République Algérienne Démocratique et Populaire
Ministére de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de cycle
En vue de l'obtention du diplôme de master en Informatique
Spécialité : Génie Logiciel

Thème

Clustering hiérarchique pour la classification et la reconnaissance des activités dans un système ubiquitaire

Réaliser par :

TAOURI Nassima

Evalué le 13 Septembre 2020 par le jury composé de :

Président	Mr	Achour	ACHROUFENE	M.C.B	U.A.M.Béjaïa
Examinateur	M^{me}	Nassima	BOUADEM	M.C.B	U.A.M.Béjaïa
Encadreur	M^{me}	Salima	SABRI	M.C.B	U.A.M.Béjaïa
CoEncadreur	M^{me}	Malika	YAICI	M.C.A	U.A.M.Béjaïa

Promotion 2019/2020

Remerciements

Mes vifs remerciements vont aux personnes qui ont contribué au bon déroulement et à l'aboutissement de ce mémoire.

Je remercier tout d'abord DIEU*, qui m'a donné le courage, la patience et le pouvoir pour accomplir ce travail.*

Ensuite, je tien à exprimer mes profonde gratitude et reconnaissance envers mon encadreur Madame SABRI Salima et mon co-encadreur Madame YAICI Malika, Enseignantes à l'université de Béjaïa, pour m'avoir proposé un thème de recherche passionnant, pour avoir su m'orienter dans mon travail. je tien à les remercier particulièrement pour leurs disponibilités constante et à l'attention quotidienne qu'elles ont apportées à ce travail.

J'exprime ainsi toute mes reconnaissance aux membres de jury, pour avoir fait l'honneur d'accepter, d'examiner et d'apporter leur jugement sur ce travail de mémoire de fin d'étude master.

J'adresse mes remerciements pour tous les enseignants de l'université de Béjaïa, et je spécifie ceux de la Faculté des Sciences Exactes, pour toutes les connaissances acquises au cours de ma formation.

Mes reconnaissance et mes affection totale vont à mes parents pour leur soutien, leur présence et encouragements à aller plus loin, qui ont toujours cru en moi, aucun mot n'exprime l'immense gratitude qui m'anime.

A ma grande soeur, un grand merci pour ton aide, ton bonne humeur et tes encouragements.

Enfin, ma famille, mes amis et toutes les personnes qui m'ont aidés d'une manière ou d'une autre au cours de mes études et qui ne sont pas citées dans ces lignes trouvent ici l'expression de mes plus sincère reconnaissance.

Table des matières

Table des matières **II**

Liste des figures **VI**

Liste des tableaux **VII**

Liste des abréviations **VIII**

Introduction générale **1**

1 Généralité sur les systèmes ubiquitaires et la reconnaissance d'activités humaines 4

1.1 Introduction 4

1.2 L'informatique ubiquitaire : Objectifs et Caractéristiques 4

1.3 Technologies liées à l'informatique ubiquitaire 7

1.3.1 Réseaux de capteurs 7

1.3.2 Plasticitè des interfaces homme-machine 7

1.4 Equipements d'un système ubiquitaire 7

1.4.1 Les dispositifs mobiles 7

1.4.2 Les réseaux filaires et sans fil 8

1.4.3 Les middleware (Intergiciel) 9

1.5 Défis de l'informatique ubiquitaire 9

1.5.1 Distribution 9

1.5.2 Mobilité 10

1.5.3 Interopérabilité . . . 10
1.5.4 Hétérogénéité . . . 10
1.5.5 Ressources limitées des dispositifs . . . 10
1.5.6 Sensibilité aux Contexte . . . 10
1.5.7 Sécurité et Confidentialité . . . 11
1.6 Domaines d'application . . . 11
1.7 La reconnaissance d'activités humaines . . . 12
1.7.1 Chaîne de reconnaissance d'activité . . . 13
1.7.2 La classification . . . 15
1.7.3 Approches de reconnaissance d'activités . . . 15
1.8 Conclusion . . . 17

2 Clustering **18**
2.1 Introduction . . . 18
2.2 Définition du clustering . . . 18
2.3 Les étapes principales du clustering . . . 20
2.4 Distance et similarité (mesure de distance et fonctions de similarité) . . . 20
2.4.1 Mesures de distance . . . 21
2.4.2 Fonctions de similarité . . . 24
2.5 Principales approches de clustering . . . 24
2.5.1 Les méthodes hiérarchiques . . . 24
2.5.2 Les méthodes de partitionnement . . . 26
2.5.3 Les méthodes basées sur la densité . . . 26
2.5.4 Les méthodes basées sur la grille . . . 27
2.6 Caractéristiques des méthodes de clustering . . . 31
2.7 Technique de validation de clustering . . . 31
2.8 Conclusion . . . 32

3 Clustering hiérarchique **33**
3.1 Introduction . . . 33

3.2 Le clustering hiérarchique 33
3.3 Les algorithmes de clustering hiérarchique 34
3.3.1 SAHN (Sequential Agglomerative Hierarchical Non-overlapping) . . 34
3.3.2 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) 39
3.3.3 CURE (Clustering Using Representatives) 42
3.3.4 ROCK (RObust Clustering using linKs) 45
3.3.5 CHAMELEON 47
3.3.6 COBWEB 51
3.4 Conclusion 54

4 Proposition **55**
4.1 Introduction 55
4.2 Jeu de données 55
4.3 Description de la proposition 56
4.4 Apprentissage 58
4.4.1 Construction des vecteurs caractéristiques 58
4.4.2 Clusterisation des vecteurs caractéristiques 60
4.4.3 Étiquetage des clusters 62
4.5 La reconnaissance 63
4.6 Validation 66
4.7 Conclusion 67

Conclusion générale **68**

chapter**Références bibliographique**

Table des figures

1.1 Vision globale d'un environnement ubiquitaire 5
1.2 Chaine de reconnaissance d'activités 13
1.3 Principe de la classification 14

2.1 Distance intra-classe et inter-classe 19
2.2 Les différentes étapes du clustering 20
2.3 Les types du clustering hiérarchiques 25
2.4 Clustering basé sur la densité 27
2.5 La représentation en grille 28

3.1 Exemple de dendrogramme et la détermination des clusters [25]. 34
3.2 La classification ascendante hiérarchique [48]. 35
3.3 Schéma de classification Single-Link [48]. 36
3.4 Schéma de classification Complete-link [49]. 37
3.5 Schéma de classification Average-Link [48]. 37
3.6 Schéma de classification Centoid-link [48]. 38
3.7 BIRCH 40
3.8 CURE (Clustering Using Representatives) 43
3.9 Présentation de CURE [54]. 44
3.10 ROCK [56]. 47
3.11 Data sets [57]. 48

4.1 Dendogramme obtenu de la validation de la méthode de clustering hiérarchique proposée. 66

Liste des tableaux

2.1 Taxonomie des méthodes de clustering 29
2.2 Avantage et inconvénients des méthodes de clustering. 30

4.1 Une partie de séquence d'évènements de capteur à changement d'état 56
4.2 Exemple d'un ensemble de données 64

Liste des abréviations

ALINK	Average-Link
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CF	Cluster Feature
CLARA	Clustering Large Applications
CLARANS	Clustering Large Applications based on RANdomized Search
CLINK	Complete-link
CURE	Clustering Using Representatives
DBSCAN	Density Based Spatial Clustering of Applications with Noise
DBCLASD	Distribution-Based Clustering Algorithm for Clustering LArge Spatial Datasets
IHM	Interface Homme-Machine
KPPV	K plus proche voisins
LAN	Local Area Network
MAN	Metropolitan Area Network
PAN	Personal Area Network
PDA	Personal Digital assistants
RFID	Radio Frequency IDentification
ROCK	RObust Clustering using linKs
SAHN	Sequential Agglomerative Hierarchical Non-overlapping
SLINK	Single-Link
STING	ST atistical INformation Grid
WAN	World Area Network

Introduction générale

Le monde dans lequel nous vivons est un monde dominé par les technologies informatiques qui ne montrent aucun signe de ralentissement. Un exemple de cette nouveauté technologique mondiale est le smartphone, qui est essentiel dans notre vie quotidienne et qui nous permet d'effectuer de diverses tâches aussi intéressantes les unes que les autres tel que passer des appels ou bien trouver des lieux utilisant des cartes en ligne avec l'aide du GPS et beaucoup d'autres. Toutes ces tâches réalisées par les dispositifs informatiques sont effectuées grâce au multiples capteurs et objets connectés qui nous permet ainsi de manipuler et d'interagir avec ces dispositifs communicants. Un tel environnement dynamique est appelé un système ubiquitaire.

L'une des caractéristiques des systèmes ubiquitaires est leurs manipulation d'un grand volume de données qui est en augmentation extrêmement rapide. Il est donc nécessaire de penser à concevoir des méthodes efficaces permettant d'extraire, de classer, de manipuler et de mettre en forme les informations qu'elles peuvent contenir afin de reconnaître les activées humaines. Avec l'augmentation des capacités de traitement d'information, plusieurs méthodes de classification de données ont été développées [1], dont l'objectif est de regrouper les objets qui se ressemblent et séparer ceux que ne se rassemblent pas (classification) et d'extraire ainsi de l'information pertinente à partir de volumes importants de données afin de prendre les bonnes décisions.

Le processus de regroupement d'un ensemble d'objets en classes d'objets similaires est

appelé clustering. Un cluster est une collection d'objets de données qui sont similaires les uns aux autres dans le même cluster et sont différents des objets d'autres clusters [2]. La classification est un moyen efficace pour distinguer des groupes ou des classes d'objets et consiste souvent de partitionner l'ensemble de données en groupes en fonction de la similitude des données, puis attribuer des étiquettes à chaque cluster [2].

Le clustering de données est en cours de développement et les domaines de recherche qui y contribuent comprennent l'exploration de données, les statistiques, l'apprentissage automatique, la technologie des bases de données spatiales, le marketing et beaucoup d'autre. En raison des énormes quantités de données collectées dans les bases de données, l'analyse de cluster est récemment devenue un sujet très actif dans la recherche d'exploration de données. Les techniques de clustering sont considérées comme des outils efficaces pour partitionner des ensembles de données afin d'obtenir des clusters d'objets homogènes et ces techniques font partie des techniques d'apprentissage automatique les plus connues. Dans notre mémoire, nous nous intéressons aux méthodes de clustering hiérarchique qui est un exemple d'apprentissage non supervisé.

Dans le domaine de la reconnaissance des activités humaines dans les systèmes ubiquitaires, la problématique consiste à pouvoir reconnaître et à comprendre les activités effectuées par les utilisateurs. Il existe de nombreuse techniques de classification de données, mais en général elles rencontrent des difficultés pour gérer les valeurs aberrantes et mènent parfois à des résultats incorrects. Dans ce mémoire de recherche, on a proposé une nouvelle méthode de clustering hiérarchiques permettant de remédier aux limites des méthodes existantes. Notre méthode consiste a choisir des vecteurs représentatifs qui sert comme des clusters distincts, puis pour chaque cluster retrouver les vecteurs qui lui sont le plus similaire.

Notre travail se répartie en quatre chapitre. Dans le premier, nous nous sommes penchés sur la description des systèmes ubiquitaires, leurs caractéristiques et objectifs, ainsi que sur les principaux concepts de la reconnaissance des activités humaines.

Nous avons consacré le deuxième chapitre pour approfondir nos connaissances dans la classification et les différentes approches de clustering de données.

Dans le troisième chapitre, nous présentons les méthodes de clustering hiérarchiques les plus connus et les plus efficaces dans la reconnaissance des activités humaines.

Le dernier chapitre, est consacré à proposer une nouvelle méthode de clustering hiérarchique afin de pallier les limites des méthodes existantes.

En fin, nous avons débouchés sur une conclusion dans laquelle on a résumé l'essentiel des résultats obtenus.

Chapitre 1

Généralité sur les systèmes ubiquitaires et la reconnaissance d'activités humaines

1.1 Introduction

Depuis toujours, l'informatique ne cesse d'évoluer et nous sommes passés du stade de l'ordinateur central associé à plusieurs individus au stade de la multiplication des objets connectés pour une même personne. Cette évolution vers l'informatique ubiquitaire, nous conduit à voir émerger de plus en plus de dispositifs mobiles communicants, équipés de capteurs permettant une instrumentation de l'environnement et capables d'interagir entre eux. Ce qui nous permet d'avoir plusieurs données et de les analyser afin de reconnaître les activités humaines [3]. C'est ce besoin d'accès à l'information de partout et à tout instant qui a fait apparaître le concept de l'informatique ubiquitaire.

Ce premier chapitre introduit des généralités sur les systèmes ubiquitaires et la reconnaissance d'activités humaines. Pour cela nous allons voir dans un premier temps la définition de l'informatique ubiquitaire à savoir ses objectifs et caractéristiques ainsi que ses défis et domaine d'application. Dans un second lieu nous décrirons le concept de la reconnaissance d'activités humaines.

1.2 L'informatique ubiquitaire : Objectifs et Caractéristiques

Le concept de l'informatique ubiquitaire a été introduit par Mark Weiser en 1991 dans son article intitulé « The computer for 21st Century ». Le terme original est « Ubiquitous computing » un mot latin qui signifie « n'importe quand, n'importe où », il désigne le fait que l'informatique est omniprésente [4].

L'informatique ubiquitaire est aussi appelé informatique diffuse, informatique pervasive, intelligence ambiante, informatique omniprésente, ou encore internet des objets. Mais l'idée reste la même, c'est l'environnement informatique dans lequel le calcul est partout et les fonctionnalités informatiques sont intégrées et reliées à tout les composants de l'environnement afin que toute personne peut communiquer, échanger et partager des informations à tout moment, à n'importe quel endroit et avec tout média. Cette vision de l'informatique est décrite par la célèbre citation de Mark Weiser :

«Les technologies les plus profondes sont celles qui disparaissent. Ils se tissent dans le tissu de la vie quotidienne jusqu'à ce qu'ils ne se distinguent pas de lui. » [4].

Dans l'informatique ubiquitaire, les capacités informatiques s'intègrent dans la vie de tous les jours, dans les différents objets qui nous entourent, Un exemple d'un tel environnement est représenté par la figure 1.1.

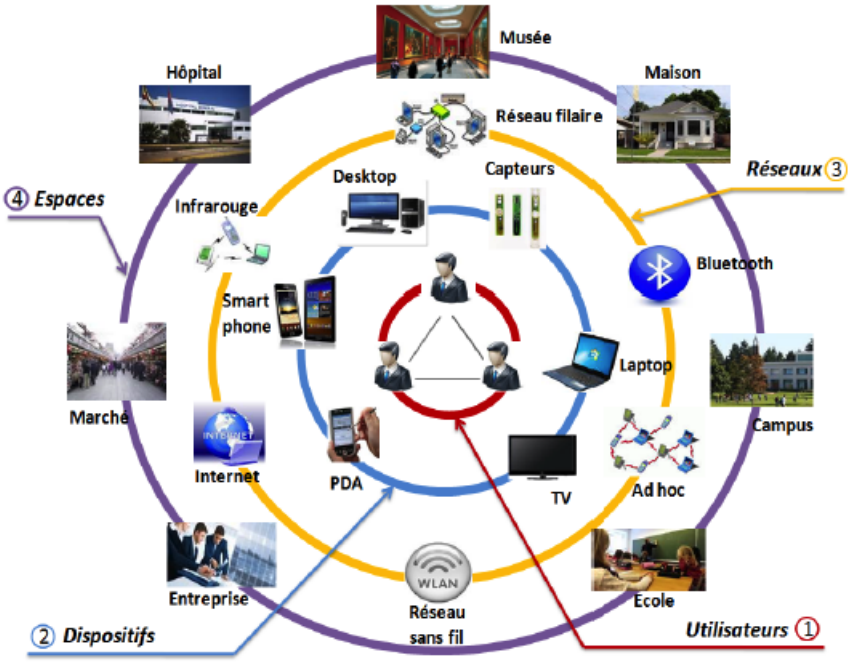

FIGURE 1.1 – Vision globale d'un environnement ubiquitaire [5].

Ainsi les objectifs des systèmes informatiques récents tel que la vidéosurveillance, les smartphones ou encore les différents capteurs ne sont plus restreints à l'exécution de tâches commandées par l'utilisateur mais à faire communiquer plusieurs systèmes mobiles ou fixes pour fournir des services personnalisés à un seul usager [3].

