > 0$, que la distance minimale D entre deux clusters C_i et C_j est au moins égale à δ :

$$\min_{\forall C_i, C_j \in \mathcal{P}, C_i \neq C_j} D(C_i, c_j) \geq \delta$$

- ▷ les contraintes de type **epsilon** (ϵ) qui indiquent $\forall C \in \mathcal{P}$ tel que le cluster C contient au moins deux points ($|C| \geq 2$) et pour tout point $x_i \in C$, qu'il doit toujours exister un point $x_j \in C$ tel que $d(x_i, x_j) \leq \epsilon$

Nous avons proposé dans [16] de nouvelles méthodes de clustering dérivées de l'algorithme Leader Ant [21] (voir la section 2.4 pour plus de détails) et reposant sur les contraintes : ML, CL et ϵ. Nous décrirons ci-après les principes de la méthodes MCLA qui intègre la gestion des contraintes ML et CL à l'algorithme Leader Ant (LA).

Intégration des contraintes ML dans LA : les contraintes ML sont utilisées pour construire des groupes initiaux avant l'exécution de l'algorithme LA traditionnel. Ces derniers vont permettre par la suite d'accélérer la convergence de la méthode. Les groupes initiaux sont déterminés comme les fermetures transitives obtenues lorsque l'on considère le jeu de données comme un ensemble de sommets d'un graphe et que les contraintes ML forment des arêtes entre certains des sommets.

Pendant la phase de construction des clusters de LA, l'affectation de la première données de chaque groupe initial, décide de l'affectation de tous les autres points. Cette heuristique un peu simple a été retenue pour conserver des temps de calcul modérés. Cependant, elle gagnerait probablement à être remplacée à terme par un système de vote sur l'ensemble des membres du groupe en fonction des nids existants au moment de l'affectation, de façon à éviter qu'une contrainte faussée n'entraîne une erreur de classification plus importante.

Intégration des contraintes CL dans LA : les contraintes CL sont utilisées lors de la phase d'affectation des fourmis artificielles au nid qui leur est le plus proche pour supprimer de la liste des nids candidats ceux qui contiennent une donnée avec laquelle la donnée considérée possède une contrainte CL.

Expérimentations et résultats : plusieurs tests ont été conduits avec les 3 approches précédentes dans [16]. Les résultats de l'approche $MCLA$ ont notamment été comparés à ceux de l'algorithme $COP - Kmeans$ présentés dans [189] comme le montre la figure 4.4. Quatre jeux de données issus du Machine Learning Repository sont considérés : Iris, Glass, Soybean et Ionosphere. Nos tests montrent que l'approche $MCLA$ donne de meilleurs résultats sauf pour le jeu de données Ionosphere où, en moyenne, $COP - Kmeans$ est légèrement plus performant. Des tests complémentaires devraient être conduits pour vérifier la validité statistique des écart observés.

4.3.2 Semi-Supervised Graph-based Clustering (SSGC)

L'algorithme Semi-Supervised Graph-based Clustering (SSGC) est un algorithme de clustering semi-supervisé qui repose sur la connaissance d'un petit nombre de données étiquetées (les graines)

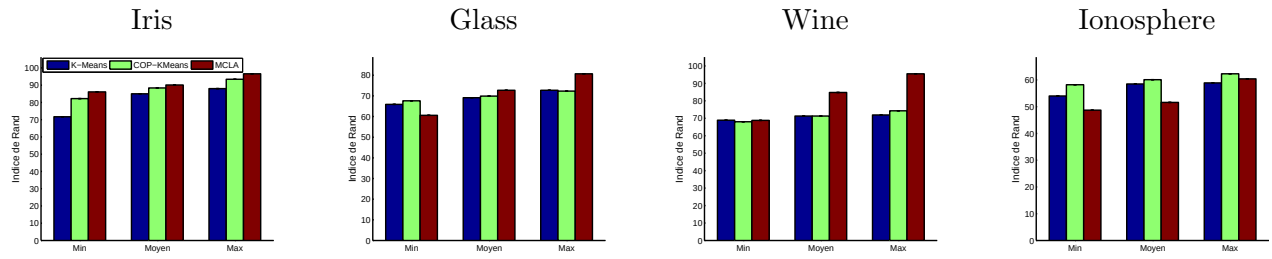

FIGURE 4.4 – Résultats comparatifs entre les algorithmes K-Means, COP-KMeans et MCLA en terme d'indice de Rand sur des jeux de données réelles.

pour produire la partition attendue. Comme dans les autres méthodes de clustering basées sur des données étiquetées, l'hypothèse est faite que l'on dispose d'au moins une graine pour chaque cluster que l'on souhaite apprendre.

La méthode SSGC repose ensuite sur la représentation du jeu de données sous la forme d'un graphe des k plus proches voisins (voir la section 4.2.1 pour une description détaillée de la construction du graphe). L'algorithme SSGC fonctionne en deux étapes principales : la phase de *partionnement* du GkPPV qui produit les clusters principaux et la phase de *raffinement* de la partition dans laquelle on construit les clusters finaux en supprimant le bruit.

Dans la première phase, l'objectif consiste à partitionner le GkPPV en composantes connexes de façon à construire un ensemble de *clusters principaux*. Ce partitionnement s'effectue de telle sorte que chaque composante connexe ne contienne que des graines avec la même étiquette de classe. Pour ce faire, un seuil γ initialement fixé à 0 va être itérativement incrémenté de 1 et utilisé pour filtrer les arcs du GkPPV en fonction de leur poids ω. Les arêtes (u, v) supprimées sont donc celles qui vérifient la condition *suppr* suivante :

$$suppr(u, v) = \omega(u, v) < \gamma \tag{4.8}$$

Enfin, dès lors que les composantes connexes ne contenant qu'un type d'étiquette de graines sont construites, l'algorithme de division s'arrête et propage les étiquettes à l'ensemble des points de chaque composante connexe. Ces composantes connexes étiquetées forment les clusters principaux.

Dans la seconde phase, les points restants sont divisés en 2 familles :
- les points qui possèdent des arêtes relatives avec un ou plusieurs clusters principaux prennent l'étiquette du cluster principal avec lequel l'arc a le poids ω le plus grand ;
- les points qui ne possèdent aucune arête avec les clusters principaux, sont dits isolés. En fonction de l'application, ils peuvent être considérés comme du bruit et supprimés, ou bien rattachés à un cluster principal en considérant l'étiquette majoritaire dans l'ensemble de leurs k plus proches voisins.

Des expérimentations comparatives entre l'algorithme SSGC et l'algorithme SSDBSCAN, rapportées dans la figure 4.5 montrent que notre nouvelle approche obtient des résultats comparables voire meilleurs que SSDBSCAN (des résultats plus complets, non présentés ici, montre que SSGC obtient de meilleurs résultats que SSDBSCAN pour 5 des 8 jeux de données considérés). De plus SSGC détermine le paramètre γ relatif à la densité automatiquement en fonction des graines qui sont fournies et ne nécessite donc aucun paramétrage. Enfin l'approche SSGC est plus rapide que l'approche SSDBSCAN du fait de sa complexité : SSGC est en $O(n^2)$ du fait du calcul du graphe des k plus proches voisins (sans optimisation) alors que la complexité de SSDBSCAN est en $O(mn^2 \log n)$.

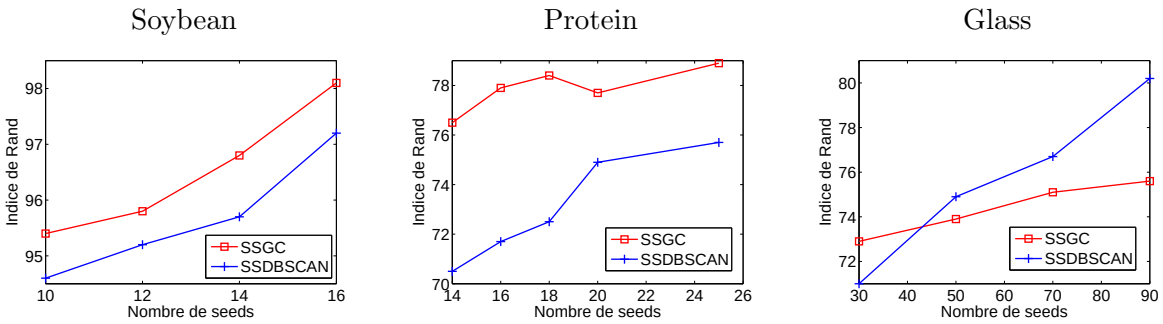

FIGURE 4.5 – Résultats comparatifs en terme d'indice de Rand entre les algorithmes SSGC et SSDBSCAN pour 3 jeux de données réelles.

4.4 Conclusion et perspectives

Nous avons proposé dans la thèse de Viet-Vu Vu des travaux visant à répondre à deux questions fondamentales du domaine du clustering semi-supervisé : (1) comment améliorer le processus de récupération des connaissances expertes de telle sorte que celles-ci améliorent les performances des algorithmes tout en minimisant l'effort d'annotation et, (2) comment fournir des algorithmes adaptés à tous types de données et possédant le moins de paramètres possibles pour les appliquer dans des cas d'usage réels ?

Pour ce faire, nous avons proposé des algorithmes actifs de sélection de contraintes ou de graines pertinentes pour tous types d'algorithmes de clustering et qui intègrent des mécanismes de propagation pour minimiser l'effort d'annotation. Les expérimentations conduites montrent que nos approches de sélection de contraintes obtiennent de meilleurs résultats que les approches de référence MMFFQS et FFQS. De même, nos approches de sélection de graines améliorent également sensiblement les performances des algorithmes Seed-KMeans et SSDBSCAN. Enfin, deux algorithmes de clustering ont été proposés pour traiter tous types de données et possédant peu (voire pas) de paramètres : d'une part l'algorithme Leader Ant avec des contraintes qui est compétitif face à l'algorithme COP-KMeans et d'autre part, l'algorithme SSGC qui est généralement meilleur que SSDBSCAN en terme d'indice de Rand et en complexité temporelle.

Cependant, l'ensemble de ces bons résultats est conditionné par la qualité de l'implémentation des requêtes de k plus proches voisins (kPPV) et par la dimension des données manipulées. Dans le cas de jeux de données avec de nombreux attributs, la résolution des requêtes de type kPPV possède une complexité quadratique avec le nombre de données n qui peut être préjudiciable dans de nombreux cas d'usages. En conséquence, il pourrait être intéressant d'étudier des méthodes approchées pour la recherche des k plus proches voisins, susceptibles de permettre la construction du GkPPV plus rapidement au prix d'une erreur modérée.

Enfin, des travaux sur la base d'image CalTech ont été conduits dans le cadre de la thèse de Viet-Vu VU et ont montré la capacité de nos méthodes à traiter des données réelles. Il serait intéressant à l'avenir d'évaluer nos méthodes semi-supervisées dans le cadre de l'analyse des usages sur Internet.

Chapitre 5

Extraction automatique de métadonnées

5.1 Introduction

D'après [144], les métadonnées - littéralement des données à propos des données - visent en premier lieu à améliorer le processus de recherche d'information, grâce la mise en place de moteurs de recherche plus précis et qui facilitent l'accès des utilisateurs aux très grandes quantités de données numériques actuellement disponibles. En second lieu, comme indiqué par [95], les métadonnées permettent une réutilisation facilitée des contenus. Ensuite, les métadonnées revêtent un intérêt crucial dans le cadre des entrepôts de données et plus largement pour le web sémantique, qui est supposé fournir aux machines les moyens de comprendre (au moins partiellement) le sens des documents qu'elles manipulent [64]. Enfin, on peut également considérer que l'enrichissement sémantique des ressources permet de mieux décrire les comportements, les préférences et les besoins d'information de leurs utilisateurs, à l'image de ce qui a été initié dans le projet DoXa par exemple (voir la section 7.4.3).

Le problème est que les métadonnées sont pour le moment souvent absentes des contenus publiés sur Internet. Par exemple, dans le cadre de l'éducation et du e-learning, un grand nombre de ressources pédagogiques a été créé et celles-ci ont été en partie étiquetées manuellement afin d'en permettre une indexation efficace en suivant des schémas de métadonnées adaptés (par exemple LOM-fr [1]). Malheureusement, l'étiquetage manuel des documents est sujet à la variabilité individuelle des annotateurs (utilisation de vocabulaires différents pour annoter la même chose) et dans tous les cas n'est pas suffisant pour couvrir l'ensemble des besoins d'annotations. En conséquence, le développement de méthodes capables d'apprendre automatiquement des métadonnées à partir de corpus HTML issus d'Internet est un problème de première importance.

