UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
BOKHABRINE AYOUB

VERS UNE PLATEFORME INFORMATIQUE POUR L'EXPERIMENTATION D'OUTILS DE CLASSIFICATION

AOÛT 2019

RÉSUMÉ

Notre projet de recherche consiste à mettre en place une plateforme pour l'expérimentation d'outils de classification. Cette plateforme doit permettre de combiner à la fois les règles d'association et la classification. Nous démontrons que grâce aux règles d'association, les résultats de la classification sont plus homogènes que si des mots avaient été utilisés comme descripteurs de textes.

Pour accomplir ce projet et réaliser notre plateforme, nous avons eu recours aux différentes technologies et techniques suivantes :
Microsoft C#, XML, Microsoft Access, RStudio.

Mots clés : Classification, règles d'association, descripteur, expérimentation.

DÉDICACES

Je dédie cet ouvrage.

À mes parents qui m'ont soutenu et encouragé durant mon parcours d'études. Espérant qu'ils trouvent dans ce mémoire le témoignage de ma profonde reconnaissance.

À ma chère femme Gulmira, que je ne trouverais jamais les mots pour la remercier.

À tous ceux qui ont partagés avec moi tous les moments d'émotion lors de la réalisation de ce projet.

À tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plein succès.

BOKHABRINE Ayoub.

REMERCIEMENTS

Je profite de ce mémoire pour exprimer mes plus sincères remerciements à tous ceux qui ont pu contribués, à leur manière, à rendre cette recherche et ce séjour au Canada si intéressant et si enrichissant.

Je tiens à remercier particulièrement Madame Ghazzali Nadia, Monsieur Biskri Ismaïl, d'avoir accepté de m'encadrer et pour leurs conseils judicieux, leurs orientations avisées ainsi que leurs soutiens permanents durant mes études.

Je ne terminerai pas, sans remercier l'Université du Québec à Trois-Rivières ainsi que tout son personnel pour leur amabilité, leur efficacité et leur constant dévouement.

BOKHABRINE Ayoub.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

SIGLES ET ABRÉVIATIONS

Acronyme	**Description**
ACP	Analyse en composantes principales
TID	Transaction Identifier
CAH	Classification Ascendante Hiérarchique
SOM	Self Organizing Map en français « Cartes Auto Adaptatives »
PMC	Perceptron MultiCouche
PDF	Portable Document Format
HTML	HyperText Markup Language
XML	Extensible Markup Language
DOC	Text Document
NBA	National Basketball Association

CHAPITRE 1 INTRODUCTION

L'évolution qu'a connu internet ces dernières décennies a contribué à l'augmentation de documents textuels électroniques facilitant la diffusion de l'information. Ces documents engendrent une quantité d'informations non structurées rendant difficile l'analyse et l'exploration de toutes les données. Plusieurs méthodes de forage de données (Text Mining) dont les règles d'association et la classification ont été proposées en vue de déceler des informations pertinentes.

De nos jours, l'extraction d'information évolue d'une façon à la fois rapide et complexe, suscitant de plus en plus l'intérêt des chercheurs dans le monde de la fouille de données. Retrouver de l'information pertinente dans des données textuelles qu'elles soient structurées, non structurées ou ambiguës nécessite de nouvelles méthodes et approches.
Dans la littérature scientifique, les règles d'association et les différentes méthodes de classification qu'elles soient supervisées ou non-supervisées font parties des approches et méthodes proposées. Néanmoins, il semble que l'identification de ce que doivent être les descripteurs d'un texte reste un enjeu central. En effet, nous constatons que le mot est souvent utilisé comme descripteur de texte, le N-Gram de caractères est également utilisé dans certains travaux où le multilinguisme est à considérer (Biskri et al. 2013).

Dans ce cadre, notre défi est d'extraire de l'information pertinente à partir des documents textuels électroniques (XML, PDF, DOC, HTML, etc). Dans ce mémoire, nous nous sommes penchés sur une approche novatrice pour résoudre cette problématique en explorant les règles d'association dans le but d'établir de nouveaux descripteurs du contenu d'un texte « itemsets fréquents ». L'application de plusieurs classifieurs sur ces descripteurs vient renforcer la pertinence de l'exploration et l'analyse des données textuelles par la mise en œuvre d'une plateforme informatique combinant l'utilisation des règles d'association et la classification automatique.