Selon Ruimin et al. [6] un système informatique ubiquitaire doit avoir les caractéristiques suivantes :

- **Pervasif :** il doit être partout, accessible de n'importe où ;
- **Embarqué (Intégré) :** il doit vivre dans notre monde, le sentir et l'affecter. Il fonctionne dans la périphérie de l'attention de l'utilisateur de manière a ce que personne ne remarquera sa présence ;
- **Mobile :** il doit permettre aux utilisateurs et aux calculs de se déplacer librement, selon leurs besoins ;
- **Adaptable :** il doit fournir la flexibilité et la spontanéité en réponse aux changements des exigences de l'utilisateur et les conditions d'exploitation ;
- **Puissant et efficace :** il doit se libérer des contraintes imposées par les ressources matérielles limitées, répondre aux contraintes du système imposé par les exigences de l'utilisateur et de la puissance de calcul disponible ou la bande passante de communication ;
- **Intentionnel :** il faut permettre aux personnes de nommer (demander) les services et les objets logiciels par intention.

1.3 Technologies liées à l'informatique ubiquitaire

1.3.1 Réseaux de capteurs

Un réseau de capteurs sans fil est un réseau ad hoc d'un grand nombre de nœuds, qui sont des micro-capteurs capables de recueillir et de transmettre des données d'une manière autonome [7].

1.3.2 Plasticitè des interfaces homme-machine

En interaction homme-machine, la plasticité dénote la capacité d'adaptation d'une Interface Homme-Machine (IHM) à son contexte d'usage dans le respect de ses bonnes propriétés pour l'humain : propriétés fonctionnelles (les bons services) et non fonctionnelles (une bonne qualité de service). Les dispositifs d'interaction se diversifient par leur forme et leur finalité. Ces dispositifs utilisent des techniques intelligentes d'interaction impliquant la participation des gestes tactile multiple, la commande vocale et le contrôle du regard etc, ce qui permet de créer des interfaces adaptées au contexte [7, 8].

1.4 Equipements d'un système ubiquitaire

L'informatique ubiquitaire a évolué grâce au développement rapide de la technologie que se soit des composants matériels ou logiciels [9]. Dans ce qui suit nous citons quelques équipements de l'informatique ubiquitaire.

1.4.1 Les dispositifs mobiles

- **Personal Digital assistants (PDA) :** C'est un ordinateur de poche sous forme de boîtier compact de petite taille qui possède un écran tactile et qui est à la fois micro-ordinateur, calculatrice, agenda, réveil, téléphone, fax, ... etc.
- **Le smartphone :** Désigne un téléphone mobile intelligent doté de fonctionnalités évoluées qui s'apparentent à celles d'un ordinateur : navigation sur Internet, lecture

de vidéos, de musique, jeux vidéo, courrier électronique, vidéoconférence, bureautique légère...

- **Tablette :** est un appareil mobile, généralement doté d'un système d'exploitation mobile et de circuits de traitement d'affichage à écran tactile, et d'une batterie rechargeable dans un boîtier unique, fin et plat. Les tablettes, étant des ordinateurs, font ce que font les autres ordinateurs personnels, mais manquent de certaines capacités d'entrée / sortie (E / S) que les autres ont. Ils sont munis d'un système de reconnaissance de l'écriture naturelle et parfois de reconnaissances vocales.
- **Les puces RFID (Radio Frequency IDentification) :** Désigne une méthode utilisée pour stocker et récupérer des données à distance en utilisant des balises métalliques appelé "Tag RFID", qui peuvent être collées ou incorporées dans des produits, réagissent aux ondes radio et transmettent des informations à distance.
- **Capteurs :** Les capteurs sont des dispositifs qui peuvent être utilisés pour détecter l'interaction entre une personne et son environnement. Il peut être un capteur de luminosité, d'humidité, de température, de son, un détecteur de présence ou encore une caméra.

1.4.2 Les réseaux filaires et sans fil

Un réseau est un ensemble d'éléments interconnectés à l'aide d'une liaison filaire ou sans fil, qui permet à des machines d'échanger des informations tel que des données informatiques, des images, des sons, des signaux de télécommandes, etc [9]. Selon leur dimension, les réseaux sont classés en quatre catégories :

- **WAN (World Area Network) :** Ce sont des réseaux à l'échelle d'une région, d'un pays voire de dimension internationale et souvent publics. Comme par exemple Internet, le réseau téléphonique, etc.
- **MAN (Metropolitan Area Network) :** Ces réseaux sont à l'échelle d'une ville, en général développés par des Collectivités locales ou par des associations de particuliers.

- **LAN (Local Area Network) :** Ce sont les plus nombreux, pour la très grande majorité privée, développés à l'échelle d'une entreprise ou d'un site industriel.
- **PAN (Personal Area Network) :** Ce sont de très petits LAN, à l'échelle résidentielle comme le réseau domestique, les ordinateurs et le téléviseur de la maison. On range aussi dans cette catégorie des réseaux de très faible portée, à la taille de l'individu. C'est par exemple la liaison entre un téléphone portable et son oreillette.

1.4.3 Les middleware (Intergiciel)

Un intergiciel (en anglais middleware) est un logiciel servant d'intermédiaire de communication entre plusieurs applications, généralement complexes ou distribuées sur un réseau informatique. Ils offrent des services de haut niveau liés aux besoins de communication des applications (temps réel, sécurisation, sérialisation, transaction informatique ...) et se situe au-dessus du système d'exploitation et en dessous des applications de son hôte [9].

1.5 Défis de l'informatique ubiquitaire

Depuis son apparition, l'informatique ubiquitaire n'a pas cessé de mettre l'accent sur la vision de la technologie de future, et comme toute autre discipline, les systèmes ubiquitaires affrontent de nombreux défis, parmi eux, ont cite :

1.5.1 Distribution

Un système distribué est un système disposant d'un ensemble d'entités communicantes, installées sur une architecture d'ordinateurs indépendants reliés par un réseau de communication [10], dans le but de résoudre en coopération une fonctionnalité applicative commune.

Autrement dit, un système distribué est défini comme étant un ensemble des ressources physiques et logiques géographiquement dispersées et reliées par un réseau de communication dans le but de réaliser une tâche commune. Cet ensemble donne aux utilisateurs une vue unique des données du point de vue logique[11].

1.5.2 Mobilité

La mobilité réfère à la capacité d'accéder a des services ou a des applications indépendamment de la localisation physique, comportements et mouvements des utilisateurs [12].

1.5.3 Interopérabilité

De manière général, le terme interopérabilité désigne la capacité que possède un produit (un objet, un système...) à fonctionner, communiquer avec d'autres produits ou systèmes différents et sans restriction [7].

1.5.4 Hétérogénéité

Un système ubiquitaire peut être constitué d'une multitude de dispositifs hétérogènes (ordinateurs portables, téléphones intelligents, tablettes, assistants numériques, etc.) ayant des capacités différentes, reliés par des réseaux de différentes caractéristiques conformes aux différentes normes. Ainsi que d'une multitude de services et de composants logiciels développés dans des langages de programmation différents et déployés sur des plates-formes diverses. Par conséquent, une caractéristique essentielle de systèmes ubiquitaires consiste à créer des technologies de base qui sont en mesure de gérer cette hétérogénéité [7].

1.5.5 Ressources limitées des dispositifs

Le développement des applications mobiles est influencé par la limitation de capacités des équipements utilisés dans l'environnement mobile (téléphone portable, PDA, etc.), y compris leurs capacités d'affichage et leur puissance de calcul [13].

1.5.6 Sensibilité aux Contexte

Le développement de l'informatique sensible au contexte constitue l'un des développements majeurs de la vision de Weiser [4]. La sensibilité aux Contexte est définie comme étant

la capacité d'un système à découvrir et à réagir aux changements dans l'environnement où il se trouve [7].

1.5.7 Sécurité et Confidentialité

la sécurité et la confidentialité sont des défis majeurs de l'informatique ubiquitaire, qui a comme objectif la protection des informations contre toutes divulgations, altération ou destruction, ainsi que la protection des composants matériels, logiciels ou informationnels de diverses menaces [14].

1.6 Domaines d'application

L'informatique ubiquitaire a pour objectif d'interconnecter tous les domaines de la vie, et donc de permettre un flux omniprésent de données, d'informations, et même de connaissance. Dans ce qui suit quelques exemples de son application.

— La maison intelligente est un domaine d'application potentiellement important pour l'informatique ubiquitaire. Les équipements dans une maison deviennent des objets intelligents interconnectés grâce a des processeurs et des capteurs intégrés [15]. L'objectif principal de la maison connectée est de faciliter le quotidien des occupants. Les propriétaires peuvent vérifier si les lumières sont éteintes même s'ils ne sont pas chez eux, d'ouvrir ou de fermer une porte à distance, de déclencher ou d'éteindre un appareil quelconque depuis leurs bureaux ;

— La santé et la médecine offrent un large domaine d'application pour l'informatique ubiquitaire, elle permet de recueillir des informations sur l'état des patients. De cette façon les médecins peuvent avoir des informations détaillées sur les conditions dans lesquelles la maladie se déclare (présence d'allergènes, conditions de température. . .), et de prévenir ainsi les comportements dangereux (incompréhension du mode de prise du médicament, oubli ou non-respect de conditions d'hygiènes propres à certains traitements...) [15] ;

— Le transport est aussi l'un des premiers produits du notre vie quotidienne dans lequel l'informatique ubiquitaire est mise à profit. Les voitures d'aujourd'hui contiennent une multitude de systèmes d'assistance au conducteur qui visent à aider le conducteur à guider le véhicule. Ce qui apporte plus de confort de conduite, de confort pour les occupants, et plus de sécurité [15].

Aujourd'hui, plusieurs domaines s'orientent de plus en plus vers l'adoption des applications ubiquitaires pour gagner la fidélité de leurs clients en leur proposant des services mobiles et intelligents, que ce soit dans le domaine public, médicale ou bien dans l'apprentissage (E-learning). Ce qui permet ainsi l'évolution de l'informatique ubiquitaire.

1.7 La reconnaissance d'activités humaines

La reconnaissance d'activités humaines est un domaine en plein développement, qui a émergé dans la recherche sur les interactions hommes-machines et sur l'informatique ubiquitaire.

La reconnaissance d'activités humaines vise à décrire, analyser, reconnaitre, comprendre et suivre les activités et les mouvements de personnes, d'animaux ou d'objets. Elle peut être utile pour détecter tôt les comportements anormaux de certaines personnes : difficultés dues à l'âge ou à une maladie. Les systèmes de reconnaissance doivent être des systèmes a temps réel, réactifs et fiables.

Plusieurs types de technologies ont été mis en œuvre pour la reconnaissance des activités des personnes à domiciles tel que les capteurs, les systèmes de reconnaissance sonore et les systèmes de reconnaissance vidéo [16].

1.7.1 Chaîne de reconnaissance d'activité

La reconnaissance des activités humaines se base sur l'extraction des données à partir des différents capteurs, tel que les bracelets et les téléphones intelligent, permettant ainsi l'extraction de nouvelle information et l'apprentissage automatique a partie des données d'entrée.

Tout le processus partant de l'extraction des données à partir des capteurs et allant jusqu'à l'entrainement du modèle de reconnaissance d'activité est appelé la Chaine de Reconnaissance d'Activité. Ce processus est classiquement divisé en différentes étapes qui sont toutes d'une importance cruciale : acquisition des données, prétraitement, segmentation, extraction de caractéristiques et classification comme représentés sur la figure 1.2 [16, 17].

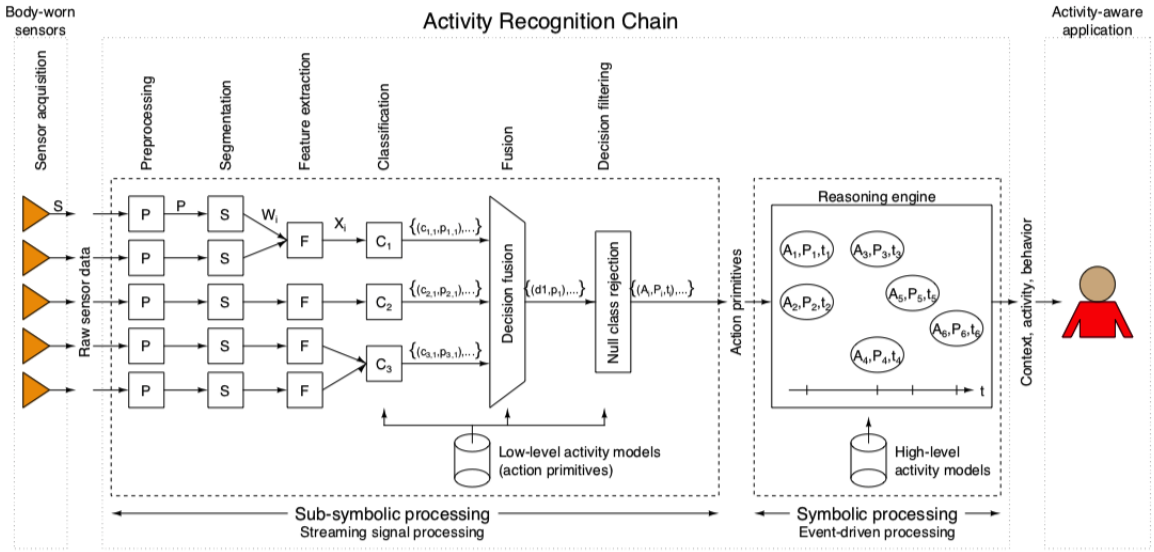

FIGURE 1.2 – Chaine de reconnaissance d'activités [17].

- **Acquisition des données :** Permet de recueillir les données d'entrée à partir de tous les capteurs ;
- **Prétraitement :** Permet le nettoyage des données brutes afin de réduire les bruits des données de capteurs ;
- **Segmentation :** Permet l'extraction des sous-ensemble de données qui contiennent une activité, sur lequel le processus de reconnaissance est appliqué. Les segments sont identifiés par leur instant de début et leur instant de fin dans le flux de données.

Cela peut être réalisé à l'aide de deux techniques : la technique de fenêtre glissante ou l'analyse de changement d'activité ;

- **Extraction de caractéristiques :** Les données contenues dans chaque segment sont transformées en un vecteur de caractéristiques ;
- **Classification :** Un classifieur utilise les caractéristiques communes comme données d'entré pour pouvoir reconnaître des classes (les activités de l'utilisateur).

Après la classification, les informations sont combinées pour avoir une décision sur l'activité qui a eu lieu. L'objectif principal de la classification est d'identifier les classes auxquelles appartiennent des entités à partir des caractéristiques utilisées comme point d'entré et elle se déroule en deux phases principales : l'apprentissage et la prise de décision. Le module d'apprentissage permet de construire un modèle qui est utilisée par la suite pour fournir la décision concernant l'appartenance d'un échantillon inconnu a l'une des classes prédéfinies.

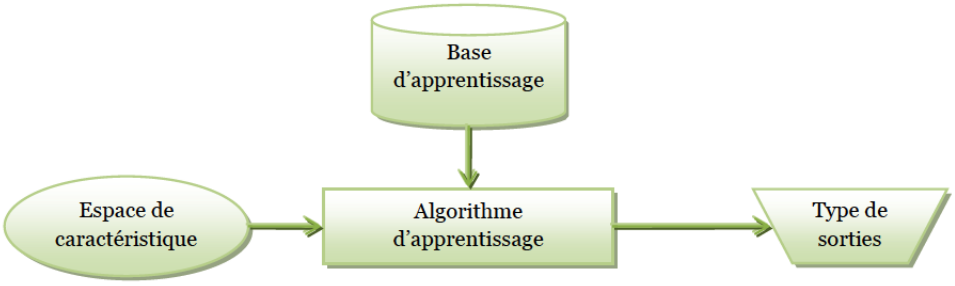

FIGURE 1.3 – Principe de la classification [18].

1.7.2 La classification

Les méthodes de classification sont subdivisées en deux grandes approches selon la connaissance des échantillons d'apprentissage ou non.

- **Classification supervisée :** Il s'agit d'examiner des caractéristiques d'un élément nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini. On dispose au départ d'un échantillon dit d'apprentissage dont le classement est connu. Cet échantillon est utilisé pour l'apprentissage des règles de classement [18]. Parmi ces méthodes on peut citer l'algorithme de Bayes, Arbre de décision, Random forest, K plus proche voisins (KPPV), Les réseaux de neurones, etc.
- **Classification non supervisée :** Dans une classification non supervisée, les classes sont encore inexistantes. Pour cette démarche, on dispose donc au départ d'un ensemble d'objets. L'idée consiste à découper l'ensemble des objets en groupes (clusters) de tel sort que les caractéristiques des objets dans un même cluster soient similaires et les caractéristiques des objets dans des clusters différents soient distinctes [18]. De même, il existe aussi dans cette approche plusieurs algorithmes : K-means, Expectation-maximization, Hierarchical clustering, Self organizing map, Isodata, etc.