Les travaux conduits durant la thèse de Sahar Changuel [42] apportent une réponse à ces problèmes et s'articulent autour de deux contributions principales. Nous avons en premier lieu proposé des approches indépendantes du contenu qui visent à déterminer s'il est possible de découvrir des relations entre les champs de métadonnées et ainsi compléter des valeurs manquantes à partir de celles qui sont connues. Nous nous sommes ensuite intéressés à des approches basées sur le contenu pour l'apprentissage automatique de métadonnées plus spécifiques du titre, de l'auteur ainsi que pour caractériser les relations de pré-requis entre les concepts d'un document pédagogique.

Ce chapitre est organisé comme suit : la section 5.2 présente succinctement les familles de

1. `http://www.lom-fr.fr/`

méthodes pour l'extraction de métadonnées, la section 5.3 détaille nos approches d'extraction indépendantes du contenu et la section 5.4 celles qui reposent sur une analyse du contenu des documents.

5.2 Méthodes d'extraction de métadonnées

D'après [96], l'extraction automatique de métadonnées est un domaine de recherche en pleine expansion qui peut se subdiviser en deux catégories de méthodes : les méthodes de récolte (*metadata harvesting*) et les méthodes d'extraction (*metadata extraction*).

La récolte de métadonnées correspond à la récupération automatique de métadonnées à partir de champs prédéfinis et dont les valeurs sont issues soit d'un étiquetage manuel, soit d'un processus semi-automatique [94]. Par exemple, les fichiers de bureautique contiennent généralement un auteur, une date de création / modification, un format ... De plus certains formats de fichiers comme le `HTML` autorisent la description de métadonnées dans des balises `<meta>`. La limite des méthodes de récolte est qu'elles ne peuvent récupérer l'information que si elle est explicitement disponible dans le document dans un format prédéterminé.

À l'inverse, les méthodes d'extraction reposent sur une analyse du contenu du document pour récupérer l'information de métadonnées. Plusieurs méthodes ont été envisagées dans ce cadre et notamment l'utilisation d'expressions rationnelles [187], des interprètes à base de règles [180] ou enfin des méthodes d'apprentissage [102].

Nos travaux présentés dans les sections suivantes s'inscrivent dans ce dernier cadre et proposent en complément des méthodes pour la construction de jeux étiquetés pour l'apprentissage des modèles de classification.

5.3 Relations entre champs de métadonnées

Comme indiqué par [123], l'extraction automatique de métadonnées depuis des contenus (texte, image ...) est extrêmement difficile et coûteuse. Nous avons donc proposé dans une première étude de voir dans quelle mesure il est possible d'apprendre des métadonnées à partir de celles déjà présentes. Pour cela, deux approches ont été utilisées [6] : une approche basée sur un problème de classification et une autre basée sur la recherche de règles d'associations.

De façon à évaluer la pertinence de nos méthodes deux entrepôts de données orientés e-learning ont principalement été utilisés : Ariadne [86] qui contient des collections de documents variés (documents, journaux, articles ...) et Ilumina [2] qui regroupe des cours du secondaire concernant différents enseignements scientifiques. L'intérêt de ces corpus provient également du fait que les ressources décrites sont variées (texte, image, vidéo). Au total, notre corpus Ariadne contient 4773 fichiers de métadonnées et notre corpus Ilumina 8563.

5.3.1 Classification de métadonnées

Le problème de la production de métadonnées est ici vu comme un problème de classification supervisée dans lequel la valeur de chaque attribut est apprise en fonction des valeurs des autres attributs. Comme plusieurs valeurs sont possibles pour chaque attribut, nous utilisons une stratégie "un contre tous" qui permet de représenter un problème multi-classes avec C classes en C problèmes de classification binaire dans lesquels chaque valeur possible de classe est apprise séparement de

2. http ://www.ilumina-dlib.org/

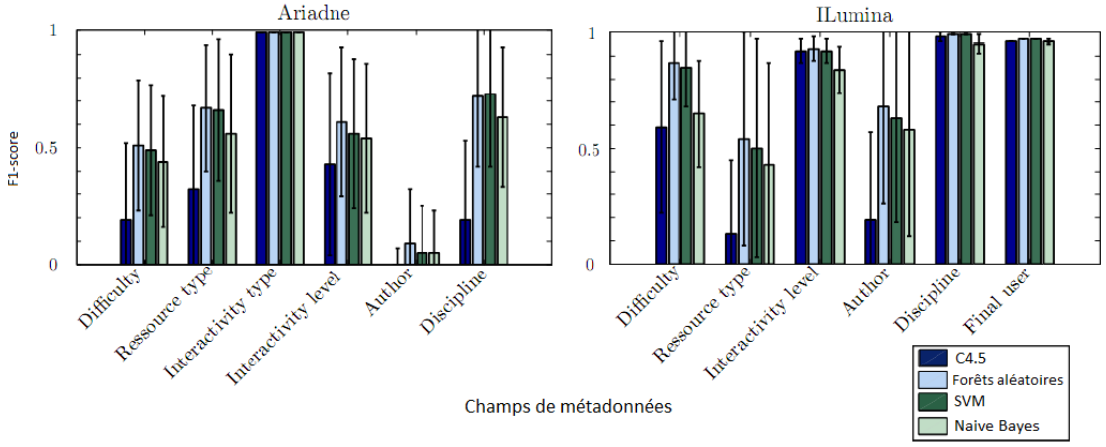

FIGURE 5.1 – Résultats obtenus en terme de F1-score par l'ensemble des algorithmes de classification évalués sur les deux corpus pour chaque champ de métadonnées.

toutes les autres classes. Plusieurs classifieurs ont été comparés : Naive Bayes [125], C4.5 [170], les forêts aléatoires [70] et les SVM [188]. La figure 5.1 présente les résultats comparatifs en terme de F1-score sur les deux corpus. Il en ressort que les forêts aléatoires obtiennent les meilleures performances avec les SVM. Cette méthode permet l'extraction de champs difficiles à apprendre à partir du seul contenu (comme le type d'interactivité ou l'utilisateur final). D'autres champs (comme l'auteur) sont plus difficiles à prédire (du fait du nombre de valeurs possibles) et justifient les méthodes basées sur le contenu que nous proposons par ailleurs (voir la section 5.4).

5.3.2 Génération de règles d'association

Nous avons également modélisé le problème de prédiction de valeurs de champs de métadonnées comme un problème de recherche de règles d'associations de la forme $A \Rightarrow B$ où A peut être une conjonction d'items et B est réduit à un seul item. Nous proposons pour cela d'utiliser l'algorithme FP-Growth [103] couplé à un mécanisme d'élagage des règles basé sur un calcul de score du χ^2 entre chaque règle et ses ancêtres (l'ensemble des règles moins spécifiques partageant la même conclusion) [145]. Nous avons pour cela proposé une optimisation calculatoire pour l'algorithme FP-Growth de façon à pouvoir déterminer plus rapidement le score $\chi^2(R_1, R_2)$ entre deux règles R_1 et R_2 en fonction des valeurs de support ($supp$) et de confiance ($conf$) :

$$\chi^2(R_2, R_1) = n \cdot supp(R_2) \frac{[conf(R_1) - supp(R_2) \cdot conf(R_1)]^2}{conf(R_2) \cdot conf(R_1)} \tag{5.1}$$

Les résultats obtenus ont montré que, plus que les valeurs manquantes, le bruit représenté par la surreprésentation de certaines valeurs et la présence de valeurs vides de sens (par exemple : interactivité "moyenne") pénalisent la découverte de règles utiles. Nous constatons cependant que les règles générées, après élagage, permettent d'annoter certains champs pour lesquels il est difficile de trouver automatiquement les valeurs à partir du contenu de la ressource. La méthode d'élagage simplifie l'interprétation des résultats en diminuant sensiblement le nombre de règles (corpus Ariadne de 90 à 25 règles et corpus Ilumina 754 à 55 règles). Enfin, les règles découvertes ont été utilisées dans un prototype de système de production de métadonnées.

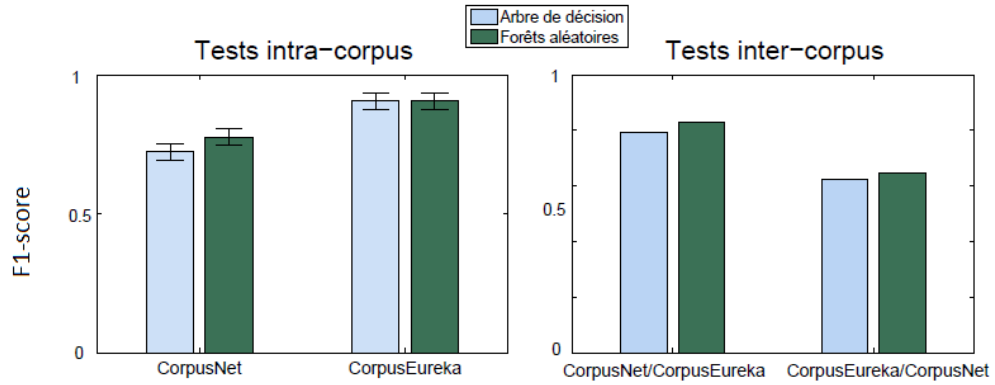

FIGURE 5.2 – Résultats comparatifs en terme de F1-score des deux algorithmes d'apprentissage C4.5 et forêts aléatoires pour l'apprentissage du titre au format texte sur deux corpus *CorpusNet* et *CorpusEureka*.

5.4 Extraction de métadonnées à partir du contenu

5.4.1 Extraction du titre et de l'auteur

Extraction du titre au format texte ou image à partir d'informations stylistiques : le titre est une des métadonnées les plus importantes dans le cadre de la recherche d'information [137, 94]. La méthode proposée se décompose en 3 principales étapes :

1. la construction d'un corpus de documents étiquetés pour apprendre le modèle de classification,
2. la construction des descripteurs,
3. et l'application de techniques d'apprentissage supervisé

L'ensemble d'apprentissage est construit à partir de pages téléchargées depuis Internet sélectionnées sur la base de mot-clés relatifs à l'éducation et qui possèdent un champ meta titre rempli. Pour l'apprentissage de titres "texte", l'ensemble d'apprentissage est construit en extrayant les 20 premiers textes de chaque page et en calculant l'intersection de chacun de ces textes avec le titre théorique de la page. Le titre qui possède le plus grand score (et la plus grande taille de police) est étiqueté 1 et les autres textes -1. Similairement, pour la tâche d'apprentissage des titres au format "image", notre méthode se base sur le contenu textuel de l'attribut `alt` de chaque image (supposé obligatoire d'après les normes du W3C) pour définir celle qui représente le titre. Plusieurs corpus sont construits : *CorpusNet*, *CorpusEureka* pour le texte et *NetImg* et *EurekaImg* pour les titres au format image. Le corpus Eureka [3] est issu du portail éducatif en ligne Eureka offrant des ressources décrites par des métadonnées au format `XML`.

Des descripteurs sont ensuite construits pour représenter chaque type d'objet (titre "texte" ou "image") en fonction du style, de la mise en page et de l'écart entre les valeurs de l'exemple et celles de son voisinage ou dans la page. Deux algorithmes d'apprentissage sont comparés pour l'apprentissage du titre comme le montre la figure 5.2. Les résultats intra ou inter-corpus (un corpus utilisé pour apprendre le modèle et l'autre pour tester) montrent que notre modèle est généralisable et donne de bons résultats par rapport à notre première étude (section 5.3) ou aux méthodes existantes [137, 94]. Similairement, l'analyse des titre au format image à l'aide de notre approche permet d'obtenir des scores de mesure F1 d'environ 0.84 sur le corpus *NetImg* avec les forêts aléatoires [15].

3. http ://eureka.ntic.org/

Extraction de l'auteur à partir d'informations contextuelles : similairement au *titre*, nous nous sommes intéressés en second lieu à l'apprentissage de la métadonnée *auteur*. Notre première étude rapportée dans la section 5.3 a montré la difficulté de prédiction de ce champ à partir des autres métadonnées. Nous avons dans ce cadre tout d'abord proposé une méthode semi-supervisée pour la création d'un corpus d'apprentissage : un ensemble d'apprentissage initial de 100 documents a été créé manuellement, sur lequel un arbre de décision est entraîné à reconnaître les auteurs sur la base de descripteurs binaires spatiaux (position dans le texte) et contextuels (présence de mot-clés à proximité comme *created by*, *written by*, email). L'ensemble d'apprentissage est ensuite étendu par l'arbre de décision. Lorsque plusieurs occurrences du même nom propre apparaissent dans le même document, nous fusionnons leurs vecteurs descripteurs par disjonction des attributs binaires. Les tests conduits montrent que notre approche obtient des résultats significativement meilleurs que les méthodes de récolte basées sur les balises méta des documents HTML [14] : jusqu'à 0.85 de F1-score pour notre approche contre 0.25 pour la méthode basée sur les balises META.