L'hypothèse de cette recherche est donc que l'utilisation des itemsets comme descripteurs du texte permet une meilleure classification, contrairement à l'utilisation des mots comme descripteurs du texte.

Pour rappel, dans la littérature scientifique, plusieurs chercheurs ont focalisé leurs efforts sur l'application des règles d'association pour des fins d'extraction de l'information, dont (Labiad, 2017), et (Nouasria, 2016).

Ce mémoire est composé de six chapitres et structuré comme suit :

Le chapitre 1 introduit le cadre de notre travail ainsi que le contenu de mémoire.

Le chapitre 2 passe en revue les règles d'association standards, maximales et séquentielles. Il fait le point sur leurs concepts de base et leurs algorithmes associés.

Au chapitre 3, nous présentons deux méthodes de classification non supervisées. La première méthode concerne la classification hiérarchique avec toutes ses propriétés, tandis que la deuxième concerne la classification non hiérarchique.

Au chapitre 4, nous abordons notre méthodologie pour l'extraction de nouveaux descripteurs du texte et l'utilisation de ces derniers dans un processus de classification, Nous donnons des détails sur les paramètres à considérer ainsi que les types de classifieurs proposés.

Nous consacrons le chapitre 5 à la présentation et à la discussion des résultats obtenus lors de la phase d'expérimentation. Celle-ci porte sur 2 types de texte, le premier est construit pour valider notre méthodologie, tandis que le second est un cas réel pour évaluer l'approche proposée. Par la suite, nous présentons la plateforme que nous avons développé pour le besoin des expérimentations. Cette plateforme est appelée IDETEX (Itemsets comme DEscripteurs de TEXtes).

Enfin, nous terminons ce mémoire par une conclusion générale, puis une présentation de différentes perspectives, qui feront certainement l'objet de travaux de recherche ultérieurs.

Ce mémoire repose sur une publication parue à la conférence ICCCI 2019 « 11th International Conference on Computational Collective Intelligence », dont le titre est « Frequent Itemsets as Descriptors of Texual Records ». Nous l'avons inclus en Annexe. Cette publication, a par ailleurs, été sélectionnée pour la parution d'une version étendue dans un journal international.

CHAPITRE 2 RÈGLES D'ASSOCIATION

2.1. Introduction

Initialement introduit sous le nom de « GUHA » par Hajek (Hajek et al. 1966), le concept de règles d'association a été popularisé par Agrawal (Agrawal et al. 1993). Cette méthode est appliquée à plusieurs domaines tels que l'Ingénierie, la Médecine, l'Astronomie, la Chimie, l'agriculture, etc.

Le développement fulgurant des données accentue le besoin d'utiliser des outils d'analyse, d'extraction des dépendances et des corrélations pertinentes entre ces données. Le tableau 2.1 illustre un exemple sur la nature de données collectées par les achats des consommateurs aux supermarchés. Chaque ligne du tableau représente une transaction identifiée par un numéro qu'on nomme « *transaction identifier* » dont l'abréviation est « TID » et contient les produits achetés.

Les Méthodes d'analyse de données permettent aux commerçants de bien comprendre le comportement de leurs clientèles.

TID	Items
1	Clavier, Écouteurs
2	Tablette, Étui, Imprimante
3	Écouteurs, Tablette, Étui
4	Clavier, Tablette, Écouteurs, Étui
5	Moniteur, Écouteurs, Tablette, Clavier

Tableau 2.1 - Achats des consommateurs

Une simple observation sur les données du tableau 2.1 montre qu'il existe une forte relation entre la vente des tablettes et la vente d'étuis. Par Conséquent, les clients qui achètent les tablettes achètent aussi les étuis.

Cette relation est représentée sous-forme d'une règle d'association comme suit :

Tablette → Étui

Cette règle est lue comme suit : si une condition existe « Antécédent », alors forcément, un résultat issu de celle-ci existe aussi « Conséquent ». (Descôteaux, 2014).