1.7.3 Approches de reconnaissance d'activités

La reconnaissance d'activités humaines a subi depuis son apparition plusieurs recherches, ce qui a permis la naissance de différentes approches et techniques qu'on peut généralement regrouper en trois catégories :

Les approches logiques

Les approches logiques se basent sur l'utilisation de la logique de premier ordre pour sélectionner, parmi les modèles d'activités, l'ensemble des activités qui peuvent expliquer un ensemble d'actions observées et de formaliser ainsi le processus d'inférence de l'activité en cours [19].

Le grand avantage des approches logiques est leur efficacité sur une très large base de connaissances. Elles permettent d'ignorer rapidement la majorité des activités et de n'en garder qu'un petit ensemble. Les traitements et les calculs se font alors sur un ensemble réduit, ce qui donne un meilleur temps de réponse. Cependant, elle ne prend pas en compte les erreurs potentielles des acteurs. De ce fait, son modèle interprète les erreurs comme des changements d'activités. De plus, l'approche logique part du principe que toutes les activités possibles sont connues et répertoriées, ceci n'est pas réaliste dans le contexte où l'acteur peut être erratique et peut se tromper [20].

Les approches probabilistes

Ces approches se basent sur des modèles probabilistes (modèles de Markov cachés et ses extensions, réseau bayésien ...). Pour les utiliser, une probabilité initiale est assignée à chaque plan de la base de données selon les actions observées. Entre autres, lorsque l'acteur fait quelque chose, la probabilité des plans qui contiennent cette action augmente. Ainsi, le plan ayant le plus haut score est probablement l'activité réalisée par l'agent observé [21].

Les approches probabilistes règlent tous les problèmes des approches logiques. Elles permettent de résoudre le problème d'irrationalité de l'acteur du fait qu'une action qui n'a aucun rapport avec l'objectif visé ne l'éliminerait plus de l'ensemble sélectionné, mais elle va juste modifier d'une façon non significative les différentes probabilités calculées. L'inconvénient majeur des approches probabilistes est la lourdeur des calculs et du traitement qui ralenti grandement le temps d'exécution alors que la reconnaissance d'activités doit se faire en temps réel pour permettre à l'agent ambiant d'intervenir au moment opportun. La complexité des calculs vient du problème du nombre d'hypothèses puisque pour chaque action observée, les probabilités de tout les modèles d'activités de la base de connaissance doivent être mises a jour [21].

Les approches basées sur l'apprentissage automatique

Les approches basées sur l'apprentissage automatique ne se fondent plus uniquement sur les actions observées pour discriminer des modèles d'activités et de garder que les plus probables, mais elles leurs permet aussi d'utiliser d'autres contraintes spatiales ou temporelles. Elles demandent l'exploitation d'un volume élevé de données, générées par les capteurs, de nature complexe comme les données temporelles. Pour cette raison, ces approches font recours au domaine de l'apprentissage automatique [22].

1.8 Conclusion

Dans ce chapitre nous nous somme concentrés sur le concept de l'informatique ubiquitaire, ainsi que sur la chaine de reconnaissance des activités humaines. Dans le chapitre 2 nous allons mettre l'accent sur la phase de la classification qui est une étape très importante dans la reconnaissance d'activités humaines et nous allons plus particulièrement nous intéresser aux techniques de classification d'activités non supervisées (clustering).

Chapitre 2

Clustering

2.1 Introduction

Dans de grandes collections documentaires, le regroupement automatique (clustering) des documents selon leurs similarités facilite l'accès aux informations, en faisant émerger des regroupements de documents organisés idéalement autour d'un même thème. Ce problème a été abordé dans de nombreux contextes et par des chercheurs dans beaucoup de disciplines, ce qui reflète son utilité et son importance dans l'analyse exploratoire des données [23].

Le clustering est un processus qui regroupe un ensemble d'objets similaires en clusters de telle sorte que les données du même cluster aient des caractéristiques identiques, et celles appartenant à des clusters distincts soient dissimilaires [24].

En raison de l'ambiguïté concernant les problèmes de clustering, différentes approches ont étés proposées dans la littérature afin d'améliorer son utilisation sur des applications spécifiques. Ces techniques se diffèrent dans leurs principes, propriétés, paramètres et formes générales du partitionnement généré.

Dans ce chapitre, en premier lieu nous présentons le clustering et ses différentes étapes. En second lieu nous nous intéressons à la notion de similarité et de distance qui jouent un rôle important dans les algorithmes de clustering. Ensuite, nous allons citer les principales approches de clustering, leur caractéristiques et les techniques de validation d'un algorithme de clustering. En dernier lieu nous parlerons des limites de ses algorithmes.

2.2 Définition du clustering

Le clustering est la tâche qui consiste à regrouper des ensembles d'objets (physiques ou abstraits) de manière automatique. Cette approche consiste à faire émerger des groupes au sein d'un ensemble d'éléments sans aucune information, c'est ce qu'on appel un regroupement non-supervisée de telle manière que les objets d'un même groupe sont plus proches les uns aux autres que celles des autres groupes. Cette tâche est appelée, selon les domaines, classification non supervisée, classification automatique ou encore le clustering. Les groupes créés sont appelés "clusters" ou "grappes".

La classification non supervisée de données permet d'organiser thématiquement une collection d'objets de façon à faciliter l'accès à l'information, ou à proposer une vue synthétique du contenu d'un ensemble d'objets [23]. L'idée est donc de découvrir des groupes au sein des données de telle sorte que les données du même cluster aient des caractéristiques similaires à l'intérieur de chaque cluster, les données sont regroupées selon une caractéristique commune. L'outil de classification est un algorithme qui mesure la proximité entre chaque élément à partir de critères définis. En clair, le clustering cherche à faire des classes telle que [23, 24] :

— Les différences intra-classe soient minimales pour obtenir des clusters.

— Les différences inter-classe soient maximales afin d'obtenir des sous-ensembles bien différenciés.

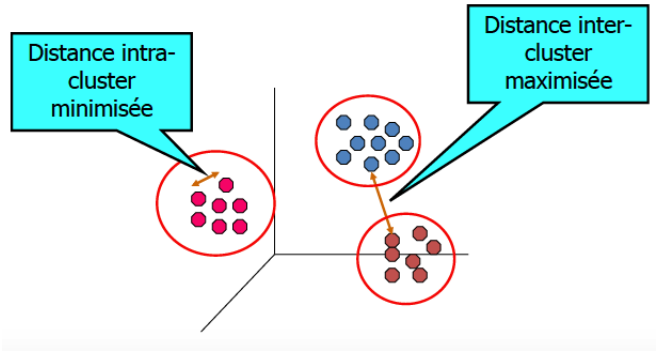

FIGURE 2.1 – Distance intra-classe et inter-classe [23].

Le but des algorithmes de clustering est de donner un sens aux données et d'extraire de la valeur à partir de grandes quantités de données structurées et non structurées. Il s'agit d'une technique d'analyse statistique des données très utilisée dans de nombreux domaines, y compris l'apprentissage automatique, la reconnaissance de formes, le traitement de signal et d'images, la recherche d'information, la bio-informatique, la compression de données et l'infographie, etc [25].

2.3 Les étapes principales du clustering

Le processus de clustering se divise en trois étapes majeures [24, 25] :

— La préparation des données : Les objets sont décrits par des variables qui sont de différentes natures (quantitatives, qualitatives, variables structurées...)

— L'algorithme de clustering : un algorithme de clustering est choisi selon la nature des variables en entrée.

— L'exploitation des résultats de l'algorithme : Cette étape permet de distinguer et d'analyser les classes pertinentes obtenue, afin d'aider à orienter le traitement suivant.

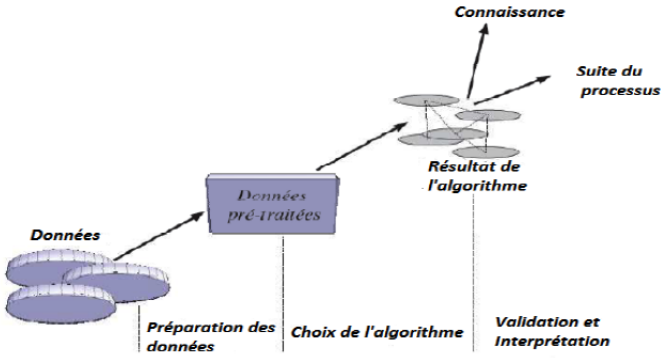

FIGURE 2.2 – Les différentes étapes du clustering [25].

2.4 Distance et similarité (mesure de distance et fonctions de similarité)

Les principes fondamentaux du concept des techniques de mise en cluster permettent aux composantes d'un cluster d'être très semblables les une aux autres. Les mesures de similarité et de dissimilitude ont une signification distincte et pratique [26, 27]. Bisson [28] a annoncé en 2000 :

« Tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou une autre, un opérateur de

similarité dont le but est d'établir les ressemblances ou les relations qui existent entre les informations manipulées ».

Ces similitudes et ces mesures distinctes des grappes ont été déterminées à l'aide des deux types principaux de mesures employées pour estimer cette relation :

— Mesures de distance.

— Fonctions de similarité.

2.4.1 Mesures de distance

Plusieurs méthodes de clustering utilisent les mesures de distance pour déterminer la similitude ou la dissimilitude de n'importe quelle paire d'objets. La mesure de distance entre deux objets xi et xj est notée d (xi, xj). Une telle mesure devrait être symétrique et atteindre sa valeur minimale (habituellement zéro) en cas de vecteurs identiques.

Une mesure de distance est appelée mesure métrique de distance si elle satisfait les propriétés suivantes [29] :

1. Inégalité de triangle : $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k) \forall x_i; x_j, x_k \in S$
2. $d(x_i, x_j) = 0 \Rightarrow x_i = x_j \forall x_i, x_j \in S$

Où : S représente l'ensemble des objets.

— **La distance de Minkowski :**La distance de Minkowski est utilisée lorsque les attributs des objets sont de type numérique.

$$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q)} \tag{2.1}$$

Avec q un entier positive non nul

Où $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ et $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ sont deux objets p-dimensionnels.

Pour q=2, on obtient la distance euclidienne.

Pour q=1, la mesure est appelé distance de Manhattan.

Pour q=∞, c'est la métrique de Tchebychev.

La distance de Minkowski est utilisée pour les attributs à évaluation continue. Dans le cas d'objets décrits par des attributs catégoriques, binaires, ordinaux ou du type mixte, une autre mesure de distance spécifique doit être définie [25].

— **Mesure de distance pour les attributs binaires :**Dans le cas des attributs binaires, la distance entre les objets peut être calculée sur la base de la table de contingence. Un attribut binaire est symétrique si ses deux états ont le même poids. Dans ce cas, on utilise le coefficient d'appariement simple pour évaluer la dissimilitude entre deux objets [24] :

$$d(x_i, x_j) = \frac{r+s}{q+r+s+t} \tag{2.2}$$

Où q est le nombre d'attributs qui sont égaux à 1 pour les deux objets, t est le nombre d'attributs qui sont égaux à 0 pour les deux objets, s et r est le nombre d'attributs qui sont inégaux pour les deux objets.

Un attribut binaire est dit asymétrique si une valeur est moins significative que l'autre, dans ce cas la dissimilarité est calculée en utilisant le coefficient de Jaccard [24] :

$$d(x_i, x_j) = \frac{r+s}{q+r+s} \tag{2.3}$$

— **Mesure de distance pour des attributs nominaux :**Lorsque les attributs sont nominaux, deux approches peuvent être utilisées [24] :

1. L'appariement simple (Simple Matching) :

$$d(x_i, x_j) = \frac{p-m}{p} \tag{2.4}$$

Où p est le nombre total d'attributs, et m le nombre de correspondances entre les objets (égalité en valeur des attributs).

2. Créer un attribut binaire pour chaque état de chaque attribut nominal et calculer la dissimilarité comme cité précédemment.

— **Mesure de distance pour des attributs ordinaux :**Lorsque les attributs sont ordinaux, l'ordre des valeurs est significatif. Dans ces cas, les attributs peuvent être traités en tant qu'attributs numériques aprés leur remplacement par leur rang respectif sur l'intervalle [0 :1], ce remplacement est effectué comme suit [24] :

$$z_{i,n} = \frac{r_{i,n} - 1}{M_n - 1} \tag{2.5}$$

Où $z_{i,n}$ est la valeur standardisée de l'attribut n de l'objet i. $r_{i,n}$ est cette dernière valeur avant standardisation et Mn la limite supérieure du domaine de l'attribut an (la limite inférieure est supposée égale à 1).

— **Mesure de distance pour des attributs mixtes :** Dans ce cas, on peut calculer la distance en combinant les méthodes mentionnées précédemment, et ce en ajoutant le carré de chaque distance trouvée par type d'attribut à la distance totale, la distance entre les objets xi et xj sera alors [24] :

$$d(x_i, x_j) = \frac{\sum_{n=1}^{p} \delta_{ij}^{(n)} d_{ij}^{(n)}}{\sum_{n=1}^{p} \delta_{ij}^{(n)}} \tag{2.6}$$

Où p est le nombre d'attributs, $\delta_{ij}^{(n)} = 0$ si l'une des valeurs est manquante (ie, il n'ya aucune mesure de la variable n pour l'objet i ou l'objet j), où $x_{in} = x_{jn} = 0$ et la variable n est binaire asymétrique, sinon, $\delta_{ij}^{(n)} = 1$. La contribution de l'attribut n à la distance entre les deux objets est calculée selon le type d'attribut :

— Si l'attribut est binaire ou catégorique, $d_{ij}^{(n)} = 0$ si $x_{in} = x_{jn}$, sinon $d_{ij}^{(n)} = 1$.

— Si l'attribut est continu, $d_{ij}^{(n)} = \frac{|x_{in} - x_{jn}|}{max_h x_{hn} - min_h x_{hn}}$ où h parcours tous les objets non-manquants pour l'attrbut n.

— Si l'attribut est ordinal, on calcule d'abord les valeurs standardisées de l'attribut,

ces valeurs seront par la suite traitées comme de type continu.

2.4.2 Fonctions de similarité

La fonction de similarité $s(x_i; x_j)$ qui compare deux vecteurs x_i et x_j constitue une alternative aux mesures de distance [26].

Cette fonction doit être symétrique ($s(x_i; x_j) = s(x_j; x_i)$), avoir une valeur élevée lorsque x_i et x_j sont similaire, et atteindre son maximum lorsque les vecteurs sont identiques [23].

2.5 Principales approches de clustering

La raison principale pour laquelle de nombreuses approches de clustering sont utilisées est le fait que la notion de «cluster» n'est pas définie avec précision. Du coup, différentes vision de classification ont été développées, chacune d'elles utilisant un principe d'induction différent.Les méthodes de clustering sont généralement classifiées en quatre catégories majeures [30] :

— Les méthodes hiérarchiques.

— Les méthodes de partitionnement.

— Les méthodes basées sur la densité

— Les méthodes basées sur la grille

2.5.1 Les méthodes hiérarchiques

Les méthodes hiérarchiques génèrent une succession de partitions emboîtées les unes dans les autres au lieu d'une seule partition de l'espace des données [31]. Un cluster peut être divisé en sous clusters, l'ensemble des clusters étant généralement représenté par un arbre. Un objet appartient à une et une seule feuille dans la hiérarchie, mais également à son nœud père, et ainsi de suite jusqu'à la racine [24]. Les méthodes de clustering hiérarchiques se subdivise en deux grand familles :

- **Le clustering de division (descendant) :** l'ensemble d'individus est décomposé en K groupes [32]. On part d'un grand cluster que l'on divise en deux progressivement de façon à optimiser un critère donné pou obtenir au final un ensemble de singletons (des groupes qui contiennent un seul individu) [33] (Voir la figure 2.3 (a)).
- **Le clustering d'agglomération (ascendant) :** l'ensemble d'individus est décomposé en une arborescence de groupes [32]. On commence avec autant de clusters qu'il y a d'objets. Ensuite, à chaque étape, on regroupe les deux éléments (clusters) qui sont jugés les plus similaires pour terminer avec un seul grand cluster englobant toutes les données [33] (Voir la figure 2.3 (b)).