5.4.2 Annotation et ordonnancement de documents pédagogiques

Enfin, les derniers travaux conduits se sont intéressés à la qualification des concepts contenus dans une ressource en distinguant ceux qui sont définis et ceux qui sont utilisés et qui sont donc des pré-requis [12, 7]. Deux méthodes ont été proposées dans la thèse de Sahar Changuel pour traiter ce problème à partir de l'extraction des concepts de chaque page : (1) une méthode à base de règles dérivée de l'algorithme $(LP)^2$ et, (2) une méthode d'apprentissage statistique utilisant l'algorithme SVM dans laquelle les concepts sont représentés par des descripteurs linguistiques et contextuels.

Dans ces travaux plusieurs problèmes théoriques sont abordés comme le déséquilibre des classes en apprentissage supervisé ou les occurrences multiples d'un même concept dans le même document. L'algorithme SVM donnant les meilleurs résultats [12, 7], il a été utilisé par la suite pour nourrir une méthode d'ordonnancement automatique des ressources pédagogiques en fonction des pré-requis. Deux méthodes ont été proposées et comparées pour ordonnancer les documents : la première classe les documents directement en fonction des relations de précédence induites par les définitions qui existent entre eux. Le score de chaque document est alors égal au nombre de documents qui en dépendent. La seconde méthode repose sur une représentation matricielle des dépendances entre documents. Dans ce second cas, 3 représentations alternatives sont proposées et deux fonctions de score sont proposées pour calculer le score de chaque document. Les résultats sont enfin évalués à partir du taux de Kendall calculé entre les ordonnancements de documents issus de nos méthodes et celui proposé sur le site web de l'enseignement. L'ensemble des outils développés durant la thèse a été intégré à une interface graphique d'aide à l'extraction de métadonnées.

5.5 Conclusion et perspectives

Les travaux conduits avec Sahar Changuel proposent des contributions dans le domaine de la génération de champs de métadonnées. Deux types d'approches ont été envisagées : des méthodes indépendantes du contenu qui définissent des règles ou des modèles de prédiction d'un champ à partir d'autres champs de métadonnées, ou bien des méthodes plus spécifiques qui reposent sur le contenu. Une autre contribution de cette thèse porte sur le problème plus général de la génération de corpus étiquetés pour la mise en œuvre de méthodes d'apprentissage supervisé. Ces travaux ont donné lieu à plusieurs publications internationales [6, 7, 12, 14, 15]. Il reste de nombreuses études qui peuvent être conduites notamment sur l'extraction de relations de précédence entre les documents (ordonnancement des réponses d'un moteur de recherche), sur la réalisation de méthodes incrémentales qui mettent à jour automatiquement les relations entre champs de métadonnées ou l'étude du passage à l'échelle des méthodes proposées. D'un point de vue analyse de traces, il serait également pertinent d'enrichir les parcours avec les métadonnées extraites.

Chapitre 6

Conclusions et perspectives

Conclusions

Ce mémoire détaille les principales contributions théoriques et applicatives de mes travaux de recherche dans le domaine de l'analyse des interactions utilisateurs selon 3 axes principaux.

En premier lieu, je me suis intéressé au problème de l'analyse des traces d'usage sur Internet qui vise à extraire des profils représentatifs des masses de sessions utilisateurs disponibles. Pour cela, les méthodes proposées reposent sur des algorithmes de classification non-supervisée pour créer des groupes d'utilisateurs exhibant des comportements similaires. Plusieurs algorithmes de clustering adaptés à l'analyse des traces ont été introduits : Leader Ant (LA) [21], Online Fuzzy C-Medoids (OFCMd) et History-Based Online Fuzzy C-Medoids (HOFCMd) [1, 11, 29]. Ces méthodes apportent une solution aux problèmes actuels du passage à l'échelle et du traitement de flux de données tout en conservant la capacité de traiter tous types de données. Ces algorithmes sont donc à même d'analyser des représentations complexes des interactions utilisateurs sous la forme de séquences d'usages, éventuellement enrichies par des métadonnées (relatives au site étudié ou bien traduisant des émotions et des opinions, voir le projet DoXa section 7.4.3). De nouvelles méthodes de similarité entre sessions ont également été proposées [18, 4, 20, 5].

Les problèmes liés au passage à l'échelle concernant la détermination et la mise à jour des représentants (médoides) des clusters ou encore la gestion des grands volumes de données ont été résolus en employant d'une part une sélection de données aléatoires et une stratégie incrémentale en une passe dans LA, et d'autre part, une sélection de candidats médoides à partir de leur valeur d'appartenance et un découpage des données en lots plus facilement manipulable en mémoire dans les approches OFCMd et HOFCMd. Enfin, les méthodes OFCMd et HOFCMd introduisent un mécanisme d'oubli qui permet de faire décroître l'influence d'une donnée au cours du temps, pour s'adapter plus spécifiquement au flux de données [1].

Lors de mes recherches, l'accent a également été mis sur le développement de chaînes complètes pour l'analyse des usages intégrant des mécanismes de prétraitement des données, d'analyse (à l'aide des algorithmes de clustering) et d'interprétation des résultats [19, 3]. Dans ce dernier cas, deux types de travaux ont été conduits : d'une part des méthodes de recherche de représentants pour résumer l'information d'un cluster (à l'aide des médoides ou de données typiques), et d'autre part des outils de visualisation interactive des parcours. Cette chaîne de traitement a été intégrée au logiciel *ILObTrack*, qui a été utilisé avec succès dans le cadre du projet Infom@gic (voir la section 7.4.2) et fait l'objet d'un dépôt d'invention entre le LIP6 et la société ILObjects.

En second lieu, je me suis intéressé au problème de clustering semi-supervisé qui suppose l'interaction d'un expert avec le système de classification pour fournir des connaissances sous la forme de contraintes ou de graines de clusters. Les travaux conduits dans la thèse de Viet-Vu Vu [43]

abordent deux aspects fondamentaux de ce problème : d'une part des algorithmes de clustering semi-supervisé capables de travailler avec tous types de données [16], et d'autre part des méthodes actives pour minimiser l'effort d'annotation de l'expert tout en améliorant sensiblement les performances en classification [2, 8, 9, 10].

Enfin, nous avons proposé des méthodes d'extraction automatique de métadonnées pour la recherche d'information dans les grandes masses de données dans le cadre de la thèse de Sahar Changuel [42]. Deux types de méthodes ont été proposées : d'une part des méthodes indépendantes du contenu [6], dans lesquelles la tâche d'extraction des métadonnées est vue comme un problème de classification ou de recherche de règles d'association, et, d'autre part, des méthodes basées sur le contenu qui combinent des méthodes d'apprentissage statistique avec des descripteurs stylistiques, textuels et/ou linguistiques pour extraire le titre [14], l'auteur [15] ou les concepts prérequis et définis dans le cadre du e-learning [7, 12].

L'ensemble de ces travaux appelle de nouvelles recherches dans le domaine de la classification de données d'une part, et dans le domaine de l'analyse des interactions utilisateurs et de l'interprétation des résultats d'analyse. Ces perspectives à moyen ou long terme, théoriques ou applicatives, sont détaillées dans la section suivante.

Perspectives

Les perspectives des travaux présentés dans ce mémoire peuvent être réparties selon deux axes : un axe théorique avec les recherches dans le domaine de la classification non-supervisée et un axe plus applicatif portant sur l'analyse des interactions utilisateurs.

Perspectives en classification non-supervisée

En premier lieu, les travaux conduits dans le cadre du traitement des flux de données pourraient être approfondis en réalisant des tests de robustesse plus poussés pour nos algorithmes OFCMd et HOFCMd. Il serait intéressant d'observer la capacité d'adaptation de nos méthodes, en fonction de leurs paramètres, à des distributions de données changeantes au cours du temps (apparition, suppression, dérive, recouvrement de clusters) similairement aux expérimentations qui peuvent être conduite dans l'outil MOA [135, 134]. Une autre piste de travail prometteuse concerne l'utilisation dans de nombreuses méthodes de la notion de *micro-cluster* qui est un résumé mis à jour continuellement pour un cluster. Les travaux actuels reposent sur le formalisme des *cluster features* issu de l'algorithme BIRCH [203] qui est limité aux données numériques. Des recherches pourraient être conduites pour voir comment maintenir à moindre coût une structure comparable pour des données non nécessairement numériques. Enfin, dans un souci d'interprétation des résultats, il serait intéressant de disposer de méthodes pour caractériser automatiquement l'information intéressante qui émerge du traitement des flux en se basant sur l'évolution des points aberrants ou la persistance des clusters observés dans le temps.

En second lieu, il est intéressant de remarquer que l'explosion des masses de données a naturellement dirigé les recherches vers les problèmes de passage à l'échelle et de traitement de flux. Je pense qu'il est pertinent de s'intéresser également à la richesse des clusters produits et leur capacité à être interprétés. Dans ce cadre, deux problèmes principaux méritent d'être étudiés. D'une part, il serait intéressant de poursuivre les recherches sur la détermination de points représentatifs pour résumer l'information propre à chaque groupe. Des travaux, dont les résultats sont prometteurs, ont été initiés sur la détermination de prototypes basée sur leur degré de typicalité. Ces méthodes restent toutefois difficilement exploitables en pratique du fait de leur complexité rédhibitoire. Il

conviendrait donc de chercher à améliorer ces méthodes en proposant des heuristiques qui garantissent une certaine qualité au prix d'une complexité sous-quadratique. D'autre part, il me paraît intéressant d'étudier les méthodes dites de subspace clustering qui visent à trouver à la fois les clusters et leurs sous-espaces d'attributs représentatifs. Dans un tel système, une donnée peut appartenir à différents clusters qui sont représentés dans des sous-espace différents. Ce type de travaux permettrait de mettre en lumière des groupes d'utilisateurs beaucoup plus discriminants et de faire apparaître, à la manière d'itemsets, des liens entre les attributs caractéristiques qui forment les sous-espaces.

Enfin, de nombreux problèmes restent ouverts dans le domaine du clustering semi-supervisé, comme par exemple la détermination du nombre optimal de requêtes qui doivent être posées à l'expert ou encore la définition de méthodes de clustering qui puissent tirer profit de graines ou de contraintes indifféremment [121]. Dans ce dernier cas, il pourrait être intéressant de disposer de méthodes de sélection de connaissance experte qui puissent estimer quel type de connaissances (contraintes ou graines) est le plus profitable à l'algorithme de clustering à chaque instant. Par exemple, dans le cas d'un algorithme de sélection de graines, il est probable que les premières informations concerneront des points au centre des clusters et que les requêtes suivantes porteront sur les frontières entre les groupes. D'un point de vue théorique, il pourrait également être opportun de lever l'hypothèse des méthodes de clustering basée sur les graines, qui impose de connaître une donnée étiquetée issue de chaque cluster, ce qui n'est généralement pas applicable dans des cas d'usage réels.

Analyse des interactions utilisateurs, interprétation et visualisation

Les interactions utilisateurs sont en train d'évoluer sous l'impulsion des nouveaux modes d'usage (comme les terminaux tactiles) ou le remplacement des interfaces logicielles classique par des interfaces dynamiques sur Internet. Désormais, la notion de page visitée n'existe plus et les menus sont contextuels. Il faut donc réfléchir à adapter et enrichir les modèles et les outils existants pour prendre en compte le contexte dans les analyses de masse des usages [78]. Des travaux ont déjà été proposés dans ce sens mais se heurtent à la richesse des traces enregistrées pour être exploitables. Les outils de visualisation doivent également évoluer pour prendre en compte cette complexité en intégrant des mécanismes de zoom contextuels qui représentent les interactions utilisateurs à différentes échelles (page, éléments d'une même page ...).

Une dernière perspective de travail concerne l'interprétation automatique des clusters. Nous avons conduits des travaux préliminaires dans ce domaine en projetant des mot-clés ou des métadonnées sur les parcours dans les données d'usage. Il est possible d'aller plus loin en représentant par exemple directement les parcours dans un espace sémantique pour caractériser les besoins d'informations des utilisateurs. Nous avons également abordé le problème de la génération automatique de résumés textuels explicatifs d'un cluster à l'aide de simples patrons de phrases. Ces travaux pourraient être étendus en proposant des méthodes d'apprentissage pour décider des informations à afficher en se reposant sur la typicalité des valeurs des attributs ou des parcours des clusters par exemple. Enfin, comme dit précédemment, le processus d'interprétation des usages gagnerait à reposer sur des analyses de type sub-space clustering, plus à même de faire ressortir des groupes d'utilisateurs pertinents.

Chapitre 7

Annexes

7.1 Table des notations

Cette section présente la table des principales notations utilisées pour décrire les algorithmes présentés dans ce mémoire.