Généralement, une règle d'association prend la forme suivante :

Antécédent → Conséquent

Dans ce chapitre, nous allons expliquer les concepts de base des règles d'association et leurs algorithmes associés. Tant pour les définitions que pour les exemples, ce chapitre est fortement inspiré du travail réalisé par Agrawal (Agrawal et al. 1993).

2.2. Notions et Définitions

Dans cette section, nous allons détailler plusieurs notions pour mieux expliquer comment extraire des règles d'association pertinentes : Transaction, Item, Support, Confiance.

2.2.1. **Transaction et Items**

Soit T un ensemble composé de n transactions tel que $T = \{t_1, t_2, \dots, t_j, \dots t_n\}$, et soit I un ensemble de m items distincts $I = \{i_1, i_2, \dots, i_k, \dots i_m\}$. Chaque transaction $t_j \in T$ est constituée d'un sous-ensemble d'items $\subseteq I$ où $t_j = \{i_1, i_2, \dots, i_l\}$.

Chaque transaction t_j est représentée comme un vecteur binaire, avec $t[k] = 1$ si l'item I_k est acheté, sinon $t[k] = 0$.

2.2.1.1. ***Exemple***

Soient I = {clavier, écouteurs, tablette, étui, imprimante, moniteur} l'ensemble de tous les items des paniers et T= $\{t_1, t_2, t_3, t_4, t_5\}$, l'ensemble de toutes les transactions. Le tableau 2.2 illustre une représentation binaire des transactions et leurs items.

TID	Clavier	Écouteurs	Tablette	Étui	Imprimante	Moniteur
1	1	1	0	0	0	0
2	0	0	1	1	1	0
3	0	1	1	1	0	0
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Tableau 2.2 - Présentation en binaire des transactions

2.2.2. **Itemset**

Dans l'analyse d'association, une collection d'un ou de plusieurs items est appelée un itemset. Si un itemset contient k items, il est appelé un k-itemset.

Une transaction t_j de taille x (x = nombre d'items formant cette transaction) contient un itemset I_1, si et seulement si $I_1 \subseteq t_j$. Le nombre d'itemsets possible est de 2^x. Pour réduire ce nombre et bien cibler les itemset fréquents, il faut que le support d'itemset soit supérieur ou égal à un seuil (Support minimum) défini par l'utilisateur, nommé *minsup*.

2.2.3. **Support**

Le support **S(X)** est un concept fondamental d'un itemset (X), on le définit par le nombre de transactions qui le contient **σ(X)** divisé par le nombre total de transactions n.

$$S(X) = \frac{\sigma(X)}{n}$$

Où :

- $\boldsymbol{\sigma(X) = |\{t_i \mid X \subseteq t_i, t_i \in T\}|}$ = *nombre de transactions qui contient X*
- $\boldsymbol{n}$ = *nombre total de transactions*

2.2.3.1. ***Exemple :***

En appliquant la formule précédente sur les données du tableau 2.1, on trouve que le support de {**Tablette**, **Étui**} est égal à $^{\mathbf{3}}/_{\mathbf{5}}$.

2.3. Règle d'association Standard

Une règle d'association est une application de la forme $A \rightarrow B$ où $A \subset X$, $B \subset X$ et $A \cap B = \phi$. La règle $A \rightarrow B$ se produit dans l'ensemble de transactions T avec un support S, qui est le pourcentage de transactions de T qui contient $A \cup B$, et une

confiance C qui est le pourcentage de transactions T contenant A qui contient également B.

Dans le contexte du forage de données et de la découverte des connaissances, une fonction d'évaluation indique la qualité des itemsets examinés à partir d'un ensemble de données à savoir le support et la confiance.

2.3.1. **Propriétés et opérations des règles d'association classiques**

Afin d'extraire des règles d'association pertinentes, on a recours à plusieurs opérations qu'on détaille ci-après.