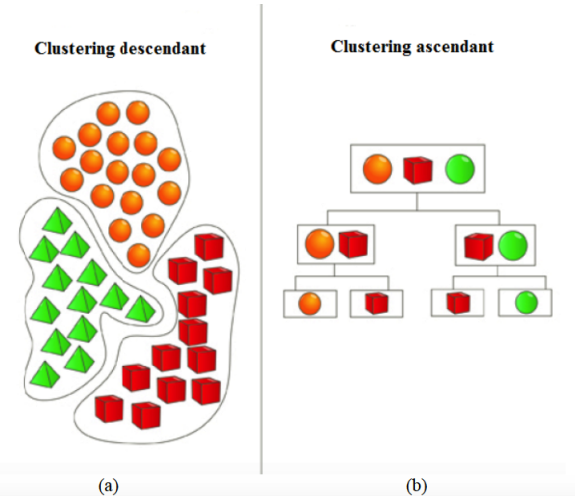

FIGURE 2.3 – Les types du clustering hiérarchiques [32].

2.5.2 Les méthodes de partitionnement

L'idée de ces méthodes est de construire k partitions et les corriger jusqu'à obtenir une similarité satisfaisante, c'est-à-dire de trouver la partition de l'espace la plus pertinente pour la formation des clusters [30].

Avec une base de données de n objets, une méthode de partitionnement construit k partions des données, où chaque partition représente un cluster, où chaque cluster doit contenir au moins un objet et chaque objet doit appartenir à un cluster exactement.

Pour obtenir une optimalité globale dans le clustering basée sur le partitionnement, il est nécessaire d'énumèrer de manière exhaustive toutes les partions possibles [25].

Dans ce cadre, des techniques sont apparues. Elles permettent d'obtenir des solutions sous-optimales mais acceptables. Parmi ces techniques les plus connues on trouve la méthode des centres mobiles K-means [34], CLARA [35] et CLARANS [36].

2.5.3 Les méthodes basées sur la densité

Le principe général de ces méthodes est l'utilisation de la densité à la place de la distance. Un point est dense si le nombre de ses voisins dépasse un certain seuil.

Ces algorithmes sont capables de découvrir des clusters de formes arbitraires, ce qui assure l'isolement des bruits et la prévention contre la construction de clusters non pertinents. Ils regroupent des objets selon des fonctions de densité spécifiques [37].

La densité des éléments dans chaque groupe est considérablement plus élevée qu'à l'extérieur du groupe et la densité dans les zones de bruit est inférieure à la densité dans l'un des groupes. Cette propriété permet facilement et sans ambiguïté de détecter des groupes et le bruit qui n'appartiennent à aucun de ces groupes [38].

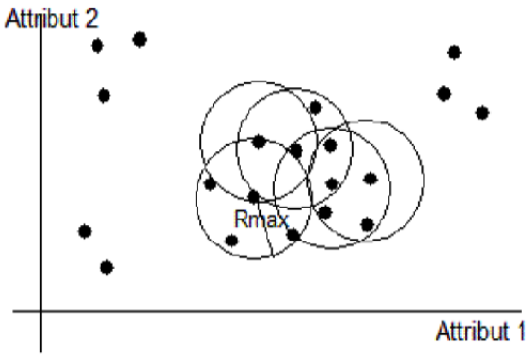

FIGURE 2.4 – Clustering basé sur la densité [37].

Cette approche se subdivise en deux types :

- **Méthodes basée sur la densité connective :** Dans cette technique de clustering, la densité et la connectivité sont mesurées en terme de distribution locale des voisins les plus proches [37].
- **Méthodes basée sur les fonctions densité :** Dans cette méthode, une fonction de densité est utilisée pour le calcul de la densité. La densité globale est définie comme la somme des fonctions de densité de tous les objets [37].

Plusieurs algorithmes exploitent les méthodes basées sur la densité dont DBSCAN [39], DBCLASD [38], . . . etc.

2.5.4 Les méthodes basées sur la grille

Ces méthodes utilisent une grille qui partitionne l'espace des données en de multiples cellules à M dimensions (M étant le nombre d'attributs). Ensuite, les densités de ces cellules bien délimitées peuvent être calculées [30]. Donc ce type d'algorithme est conçu pour des données spatiales. Une cellule peut être un cube, une région ou un hyper rectangle. En fait, elle est un produit cartésien de sous intervalles d'attributs de données [40]. Deux types d'approches sont possibles :

- **Détection de zones denses :** approche qui cherche à détecter les clusters comme des zones denses dans l'espace des données. On fusionne donc des cellules de sorte que leur regroupement ait une densité supérieure à une valeur seuil fixée.
- **Détection de zones peu denses :** approche qui vise à déceler des zones inoccupées de l'espace afin d'établir les frontières entre clusters. On se base donc sur l'existence de changements de densités au travers des limites des clusters afin de reconstituer ceux-ci.

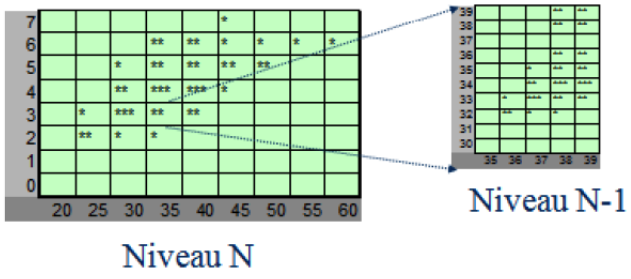

FIGURE 2.5 – La représentation en grille [30].

Comme montré sur la figure 2.5, le principe général est au lieu d'opérer sur les données on opère sur les partitions de l'espace des attributs, la partition de données étant déduite à partir de ces cellules. La principale difficulté de ces méthodes est de choisir convenablement les paramètres liés à la taille des cellules. En effet, des cellules de trop grande taille reprendront beaucoup d'objets qui ne sont pas forcément bien uniformément répartis. On pourra donc avoir des clusters peu homogènes, nécessitant d'être subdivisés par la suite. Il y a donc sous-partitionnement. Par contre, des cellules de petite taille seront généralement très denses à partir du moment où elles contiennent des objets. On aura alors tendance à détecter des frontières qui n'auraient pas lieu d'être et à avoir un sur-partitionnement [41].

Les méthodes basées sur la grille ont été proposées pour réduire l'explosion combinatoire des méthodes à base de densité qui fait suite à l'augmentation du nombre d'objets. Les algorithmes les plus connus dans cette catégorie sont STING [33], CLIQUE [42] et WaveCluster [43].

Le tableau 2.1 présente les algorithmes les plus connus qui ont été développés dans les différentes approches du clustering :

Approches de clustering	Algorithmes
Approche hiérarchique	Classification hiérarchique ascendante, Classification hiérarchique descnedante, CURE, BIRCH, L'algorithme de ward, etc.
Approche par partitionnement	K-Means, Fuzzy C-Means, IsoData, Fast Global K-Means, K-Means++, CLARA, CLARANS, etc.
Approche basé sur la densité	Denclust, Mean-scihft, DBSCAN, DBCLASD, OPTICS, etc.
Approche basé sur la grille	STING, CLIQUE, WaveCluster, etc.

TABLE 2.1 – Taxonomie des méthodes de clustering

Le tableau ci-dessous récapitule quelques avantages et inconvinients des méthodes de clustering que nous avons pu établir :

Méthodes	Avantages	Inconvénients
Les méthodes hiérarchiques	• Le résultat est représenté souvent sous la forme d'un arbre qui est facile à lire et à comprendre ; • Facilité de manipuler toutes formes de similitude ou de distance ;	• Imprécision sur les critères d'arrêt ; • La plupart des algorithmes hiérarchique ne revisitent pas les clusters une fois construits en vue de l'amélioration des résultats.
Les méthodes de partitionnement	• Facilité d'implémentation ; • Insensibles à l'ordre des données.	• Ils ne sont pas applicables en présence d'attributs non numérique ; • Le résultat final dépend du nombre d'itération effectué ; • Sensibles à la présence de bruits.
Les méthodes basées sur la densité	• Gestion des bruits ; • Effecacité du temps de calcul ; • Il n'est pas nécessaire de spécifier le nombre de cluster à l'avance. • Gestion de clusters de différentes formes et tailles.	• Ils ne sont pas capables de gérer des clusters de densités différentes.
Les méthodes basées sur la grille	• Parallélisable, mise à jour incrémentale.	• Les bords des clusters sont soit horizontaux soit verticaux, pas de diagonale.

TABLE 2.2 – Avantage et inconvénients des méthodes de clustering.

2.6 Caractéristiques des méthodes de clustering

Un algorithme de clustering doit répondre aux caractéristiques suivantes [25] :

— La capacité à gérer différents types d'attributs ;

— La découverte de clusters avec des formes arbitraires : Les clusters peuvent avoir des formes diverses.

— Besoin minimum de connaissances du domaine pour déterminer les paramètres ;

— La capacité à gérer le bruit et les exceptions ;

— Non sensible à l'ordre des données en entrée ;

— La capacité de gérer des données volumineuse ;

— Les résultats de clustering doivent être interprétables, compréhensibles et bien évidement utilisables.

2.7 Technique de validation de clustering

L'objectif principal de la validation de clusters est d'évaluer le résultat de clustering afin de trouver le meilleur partitionnement des données [40]. Chaque algorithme peut diviser les données, mais différents algorithmes ou paramètres d'entrée produisent différents résultats [44]. Ils existent en général trois approches de validation des algorithmes de clustering [44] :

- **L'évaluation externe :** il s'agit de confronter un schéma avec une classification prédéfinie. L'évaluation porte donc sur l'adéquation entre le schéma obtenu et une connaissance externe sur les données.
- **L'évaluation interne :** ce type d'évaluation utilise les données d'entrées comme référence. Ainsi, par exemple, parmi plusieurs schémas, le meilleur sera celui qui conserve le maximum d'informations.
- **L'évaluation manuelle :** elle se fait en faisant appel à un expert qui nous dira si un algorithme de clustering est bon ou non. Mais cela reste applicable sur des données de petite dimension.

Il existe deux critères qui ont été largement considérés suffisants pour mesurer la qualité du partitionnement de données [44] :

- **Compacité :** Les objets situés dans un cluster doivent être similaires entre eux et différents des objets appartenant aux autres clusters. La variance des objets dans un cluster est un indice de compacité.
- **Séparation :** Les cluster doivent être bien séparés entre eux. La distance Euclidienne entre les centroïdes des clusters donne une indication sur le degré de séparation.

2.8 Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon des principaux concepts, définitions et approche relatifs au problème du clustering qui est une technique non supervisée largement utilisé dans la classification des données et permet d'obtenir des informations sans aucune connaissance préalable. Nous avons également vue les étapes majeurs de ce processus, à savoir : la préparation des données, le clustering et l'exploitation des résultats de l'algorithme. Nous avons focalisé notre attention sur les approches de la classification automatique (clustering) et sur les techniques d'évaluation de leur algorithmes. Le chapitre suivant sera consacré pour examiner les différentes méthodes de clustering hiérarchique qui sont plus flexibles et plus pratiques pour la reconnaissance et la compréhension des activités humaines complexes et de haut niveau.

Chapitre 3

Clustering hiérarchique

3.1 Introduction

Les clustering hiérarchiques visent à décrire les activités de haut niveau en reconnaissant les sous-activités ou sous-événements de bas niveau. Par exemple, l'activité "Préparer le café" peut être reconnue si les tâches "Prendre du café", "Verser de l'eau dans la bouilloire",... est observée. Les approches hiérarchiques présentent plusieurs avantages, ce qui en fait un choix approprié pour modéliser la structure complexe des activités humaines à long terme [3]. De plus, les modèles hiérarchiques sont plus flexibles et plus pratiques pour intégrer les connaissances antérieures, ce qui les rend plus compréhensibles. Ils aident également à mieux comprendre la structure des activités[45, 46].

Les activités humaines complexes de haut niveau telles que les activités de la vie quotidienne sont plus faciles à décrire avec des approches hiérarchiques. De ce fait, ils sont plus efficace dans les phases de reconnaissance automatique des activités humaines.

Afin d'apporter des améliorations pour le clustering hiérarchique, une large collection de méthodes s'est répandue dans la littérature au fil du temps. Dans la suite du chapitre, nous allons passer en revue quelques méthodes connus dans la littérature traitant la problématique de clustering hiérarchique.

3.2 Le clustering hiérarchique

Les approches hiérarchiques génèrent une succession de partitions emboîtées les unes dans les autres au lieu d'une seule partition de l'espace des données. Celles-ci sont souvent représentées sous la forme d'un dendrogramme dont la racine de l'arborescence contient toutes les instances et les feuilles représentent chacune une seule instance[25]. Selon que l'on parcourt le dendrogramme "de haut en bas" ou "de bas en haut", la méthode sera dite divisive (descendante) ou agglomérative (ascendante).

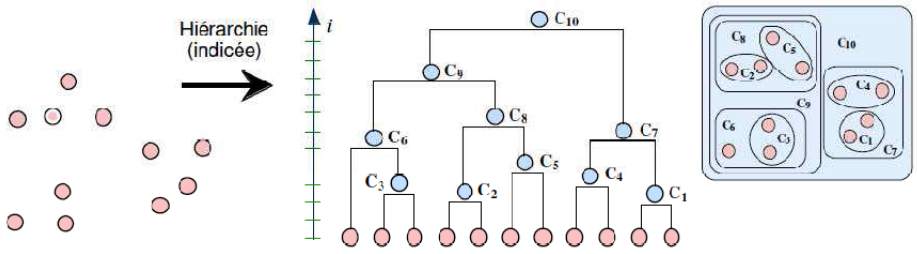

FIGURE 3.1 – Exemple de dendrogramme et la détermination des clusters [25].

Dans les deux cas, un dendrogramme représente les différentes étapes successives de la recherche des clusters. Il présente à chaque niveau quels éléments ont été rassemblés dans une approche agglomerative ou au contraire quels éléments ont été créés dans une approche de divisive[25]. Dans ce qui suit, nous nous intéressons à présenter les méthodes hiérarchiques les plus connus.

3.3 Les algorithmes de clustering hiérarchique

3.3.1 SAHN (Sequential Agglomerative Hierarchical Non-overlapping)

SAHN est une approche de clustering qui regroupe les méthodes de classification hiérarchique agglomératif (ascendant) qui sont considérés comme technique de clustering hiérarchique les plus populaire. Cette stratégie ascendante commence par placer chaque objet dans son propre cluster, puis fusionne ces clusters atomiques en clusters de plus en plus grands, jusqu'à ce que tous les objets soient dans un cluster unique en fonction de la distance entre les clusters ce qui donne naissance à quatre types importants d'algorithmes de liaison et qui sont : Single-link (SLINK), Average-link (ALINK), Complete-link (CLINK) et Centoid-link [47, 48]. Ces méthodes diffèrent par la manière dont leur matrice de similarité est initialement calculée ou dans leurs stratégies de regroupement des clusters à chaque itération.

Algorithme

La description simplifié de cet algorithme est la suivante [48] :

1. Chaque point de donnée est considéré comme un cluster ;
2. On calcule les distances entre les clusters ;
3. Les deux classes les plus proches sont fusionnées et remplacées par une seul ;
4. Le processus reprend en 2 jusqu'à n'avoir plus qu'une seule classe, qui contient toutes les observation.

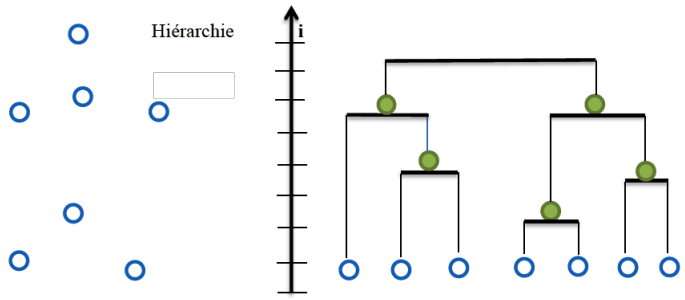

FIGURE 3.2 – La classification ascendante hiérarchique [48].

Algorithme Single-Link

En SLINK la distance entre deux clusters est représentée par la distance minimum entre toutes les paires de données entre les deux clusters (paire composé d'un élément de chaque cluster), nous parlons alors de saut minimum [48]. Le point fort de cette approche est qu'elle sait très bien détecter les classes allongées, mais son point faible est sa sensibilité à la présence de valeurs aberrantes et à la difficulté de faire face à de fortes différences dans la densité des clusters. D'un autre côté, elle affiche une insensibilité totale à la forme et à la taille des clusters [49].

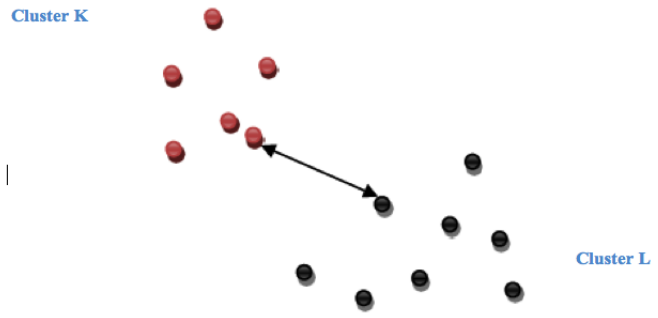

FIGURE 3.3 – Schéma de classification Single-Link [48].