Notations	**Explications**
$\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$	Jeu de données contenant n objets qui peuvent être indifféremment des données spatiales numériques, des données catégorielles ou relationnelles
$\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$	Ensemble des poids associés aux objets de l'ensemble $\mathcal{O}$
$\mathcal{P} = \{C_1, C_2, \ldots, C_k\}$	Partition composée de k clusters
$\mathcal{V} = \{v_1, v_2, \ldots, v_k\}$	Ensemble des centres des k clusters
d^2	Norme euclidienne

TABLE 7.1 – Notations utilisées dans les descriptions des algorithmes de classification non supervisée

7.2 Évaluation des méthodes de clustering

Il existe de nombreux travaux portant sur l'analyse de la qualité d'une partition produite par un algorithme de partitionnement [101]. Il existe par exemple l'indice de Davies–Bouldin [45] ou encore l'indice de Dunn [44] qui définissent des ratios entre la compacité et la séparabilité des clusters ou encore l'indice de Xie–Beni [46] plus spécifiquement dévolu au traitement des partitions floues. Nous décrivons dans cette section les indicateurs que nous avons utilisés pour valider et comparer nos approches à l'état de l'art ainsi que les jeux de données utilisés.

7.2.1 Indice de Rand

Le premier indicateur de performance retenu est l'indice de Rand [171] qui mesure l'appariement entre deux partitions à partir de $n \times (n-1)/2$ comparaisons deux à deux entre les n points qui composent le jeu de données $\mathcal{X}$. Soient deux partitions de $\mathcal{X}$ notées P_1 et P_2, et soient a le nombre de couples de points classés dans le même groupe dans les deux partitions et b le nombre de

couples de points classés dans des groupes différents dans les deux partitions. Une mesure d'erreur $\mathcal{E}_{rand}(P_1, P_2)$ basée sur l'indice de Rand peut être définie comme suit :

$$RI(P_1, P_2) = 1 - \frac{2 \times (a+b)}{n \times (n-1)} \tag{7.1}$$

où $RI \in [0, 1]$; $RI = 0$ quand les partitions sont identiques et 1 lorsqu'elles sont complètement différentes. Par définition, la mesure de Rand est sensible aux différences entre les nombres de groupes dans chaque partition. Dans nos résultats, nous considérerons parfois l'erreur de Rand (le plus bas score est le meilleur) ou bien l'indice de Rand (le plus haut score est le meilleur) en fonction des études qui ont été conduites.

7.2.2 Matrice de confusion

La matrice de confusion C est une matrice de dimensions $c \times k$ qui indique pour chacune des c classes théoriques d'un jeu de données $\mathcal{X}$, sa répartition dans les k clusters découverts par un algorithme de partitionnement. Autrement dit, chacune de ses valeurs $C(i, j)$ indique le nombre de données issues de la classe $i \in [1, c]$ qui apparaissent dans le cluster $j \in [1, k]$. Nous utilisons deux mesures comparables dérivées de cette matrice, qui considèrent pour chaque classe $i \in [1, c]$ que le nombre de points bien classés correspond au cluster découvert $j \in [1, k]$ qui est majoritaire pour cette classe.

La première mesure, nommée erreur de confusion, est définie comme suit :

$$Err(C) = 1 - (\frac{1}{n} \times \sum_{\forall i \in [1,c]} max_{\forall j \in [1,k]} C(i, j)) \tag{7.2}$$

La seconde mesure, nommée pureté, est utilisée dans les travaux de [72] pour l'évaluation des algorithmes de traitement de flux de données :

$$pur = 1 - \frac{1}{k} \times \sum_{j=1}^{k} \frac{|C_j^d|}{|C_j|} \tag{7.3}$$

où k est le nombre de clusters, $|C_j^d|$ le nombre de points issus de la classe dominante dans le cluster j et $|C_j|$ la taille du cluster j. Par la suite, *erreur de pureté* et *pureté* sont utilisés indifféremment.

7.2.3 F1-score

Nous utilisons également la mesure traditionnelle du F1-score telle que présentée dans [196] et utilisée dans l'outil *MOA* [134] :

$$F = \sum_{i \in [1,c]} \frac{C(i, .)}{n} \max_{j \in [1,k]} \frac{2C(i, j)}{C(., j) + C(i, .)} \tag{7.4}$$

où c désigne le nombre de classes de la partition théorique, $C(i, j)$ désigne le nombre d'objets du cluster j qui appartient à la classe originale i, $C(i, .)$ (resp. $C(., j)$) est le nombre d'objets dans la classe i (resp. le cluster j) et n indique le nombre total d'objets.

Compacité et séparabilité des clusters

Le dernier indicateur utilisé est comparable aux indices de Davies-Bouldin [45] et de Dunn [44] et est exprimé comme le ratio entre la compacité (ou distance intra-cluster) et la séparabilité (ou distance inter-cluster). La différence repose ici sur la définition de la compacité et de la séparabilité des clusters : la compacité D_{intra} est calculée comme la moyenne des distances entre les points

appartenant au même cluster et similairement la séparabilité D_{inter} est définie comme la moyenne des distances des points appartenant à des clusters différents comme le montrent les équations suivantes.

$$DI(P) = \frac{D_{intra}(P)}{D_{inter}(P)} \tag{7.5}$$

$$D_{intra}(P) = \frac{\sum_{\forall(i,j)\in\mathcal{X}^2, i<j} \delta_{c_i=c_j} \times D(i,j)}{\sum_{\forall(i,j)\in\mathcal{X}^2, i<j} \delta_{c_i=c_j}} \tag{7.6}$$

$$D_{inter}(P) = \frac{\sum_{\forall(i,j)\in\mathcal{X}^2, i<j} \delta_{c_i\neq c_j} \times D(i,j)}{\sum_{\forall(i,j)\in\mathcal{X}^2, i<j} \delta_{c_i\neq c_j}} \tag{7.7}$$

où c_i (resp. c_j) est le cluster du point i (resp. j) dans la partition P, δ est la fonction identité et $D(i,j)$ une mesure de distance définie dans l'espace des objets. Contrairement à l'indice de Rand, cette mesure ne vise pas à comparer deux partitions à partir des affectations réalisées mais à les différencier sur la base de la qualité du partitionnement produit. Cette mesure de qualité d'une partition est donc principalement destinée à valider nos approches sur des données réelles (comme les sessions d'usage sur Internet) pour lesquelles il n'existe pas d'étiquetage utilisable pour comparer la partition découverte à celle qui était attendue.

7.3 Jeux de données de test

Nous utilisons des jeux de données artificielles et réelles dans nos tests. Les données artificielles se nomment $Art_{1,2,3,4,5,6}$ et ont été générées selon des distributions de valeurs gaussiennes ou uniformes avec différentes difficultés (attributs inutiles, chevauchement entre groupes). Ces jeux de données dont la taille est variable nous ont permis d'étudier le paramétrage de nos algorithmes. Cependant, de façon à pouvoir évaluer la capacité des méthodes à traiter de plus grands jeux de données des versions étendues de certains jeux artificiels ont été produits selon des distributions de points identiques aux jeux $Art_{1,2,3,4,5,6}$. Les jeux de données résultants sont nommés : $LArt_1$, $LArt_2$, $LArt_5$ et $LArt_6$.

Les tables 7.2 et 7.3 détaillent les paramètres des distributions uniformes $\mathcal{U}$ ou normales $\mathcal{N}$ utilisées pour générer les jeux de données artificielles $Art_{1...6}$ et $LArt_1$, $LArt_2$, $LArt_5$ et $LArt_6$.

La figure 7.1 présente un aperçu des jeux de données artificielles représentés en 2 dimensions (3 dimensions pour Art_6).

Similairement aux données artificielles, des jeux de données classiques issus du UCI Machine Learning Repository [55] ont été utilisés pour nos tests et se répartissent également en deux groupes principaux en fonction du nombre de données à traiter. Les plus "petits" ont été utilisés pour évaluer les algorithmes non incrémentaux et notre approche Leader Ant alors que les plus grands ont été utilisés pour valider nos approches floues incrémentales. D'autres enfin ont été utilisés lors des travaux sur le clustering semi-supervisé présentés dans le chapitre 4. La table 7.4 présente pour chacun des jeux de données son nombre d'objets (n), sa dimensionalité (nombre d'attributs n_{att}) et le nombre de clusters k attendus.

Enfin, afin de pouvoir estimer la capacité de nos approches à traiter des flux de données, nous avons générés deux flux artificiels nommés $Stream_1$ et $Stream_2$. Ces deux jeux de données sont composés chacun de 10 sous jeux de données qui sont supposés être générés au cours du temps. Dans les deux cas, le premier sous-jeu à être proposé possède une distribution de points similaire à celle de Art_1. Puis les deux flux se différencient : $Stream_1$ correspond au cas le plus simple dans lequel les nouvelles données produites au cours du temps sont identiquement distribuées au premier sous jeu de données. Dans le flux $Stream_2$ une déviation sur la moyenne des lois normales

Données	**n**	**natt**	**k**	$\|k_i\|$	**Distributions**
Art_1	400	2	4	100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
Art_2	1000	2	2	500	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
				500	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
Art_3	1100	2	4	500	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
				50	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
				500	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				50	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
Art_4	200	2	2	100	$(\mathcal{U}([-1, 1]), \mathcal{U}([-10, 10]))$
				100	$(\mathcal{U}([2, 3]), \mathcal{U}([-10, 10]))$
Art_5	900	2	9	100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(1.4, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
				100	$(\mathcal{N}(1.4, 0.2), \mathcal{N}(0.8, 0.2))$
				100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(1.4, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(1.4, 0.2))$
				100	$(\mathcal{N}(1.4, 0.2), \mathcal{N}(1.4, 0.2))$
Art_6	400	8	4	100	$(\mathcal{N}(0.2, 0.2), \ldots, \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2), \ldots, \mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2), \ldots, \mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \ldots, \mathcal{N}(0.8, 0.2))$

TABLE 7.2 – Description des jeux de données artificielles avec pour chacun sa taille n, son nombre d'attributs *natt*, le nombre de clusters k et pour chacun des clusters son effectif $|k_i|$ et les distributions de points associées.

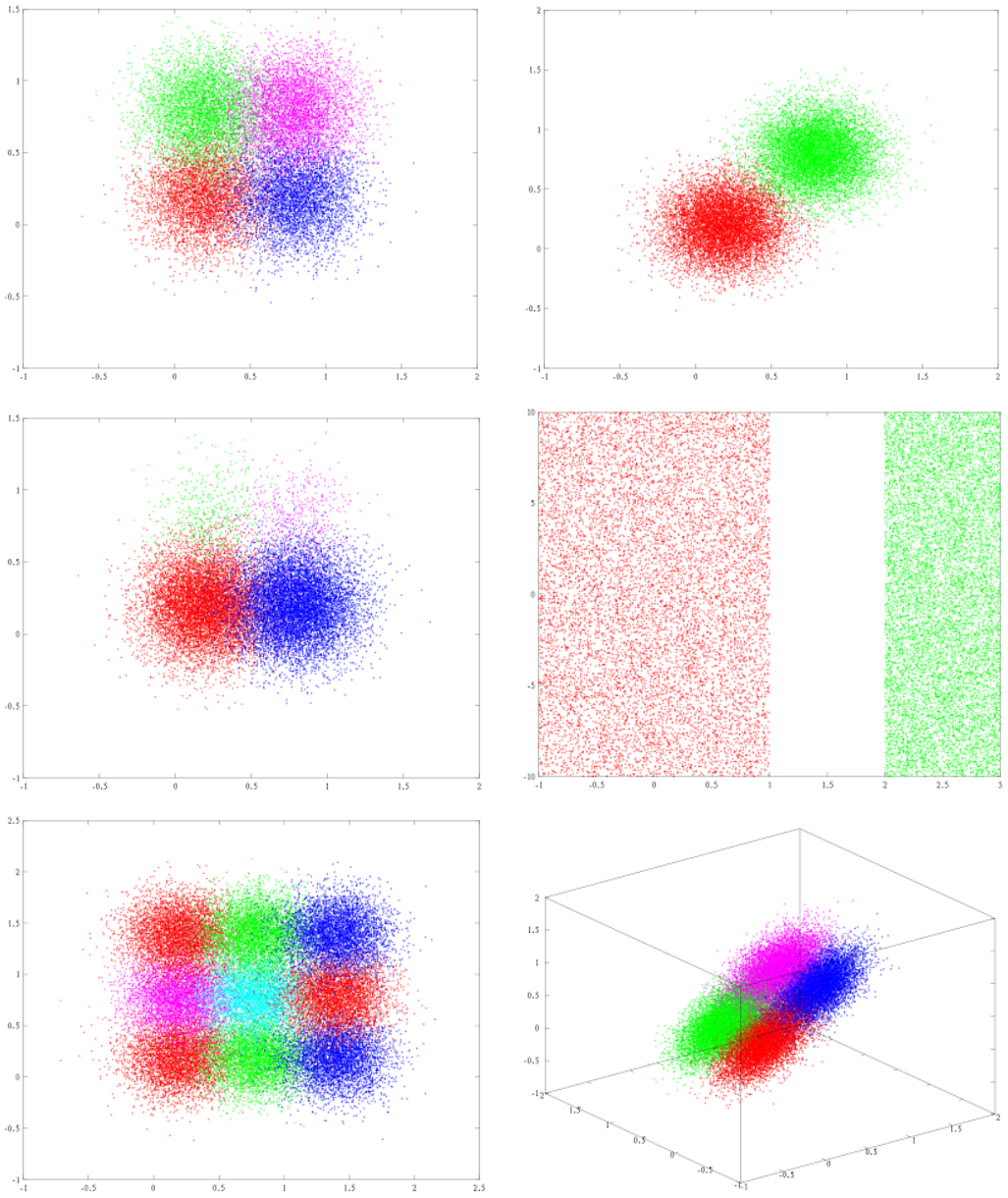

FIGURE 7.1 – Représentation des jeux de données : ligne haut Art_1 (gauche) et Art_2 (droite), milieu Art_3 (gauche) et Art_4 (droite) et bas Art_5 (gauche) et Art_6 (droite).