2.3.1.1. ***Support d'une règle d'association***

Le support d'une règle d'association **A → B** est le nombre de transactions qui contiennent **A ∪ B**, divisé par le nombre total **n** des transactions.

$$S(A \rightarrow B) = \frac{|\{t \in D/(A \cup B) \subseteq t\}|}{n}$$

Ou bien : $$S(A \rightarrow B) = \frac{\sigma(A \cup B)}{n}$$

Le tableau 2.3 représente la mesure de support de quelques règles d'association générées à partir des transactions du tableau 2.1.

Règles	Support
{**Clavier**} → {**Écouteurs**}	3/5 = 0.6
{**Tablette**} → {**Étui**}	3/5 = 0.6
{**Tablette**, **Étui**} → {**Écouteurs**}	2/5 = 0.4
{**Tablette**} → {**Moniteur**}	1/5 = 0.2

Tableau 2.3 - Mesure du support d'une règle d'association

Considérant la règle {Tablette} → {Étui}, le support de l'ensemble {*Tablette*, Étui} égale à 3 et le nombre total des transactions qui est égal à 5, ce qui induit un support de 3/5 = 0.6.

2.3.1.2. *Confiance d'une règle d'association*

La confiance d'une règle d'association C ($A \rightarrow B$) représente le ratio du nombre de transactions supportant la règle (antécédent et conséquent) divisé par le nombre de transactions supportant l'antécédent. On note comme suit :

$$C(A \rightarrow B) = \frac{S(A \rightarrow B)}{S(A)}$$

Le Tableau 2.4 représente la mesure de confiance de quelques règles d'association générées à partir des transactions du tableau 2.1.

Règles	Confiance
{**Clavier**} → {**Écouteurs**}	3/3 = 1
{**Tablette**} → {**Étui**}	3/4 = 0.75
{**Tablette**, **Étui**} → {**Écouteurs**}	2/3 = 0.67
{**Tablette**} → {**Moniteur**}	1/4 = 0.25

Tableau 2.4 *- Mesure de la confiance d'une règle d'association*

La confiance mesure la fiabilité et la précision d'une règle d'association, Considérant la règle d'association {Tablette} → {Étui}, la confiance est obtenue en divisant le support de l'ensemble {Tablette, Étui} qui est 3, par le support de l'ensemble {Tablette}. Comme il y a 4 transactions contenant {Tablette}, la confiance de cette règle {Tablette} → {Étui} est de : $\frac{3}{4} = 75\%$, cela signifie que 75% des transactions qui contiennent {Tablette} contiennent aussi {Étui}.

La mesure de support est souvent utilisée pour éliminer les règles triviales, comme dans le cas de la règle {Tablette}→{Moniteur} du tableau 2.3 qui a un support faible.

La confiance mesure ainsi la pertinence de l'inférence faite par une règle. Plus la confiance de $X \rightarrow$ Y est élevée, plus la probabilité d'observer Y avec X est forte. La confiance donne aussi une estimation de la probabilité conditionnelle de Y sachant X.

Les règles de la forme $X \rightarrow$ Y sont dites pertinentes si $S(X \rightarrow Y) > S_0$ et $C(X \rightarrow Y) > C_0$.
Où : S_0 et C_0 sont des seuils minimums définis au préalablement par l'utilisateur.

2.4. Processus d'extraction des règles d'association

Il y a plusieurs façons de réduire le coût de la recherche d'ensembles d'itemsets fréquents :

- **Réduire le nombre d'ensembles d'items candidats :** Le principe Apriori, décrit dans la section suivante, est un moyen efficace d'éliminer certains candidats sans évaluer leur support.
- **Réduire le nombre de comparaisons :** Au lieu de comparer chaque ensemble d'items candidats à toutes les transactions, on peut réduire le nombre de comparaisons en utilisant des structures de données plus complexes, soit pour stocker les ensembles d'items, soit pour compresser la base de données.

2.4.1. L'algorithme Apriori

L'algorithme Apriori représente la base de tous les algorithmes de recherche des règles d'association. Il extrait les itemsets fréquents pour les règles d'association. Cet algorithme est proposé par Agrawal et Srikant en 1994.