Algorithme Complete-link

En CLINK la distance entre deux clusters est représentée par la distance maximum entre toutes les paires de données entre les deux clusters, nous parlons alors de saut maximum ou de critère du diamètre. Par définition cette approche est très sensible aux point aberrants donc elle est peu utilisé [48,49].

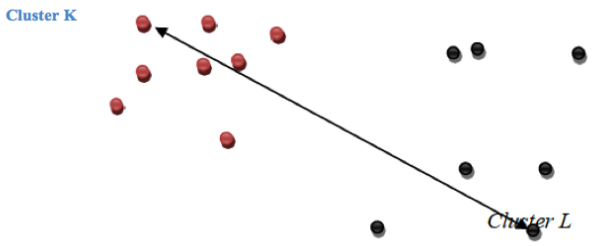

FIGURE 3.4 – Schéma de classification Complete-link [49].

Algorithme Average-Link

ALINK propose de calculer la distance entre deux clusters en prenant la valeur moyenne des distances entre tous les couples d'objets des deux clusters. Nous parlerons ausssi de saut moyen [48]. Cette approche tend à produire des classe de même variance. La liaison moyenne est sensible à la forme et à la taille des grappes. Ainsi, il peut facilement échouer lorsque les grappes ont des formes compliquées s'écartant de la forme hyper sphérique [50].

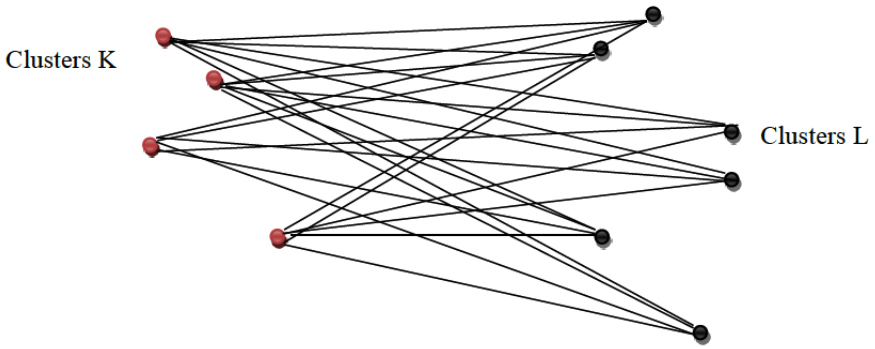

FIGURE 3.5 – Schéma de classification Average-Link [48].

Algorithme Centoid-link

Il définit, quant à lui , la distance entre deux clusters comme la distance entre leur centre de gravité. Une telle méthode est plus robuste au points aberrantes. Toutefois, elle est aussi limitée aux données quantitatives numériques pour lesquelles le calcul du centre de gravité est possible [48, 50].

FIGURE 3.6 – Schéma de classification Centoid-link [48].

Les Avantages de SAHN

La classification ascendante hiérarchique est une méthode de classification qui présente les avantages suivants :

— On travaille à partir des dissimilarités entre les objets que l'on veut regrouper. On peut donc choisir un type de dissimilarité adapté au sujet étudié et à la nature des données.

— L'un des résultats est le dendrogramme, qui permet de visualiser le regroupement progressif des données. On peut alors se faire une idée d'un nombre adéquat de classes dans lesquelles les données peuvent être regroupées.

— L'algorithme du SAHN en générale consiste à fournir un ensemble de clusters de moins en moins fines obtenues par regroupement successifs de parties.

3.3.2 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH est une méthode de clustering crée en 1996, elle est adaptée particulièrement aux très grandes bases de données. Elle permet de produire la meilleure qualité de clustering avec les ressources disponibles (la mémoire disponible et les contraintes de temps). BIRCH peut généralement trouver un bon clustering avec une seule analyse de l'ensemble de données et un ou plusieurs passages supplémentaires peuvent éventuellement être utilisés pour améliorer encore la qualité. BIRCH est également le premier algorithme de clustering proposé dans le domaine de base de données pour gérer efficacement le bruit c'est-à-dire les points de données qui ne font pas partie du modèle sous-jacent [51].

L'idée principale de BIRCH est la classification qui est effectuée sur un résumé vraiment compact des données, au lieu des données originales. Il essaie de minimiser le coût d'entrée/sortie en organisant les données traitées en une structure d'arbre équilibré avec une taille limitée. BIRCH utilise la notion Cluster Feature (CF) pour représenter une classe ou une sous-classe [52]. C'est un vecteur dont les informations stockées dépendent du type de données qu'on traitera en respectant les deux exigences suivantes [53] :

— La similarité (ou la distance) entre un couple de classes est facile à calculer à l'aide des CFs ;

— Le CF de chaque classe est facilement mis à jour lors d'une insertion d'un nouveau membre ou d'une fusion des classes. Dans BIRCH, on utilise l'addition comme opération pour la mise à jour du CF d'une classe.

CF est un vecteur tridimensionnel résumant des informations sur les grappes d'objets. Étant donné n objets ou points en d-dimensions dans un cluster, xi, alors le CF du cluster est défini comme suite : **CF = (N, LS, SS)**

où :

— N : le nombre de points de la grappe ;

— LS : la somme linéaire des n points, c'est-à-dire : $\sum_n^i x_i$;

— SS : la somme carrée des points de données, c'est-à-dire : $\sum_n^i x_i^2$

Par exemple, supposons que nous ayons deux clusters disjoints, C1 et C2, ayant respectivement les fonctionnalités de clustering, CF1 et CF2. La fonction de clustering pour le cluster formé par les méthodes hiérarchiques fusionnant C1 et C2 est simplement CF1 + CF2.

Les fonctionnalités de clustering sont suffisantes pour calculer les mesures nécessaires pour prendre des décisions de regroupement dans BIRCH.

Exemple de fonction de clustering :

— Supposons qu'il y ait trois points (2, 5), (3, 2) et (4, 3), dans un cluster, C1. La fonctionnalité de clustering de C1 est :

$CF_1 = [3, (2+3+4, 5+2+3), (2^2+3^2+4^2, 5^2+2^2+3^2)] = [3, (9, 10), (29, 38)]$

— Supposons que C1 est joint à un deuxième groupe C2, où CF2 = [3, (35, 36), (417, 440)].

— La fonction de clustering d'un nouveau cluster C3 qui est formé par la fusion de C1 et C2, est dérivée en additionant CF1 et CF2 :

$CF_3 = [3+3, (9+35, 10+36), (29+417, 38+440)] = [6, (44, 46), (446, 478)]$

L'algorithme de clustering BIRCH

La figure 6 présente la vue d'ensemble de BIRCH qui est composé des 4 phases suivante [51] :

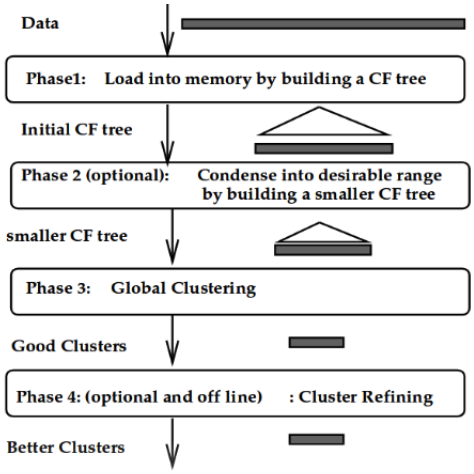

FIGURE 3.7 – BIRCH [51].

Phase 1 (Le chargement des données en mémoire en construisant un arbre CF) : cette phase consiste à analyser toutes les données et à créer une arborescence CF initiale en mémoire en utilisant la quantité disponible de mémoire et d'espace de recyclage sur le disque. Cet arbre CF essaie de refléter les informations de clustering de l'ensemble de données aussi fin que possible sous la limite de mémoire en créant ainsi un résumé en mémoire des données.

Phase 2 (Condenser les données dans la plage souhaitée en construisant un arbre CF plus petit) : elle est optionnelle et elle permet d'analyser les entrées de feuille dans l'arborescence CF initiale pour reconstruire un arbre CF plus petit, tout en supprimant plus de valeurs aberrantes et en regroupant les sous-groupes surpeuplés.

Phase 3 (clustering global) : ici un algorithme global ou semi-global est utilisé pour regrouper toutes les entrées de feuille. Nous pouvons appliquer un algorithme existant directement aux sous-grappes car les informations dans leurs vecteurs CF sont généralement suffisantes pour calculer la plupart des distances et mesures de qualité et pour obtenir au final un ensemble de clusters qui capture le modèle de distribution principal dans les données.

Phase 4 (raffinage de cluster) : Jusqu'à présent, bien que l'arborescence ait pu être reconstruite plusieurs fois, les données d'origine n'ont été analysées qu'une seule fois. La phase 4 implique des passages supplémentaires sur les données pour corriger les inexactitudes causées par le fait que l'algorithme de clustering est appliqué à un résumé grossier des données. La phase 4 nous offre également la possibilité d'éliminer les valeurs aberrantes.

Les avantages de BIRCH

— BIRCH est local car chaque décision de clustering est prise sans analyser tous les points de données ou tous les clusters existants. Il utilise des mesures qui reflètent la proximité naturelle des points et, en même temps, peuvent être maintenues de manière incrémentielle pendant le processus de clustering.

— BIRCH exploite l'observation selon laquelle l'espace de données n'est généralement pas occupé uniformément, et donc chaque point de données n'est pas également important à des fins de clustering. Une région dense de points est traitée collectivement comme un seul cluster. Les points dans les régions clairsemées sont traités comme

des points extrêmes et supprimés éventuellement.

— BIRCH utilise pleinement la mémoire disponible pour dériver les meilleurs sous-clusters possibles afin de garantir la précision, tout en minimisant les coûts d'E/S et garantir ainsi l'efficacité). Le processus de regroupement et de réduction est organisé et caractérisé par l'utilisation d'une structure arborescente en mémoire, équilibrée en hauteur et très occupée. En raison de ces fonctionnalités, son temps de fonctionnement est évolutif de manière linéaire.

Les limites de BIRCH

— Une limite importante de l'algorithme BIRCH est son impossibilité de faire face à des classes de tailles variées, de forme non sphériques, ou reliées par un ensemble de points ;

— Ne peut traiter que les attributs métriques dont les valeurs peuvent être représentées par des coordonnées explicites dans un espace euclidien.

3.3.3 CURE (Clustering Using Representatives)

Il a été proposé par Guha et al en 1998. CURE est un algorithme de clustering hiérarchique ascendant efficace pour les grandes bases de données qui adopte un compromis entre l'approche centroïde et l'approche en tous points. Il est basé sur les représentants qui sont un sous ensemble de points de chaque cluster choisis pour bien le couvrir. CURE adopte un juste milieu entre le centroïde et les extrêmes de tous les points en représentant chaque cluster par un certain nombre fixe de points qui sont générés en sélectionnant des points bien dispersés dans le cluster, puis il les rétrécissent vers le centre du cluster en utilisant un facteur de rétrécissement **"a"** spécifiée. Ces points dispersés après rétrécissement sont utilisés comme représentants du cluster. Le fait d'avoir plus d'un point représentatif par cluster permet à CURE de bien s'adapter à la géométrie des formes non sphériques et le rétrécissement aide à soulager les effets des valeurs aberrantes. Les clusters avec la paire de points représentatifs la plus proche sont les clusters qui sont fusionnés à chaque étape de l'algorithme de clustering

hiérarchique de CURE [53, 54].

Pour gérer de grandes bases de données, CURE utilise une combinaison d'échantillonnage aléatoire et de partitionnement. Un échantillon aléatoire tiré de l'ensemble de données est d'abord partitionné et chaque partition est partiellement clusterisée. Les clusters partiels sont ensuite regroupés dans un deuxième passage pour donner les clusters souhaités [54].

L'algorithme de clustering commence avec chaque point d'entrée en tant que cluster séparé, et à chaque étape successive fusionne la paire de clusters la plus proche comme montré sur la figure 3.8 [54] :

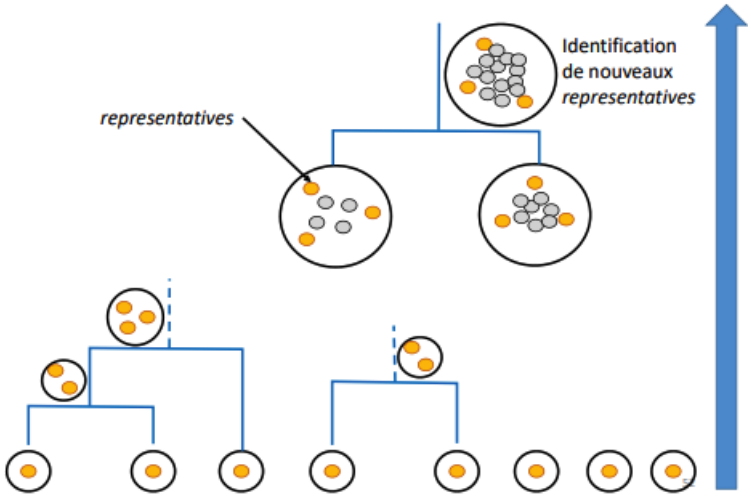

FIGURE 3.8 – CURE (Clustering Using Representatives) [54].

Les étapes de l'algorithme

Les étapes impliquées dans le clustering à l'aide de CURE sont décrites dans la figure 3.9 [54, 55] :

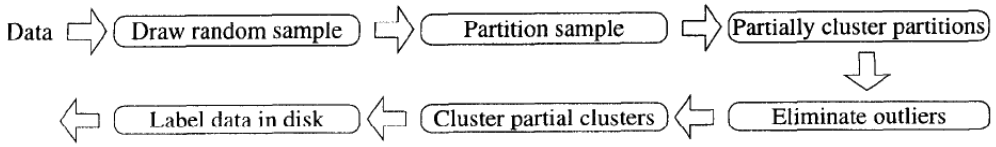

FIGURE 3.9 – Présentation de CURE [54].

- **Échantillonnage aléatoire :** permet de tirer un échantillon aléatoire de la base de données. Les échantillons aléatoires de tailles modérées préservent assez précisément les informations sur la géométrie des clusters, permettant ainsi à CURE de regrouper correctement les données d'entrées.
- **Partitionnement des échantillons et regroupement :** afin d'accélérer davantage le clustering, CURE partitionne d'abord l'échantillon aléatoire et regroupe partiellement les points de données dans chaque partition.
- **Eliminer les valeurs aberrantes :** ici CURE élimine toutes les données et valeurs où une ou plusieurs observations sont distante des autres observations effectuées sur le même ensemble de données, c'est-à-dire qu'elles contrastent grandement avec la majorité des valeurs.
- **Regrouper partiellement les clusters :** après avoir éliminé les valeurs aberrantes, les données pré-classées dans chaque cluster sont ensuite regroupées en un dernier passage pour générer les clusters finaux.
- **Etiqueter les données sur les disques :** une fois le regroupement de l'échantillon aléatoire terminé, au lieu d'un seul centroïde, plusieurs points représentatifs de chaque cluster sont utilisés pour étiqueter le reste de l'ensemble de données.

Les avantages de CURE

— Il identifie les clusters ayant des formes non sphériques et de grandes variations de taille.
— Il est plus robuste à la présence de valeurs aberrantes.
— L'algorithme a des exigences de stockage linéaire et une complexité temporelle de $O(n^2)$ pour les données de faible dimension.
— Il offre une meilleure qualité de clustering et un temps d'exécution nettement meilleurs.

Les limites de CURE

— CURE arrive à clusteriser des ensembles de points complexes mais reste sensible au réglage de ses paramètres.
— CURE ne parvient pas à expliquer l'interconnectivité des objets dans les clusters.
— Il est très séquentiel.

3.3.4 ROCK (RObust Clustering using linKs)

ROCK est un algorithme de clustering hiérarchique agglomératif basé sur la notion de voisins et de liens. Avec des ensembles de données réels, la qualité des clusters générés par ROCK est de loin supérieure aux clusters produits par l'algorithme de clustering hiérarchique traditionnel basé sur les centroïdes [48]. Les voisins d'un point peuvent être définit par les points qui sont considérablement similaires. Si le point A est voisin du point C et que le point B est voisin du point C, les points A et B sont liés, même s'ils ne sont pas eux-mêmes voisins. Soit $sim(p_i, p_j)$ une fonction de similitude normalisée qui capture la proximité entre la paire de points p_i et p_j, de sorte que $sim(p_i, p_j) = 1$ lorsque $p_i = p_j$ et $sim(p_i, p_j) = 0$ s'ils sont complètement différents. Nous considérons alors p_i et p_j comme des voisins si $sim(p_i, p_j) \geq \theta$, où θ est un paramètre fourni par l'utilisateur [56].