Données	n	natt	k	$\|k_i\|$	Distributions
L_{Art1}	20 000	2	4	5 000	$(\mathcal{N}(0.2,0.2),\mathcal{N}(0.2,0.2))$
				5 000	$(\mathcal{N}(0.2,0.2),\mathcal{N}(0.8,0.2))$
				5 000	$(\mathcal{N}(0.8,0.2),\mathcal{N}(0.2,0.2))$
				5 000	$(\mathcal{N}(0.8,0.2),\mathcal{N}(0.8,0.2))$
L_{Art2}	20 000	2	2	10 000	$(\mathcal{N}(0.2,0.2),\mathcal{N}(0.2,0.2))$
				10 000	$(\mathcal{N}(0.8,0.2),\mathcal{N}(0.8,0.2))$
L_{Art5}	45 000	2	9	5 000	$(\mathcal{N}(0.2,0.2),\mathcal{N}(0.2,0.2))$
				5 000	$(\mathcal{N}(0.8,0.2),\mathcal{N}(0.2,0.2))$
				5 000	$(\mathcal{N}(1.4,0.2),\mathcal{N}(0.2,0.2))$
				5 000	$(\mathcal{N}(0.2,0.2),\mathcal{N}(0.8,0.2))$
				5 000	$(\mathcal{N}(0.8,0.2),\mathcal{N}(0.8,0.2))$
				5 000	$(\mathcal{N}(1.4,0.2),\mathcal{N}(0.8,0.2))$
				5 000	$(\mathcal{N}(0.2,0.2),\mathcal{N}(1.4,0.2))$
				5 000	$(\mathcal{N}(0.8,0.2),\mathcal{N}(1.4,0.2))$
				5 000	$(\mathcal{N}(1.4,0.2),\mathcal{N}(1.4,0.2))$
L_{Art6}	40 000	8	4	10 000	$(\mathcal{N}(0.2,0.2),\ldots,\mathcal{N}(0.2,0.2))$
				10 000	$(\mathcal{N}(0.2,0.2),\mathcal{N}(0.8,0.2),\ldots,\mathcal{N}(0.2,0.2),\mathcal{N}(0.8,0.2))$
				10 000	$(\mathcal{N}(0.8,0.2),\mathcal{N}(0.2,0.2),\ldots,\mathcal{N}(0.8,0.2),\mathcal{N}(0.2,0.2))$
				10 000	$(\mathcal{N}(0.8,0.2),\ldots,\mathcal{N}(0.8,0.2))$

TABLE 7.3 – Description des grands jeux de données artificielles avec pour chacun sa taille n, son nombre d'attributs *natt*, le nombre de clusters k et pour chacun des clusters son effectif $|k_i|$ et les distributions de points associées.

Données	n	n_{att}	k
$Iris$	150	4	3
$Glass$	214	9	6
$Pima$	798	8	2
$Soybean$	47	35	4
$Thyroid$	215	5	3
$Wine$	177	13	3
$Breast$	569	30	2
$Protein$	116	20	6
$LetterIJL$	227	16	3
Zoo	101	16	7
$Ionosphere$	351	34	2
$Statlog_{segment}$	2310	19	7
$Statlog_{vehicule}$	846	18	4
$Waveform$	5000	21	3
$Statlog_{shuttle}$	14500	8	7
$Letter_{recognition}$	20000	16	26
$kddcup99$	494000	34	23

TABLE 7.4 – Caractéristiques principales des jeux de données réelles issues du répartoire UCI Machine Learning Repository [55].

Données	Nb. lots	Taille des lots	k	$\|k_i\|$	Distributions
$Stream_1$	10	400	4	100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.2, 0.2), \mathcal{N}(0.8, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.2, 0.2))$
				100	$(\mathcal{N}(0.8, 0.2), \mathcal{N}(0.8, 0.2))$
$Stream_2$	10	400	4	100	$(\mathcal{N}(0.2 + 0.1 * (j-1), 0.2), \mathcal{N}(0.2 + 0.1 * (j-1), 0.2))$
				100	$(\mathcal{N}(0.2 + 0.1 * (j-1), 0.2), \mathcal{N}(0.8 + 0.1 * (j-1), 0.2))$
				100	$(\mathcal{N}(0.8 + 0.1 * (j-1), 0.2), \mathcal{N}(0.2 + 0.1 * (j-1), 0.2))$
				100	$(\mathcal{N}(0.8 + 0.1 * (j-1), 0.2), \mathcal{N}(0.8 + 0.1 * (j-1), 0.2))$

TABLE 7.5 – Description des flux de données avec pour chacun le nombre de lots et pour chacun des lots $j \in [1, \text{nb. lots}]$ la distribution des points dans les différents clusters

est appliquée selon les deux dimensions du jeu de données afin de progressivement déplacer les clusters. Le tableau 7.5 indique les détails relatifs à ces autres jeux de données.

7.4 Description des projets collaboratifs

En parallèle de mes travaux théoriques sur les méthodes de clustering, sur l'analyse de traces utilisateurs, ou l'extraction de métadonnées, j'ai été amené à participer à plusieurs projets collaboratifs qui m'ont apporté un ensemble de cas d'usages réels très intéressants et qui m'ont aidé à développer plus avant mes contributions.

7.4.1 Projet webCSTI

Description : WEBCSTI est un projet sélectionné dans l'appel d'offres "Usages de l'Internet" du Ministère de la Recherche. Ce projet a été réalisé en partenariat avec le Laboratoire des Usages en Technologies d'Information Numériques (*LUTIN*) et l'AMCSTI (Association des Musées et Centres pour le développement de la Culture Scientifique, Technique et Industrielle). L'objectif du projet WebCSTI était double :

- ▷ recenser et définir des méthodes de compréhension des pratiques des usagers du Web, en combinant des approches diverses et multidisciplinaires (informatique, sciences cognitives, sémiotique) ;
- ▷ proposer une analyse de la dynamique propre au champ de la Culture Scientifique, Technique et Industrielle (CSTI) sur le Web et fournir des résultats sur le domaine et sur certains sites qui soient utiles pour les stratégies éditoriales dans le domaine.

Durée : 18 mois, de janvier 2005 à juillet 2006

Responsabilité : projet découpé en 5 sous-projets : responsable du sous projet 3 "Analyse des parcours Web grâce à l'exploitation des traces informatiques" et réalisation des analyses en collaboration avec ma collègue du LIP6 Maria Rifqi.

Réalisation : les travaux conduits ont conduit au développement de l'algorithme Leader Ant et ont permis de mettre en lumière différents rôles pour les sites de la CSTI (préparation de visite, téléchargement de documentation pédagogique). Voir également la section **??**.

7.4.2 Projet Infom@gic

Description : INFOM@GIC est un projet structurant du pôle de compétitivité CAP DIGITAL de la région Île de France. Il s'agit d'un projet coopératif visant à élaborer une plate-forme de recherche, d'extraction, de fusion et d'analyse de données multi-types et multimédia. Le projet Infom@gic a exploré des sujets majeurs en traitement de données pluri média et implémenté des prototypes logiciels de recherche et d'analyse intelligente de données issues du Web.

Durée : 36 mois de janvier 2005 à juin 2008

Responsabilité : responsable de la sous-tâche ST4.12 puis ST3.62 ayant trait à l'analyse des usages sur le site de la société Maxicours, partenaire du projet.

Réalisation : Analyse des traces utilisateurs pour caractériser les profils d'apprenants sur un site de soutien scolaire et interventions dans d'autres sous projets (plateforme logicielle Infom@gic) ; le logiciel d'analyse et de visualisation ILObTrack a été développé dans le cadre de ce projet en collaboration avec Lionel Yaffi de la société ILObjects. Voir également la section **??**.

7.4.3 Projet DoXa

Description : DoXa est un projet du pôle de compétitivité Cap Digital de la région Île de France visant à fournir des outils pour le suivi et l'analyse des opinions dans des textes, sur des blogs et des sites Internet et également à étudier la dynamique de la propagation des opinions sur Internet.

Durée : 36 mois de janvier 2009 à janvier 2012

Responsabilité : co-responsable avec Lionel Yaffi (Intelligent Learning Objects) du SP 5.42 relatif à la prise en compte des opinions et des sentiments dans l'analyse des traces utilisateurs.

Réalisation : Développement d'un système d'analyse de traces utilisateurs enrichies par des concepts et des opinions. Des mesures de similarité adéquates ont été proposées pour comparer les thèmes (concepts rattachés à une ontologie) et les opinions (valeurs positives ou négatives). Mise en place de cas d'usages avec ILObjects et Opinion Way et notamment le laboratoire des usages qui sert de base à l'étude conduite sur les jeux vidéos FIFA et PES. Voir également la section **??**.

7.4.4 Projet PURPLE : Routage intelligent dans un réseau pair-à-pair

Description : Le projet PURPLE a été initié pour favoriser la cohésion entre les équipes "base de données" et "apprentissage" du LIP6 lors de la création du département DAPA. L'objectif était d'optimiser le routage des requêtes dans un réseau pair à pair.

Durée : 24 mois de octobre 2005 – octobre 2007

Responsabilité : Responsabilité partagée à égalité avec les autres participants (33 %). Recrutement et encadrement de stagiaires de L3 et M2 recherche sur le sujet.

Réalisation : Le projet Purple a permis la réalisation d'un premier prototype logiciel de réseau P2P basé sur le système PeerSim et intégrant nos modèles de routage intelligent dépendant des modèles que chaque pair maintient concernant ses pairs voisins.

7.4.5 Projet Gamelab : analyse des interactions des joueurs dans un jeu vidéo

Description : Projet visant à diagnostiquer la qualité du gameplay d'un jeu vidéo à partir de traces utilisateurs (signaux physiologiques de type EEG, ECG, sudation, conductance de la peau, traces oculaires) et de questionnaires. Ce projet a été fait en coopération avec le Laboratoire des Usages LUTIN, Capital Games (structure regroupant la plupart des entreprises de développement de jeux vidéo en Ile de France), le laboratoire Chart de Paris 8 et la collaboration de Frédéric Lallemand (Ingénieur ILObjects).

Durée : 26 mois de février 2007 – mars 2009

Responsabilité : Co-gestion du projet pour le LIP6 avec Marie-Jeanne Lesot (MCF, LIP6) et Marc Damez (Dr, Ingénieur LIP6), recrutement et encadrement des stagiaires de master 2, réalisation des livrables de fin de projet. Le stage de François Nel a ensuite donné lieu à une thèse (soutenue) au LIP6.

Réalisation : Notre tâche dans le projet visait à établir un expert artificiel pour le diagnostic du gameplay d'un jeu vidéo. La solution proposée définit la qualité du gameplay comme un ensemble d'objectifs que les concepteurs du jeu souhaitent atteindre en terme de ressenti et de plaisir de jeu et repose sur la méthode des gabarits. Ce projet a donné lieu à l'encadrement de plusieurs stagiaires et notamment à la rédaction d'un article accepté à la conférence internationale IPMU 2008 [17].

7.4.6 Projet TOPOS

Description : Solution logicielle multi-plateformes pour la concertation publique et la réflexion participative réalisée en collaboration entre l'Adreva, le LIP6, ILObjects, Pertimm et Prylos.

Durée : 18 mois de décembre 2009 - juin 2011

Réalisation : Intervention dans la sous-tâche 2.2 visant à définir des modèles utilisateurs pour réaliser une interface intelligente. La nouveauté de l'approche retenue repose sur l'emploi de profils utilisateurs mixtes qui s'intéressent à la fois aux actions de l'utilisateur (fonctionnalités préférées dans l'interface) et aux contenus les plus pertinents (grâce à l'emploi de méthodes d'analyse sémantique latente de type LSA).

Chapitre 8

Publications

Revues :	5
Conférences internationales :	23
Conférences nationales :	6
Autres (forums, rapports techniques) :	4

TABLE 8.1 – Résumé des publications

Revues

[1] N. Labroche. Online fuzzy medoid based clustering algorithms. *Neuro Computing - Special Issue on Online Data Processing*, 2012.

[2] V.V. Vu, N. Labroche, and B. Bouchon-Meunier. Improving constrained clustering with active query selection. *Pattern Recognition*, 2011.