Le lien entre p_i et p_j $link(p_i, p_j)$ est définit comme étant le nombre de voisins communs entre p_i et p_j. La fonction de similitude peut être une distance euclidienne, le coefficient de Jaccard, ou toutes autre fonction de similitude fourni par un expert [56].

Le meilleur ensemble de clusters est caractérisé par l'ensemble de clusters qui maximise une fonction critère E1. La première approche consiste à maximisez le nombre de liens entre les paires de points dans chaque cluster en utilisant la fonction suivante [54] :

$$E1 = \sum_{i=1}^{k} \sum_{p_q, p_r \in C_i} link(p_q, p_r) \tag{3.1}$$

Cette fonction conserve les points qui partagent de nombreux liens dans le même cluster, mais ne force pas les points avec peu de liens à se diviser en différents clusters. Une approche améliorée permet de diviser le nombre réel de liens par le nombre attendu de liens, ce qui empêche les points avec peu de liens d'être placés dans le même cluster et permet ainsi de définir la fonction de critère final [56] :

$$E1 = \sum_{i=1}^{k} n_i \sum_{p_q, p_r \in C_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}} \tag{3.2}$$

ROCK défini une mesure de qualité basée sur la fonction de critère ci-dessus :

$$g(C_i, C_j) = \frac{link[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \tag{3.3}$$

A chaque étape l'algorithme fusionnent la paire de clusters qui maximisent cette fonction.

Les étapes de l'algorithme

Les étapes impliquées dans le clustering à l'aide de ROCK sont décrites dans la figure 3.10. Après avoir tiré un échantillon aléatoire de la base de données, un algorithme de clustering hiérarchique qui utilise des liens est appliqué aux échantillons c'est-à-dire fusionner itérativement les clusters C_i, C_j qui maximisent la qualité et le fusionnement s'arrête une fois qu'il n'y a plus de liens entre les clusters ou que le nombre requis de clusters a été atteint. Enfin, les clusters impliquant uniquement les points échantillonnés sont utilisés pour affecter les points de données restants sur le disque aux clusters appropriés [52, 56].

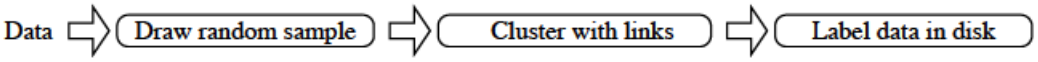

FIGURE 3.10 – ROCK [56].

3.3.5 CHAMELEON

Chameleon est un nouvel algorithme agglomératif qui surmonte les limites des algorithmes de clustering existants. La caractéristique clé de l'algorithme Chameleon est qu'il tient compte à la fois de l'inter-connectivité et de la proximité dans l'identification de la paire de clusters la plus similaire et utilise une nouvelle approche pour modéliser le degré d'interconnectivité et de proximité entre chaque paire de grappes. Cette approche tient compte des caractéristiques internes des grappes elles-mêmes. Ainsi, il ne dépend pas d'un modèle statique fourni par l'utilisateur et peut s'adapter automatiquement aux caractéristiques internes des clusters fusionnés [57].

Chameleon fonctionne sur un graphique clairsemé dans lequel les nœuds représentent les éléments de données et les bords pondérés représentent les similitudes entre les éléments de données. Cette représentation permet à Chameleon de s'adapter à de grands ensembles de données et d'utiliser avec succès des ensembles de données qui ne sont disponibles que dans l'espace de similarité (qui ne fournit que des similitudes entre les éléments de données) et

non dans les espaces métriques qui ont un nombre fixe d'attributs pour chaque élément de données [57].

Chameleon trouve les clusters dans l'ensemble de données en utilisant un algorithme à deux phases. Au cours de la première phase, Chameleon utilise un algorithme de partitionnement de graphe pour regrouper les éléments de données en plusieurs sous-grappes relativement petites. Au cours de la deuxième phase, il utilise un algorithme pour trouver les clusters authentiques en répétant la combinaison de ces sous-grappes [14]. Deux grappes ne sont combinées que si la proximité et l'interconnectivité entre elles sont élevées par rapport à l'interconnectivité interne des éléments de proximité et de grappe au sein des grappes. Chameleon est un algorithme de clustering basé sur l'algorithme K-le plus proche voisin (KNN). Le processus de base de l'algorithme de Chameleon est illustré par la figure 3.11 [57, 58].

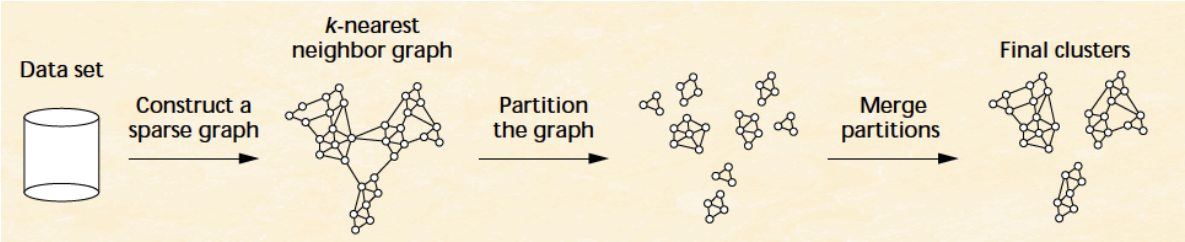

FIGURE 3.11 – Data sets [57].

L'algorithme de Chameleon crée d'abord un graphique KNN de l'ensemble de données donné tel que les points p et q sont connectés si q est parmi les k premiers voisins les plus proches de p, puis il utilise un algorithme de partitionnement de graphe à plusieurs niveaux pour trouver les sous-clusters initiaux et détermine l'inter-connectivité relative entre la proximité relative des deux clusters au sein d'un cluster. Ce processus de fusion de cluster se poursuit jusqu'à atteindre le nombre spécifié de clusters créés. Chameleon utilise une méthode à deux niveaux pour trouver les grappes finales. Dans le niveau initial, il utilise un algorithme de partitionnement de graphe et dans le cluster de second niveau les éléments de données en de nombreux sous-cluster relativement minuscules. L'algorithme Chameleon comprend des grappes de différentes densités, formes, bruit, taille et artefacts et il contient également des points dans l'espace 2D [58].

Les étapes de l'algorithme

Étant donné un ensemble de points, construisez le graphique k-plus proche voisin (k-NN) pour capturer la relation entre un point et ses k voisins les plus proches [58].

Phase 1 : utilisez un algorithme de partitionnement de graphe à plusieurs niveaux sur le graphe pour trouver un grand nombre de grappes de sommets bien connectés. C'est-à-dire que chaque cluster doit contenir principalement des points d'un "vrai" cluster.

Phase 2 : utilisez le clustering agglomératif hiérarchique pour fusionner les sous-clusters. Deux clusters sont combinés si le cluster résultant partage certaines propriétés comme l'inter-connectivité relative et la proximité relative avec les clusters constituants.

Chameleon utilise un cadre de modélisation dynamique pour déterminer la similitude entre les paires de grappes en examinant leur interconnectivité relative (RI) et leur proximité relative (RC). Chameleon sélectionne les paires à fusionner pour lesquelles RI et RC sont élevés. Autrement dit, il sélectionne des clusters qui sont bien interconnectés ainsi que rapprochés.

— **Interconnectivité relative :** c'est l'inter-connectivité absolue normalisée de C_i et C_j sur leurs inter-connectivité interne [58].

$$RI(C_i, C_j) = \frac{\mid EC(C_i, C_j) \mid}{\frac{|EC(C_i)|+|EC(C_j)|}{2}} \tag{3.4}$$

Où :

— $EC(C_i, C_j)$: l'inter-connectivité absolue qui est la somme du poids des arêtes qui relient les sommets en C_i aux sommets en C_j.

— L'inter-connectivité interne est la somme pondérée des arêtes qui partitionnent le graphique en deux parties à peu près égales.

— **Proximité relative :** la proximité relative entre une paire de cluster C_i et C_j est la proximité absolue entre C_i et C_j normalisée sur la proximité interne de ces deux clusters [53].

$$RC(C_i, C_j) = \frac{\bar{S}EC(C_i, C_j)}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}EC(C_i) + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}EC(C_j)} \tag{3.5}$$

Où : $\bar{S}EC(C_i)$ et $\bar{S}EC(C_j)$ les poids moyens des arêtes qui appartiennent à la bissectrice min-cut des grappes C_i et C_j, respectivement, et $\bar{S}EC(C_i, C_j)$ est le poids moyen des arêtes qui relient les sommets en C_i aux sommets en C_j.

CHAMELEON Fusionne uniquement les paires de clusters dont RI et RC sont tous deux supérieurs aux seuils spécifiés par l'utilisateur c'est-à-dire il fusionne ceux qui maximisent la fonction qui combine RI et RC. La décision de fusionner est réalisée en se basant sur la combinaison des mesures relatives locales comme suit :

$$I(C_i, C_j) = RI(C_i, C_j) * RC(C_i, C_j)^{\alpha} \tag{3.6}$$

où α est un paramètre spécifié par l'utilisateur. Si $\alpha > 1$, alors Chameleon donne une importance plus élevée à la proximité relative, et quand $\alpha < 1$, il donne une importance plus élevée à l'interconnectivité relative.

Les avantages de CHAMELEON

— CHAMELEON est capable de découvrir des grappes de forme arbitraire et de haute qualité.

— Il prend en compte le rapport de dispersion des points à l'intérieur d'un groupe en se basant sur l'utilisation des distances d'inter-connectivité et la proximité relative.

Les limites de CHAMELEON

— Il est sensible au bruit et aux points aberrants.
— Le coût de traitement pour des données de grande dimension peut nécessiter $O(n^2)$ de temps pour n objets dans le pire des cas.
— L'algorithme est non incrémental.

3.3.6 COBWEB

Il s'agit d'un algorithme de clustering conceptuel qui a été développé par des chercheurs en apprentissage automatique dans les années 1980 et fonctionne progressivement en mettant à jour les clusters objet par objet. Les grappes décrites de manière probabiliste sont organisées sous forme d'arbre pour former un regroupement hiérarchique appelé dendrogramme. Il gère l'incertitude associé à des attributs catégoriels dans le clustering à travers un cadre probabiliste. Le dendrogramme de cet algorithme est également appelé arbre de classification et les nœuds sont appelés concepts [59].

Le clustering conceptuel est utilisé pour découvrir des classes d'objets ayant des caractéristiques communes dans de grandes quantités de données. Un système employé dans cette tâche reçoit en entrée un ensemble d'observations et sort un ensemble de classes. Les observations sont décrites par un ensemble d'attributs prédéfinis qui prennent une valeur à partir d'un ensemble de valeurs donné. Les attributs sont choisis de manière à représenter les caractéristiques des observations et à aider de cette manière à former un schéma de regroupement significatif. Les systèmes de clustering conceptuels évaluent ces propriétés à l'aide d'une fonction objective appropriée et tentent d'améliorer la qualité du clustering en utilisant une stratégie de contrôle. L'algorithme COBWEB connecte des objets pour former des grappes en fonction de leur distance [60].

Algoritme de COBWEB

L'algorithme COBWEB construit un arbre de classification de manière incrémental en insérant les objets un par un dans l'arbre de classification. Lors de l'insertion d'un objet dans l'arbre de classification, l'algorithme COBWEB parcourt l'arborescence de haut en bas à partir du nœud racine. À chaque nœud, l'algorithme COBWEB considère 4 opérations possibles : insérer, créer, fusionner ou fractionner, et en sélectionne une qui produit la fonction d'utilité (CU) de catégorie la plus élevée. CU tente de maximiser à la fois la probabilité que deux instances de la même catégorie aient des valeurs d'attribut en commun et la probabilité que les instances de différentes catégories aient des valeurs d'attribut différentes [60] :

$$CU = \sum_{C} \sum_{A} \sum_{v} P(A = v)P(A = v|C)P(C|A = v) \tag{3.7}$$

$P(A = v|C)$ est la probabilité qu'une instance ait la valeur v pour son attribut A, étant donné qu'elle appartient à la catégorie C. Plus cette probabilité est élevée, plus il est probable que deux instances d'une catégorie partagent les mêmes valeurs d'attribut. $P(C|A = v)$ est la probabilité qu'une instance appartienne à la catégorie C, étant donné qu'elle a la valeur v pour son attribut A. Plus cette probabilité est grande, moins les instances probables de différentes catégories auront des valeurs d'attribut en commun. $P(A = v)$ est un poids, garantissant que les valeurs d'attribut fréquentes auront une plus grande influence sur l'évaluation [60].

Après avoir appliqué la règle de Bayes à l'équation 3.8, nous obtenons :

$$CU = \sum_{C} \sum_{A} \sum_{v} P(C)P(A = v|C)^2 \tag{3.8}$$

$\sum_A \sum_v P(A = v|C)^2$ est le nombre attendu de valeurs d'attribut que l'on peut correctement deviner pour un membre arbitraire de la classe C. Cette attente suppose qu'une stratégie d'appariement de probabilité, dans laquelle on devine une valeur d'attribut avec une probabilité égale à sa probabilité d'occurrence. Sans connaître la structure du cluster, le terme

ci-dessus est $\sum_A \sum_v P(A=v)^2$

La CU finale est définie comme l'augmentation du nombre attendu de valeurs d'attribut qui peuvent être correctement devinées, étant donné un ensemble de n catégories, par rapport au nombre attendu de suppositions correctes sans cette connaissance. C'est [60] :

$$CU = \frac{1}{n} \sum_C p(C) \sum_A \sum_v [P(A=v|C)^2 - P(A=v)^2] \tag{3.9}$$

L'expression ci-dessus est divisée par n pour permettre de comparer des grappes de tailles différentes.

Les avantages de COBWEB

— COBWEB donne une meilleure complexité temporelle.

— Le nombre de clusters est non connu au préalable.

Les limites de COBWEB

— L'algorithme COBWEB ne gère pas bien les valeurs aberrantes.

— COBWEB a tendance de faire des arbres épais et trop chargés, et donc que les niveaux supérieurs de l'arbre finissent par être les catégories de classes les plus importantes.

3.4 Conclusion

Le clustering hiérarchique est un outil, parmis d'autres, utilisé pour comprendre le monde. Il est nécessaire dans la prise de décisions et il est basé sur une reconnaissance des similitudes entre les objets.

Il existe plusieurs méthodes de clustering hiérarchiques dans la littérature qui sont comme toutes les autres méthodes de classification, ont des avantages et aussi des faiblesses et on a présenté les plus intéressantes ci-dessus, ce qui nous a permis d'élaborer une proposition d'une méthode de clustering hiérarchique. Le chapitre suivrant sera consacré à détailler cette méthode ainsi que notre contribution dans le domaine de clustering hiérarchique dans les systèmes ubiquitaires.

Chapitre 4

Proposition

4.1 Introduction

Dans ce chapitre nous allons présenter une nouvelle méthode de clustering hiérarchique pour la reconnaissance des activités humaines dans les systèmes ubiquitaires. En premier lieu nous décrivons le jeu de données utilisées pour la validation. En second lieu nous allons nous intéresser à présenter les différentes phases de notre système de reconnaissance. En dernier lieu nous exposons les résultats de notre travail.

4.2 Jeu de données

Dans cette section, nous décrivons la démarche que nous avons pris pour construire notre système de reconnaissance d'activités humaines sur un flux de données binaires. Comme on l'a déjà cité précédemment, la reconnaissance d'activités humaines est une tâche de classification à deux étapes : l'apprentissage et la reconnaissance.

Généralement les expériences de la reconnaissance des activités humaines se déroulent dans des maisons laboratoires plutôt que dans des maisons réelles. L'occupant de la maison laboratoire effectue un ensemble d'activité, puis les activités sont segmentées pour être soumise à un algorithme d'apprentissage qui génère les classes d'activité [61].

Dans notre cas, des capteurs à changement d'états ont été mis en place afin de pouvoir récolter un ensemble de données indiquant des tâches effectuées par l'occupant de la maison laboratoire. Un capteur a changement d'état peut être comme son nom l'indique dans deux états : état activé (1) et l'état non-activé (0), il est activé lorsque l'utilisateur touche l'objet portant ce capteur. Un exemple d'une partie de séquence d'évènement de capteurs est présenté dans le tableau 4.1. Chaque ligne représente un évènement de capteur.