[3] M.J. Lesot, N. Labroche, and L. Yaffi. Analyse et visualisation interactive de sessions web. *RIA, Revue d'Intelligence Artificielle, numéro spécial Visualisation et extraction des connaissances*, 3-4(22) :369–382, 2008.

[4] N. Labroche. Clustering web pages sequences with artificial ants. *IADIS International Journal on www/Internet (ISSN : 1645-7641)*, 5 Issue 1, 2007.

[5] N. Labroche. Mesure d'audience sur internet par population de fourmis artificielles. *Revue des Nouvelles Technologies de l'Information (RNTI-E-5) Extraction des connaissances : Etat et perspectives*, pages 119–124, 2005.

Conférences internationales

[6] S. Changuel and N. Labroche. Content independant metadata production as a machine learning problem. In Petra Perner, editor, *Proceedings of the 8th International Conference in Machine Learning and Data Mining in Pattern Recognition (MLDM 2012)*, pages 306–320, Berlin, Germany, July 2012. Springer.

[7] S. Changuel and N. Labroche. Distinguishing defined concepts from prerequisite concepts in learning resources. In *2011 IEEE Symposium on Computational Intelligence and Data Mining, SSCI 2011 Conference*, 2011.

[8] V.V. Vu, N. Labroche, and B. Bouchon-Meunier. Boosting clustering by active constraint selection. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010)*, pages 297–302, Lisbon, Portugal, August 2010. IOI Press.

[9] V.V. Vu, N. Labroche, and B. Bouchon-Meunier. An efficient active constraint selection algorithm for clustering. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR-2010)*, pages 2969–2972, Istanbul, Turkey, August 2010. IEEE.

[10] V.V. Vu, N. Labroche, , and B. Bouchon-Meunier. Active learning for semi-supervised k-means clustering. In *Proceedings of the 22th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2010)*, Arras, France, October 2010. IEEE.

[11] N. Labroche. New incremental fuzzy c medoids clustering algorithms. In *Proc. of the North American Fuzzy Information Processing Society 2010*, pages 145–150, Toronto, Canada, 2010.

[12] S. Changuel, N. Labroche, and B. Bouchon-Meunier. Automatic concept type identification from learning resources. In *Proc. of IEEE World Congress on Computational Intelligence, Special Session : Medical and Educational Applications of Computer Intelligence to Benefit Societ*, 2010.

[13] N. Labroche and C. Marsala. Optimization of a fuzzy decision trees forest with artificial ant based clustering. In *Proc. of the SOCPAR 2010 Conference*, 2010.

[14] S. Changuel, N. Labroche, and B. Bouchon-Meunier. Automatic web pages author extraction. In *FQAS 2009 (Flexible Query Answering Systems)*, pages 300–311, 2009.

[15] S. Changuel, N. Labroche, and B. Bouchon-Meunier. A general learning method for automatic title extraction from html pages. In *International Conference on Machine Learning and Data Mining MLDM'2009*, pages 704–718, 2009.

[16] V.-V. Vu, N. Labroche, and B. Bouchon-Meunier. Leader ant clustering with constraints. In *IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies, RIVF09*, pages 79–86, 2009.

[17] F. Nel, M.J. Lesot, N. Labroche, and M. Damez. Automated video games evaluation based on the template formalism. In *IPMU 2008 Conference*, 2008.

[18] N. Labroche. Learning web users profiles with relational clustering algorithms. In *Workshop On Intelligent Techniques for Web Personalization, AAAI 2007 Conference*, pages 54–64, Vancouver, Canada, 2007.

[19] N. Labroche, M.J. Lesot, and L. Yaffi. A new web usage mining and visualization tool. In *IEEE Internal Conference on Tools with Artificial Intelligence, ICTAI 2007*, pages 321–328, Patras, Greece, 2007.

[20] N. Labroche. Clustering web pages sequences with artificial ants. In *IADIS International WWW/Internet Conference*, pages 503–510, Murcia, Spain, 2006.

[21] N. Labroche. Fast ant-inspired clustering algorithm for web usage mining. In *IPMU 2006 Conference*, pages 2668–2675, Paris, France, 2006.

[22] J.F. Omhover, N. Labroche, and B. Bouchon-Meunier. Sensorial approaches to multimedia information retrieval. In *SEIT 2005 Workshop, 17th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, Paris, France, 2005.

[23] N. Labroche, C. Guinot, and G. Venturini. Fast unsupervised clustering with artificial ants. In Xin Yao et al. Editors, editor, *Proceedings of the Parallel Problem Solving from Nature 2004 (PPSN VIII)*, pages 1143–1152, Birmingham, England, 18-22 september 2004.

[24] N. Labroche, N. Monmarché, and G. Venturini. Visual clustering with artificial ants colonies. In R.J. Howlett V. Palade and L.C. Jain Editors, editors, *Proceedings of the KES 2003 International Conference*, Oxford, England, september 3-5 2003.

[25] N. Labroche, N. Monmarché, and G. Venturini. Web sessions clustering with artificial ants colonies. In *Proceedings of the World Wide Web Conference 2003*, Budapest, Hungary, may 20-24 2003.

[26] N. Labroche, N. Monmarché, and G. Venturini. Antclust : Ant clustering and web usage mining. In *Proceedings of the Genetic and Evolutionary Computation Conference (Gecco 2003)*, Chicago, USA, july 12-16 2003.

[27] N. Labroche, N. Monmarché, and G. Venturini. A new clustering algorithm based on the chemical recognition system of ants. In *Proc. of 15th European Conference on Artificial Intelligence (ECAI 2002), Lyon FRANCE*, pages 345–349, 2002.

[28] N. Labroche, FJ. Richard, N. Monmarché, A. Lenoir, and G. Venturini. Modelling the chemical recognition system of ants. In C.K. Hemelrijk, editor, *International Workshop on Self-Organization and Evolution of Social Behaviour*, pages 283–292, Monte Verità, Ascona, Switzerland, september 8-13 2002.

Conférences nationales

[29] N. Labroche. Classification non supervisée incrémentale de données relationnelles. In *Rencontres francophones sur la Logique Floue et ses Applications*, Lannion, France, 2010.

[30] M.J. Lesot, N.Labroche, and L. Yaffi. Analyse et visualisation de sessions web. In *Atelier Visualisation et Extraction de Connaissances, Conférence EGC 2008*, 2008.

[31] N. Labroche. Classification supervisée par population de fourmis artificielles. In *EGC 2007 - Atelier Fouille de Données et Algorithmes Biomimétiques*, pages 37–48, Namur, Belgique, 2007.

[32] N. Labroche. Mesure d'audience sur internet par population de fourmis artificielles. In *Workshop Modélisation de l'Utilisateur, Conférence EGC 2005*, Paris, France, 2005.

[33] N. Labroche, N. Monmarché, and G. Venturini. Modélisation de la fermeture coloniale chez les fourmis pour la classification non-supervisée. In *Conférence d'apprentissage (CAp 2002)*, pages 137–148, Orléans, France, 17-19 juin 2002. Presses Universitaires de Grenoble.

[34] N. Monmarché, D. Laugt, M. Mestre, N. Labroche, A. Oliver, and G. Venturini. Classification et visualisation dynamique de données par nuage d'insectes volants. In *Journées de la Société Francophone de classification*, Pointe à Pitre, Guadeloupe, 17-21 décembre 2001.

Forums avec comité de relecture

[35] N. Labroche. Visualisation et classification non-supervisée grâce à une population de fourmis artificielles. In *Forum de l'École Doctorale de l'Université de Tours*, mai 2002.

[36] N. Labroche. Classification non-supervisée de données issues d'internet à partir de populations de fourmis artificielles. In *Forum de l'École Doctorale de l'Université de Tours*, mai 2001.

[37] E. Dellandrea, N. Labroche, P. Makris, M. Boiron, and N. Vincent. Fusion de données radiologiques et sonores dans l'étude du traitement chirurgical du reflux gastro-œsophagien. In *10ème Forum des jeunes chercheurs Génie Biomédical, Biophysique et traitement d'images, TOURS (FRANCE)*, pages 62–63, mai 2000.

Rapport technique

[38] N. Labroche, FJ. Richard, N. Monmarché, A. Lenoir, and G. Venturini. Description et modélisation du système d'identification chimique des fourmis. Technical Report 254, Laboratoire d'Informatique de l'Université de Tours, avril 2002. 47 pages.

Mémoires

[39] N. Labroche. *Modélisation du système de reconnaissance chimique des fourmis pour le problème de la classification non-supervisée : application à la mesure d'audience sur Internet.* PhD thesis, Laboratoire d'Informatique de Tours, UPRES EA 2101, 4 décembre 2003. Responsable : Pr. Gilles Venturini.

[40] N. Labroche. Étude chez l'homme des bruits gastriques liés à la déglutition, septembre 2000. DEA Signaux et images en biologie et médecine, Responsable : Pr. Nicole Vincent.

Encadrement de thèses

[41] Vicent Labbé. *Modélisation et apprentissage des préférences appliqués à la recommandation dans les systèmes d'impression.* PhD thesis, Université Pierre et Marie Curie, Laboratoire d'Informatique de Paris 6, 22 septembre 2009. Sous la direction de Bernadette Bouchon-Meunier et encadrée par Nicolas Labroche.

[42] Sahar Changuel. *Métadonnées pour la personnalisation et l'accès à la connaissance.* PhD thesis, Université Pierre et Marie Curie, Laboratoire d'Informatique de Paris 6, 3 mai 2011. Sous la direction de Bernadette Bouchon-Meunier et encadrée par Nicolas Labroche.

[43] Viet-Vu Vu. *Clustering semi-supervisé et apprentissage actif.* PhD thesis, Université Pierre et Marie Curie, Laboratoire d'Informatique de Paris 6, 5 juillet 2011. Sous la direction de Bernadette Bouchon-Meunier et encadrée par Nicolas Labroche.

Chapitre 9

Références

[44] A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. of Cybernetics*, 3 :32–57, 1973.

[45] A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1 :224–227, 1979.

[46] Validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 3(3) :841–846, 1991.

[47] C. Aggarwal, J. Han, J. Wang, and P. Yu. A framework for projected clustering of high dimensional data streams. In *Proc. of the 30th International Conference on Very Large Databases*, pages 852–863, 2004.

[48] Charu C. Aggarwal, T. J. Watson, Resch Ctr, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In *In VLDB*, pages 81–92, 2003.

[49] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of ACM SIGMOD*, pages 94–105, 1998.

[50] L. Ahmad and A. Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data and Knowledge Engineering*, vol. 63 :503–527, 2007.

[51] P. Angelov. Fuzzily connected multimodel systems evolving autonomously from data streams. *IEEE Transactions on Systems, Man and Cybernetics - Part B : Cybernetics*, 41(4) :898–910, 2011.

[52] P. Angelov and D.P. Filev. Simpl_ets : a simplified method for learning evolving takagi-sugeno fuzzy models. In *International conference on fuzzy systems*, pages 1068–1072, 2005.

[53] M. Ankerst, M. Breunig, H.P. Kriegel, and J. Sander. Optics : Ordering points to identify clustering structure. In *Proc. of the ACM SIGMOD*, pages 49–60, Philadenphia, USA, 1999.

[54] V. Antoine, B. Quost, M.-H. Masson, and T. Denœux. Cecm : Constrained evidential -means algorithm. *Computational Statistics & Data Analysis*, 56(4) :894 – 914, 2012.

[55] A. Asuncion and D.J. Newman. UCI machine learning repository – University of California, Irvine, School of Information and Computer Sciences. http ://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[56] R. Baraglia and P. Palmerini. Suggest : A web usage mining system. In *Proc. of IEEE Int. Conf. on Information Technology : Coding and Computing*, pages 282–287, 2002.

[57] Maria Camila N. Barioni, Humberto L. Razente, Agma J. M. Traina, and Caetano Traina Jr. An efficient approach to scale up k-medoid based algorithms in large databases. In *Proceedings of the XXI Simpósio Brasileiro de Banco de Dados*, 2006.

[58] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *In Proceeding of the 19th International Conference on Machine Learning (ICML)*, pages 27–34, 2002.

[59] S. Basu, A. Banerjee, and R.J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining*, pages 333–344, 2004.

[60] Sugato Basu, Arindam Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*, 2002.

[61] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger. The r*-tree : An efficient and robust access method for points and rectangles. In *In Proc. of ACM SIGMOD Int. Conf. on Management of Data*, 1990.

[62] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke. Partially supervised clustering for image segmentation. *Pattern Recognition*, 29(5), 1996.

[63] Amine M. Bensaid, Lawrence O. Hall, James C. Bezdek, and Laurence P. Clarke. Partially supervised clustering for image segmentation. *Pattern Recogn.*, 29(5) :859–871, 1996.

[64] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American Magazine*, May 17, 2001.

[65] James Bezdek, James Keller, Raghu Krishnapuram, and Nikhil R. Pal. *Fuzzy Models And Algorithms For Pattern Recognition And Image Processing.* The Handbook of Fuzzy Sets Series, didier dubois and henri prade edition, 1999.

[66] James C. Bezdek, Richard J. Hathaway, Jacalyn M. Huband, Christopher Leckie, and Ramamohanarao Kotagiri. Approximate clustering in very large relational data. *International Journal of Intelligent Systems*, 21(8) :817–841, 2006.

[67] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[68] Christian Böhm and Claudia Plant. Hissclu : a hierarchical density-based method for semi-supervised clustering. In *In Proceeding of the 11th International Conference on Extending Database Technology*, pages 440–451, 2008.

[69] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Intl. Conference on Machine Learning, ICML 2004*, pages 81–88, 2004.

[70] Leo Breiman and Leo Breiman. Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.

[71] B.Scholkof, A. Smola, and K.Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computat*, vol. 10(5) :1299–1319, 1998.

[72] Feng Cao, Martin Ester, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *In 2006 SIAM Conference on Data Mining*, pages 328–339, 2006.

[73] E. Chi, J. Pitkow, J. Mackinlay, P. Pirolli, R. Gossweiler, and Stuart K. Card. Visualizing the evolution of web ecologies. In *Proc. of the Conf. of CHI'98*, 1998.

[74] E. Chi, A. Rosien, and J. Heer. Lumberjack : Intelligent discovery and analysis of web user traffic composition. In *WEBKDD 2002 : Web Mining for Usage Patterns and User Profiles*, 2002.

[75] M. Ming-Tso Chiang and Boris Mirkin. Intelligent choice of the number of clusters in k-means clustering : An experimental study with different cluster spreads. *Journal of Classification*, 27.

[76] R. Cooley, B. Mobasher, and J. Srivastava. Web mining : Information and pattern discovery on the world wide web. In *Proc. of the ICTAI Conference*, pages 558–567, 1997.

[77] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1) :5–32, 1999.

[78] Marc Damez-Fontaine. *De l'apprentissage artificiel pour l'apprentissage humain : de la récolte de traces à la modélisation utilisateur.* PhD thesis, Université Pierre et Marie Curie, 2008.

[79] I. Davidson and S. Basu. A survey of clustering with instance level constraints. *ACM Transactions on Knowledge Discovery from data*, pages 1–41, 2007.

[80] I. Davidson and S.S. Ravi. Agglomerative hierarchical clustering with constraints : Theoretical and empirical results. In *Proceeding of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD-2005*, pages 59–70, 2005.

[81] I. Davidson and S.S. Ravi. Clustering with constraints : Feasibility issues and the k-means algorithm. In *Proceedings of the SIAM International Conference on Data Mining*, 2005.

[82] I. Davidson, K.L. Wagstaff, and S. Basu. Measuring constraints-set utility for partitional clustering algorithms. In *Proceeding of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 577–584, 2006.

[83] Ian Davidson. Efficient incremental constrained clustering. In *in Proceedings of the KDD Conference*, pages 240–249, 2007.

[84] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, Series B (Methodological) 39 (1) :1–38, 1977.

[85] M. Detyniecki. *Mathematical aggregation operators and their application to video querying.* PhD thesis, Université de Paris VI, 2000.

[86] E. Duval, E. Forte, K. Cardinaels, B. Verhoeven, R. Van Durm, K. Hendrikx, M. W. Forte, N. Ebel, M. Macowicz, K. Warkentyne, and F. Haenni. The Ariadne knowledge pool system. *Communications of the ACM*, 44(5) :72–78, 2001.

[87] M. Ester, H.P. Kriegel et J. Sander, M. Wimmer, and X. Xu. Incremental clustering for mining in a data warehousing environment. In *In Proc. of the 24th VLDB conference*, 1998.

[88] M. Ester, H.-P Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, USA, 1996. AAAI Press.

[89] Vladimir Estivill-Castro and Jianhua Yang. Categorizing visitors dynamically by fast and robust clustering of access logs. In *Lecture Notes in Computer Science, Proceedings of the First Asia-Pacific Conference on Web Intelligence : Research and Development*, pages 498–507, 2001.

[90] Density-Based Clustering for Real-Time Stream Data. Y. chen and l. tu. In ACM, editor, *Proceedings of KDD'07 Conference*, pages 133–142, 2007.

[91] Y. Fu, K. Sandhu, and M. Shih. Clustering of web users based on access patterns. In Springer-Verlag, editor, *Proceedings of the 1999 KDD Workshop on Web Mining*, San Diego, CA, 1999.

[92] Y. Fu, K. Sandhu, and M. Shih. A generalization-based approach to clustering of web usage sessions. In *Web Usage Analysis and User Profiling*, pages 21–38. Springer, 2000.

[93] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams : a review. *ACM SIGMOD Record*, 34(2), June 2005.

[94] Jane Greenberg. Metadata extraction and harvesting : A comparison of two automatic metadata generation applications. *Journal of Internet Cataloging*, 6(4) :59–82, 2004.

[95] Jane Greenberg and W. Davenport Robertson. Semantic web construction : an inquiry of authors' views on collaborative metadata generation. In *Proc. of the 2002 Int. Conf. on Dublin core and metadata applications : Metadata for e-communities : supporting diversity and convergence*, pages 45–52. Dublin Core Metadata Initiative, 2002.

[96] Jane Greenberg, Kristina Spurgin, and Abe Crystal. Functionalities for automatic metadata generation applications : a survey of metadata experts' opinions. *International Journal of Metadata, Semantics and Ontologies*, 1(1) :3–20, 2006.

[97] S. Guha, R. Rastogi, and K. Shim. Cure : An efficient clustering algorithm for large databases. In *Proc. of ACM SIGMOD International Conference on management of Data*, pages 73–84, 1998.

[98] S. Guha, R. Rastogi, and K. Shim. Rock : A robust clustering algorithme for categorical attributes. *Inf. Sys.*, vol. 25 no. 5 :345–366, 2000.

[99] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams : Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3) :515–528, May-June 2003.

[100] Naoki Haga, Katsuhiro Honda, Hidetomo Ichihashi, and Akira Notsu. Linear fuzzy clustering of relational data based on extended fuzzy c-medoids. In *IEEE Intl. Conference on Fuzzy Systems, 2008. FUZZ-IEEE 2008*, pages 1098–7584, Hong Kong, 2008.

[101] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems, Kluwer Academic Publishers. Manufactured in The Netherlands*, 7 :2/3 :107–145, 2001.

[102] Hui Han, C. Lee Giles, Eren Manavoglu, Hongyuan Zha, Zhenyue Zhang, and Edward A. Fox. Automatic document metadata extraction using support vector machines. In *Proc. of the 3rd ACM/IEEE-CS joint conference on Digital libraries, JCDL '03*, pages 37–48. IEEE Computer Society, 2003.

[103] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *SIGMOD Record*, 29 :1–12, May 2000.

[104] P. Hansen and N. Mladenovic. J-means : A new local search heuristic for minimum sum-of-squares clustering. *Pattern Recognition*, 34(2) :405–413, 2001.

[105] J.A. Hartigan. *Clustering algorithms.* John Wiley & Sons Inc., 1975.

[106] M.A. Hasan, V.Chaoji, S. Salem, and M.J. Zaki. Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition Letters 30(11) : 994-1002*, 2009.

[107] R.J. Hathaway and J.C. Bezdek. Nerf c-means : Non-euclidian relational fuzzy clustering. *Pattern Recognition*, 27 :429–437, 1994.

[108] R.J. Hathaway, J.C. Bezdek, and J.W. Davenport. On relational data versions of c-means algorithms. *Pattern Recognition Letters*, 17 :607–612, 1996.

[109] R.J. Hathaway, J.W. Davenport, and J.C. Bezdek. Relational dual of the c-means clustering algorithms. *Pattern Recognition*, 22(2) :205–212, 1989.

[110] J. Heer and E. Chi. Identification of web user traffic composition using multi-modal clustering and information scent. In *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining*, pages 51–58, 2001.

[111] J. Heer and E. Chi. Separating the swarm : Categorization methods for user sessions on the web. In *Proceedings of CHI 2002, Human Factors in Computing Systems*, 2002.

[112] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization : A survey. *IEEE Trans. on Visualization and Computer Graphics*, 6(1) :24–43, 2000.

[113] A. Hinneburg and A.A Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proc. of Knowledge Discovery and Data Mining*, pages 58–65, 1998.

[114] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2) :180–184, 1985.

[115] B. Hölldobler and E.O. Wilson. *The Ants*, chapter Colony odor and kin recognition, pages 197–208. Springer Verlag, Berlin, Germany, 1990.

[116] J. I. Hong, J. Heer, S. Waterson, and J. A. Landay. Webquilt : a framework for capturing and visualizing the web experience. In *World Wide Web*, pages 717–724, 2001.

[117] P. Hore, L.O. Hall, D.B. Goldgof, and W. Cheng. Online fuzzy c means. In *Fuzzy Information Processing Society, NAFIPS 2008*, pages 1–5, 2008.

[118] Y. Hu, J. Wang, N. Yu, and X.-S. Hua. Maximum margin clustering with pairwise constraints. In *Procedings of the Eighth IEEE International Conference on Data Mining (ICDM)*, pages 253–262, 2008.

[119] A.K. Jain and R.C. Dubes. *Algorithms for clustering Data*. Prentice Hall Advanced Reference series, 1988.

[120] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering : A review. *ACM Computing Surveys*, 31(3) :264–323, 1999.

[121] Anil K. Jain. Data clustering : 50 years beyond k-means. *Pattern Recognition Letters*, 31(8) :651–666, 2010.

[122] R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computer*, (11), 1973.

[123] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proc. of the 26th annual international ACM SIGIR Conf. on Research and development in informaion retrieval*, SIGIR '03, pages 119–126, New York, NY, USA, 2003. ACM.

[124] X. Jin, Y. Zhou, and B. Mobasher. A unified approach to personalization based on probabilistic latent semantic models of web usage and content. In *AAAI 2004 Workshop on Semantic Web Personalization (SWP'04)*, 2004.

[125] George John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *In Proc. of the Eleventh Conf. on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann, 1995.

[126] C.F. Juang and C.T. Lin. An on-line self constructing neural fuzzy inference network. *IEEE Transactions on Neural Networks*, 6(1) :12–32, 1998.

[127] S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *Proceeding of the Joint Conference on Artificial Intelligence, IJCAI-2003*, pages 561–566, 2003.

[128] G. Karypis, E.H. Han, and V. Kumar. Chameleon : Hierarchical clustering using dynamic model. *Computer*, vol. 32(8) :68–75, 1999.

[129] G. Karypis and V. Kumar. hmetis 1.5 : A hypergraph partitioning package. Technical report, Dpt. of Computer Sciences, University of Minnesota, 1998.

[130] N. Kasabov and Q. Song. Denfis : Dynamic evolution neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2) :144–154, 2002.

[131] L. Kaufman and P.J. Rousseeuw. Clustering by means of medoids. In *Statistical Data Analysis based on the L1 Norm*, pages 405–416. Elsevier, 1987.

[132] L. Kaufman and P.J. Rousseeuw. Finding groups in data : An introduction to cluster analysis. In *John Wiley and Sons*, 1990.

[133] Judy Kay, Nicolas Maisonneuve, Kalina Yacef, and Osmar Zaïane. Mining patterns of events in students' teamwork data. In *In Educational Data Mining Workshop, held in conjunction with Intelligent Tutoring Systems (ITS*, pages 45–52, 2006.

[134] P. Kranen, H. Kremer, T. Jansen, T. Seidl, A. Bifet, G. Holmes, and B. Pfahringer. Clustering performance on evolving data streams : Assessing algorithms and evaluation measures within moa. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on Data Mining Workshops*, pages 1400–1403, 2010.

[135] Hardy Kremer, Philipp Kranen, Timm Jansen, Thomas Seidl, Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. An effective evaluation measure for clustering on evolving data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 868–876, New York, NY, USA, 2011. ACM.

[136] R. Krishnapuram, A. Joshi, O. Nasraoui, and L. Yi. Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE-FS*, 9 :595–607, 2001.

[137] A. Krowne, K. Skinner, M. Halbert, S. Ingram, U. Gadi, and S. Pathak. Metacombine project interim report. rapp. tech., emory university, 2006.

[138] A. Laurent and M.-J. Lesot, editors. *Scalable Fuzzy Algorithms for Data Management and Analysis : Methods and Design*. IGI Global, 2009.

[139] D.-D. Le and S. Satoh. Unsupervised face annotation by mining the web. In *Proc. 9th IEEE ICDM*, 2008.

[140] Levi Lelis and Jörg Sander. Semi-supervised density-based clustering. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 842–847, Washington, DC, USA, 2009. IEEE Computer Society.

[141] M.-J. Lesot. *Classification non supervisée pour la visualisation de données structurées et la construction de prototypes*. PhD thesis, Université Paris VI, 31 janvier 2005.