Début	**Fin**	**ID**	**Valeur**
19-Nov-2008 22 :47 :46	19-Nov-2008 22 :49 :17	5	1
19-Nov-2008 22 :49 :21	19-Nov-2008 22 :49 :22	5	1
19-Nov-2008 22 :49 :24	19-Nov-2008 22 :50 :14	5	1
19-Nov-2008 22 :50 :19	20-Nov-2008 11 :14 :11	13	1
21-Nov-2008 22 :51 :03	22-Nov-2008 08 :51 :04	6	1
22-Nov-2008 09 :51 :05	22-Nov-2008 10 :51 :46	3	1
22-Nov-2008 10 :52 :05	22-Nov-2008 10 :59 :46	13	1

TABLE 4.1 – Une partie de séquence d'évènements de capteur à changement d'état

Il est important d'avoir un grand nombre d'observations pour avoir un modèle plus efficace, Nous allons procéder par former à partir des observations nos vecteurs caractéristiques qui à chaque instant t donne la situation de la personne (les gestes reconnus, les capteurs activés, la localisation, etc). Les données récoltées une fois synchronisées donne lieu à un flux vectoriel qui doit être segmentées, puis soumis au système de reconnaissance qui cherche à reconnaître l'activité effectuée.

Dans la suite de ce travail, nous n'allons considérer que les deux activités : préparer le diner et faire la vaisselle pour des raisons de simplification.

4.3 Description de la proposition

L'étude de l'état de l'art sur les méthodes de clustering hiérarchqiues, nous a permis d'approfondir et d'enrichir nos connaissances en ce qui concerne les techniques de classification non supervisées.

Il existe deux types de clustering hiérarchique : Agglomératif (ascendant) et Divisif (descendant). Dans le premier cas, les points de données sont regroupés en utilisant une approche ascendante en commençant par des points de données individuels. Tandis que dans la seconde, une approche descendante est suivie où tous les points de données sont traités comme un seul grand cluster et le processus de regroupement implique de diviser le seul grand cluster

en plusieurs petits groupes [60]. Pour les deux types l'objectif est le même, nous cherchons à ce que les individus regroupés au sein d'un même cluster soient le plus semblables possibles tandis que les clusters soient le plus dissemblables.

Notre proposition est basée sur le clustering hiérarchique ascendant. Le principe de base de ce type de clustering est de rassembler des objets selon un critère de ressemblance défini au préalable qui s'exprimera sous la forme d'une matrice de distances qui calcul la distance existant entre chaque paire d'objets. Deux objets identiques auront une distance nulle. Plus les deux objets seront dissemblables, plus la distance sera importante. Le clustering hiérarchique ascendant va ensuite rassembler les individus de manière itérative afin de produire un dendrogramme ou un arbre de classification. La classification est ascendante car elle part des objets individuels ; elle est hiérarchique car elle produit des clusters de plus en plus vastes. En découpant ce dendrogramme à une certaine hauteur choisie, on produira la partition désirée [62 ; 63].

Notre approche consiste à tirer des vecteurs représentatifs à partir de l'ensemble des vecteurs caractéristiques et cela en choisissant les vecteurs les plus dissemblables. Une fois qu'on a les vecteurs représentatifs, la matrice de distance entre ces vecteurs représentatifs et les autre vecteurs caractéristiques est calculé afin de regrouper chaque paire de vecteurs les plus proches. Ce processus est répété jusqu'à avoir regrouper tout les vecteurs.

Comme tout les techniques de clustering hiérarchique ascendant, notre proposition se fait en deux phase principales : l'apprentissage et la reconnaissance dont un petit aperçu est donnée ci-dessous avant d'entrer dans les détails de chaque phase :

1. **L'apprentissage** dans laquelle les modèles d'activités sont appris ;
2. **La reconnaissance** dans laquelle les modèles d'activités déjà appris sont utilisés pour prédire les classes d'activités de nouvelles données de capteur.

4.4 Apprentissage

Les étapes de l'apprentissage d'une activité A sont :

1. Construction des vecteurs caractéristiques ;
2. Clusterisation des vecteurs caractéristiques ;
3. Étiquetage des clusters.

4.4.1 Construction des vecteurs caractéristiques

Les données d'entrées de notre algorithme indiquent si un capteur est activé ou non à chaque fois qu'une activité est réalisée. Donc on a une instance de chaque activité effectuée ce qui va nous permettre de dresser l'état de chaque capteur mis en place pour chaque activité et cela donne naissance à des vecteurs caractéristiques v dont la dimension est d représentant le nombre de capteurs.

$$v = \begin{bmatrix} \textbf{IDCapteur} & \textbf{État} \\ 1 & 1 \\ 2 & 1 \\ 3 & 0 \\ ... & ... \\ ... & ... \\ n & 0 \end{bmatrix} \Rightarrow v = \begin{bmatrix} 1 \\ 1 \\ 0 \\ ... \\ ... \\ 0 \end{bmatrix}$$

exemple : Pour les deux activités "préparer le diner" et "Faire la vaisselle" on regard l'état de tous les capteurs mis en place dans la cuisine.

Préparer le diner

ID	Désignation	Val
1	$Fourmicroonde$	1
4	"$Armoiretasses$"	0
5	"$Frigo$"	1
9	"$Armoireassiettes$"	1
7	"$Lave-vaisselle$"	0
8	"$Chassed'eau$"	1
9	"$Conglateur$"	1
10	"$Placardcasseroles$"	0
12	"$Placardpicerie$"	1

Faire la vaisselle

ID	Désignation	Val
1	"$Fourmicroonde$"	0
4	"$Armoiretasses$"	1
5	"$Frigo$"	0
9	"$Armoireassiettes$"	1
7	"$Lave-vaisselle$"	1
8	"$Chassed'eau$"	1
9	"$Conglateur$"	1
10	"$Placardcasseroles$"	1
12	"$Placardpicerie$"	1

Pour éviter qu'on manipule des vecteurs caractéristiques avec de très grande dimension, il est nécessaire de choisir les capteurs les plus important sur lesquelles le processus de reconnaissance va être effectuer. Dans notre cas on a considéré la mise en place de cinq capteurs seulement et ils correspondent aux capteurs qui s'activent lors de la réalisation des deux activité qu'on a considéré à savoir préparer le diner et faire la vaisselle.

Pour des raisons de simplification, on va considérer que pour chaque intervalle de temps fixé il y'a qu'une seule activité en cours d'exécution et chaque activité est effectuée par une seule personne (pas d'activités de groupes).

4.4.2 Clusterisation des vecteurs caractéristiques

Maintenant on va regrouper l'ensemble des vecteurs caractéristiques en clusters. Chaque cluster contient les vecteurs les plus similaires. L'algorithme choisi pour réaliser la clusterisation est l'algorithme de clustering hiérarchique ascendant où chaque objet est considéré comme un cluster, puis à chaque étape on regroupe les clusters les plus similaires. Notre approche se base sur le choix de vecteurs représentatifs les plus dissimilaires avant d'appliquer le clustering hiérarchique ascendant, puis on va comparer chaque vecteur représentatif à un autre vecteur afin de générer k clusters. Ce choix de vecteur dans notre approche ne va pas être un choix aléatoire mais on choisit les vecteurs les plus dissimilaire et de cette façon on s'assure qu'ils désignent des activités distinctes.

Donc si on veut générer à la fin de notre partitionnement k clusters, d'abord on calcule les distances entre chaque couple de vecteur, puis on choisit les k couples les plus dissimilaires et leurs vecteurs vont être des vecteurs représentatifs, ce qui donnent $2K$ vecteurs représentatifs, ensuite pour chaque vecteur représentatif on cherche les vecteurs les plus proche de lui (les plus similaire) ce qui donne à la fin de cette étape $2K$ clusters. Finalement on applique le clustering hiérarchiques ascendant sur les $2K$ clusters et on arrête lorsque le nombre de clusters est égale à k.

Dans notre cas, les données qu'on manipule sont binaires donc on utilise l'équation suivante pour calculer la dissimilarité entre chaque paire de vecteur V_i, V_j [22] :

$$d(V_i, V_j) = \frac{r + s}{q + r + s + t} \tag{4.1}$$

Où :

- s et r est le nombre d'attributs qui sont inégaux pour les deux objets ;
- q est le nombre d'attributs qui sont égaux à 1 pour les deux objets ;
- t est le nombre d'attributs qui sont égaux à 0 pour les deux objets.

Algorithme de clustering ascendant

Entrées :

- **S :** Ensemble de données (les vecteurs caractéristiques)
- **K :** Le nombre de clusters à générer

Étape 1 :

- Calculer les distances entre chaque couple de vecteur
- Identifier les K distance d (Vi, Vj) maximal (donc 2K vecteurs représentatifs)

Étape 2 : Pour chaque vecteur représentatif Vr calculer la distance avec les autres vecteurs

1. Identifier la distance d (Vr, Vj) minimale
2. Regrouper les clusters Vr et Vj
3. Calculer les distances entre le nouveau cluster et les autres clusters (Les autres distances restent inchangées)
4. Identifier la distance minimale
5. Regrouper des clusters associés
6. **Si** nbr de clusters > 2k : retour au point 3
   Sinon : fin (Comme ça on obtient pour chaque vecteur représentatif un cluster ce qui donne 2K clusters)

Étape 3

- Calculer les distances entre les 2K clusters d (Vi, Vj)
- Revenir à l'étape 2 - 1 mais ici on s'arrête lorsque on obtient un seul cluster (c.-à-d. si le nbr de clusters > 1 : retour au point 3).

Sortie :

- Un ensemble de K clusters

Fin de l'algorithme

4.4.3 Étiquetage des clusters

Une fois que l'ensemble des données est regroupé en K clusters, il est nécessaire de donner à chaque clusters une étiquette. Plusieurs points représentatifs de chaque cluster sont utilisés pour étiqueter le reste de l'ensemble de données.

Dans chaque cluster générer par l'algorithme du clustering hiérarchique on peut avoir un ou plusieurs vecteurs représentatifs, donc pour annoter nos clusters il est intéressant de donner des étiquettes au vecteurs représentatifs de chaque cluster. Une fois que tous les vecteurs représentatifs d'un seul cluster sont étiquetés, on les compare entre eux. S'il s'agit de la même étiquette on l'attribut au clusters, sinon on repartitionne une deuxième fois ce clusters, c'est-à-dire on applique l'algorithme de clustering cité ci-dessus une deuxième fois sur ce clusters en choisissant une nouvelle valeur pour Kqui va être le nombre de vecteurs caractéristiques différemment annotés qui existe dans ce cluster et dans ce cas on applique notre algorithme à partir de l'étape 2 parce que on possède déjà les vecteurs représentatifs.

Pour attribuer une étiquette à un cluster, on a besoin d'un ensemble d'apprentissage qui contient les vecteurs d'activités et leurs labels associés et qui va être notre référence.

Algorithme d'étiquetage

Entrées :

- K clusters
- 2K vecteurs représentatifs
- La base d'apprentissage

Début

Pour chaque vecteur représentatif :

- Attribuer l'étiquette appropriée en se basant sur l'ensemble d'apprentissage.

Fin Pour

Pour chaque cluster :

- Comparer l'étiquette de ces vecteurs représentatifs
- **Si** les vecteurs représentatifs on la même étiquette :

  Attribuer cette étiquette au cluster

  Sinon Clusterisation de ce cluster

Sortie :

- L'ensemble des clusters étiquetés

Fin de l'algorithme

4.5 La reconnaissance

La reconnaissance d'un nouvel élément dans notre système à partir du modèle d'apprentissage va être en temps réel où chaque évènement de capteur est classifié au moment de son arrivée. Ce processus de reconnaissance en temps réel est assuré en choisissant la durée de nos segments à l'avance, c'est-à-dire le delta(t) est choisi avant d'avoir les données et pour chaque delta(t) on récolte les données enregistrées par les capteurs à changement d'état puis on applique notre algorithme de reconnaissance sur ce segment.

Donc pour chaque durée delta(t) on regard l'état des capteurs a changement d'état ce qui donne les vecteurs suivants :

Début	**Fin**	**IDCapteur**	**Val**

L'analyse de l'état des cinq capteurs à changement d'état utilisé dans notre étude et qui sont installés dans la cuisine (Frigo, four micro onde, armoire à assiettes, lave-vaisselle et la chasse d'eau) afin de reconnaitre les deux activité qu'on a considéré à savoir préparer le diner et faire la vaisselle, pour une durée prédéfini delta(t) donne les vecteurs représentatifs décrit dans le tableau 4.2 :

Capt1	**Capt2**	**Capt3**	**Capt4**	**Capt5**
1	1	0	1	0
0	0	1	1	0
1	1	1	1	1
0	0	0	0	0
1	0	0	0	1
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
1	0	1	0	0
1	0	0	0	0
1	0	0	0	1
0	0	0	1	0
0	1	0	1	0
0	1	0	0	0
0	1	0	1	0

TABLE 4.2 – Exemple d'un ensemble de données

Notre approche de reconnaissance se base sur le calcul de la distance entre les vecteurs représentatifs et le vecteur à reconnaitre et on choisit la distance minimale pour attribuer une étiquette à ce dernier.

Si tous les capteurs d'un segment sont désactivés alors on élimine ce segment et on ne va pas l'étiqueter car il peut désigner une période d'inactivité, un silence, une transition entre les activités ou encore des valeurs aberrantes.

Algorithme de reconnaissance

Entrées :

- Les vecteurs représentatifs
- Le vecteur à reconnaitre

Début

Si tous les capteur d'un segment sont désactivés (vecteur à reconnaitre= [0]) alors :

- Éliminer se vecteur

Sinon

Pour chaque vecteur représentatif

- Calculer sa distance avec le vecteur à reconnaitre

Fin pour

- Chercher le vecteur représentatif ayant la distance minimale
- Attribuer l'étiquette de ce vecteur au vecteur à reconnaître

Sortie :

- Le vecteur reconnu

Fin de l'algorithme

4.6 Validation

En appliquant la méthode de clustering hiérarchique proposée dans les section présidente sur un jeu de données qui est composé de 50 instances de vecteurs caractéristiques qui décrit des activités effectués, on obtient le résultat présenté par le dendrogramme de la figure 4.1 :

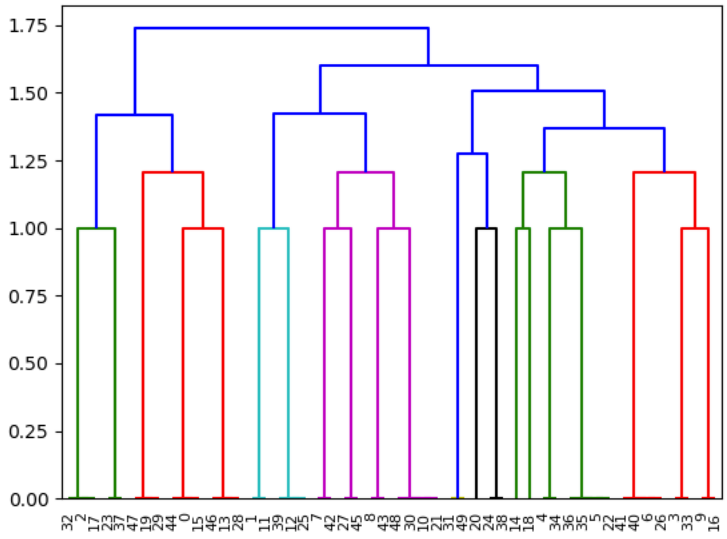

FIGURE 4.1 – Dendogramme obtenu de la validation de la méthode de clustering hiérarchique proposée.

Pour obtenir le nombre de clusters pour le clustering hiérarchique, nous utilisons un concept appelé un dendrogramme. Un dendrogramme est un diagramme fréquemment utilisé pour illustrer l'arrangement de clusters générés par un regroupement hiérarchique.

A chaque fois que nous fusionnons deux clusters, un dendrogramme enregistrera la distance entre ces clusters et la représentera sous forme de graphique tel que les échantillons du jeu de données sur l'axe des x et la distance sur l'axe des y. Chaque fois que deux clusters sont fusionnés, nous les rejoindrons dans ce dendrogramme et la hauteur de la jointure sera la distance entre ces points.

Nous pouvons clairement visualiser les étapes du clustering hiérarchique sur un dendrogramme. Plus la distance des lignes verticales dans le dendrogramme est plus grande est la distance entre ces groupes.

Maintenant, il faut définir une distance de seuil de telle sorte qu'il coupe la ligne verticale la plus haute et tracer une ligne horizontale. Le nombre de grappes sera le nombre de lignes verticales qui sont coupées par la ligne tracée à l'aide du seuil.