[142] M.-J. Lesot, L. Mouillet, and B. Bouchon-Meunier. Fuzzy prototypes based on typicality degrees. In *Proc. of the 8th Fuzzy Days 2004*, pages 125–138. Springer, 2005.

[143] M.-J. Lesot, M. Rifqi, and B. Bouchon-Meunier. Fuzzy prototypes : From a cognitive view to a machine learning principle. In *Fuzzy Sets and Their Extensions : Representation, Aggregation and Models*. Springer, 2007.

[144] Elizabeth Liddy, Jiangping Chen, Christina Finneran, Anne Diekema, Sarah Harwell, and Ozgur Yilmazel. Generating and evaluating automatic metadata for educational resources. In *Research and Advanced Technology for Digital Libraries*, volume 3652 of *Lecture Notes in Computer Science*, pages 513–514. Springer Berlin / Heidelberg, 2005.

[145] Bing Liu, Wynne Hsu, and Yiming Ma. Pruning and summarizing the discovered associations. In *Proc. of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 125–134. ACM Press, 1999.

[146] W. Liu and J. O. Yang. Clustering algorithm for high dimensional data stream over sliding windows. In *International Joint COnference of IEEE TrustCom-11*, pages 1537–1542, 2011.

[147] U.V. Luxburg. A tutorial on spectral clustering. Technical Report TR-149, Max Planck Institute for Biological Cybernetics, Germany, August 2006.

[148] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In University of California Press, editor, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, 1967.

[149] P.K. Mallapragada, R. Jin, and A.K. Jain. Active query selection for semi-supervised clustering. In *Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[150] F. Masseglia, P. Poncelet, and R. Cicchetti. Webtool : An integrated framework for data mining. In *Database and Expert Systems Applications*, pages 892–901, 1999.

[151] F. Masseglia, P. Poncelet, and M. Teisseire. Using data mining techniques on web access logs to dynamically improve hypertext structure. *ACM SigWeb Letters*, 8(3) :1–19, 1999.

[152] D. Mavroeidis. Accelerating spectral clustering with partial supervision. *Data Mining and Knowledge Discovery*, 21 :241–258, 2010.

[153] B. Mobasher. Data mining for personalization. In *The Adaptive Web : Methods and Strategies of Web Personalization*, volume 4321 of *LNCS*, pages 90–135. Springer, 2006.

[154] Bamshad Mobasher. *Web Data Mining : Exploring Hyperlinks, Contents and Usage Data*, chapter Web Usage Mining, pages 449–483. Springer Berlin-Heidelberg, 2006.

[155] O. Nasraoui, H. Frigui, A. Joshi, and R. Krishnapuram. Mining web access logs using relational competitive fuzzy clustering. In *Eight International Fuzzy Systems Association World Congress - IFSA 99*, 1999.

[156] O. Nasraoui, A. Joshi, and R. Krishnapuram. Relational clustering based on a new robust estimator with application to web mining. In *Proc. of Int. Conf. North American Fuzzy Info. Proc. Society, NAFIPS'99*, New York, 1999.

[157] O. Nasraoui, R. Krishnapuram, A. Joshi, and T. Kamdar. Automatic web user profiling and personalization using robust fuzzy relational clustering. In J. Kacprzyk, editor, *E-Commerce and Intelligent Methods*. Springer, 2002.

[158] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering : Analysis and an algorithm. In *Proc. of NIPS*, pages 849–856, 2001.

[159] R. Ng and J. Han. Clarans : A method for clustering objects for spatial data mining. *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 5 :1003–1016, 2002.

[160] Clark F. Olson. Parallel algorithms for hierarchical clustering. Technical report, University of California at Berkeley, 1993.

[161] W. Pedrycz. Algorithm of fuzzy clustering with partial supervision. *Pattern Recognition Letters*, 3 :13–20, 1985.

[162] W. Pedrycz. Algorithm of fuzzy clustering with partial supervision. *Pattern Recognition Letters*, 3, 1985.

[163] W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Transactions on systems, Man, and Cybernetics*, 27(5) :787–795, 1997.

[164] W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Transactions on systems, Man, and Cybernetics*, 27(5) :787–795, 1997.

[165] R.G. Pensa and J.-F. Boulicaut. Co-classification sous contraintes par la somme des résidus qudratiques. In *Actes des 8ème Journées Francophones Extraction et Gestion de Connaissances EGC'08*, pages 655–666, 2008.

[166] R.G. Pensa, C. Robardet, and J.-F. Boulicaut. Co-classification sous contraintes. In *Actes CAp'06*, pages 155–170, Trégastel, France, 2006. Presses Universitaires de Grenoble.

[167] Corinna Baldauf Philipp Kranen, Ira Assent and Thomas Seidl. The clustree : indexing micro-clusters for anytime stream mining. *Knowledge and Information Systems*, 29(2) :249–272, 2011.

[168] J. Pitkow and K. Bharat. Webviz : A tool for world wide web access log analysis. In *Proc. of the First Int. World-Wide Web Conference*, pages 271–277, 1994.

[169] P. Plant and C. Boehm. *Novel Trends in Clustering*, chapter 9, pages 185–211. Evolving Application Domains of DataWarehousing and Mining : Trends and Solutions. IGI Global Press, 2009.

[170] Ross Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, 1st edition, 1993.

[171] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association*, vol. 66, 1971.

[172] M. Rifqi. Constructing prototypes from large databases. In *Proc. of IPMU'96*, 1996.

[173] E. Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and categorization*, pages 27–48. Lawrence Erlbaum associates, 1978.

[174] J. J. Rubio, D. M. Vàsquez, and J. Pacheco. Backpropagation to train an evolving radial basis function neural network. *Evolving Systems*, 1 :173–180, 2010.

[175] C. Ruiz, E. Menasalvas, and M. Spiliopoulou. C-denstream : Using domain knowledge on a data stream. In Springer-Verlag Berlin Heidelberg, editor, *DS 2009, LNAI 5808*, pages 287–301, 2009.

[176] C. Ruiz, M. Spiliopoulou, and E. Menasalvas. C-dbscan : Density-based clustering with constraints. In *Proceedings of the International Conference on Rough Sets Fuzzy Sets Data Mining and Granular Computing*, pages 216–223, 2007.

[177] C. Ruiz, M. Spiliopoulou, and E. Menasalvas. Density-based semi-supervised clustering. *Data Mining and Knowledge Discovery*, 21(3) :345–370, 2010.

[178] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[179] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster : A multiresolution clustering approach for very large spatial databases. In *Proc. of the 24 conference on VLDE*, pages 428–439, 1998.

[180] Eddie C. Shek and Jihoon Yang. Knowledge-based metadata extraction from postscript files. In *Proc. of the 5th ACM Conference on Digital Libraries*, pages 77–84. ACM Press, 2000.

[181] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, 2000.

[182] Ahu Sieg, Bamshad Mobasher, and Robin Burke. Ontological user profiles for personalized web search. In *5th Workshop on Intelligent Techniques for Web Personalization*, 2007.

[183] P. H. A. Sneath and R. R. Sokal. Numerical taxonomy - the principles and practice of numerical classification. Technical report, W. H. Freeman, San Francisco, 1973.

[184] Y. Song, W.-Y Chen, H. Bai, C.-J Lin, and E.-Y. Chang. Parallel spectral clustering. In *Proc. of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases-ECML08*, Antwerp, Belgium, September 2008.

[185] M. Spiliopoulou and L.C. Faulstich. Wum : a web utilization miner. In *Workshop on the Web and Data Bases, WebDB'98*, pages 109–115, 1998.

[186] M. Su and C. Chou. A modified version of the k-means algorithm with a distance based on cluster symmetry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) :674–680, 2001.

[187] Xiaoyu Tang, Qingtian Zeng, Tingting Cui, and Zeze Wu. Regular expression-based reference metadata extraction from the web. In *IEEE 2nd Symposium on Web Society (SWS)*, pages 346–350, 2010.

[188] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2 edition, 1999.

[189] K. Wagstaff. When is constrained clustering beneficial, and why ? In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI)*, July 2006.

[190] K. L. Wagstaff. Value, cost, and sharing : Open issues in constrained clustering. In *Proceeding of the 5th International Workshop on Knowledge Discovery in Inductive Databases, KDID-2007*, pages 1–10, 2007.

[191] K. L. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *In Proceedings of the 17th International Conference on Machine Learning, ICML*, pages 1103–1110, 2000.

[192] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *In Proceedings of the 18th International Conference on Machine Learning, ICML-2001*, pages 577–584. Morgan Kaufmann, 2001.

[193] W. Wang and O. Zaiane. Clustering web sessions by sequence alignment. In *3rd Intl. Workshop on Management of Information on the Web in conjunction with DEXAí2002*, pages 394–398, Aix en Provence, France, 2002.

[194] Weinan Wang and Osmar R. Zaïane. Clustering web sessions by sequence alignment. In *In Proceedings of the 13th international workshop on database and expert systems applications (DEXA 2002). Aix-en-Provence*, pages 394–398. Springer-Verlag, 2002.

[195] X. Wang and I. Davidson. Flexible constrained spectral clustering. In *Proceeding of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD-2010*, pages 563–572, 2010.

[196] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *Proceedings of the KDD 2009 Conference*, pages 877–885. ACM, 2009.

[197] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey McLachlan, Angus Ng, Bing Liu, Philip Yu, Zhi-Hua Zhou, Michael Steinbach, David Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1) :1–37, 2008.

[198] Rui Xu and Donald Wunsch II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3) :645–678, May 2005.

[199] Tak Woon Yan, Matthew Jacobsen, Hector Garcia-molina, and Umeshwar Dayal. From user access patterns to dynamic hypertext linking. pages 1007–1014, 1996.

[200] A. Youssefi, D. Duke, and M. J. Zaki. Visual web mining. In ACM, editor, *Poster Proc. of WWW'04 Conference*, 2004.

[201] O.R. Zaiane, C.H. Lee A. Foss, and W. Wang. On data clustering analysis : Scalability, constraints and validation. In *Proc. of the Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. Lecture Notes in Artificial Intelligence 2336, Advances in Knowledge Discovery and Data Mining, Springer-Verlag, 2002.

[202] B. Zhang and M. Hsu. K-harmonic means - a data clustering algorithm. Technical Report HPL-1999-124, Hewlett-Packard Labs, 1999.

[203] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Canada, 1996.

Résumé :

Les travaux de recherche présentés dans ce mémoire s'intéressent au développement de méthodes d'apprentissage pour l'étude des interactions utilisateurs selon trois principaux niveaux. En premier lieu, nous nous sommes intéressés à l'analyse des interactions utilisateurs sous la forme de traces d'usage. Nous avons développé pour cela des méthodes de clustering incrémentales qui autorisent le passage à l'échelle ainsi que le traitement de flux de données. Contrairement aux approches existantes, nos méthodes ne sont pas limitées au traitement de données numériques et sont couplées à des outils de recherche de parcours typiques et de visualisation dynamique pour faciliter l'interprétation des analyses. En second lieu, nous nous sommes intéressés aux méthodes de clustering semi-supervisé qui permettent l'intégration de connaissances expertes dans le processus de construction des clusters. Dans ce cadre, nous avons notamment proposé des algorithmes actifs qui optimisent les interactions de l'expert avec l'algorithme d'apprentissage pour en améliorer les performances. Enfin, nous nous sommes intéressés au problème de l'extraction automatique de métadonnées à partir de corpus structurés dans le cadre de la recherche d'information. La spécificité de ces travaux est double : d'une part nous avons introduit des méthodes indépendantes du contenu et d'autre part, nous avons proposé des méthodes basées sur le contenu qui combinent apprentissage statistique et descripteurs contextuels, stylistiques et linguistiques.

Mot-clés : classification non-supervisée, analyse de traces sur Internet, visualisation, clustering semi-supervisé, apprentissage automatique de métadonnées.

Abstract :

The research described in this habilitation document proposes new machine learning methods dedicated to the study of users interactions on three main levels. First, we are interested in the analysis of user interaction in the form of users' activity traces. To this aim, we develop new incremental clustering algorithms for large scale data sets or data streams. Unlike existing approaches, our methods are not limited to numerical data processing and are paired with tools to research typical user profile and that provide dynamic web paths visualization to facilitate interpretation of the analyzes. Secondly, we are interested in semi-supervised clustering methods which allow the integration of expert knowledge in the clustering process. In this context, we have proposed active learning algorithms that optimize interactions between the human expert and the classification algorithms to improve their performance. Finally, we are interested in the problem of automatic extraction of metadata from structured corpus in the context of information retrieval. The specificity of this work is twofold : on one hand we have introduced methods that are independent of the document's content and, on the other hand, we have proposed methods based on the content that combine statistical learning algorithms and contextual, stylistic and linguistic features.

Keywords : clustering, web usage mining, visualization, semi-supervised clustering, metadata extraction.