C'est ainsi que nous pouvons décider du nombre de clusters à l'aide d'un dendrogramme dans le clustering hiérarchique.

Les résultats sont intéressants car la majorité des vecteurs tests sont correctement classifier et notre approche a pu regrouper les vecteurs jugés comme les plus similaires dans un seul cluster.

4.7 Conclusion

Dans ce dernier chapitre qui mis fin a notre mémoire, nous avons d'abord présenté notre contribution dans les techniques de clustering hiérarchiques appliqué à la reconnaissance des activités humaines dans les systèmes ubiquitaires. Puis, on a illustré les résultats de validation.

Conclusion générale

Le présent travail est une description d'une nouvelle méthode de clustering hiérarchique afin de reconnaître les activités humaines dans un système ubiquitaire donné. Notre motivation a été d'étudier la problématique de clustering hiérarchique, qui a un champs d'applications très larges dans la reconnaissance des activités humaines. Pour cela, nous avons étudié les systèmes ubiquitaires dans le chapitre introductif afin de se familiarisé avec ces différents concepts. Puis, le clustering est abordé avec plus de détails dans le second chapitre, où on a exposé les propriétés liées au clustering, ses différents approches et ses caractéristiques. Nous avons vu un aperçu sur les méthodes de clustering hiérarchiques les plus utilisées dans le troisième chapitre ce qui nous a permis de dresser leurs points forts ainsi que leurs limites. Enfin, dans le dernier chapitre, nous expliquons notre contribution dans ce domaine.

Nous avons proposé une solution qui permet d'établir des classes d'objets les plus homogènes possibles, donner un sens à ces classes, et gagner en temps en se basant sur des vecteurs représentatifs choisi avant de commencer le processus de classification. Nous avons constaté que notre approche a permis d'obtenir de bon résultats. Elle a permis essentiellement d'avoir des clusters dont les objets sont les plus similaires les uns des autres.

La partie pratique de ce mémoire nous a permis de nous rendre compte des difficultés et du temps que demande un projet de clusterisation de données. L'évaluation de notre méthode de clustering hiérarchique sur un ensemble d'activités effectués au sein d'une maison laboratoire nous a permis d'aboutir à des résultats remarquables et de reconnaître les activités

effectués par ces occupants. De nombreux méthodes de clustering hiérarchiques existe mais sont généralement complexe. Notre méthode se manifeste par sa simplicité et sa capacité de donner des résultats assez satisafaisant.

Toutefois, ce travail nous a conduit à développer des perspectives très intéressantes qui s'inscrivent parfaitement dans la continuité des recherches que nous souhaitons aborder dans le futur afin d'améliorer notre algorithme :

- En étendant notre approche aux autres types d'attributs autres que binaires ;
- En étudiant la possibilité d'adapter notre algorithme aux ensembles de données plus vaste et à haute dimension ;
- En effectuant plusieurs activités au même temps ;
- En utilisant les capteurs portatifs sur le corps humain au lieux des capteurs à changement d'état ;
- En améliorant le temps de calcul.

Références bibliographiques

[1] L. Laouamer ; Approche exploratoire sur la classification appliquée aux images ; Thèse de Master ; université du québec ; Avril 2006.

[2] K. Abdessalem ; Classification non supervisée de textes arabes appliquée à la recherche documentaire ; These de Magister ; Université du 08 mai 45, Guelma ; 2007.

[3] A. Ben Cheikh ; E-CARe : une méthode d'ingénierie des systèmes d'information ubiquitaires ; Thèse de doctorat en Informatique ; Université de Grenoble ; Juin 2012.

[4] M. Weiser ; The computer for the 21st century ; Scientific American ; September 1991.

[5] J. Zhou, J. Riekki, M. Ylianttila, F. Tang, M. and Guo ; State of the art on pervasive service computing. In Proceedings of International Workshop on Ubiquitous Healthcare and Welfare Services and Supporting Technologies ; 2010 ; pages 1 - 6.

[6] L. Ruimin, C. Feng, Y. Hongji, C. William, and L. Yu-Bin ; Agent-based web services evolution for pervasive computing ; Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04) ; IEEE ; November 2004 ; pages 726 - 731.

[7] M. F. Khalfi, S. M. Benslimane ; Systèmes D'information Pervasifs : Architecture et Challenges ; Conference Paper ; Juin 2014.

[8] M. Beigl, A. Krohn, T. Zimmer, C. Decker ; Situation Aware Context Communication ; International conference on Ubiquitous Computing ; Seattle ; USA ; 2003 ; pages 157 - 160.

[9] S. Ghanem, Z. Bouanani. Gestion de contexte et découverte de service sensible au contexte dans un environnement ubiquitaire ; Thèse de master ; Université A.Mira, Béjaia ; 2012.

[10] J. Bourcier ; Auto-Home : une plate-forme pour la gestion autonomique d'applications pervasives ; Thèse de doctorat en Informatique ; 2008.

[11] A. S. Tanenbaum and M. Van Steen ; Distributed Systems : Principles and Paradigms ; Prentice Hall ; 1999.

[12] N. Plouzniko, J. M. Robert ; Caractéristiques, enjeux et défis de l'informatique portée ; École polytechnique de Montréal ; 2005.

[13] F. Boudjadi, L. Cheklat ; Conception et réalisation d'une application mobile sensible au contexte pour un musée ; Université A.Mira de Béjaïa ; 2013.

[14] M. A.Takerabet, A. Oussayah ; Gestion de la sécurité dans les systèmes sensibles aux contexte ; Thèse de master ; Université A.Mira de Béjaïa ; 2014.

[15] A. Azoui ; L'orientation service dans les systèmes ubiquitaires - SOA-Ubicomp : Application à la gestion de crise ; Mémoire de Magistère ; Université A.Mira, Béjaïa ; 2012.

[16] M. Selmi ; Reconnaissance d'activités humaines à partir de séquence vidéo ; Réseaux et télécommunication ; Institut National des Télécommunication ; 2014.

[17] D. Roggen, S. Magnenat, M. Waibel, and G. Troster ; Designing and sharing activity recognition systems across platforms ; IEEE Robotics et Automation Magazine ; 2011 ; pages 5 - 6.

[18] H. Khoufi Zouari ; Contribution à l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation ; Thèse de doctorat ; université de ROUEN U.F.R. des sciences et techniques ; 2004.

[19] H. Kautz, J. Allen ; Generalized plan recognition ; National Conference on Artificial Intelligence (AAAI) ; 1986 ; pages 32 - 37.

[20] H. A. Kautz ; A formal theory of plan recognition and its implementation ; Reasoning About Plans ; Jul 1991 ; pages 69 - 124.

[21] L. R. Rabiner ; A tutorial on hidden Markov models and selected applications in speech recognition ; in Readings in speech recognition ; ed : Morgan Kaufmann Publishers Inc ; 1990 ; pages 267 - 296.

[22] H. K. Donald, J. Patterson, Lin Liao, Dieter Fox ; Inferring High-Level Behavior from Low-Level Sensors ; Proc. Fifth Int. Conf. Ubiquitous Comput ; 2003 ; pages 73 - 89.

[23] R. Besançon, A. L. Daquo ; Clustering de documents dans des collections hétérogènes ; Document numérique ; Vol 18 ; 2015 ; pages 81 - 100.

[24] J. Han, M. Kamber ; Data mining : Concepts and techniques, Management Systems (The Morgan Kaufmann Series in Data Management Systems) ; MORGAN KAUFFMAN ; ISBN 1-55860-901-6 ; 2006.

[25] A. Yahi ; Clustering des données de puces à ADN ; Thèse de master ; Université M. BOUDIAF ; M'SILA ; 2019.

[26] F. Long, H. Zhang, D. D. Feng ; Fundamentals of content-based image retrieval ; Multimedia Information Retrieval and Management ; Springer Berlin Heidelberg ; 2003.

[27] C. H. C. Leung, Y. Li ; Semantic Image Retrieval using collaborative indexing and filtring ; In web Intellegent Agent Technology (WI-IAT) ; 2012 IEEE/WIC/ACM International Conferences ; vol 3 ; 2012 ; pages 261 - 264.

[28] G. Bisson ; La similarité : une notion symbolique/numérique ; Chap. XX of :

Apprentissage symbolique-numérique (tome 2) ; Editions CEPADUES ; 2002.

[29] L. Rokach, O. Maimon ; Clustering Methods ; Tel-Aviv University ; 2002.

[30] G. Cleuzieu ; une méthode de classification non-suppervisé pour l'apprentissage de règle et la recherche d'information ; Thèse de doctorat ; université d'orléans ; 2004.

[31] N. Beck ; Application de méthodes de clustering traditionnels et extension au cadre multicritère ; thèse de magister ; Université libre de Bruxelles ; 2006.

[32] M. Koudri ; Segmenter via Expectation Maximization ; Thèse de master ; université de Tlemcen ; pages12 - 45.

[33] K. Zeitouni ; Techniques de data mining ; cours de Master Profssionnel ; ASS ; Edition 2009.

[34] J. McQueen ; Some methods for classification and analysis of multivariate observations ; Proc. 5th Berkeley Symp ; Math, Statistics and Probability ; 1967 ; pages 281 - 296.

[35] L. Kaufman, P. J. Rousseeuw ; Finding Groups in Data : an Introduction to Cluster Analysis ; John Wiley & Sons ; 1990.

[36] D. P. Mercer ; Clustering large datasets : Linacre College ; raport de recherche ; 2003.

[37] P. Rai, S. Singh ; A survey of clustering techniques ; International Journal of Computer Applications (0975 - 8887) Volume 7- No.12 ; October 2010.

[38] X. Xu, M. Ester, H. P. Kriegel, J. Sander ; A distribution-based clustering algorithm for mining in large spatial databases ; In 14th International Conference on Data Engineering ; USA ; 1998 ; pages 324 - 331.

[39] B. Deveze, M. Fouquin ; Data Mining C4.5 -DBSCAN ; Cours, Ecole d'ingénieurs en informatique EPITA ; France ; 2004.

[40] D. Renaudie ; méthodes d'apprentissage automatique pour la modélisation de l'élève en algèbre ; thèse de doctorat ; institut national Polytechnique de grenoble ; 2005.

[41] O. OUAMRI ; Contribution des arbres dirigés et les k-means pour l'indéxation et recherche d'images par contenu ; Mémoire de Magister ; Université des Sciences et de la Technologie d'Oran ; 2007.

[42] R. Agrawal, J. Gehrke, D. Gunopoulos, P. Raghavan ; Automatic Subspace Clustering of High Dimensional Data for Data-mining Applications ; Proceedings of ACM SIGMOD Int ; Conf on Management of Data ; 1998 ; pages 5 - 33.

[43] G. Sheikholeslami, S. Chatterjee, A. Zhang ; WaveCluster : A Multi-Resolution Clustering Approach for Very Large Spatial Databases ; Procs of 24th VLDB Conf ; USA ; 1998 ; pages 289 - 304.

[44] Y. Batistakis, M.vazirgiannis, M. Halkidi ; Clustering validity checking methods ;

SIGMOD Record ; 2001.

[45] T. L. M. van Kasteren, G. Englebienne, B.J.A. Kröse ; Hierarchical Activity Recognition Using Automatically Clustered Actions ; Department of Computer Engineering ; Amsterdam ; 2011.

[46] S. Kafle ; A heterogeneous clustering approach for Human Activity Recognition ; Big Data Analytics and Knowledge Discovery ; 2016 ; pages 68 - 81.

[47] P. H. A. Sneath, R. R. Sokal ; Numerical taxonomy - the principles and practice of numerical classiffication ; Technical report ; San Francisco ; 1973.

[48] R. Yogita, R.Harish ; A Study of Hierarchical Clustering Algorithm ; International Journal of Information and Computation Technology ; 2013.

[49] A.Blum, T.Mitchell ; combining labeled and unlabeled data with co-training ; COLT : Proceedings of the Workshop on Computational Learning Theory ; Morgan Kaufmann ; 1998 ; pages 92-100.

[50] S. TUFFÉRY ; Data mining et statistique décisionnelle : l'intelligence dans les bases decisionnelle ; Edition Technip ; Paris ; 2005 ; pages 133 - 146.

[51] T. Zhang, R. Ramakrishnan, M. Linvy ; BIRCH : an efficient data clustering method for large databases ; International Conference on Management of Data ; Montreal ; 1996.

[52] R. Yogita, R.Harish ; A Study of Hierarchical Clustering Algorithm ; International Journal of Information and Computation Technology ; 2013.

[53] N. Masmoudi ; Modèle bio-inspiré pour le clustering de graphes : application à la fouille de données et à la distribution de simulations ; Thèse de doctorat en Informatique ; Université de Normandie ; Université de Sfax (Tunisie) ; 2017.

[54] S. Guha, R. Rastogi, K. Shim ; Cure : An efficient clustering algorithm for large databases ; Conf. Management of Data ; Seattle, WA ; June 1998.

[55] S. S. Mary, T. Selvi ; A Study of K-Means and Cure Clustering Algorithms ; International Journal of Engineering Research & Technology (IJERT) ; Février 2014 ; pages 1 -3.

[56] S. Guha, R. Rastogi, K. Shim ; ROCK : A Robust Clustering Algorithm for Categorical Attributes ; Information Systems ; vol.25 ; No 5 ; 2000 ; pages 1 - 10.

[57] G.Karypis, E. H. Han, V. Kumar ; CHAMELEON : Hierarchical Clustering Using Dynamic Modeling ; Computer ;vol.32 ; No 8 ; 1999 ; pages 1 - 8.

[58] G. Cleuziou ; Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information ; Thèse de doctorat en Informatique ; Université d'Orléans ; 2004.

[59] D. Fisher ; Knowledge acquisition via incremental conceptual clustering. Machine Learning ; vol.2 ; 1987 ; pages 139 - 172.

[60] A. Satyanarayana, V. Acquaviva ; Enhanced Cobweb Clustering for Identifying Analog Galaxies in Astrophysics ; IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) ; Canada ; 2014 ; pages 1 - 4.

[61] N. Yala ; Contribution aux méthodes de classification de signaux de capteurs dans un habitat intelligent ; Thèse de Doctorat ; Université des sciences et de la technologie Houari Boumedienne ; 2019.

[62] B.S. Everitt ; Cluster analysis ; Edward Arnold and Halsted Press ; 1993.

[63] A. K. Jain, R. C. Dubes ; Algorithms for clustering data ; Prentice Hall Advanced Reference Series ; 1988.

Résumé

L'apparition des Systèmes ubiquitaires ou pervasifs est issue de l'émergence de nouvelles technologies fournissant au système une vision de son environnement, de l'environnement de ses utilisateurs ainsi que de leurs profils. Grâce à ces données formant le contexte de l'application, il est possible de fournir des services personnalisés, pertinents et ciblés. Mais, le problème qui se pose à ce niveau concerne la reconnaissance des activités effectuées par les utilisateurs tout en tenant compte du grand nombre de données.

La classification des données est la phase la plus importante dans la chaine de reconnaissances des activités humaines. Dans notre étude nous avons mis l'accent sur les techniques de classification non supervisée (Le clustering).

Le clustering est un problème d'une grande importance dans de nombreuses applications et il devient plus difficile lorsqu'on manipule un grand volume de données. Ce travail consiste à présenter les principales méthodes de clustering hiérarchiques utilisées dans l'exploration de données, leurs caractéristiques, avantages et inconvénients.

Ce mémoire contient une partie théorique sur les systèmes ubiquitaires et le clustering en général. Puis nous avons décrit notre contribution au développement d'un algorithme de clustering hiérarchique.

Mots clés : systèmes ubiquitaires, reconnaissance d'activités humaines, clustering hiérarchique, similarité, dissimilarité.

Abstract

The appearance of ubiquitous or pervasive Systems is the result of the emergence of new technologies providing the system with a vision of its environment, the environment of its users as well as their profiles. Thanks to this data forming the context of the application, it is possible to provide personalized, relevant and targeted services. However, the problem which arises at this level concerns the recognition of the activities carried out by the users while taking into account the large amount of data.

The classification of data is the most important phase in the chain of recognition of human activities. In our study we focused on unsupervised classification techniques (clustering).

Clustering is a problem of great importance in many applications and it becomes more difficult when dealing with a large volume of data. This work consists in presenting the main hierarchical clustering methods used in data mining, their characteristics, advantages and disadvantages.

This thesis contains a theoretical part on ubiquitous systems and clustering in general. Then we described our contribution of the development of a hierarchical clustering algorithm.

Key words : ubiquitous systems, recognition of human activities, hierarchical clustering, similarity, dissimilarity.