2.4.1.1. *Explication de l'algorithme Apriori*

L'algorithme Apriori est donné par la suite d'instructions suivantes :

$k \leftarrow 1;$
$L_k \leftarrow I;$
TANTQUE $(L_{k-1} <> 0)$ FAIRE
$C_k = \boldsymbol{apprioriGen}(L_k);$
TANTQUE $t \in T$ FAIRE
$C_t = \boldsymbol{sousensemble}(C_k, t);$
TANTQUE $c \in C_t$ FAIRE
$c.count++;$
FIN TANTQUE
FIN TANTQUE
$L_k \leftarrow \{c \in C_k \mid c.count \geq minsup\};$
$k \leftarrow k + 1;$
FIN TANTQUE
Retourner $\bigcup_k F_k$

Générations des règles d'association

Figure 1 - Pseudo-code de l'algorithme Apriori

Les données C_k et L_k sont des ensembles d'enregistrements contenant deux champs :

C_k : Cette variable contient toutes les combinaisons d'itemsets possibles.

L_k : Cette variable ne contient que les itemsets fréquents dont les leurs supports sont supérieurs au support minimum.

$minsup$: Support minimum fixer par l'utilisateur.

L'algorithme Apriori permet de découvrir les sous-ensembles d'items fréquents en partant de ceux dont la longueur est 1 et en augmentant la longueur au fur et à mesure. Cet algorithme est fondé sur la propriété des sous-ensembles d'items fréquents. Il fait appel à des fonctions :

apprioriGen : L'algorithme apprioriGen est constitué de deux phases :

- La première phase nommée **Joindre,** trouve tous les candidats possibles de longueur K à partir de l'ensemble L_{k-1}.
- La deuxième phase nommée **Effacer,** efface de C_k les éléments qui ne vérifient pas la propriété des sous-ensemble fréquents.

Sousensemble : cet algorithme calcule le sous-ensemble $C_T \subseteq C_K$ qui correspond à des sous-ensembles présents dans les transactions contenues dans T.

Générations des règles d'association: A partir d'un itemset fréquent I, l'algorithme construit toutes les règles de la forme $X \longrightarrow Y$ où X et Y, sont deux sous-itemsets de I qui ne possèdent pas d'items en commun et qui redonnent I par conjonction : $X \cap Y = I$ (Blanchard, 2005).

La confiance d'une telle règle est calculée de la manière suivante :

$$C(X \rightarrow Y) = \frac{S(X \longrightarrow Y)}{S(X)}$$

Finalement, l'algorithme fournit l'ensemble des itemsets fréquents et les règles validées par le seuil de confiance. L'indice de support pour chaque itemset est conservé et sera utilisé pour le calcul des différentes mesures d'intérêts qui enrichissent les règles extraites.

2.4.1.1. *Principe de Apriori*

L'algorithme Apriori utilise une approche itérative, où k-itemsets sont employés pour explorer les (k + 1) - itemsets. D'abord, les 1-Itemsets sont trouvés par un balayage de la base de données pour calculer le support de chaque item et la collecte de ces itemsets qui ont un support $\geq minsup$. L'ensemble résultant est noté L_1 (chaque L_k

servit à construire l'étape suivante), puis utilisé pour trouver L_2 , aussi l'ensemble résultant les 2-itemsets est utilisé pour trouver L_3 , et ainsi de suite jusqu'à ce qu'aucun k-itemsets ne puisse être trouvé. L'obtention de chaque L_k nécessite une analyse complète de la base de données (Han et Kamber, 2006).

Si un ensemble d'items est fréquent, alors tous ses sous-ensembles sont aussi fréquents.

2.4.1.2. *Exemple d'explication de l'algorithme Apriori*

Dans cet exemple, on simplifie le fonctionnement de l'algorithme Apriori tout en se basant sur les données du tableau 2.2 qui contiennent 5 transactions. On suppose que le support minimum requis est égal à 35%, soit, minsup = 35%.

1 Dans la première itération de l'algorithme (figure 2), chaque item appartient à l'ensemble l-itemsets C_1. Par la suite, il procède au calcul de l'occurrence de chaque élément de C_1. Après avoir calculé les occurrences, il compare chaque valeur avec la valeur minsup afin d'éliminer les items qui ne satisferont pas le minsup. Le résultat obtenu détermine l'ensemble fréquent L_1.

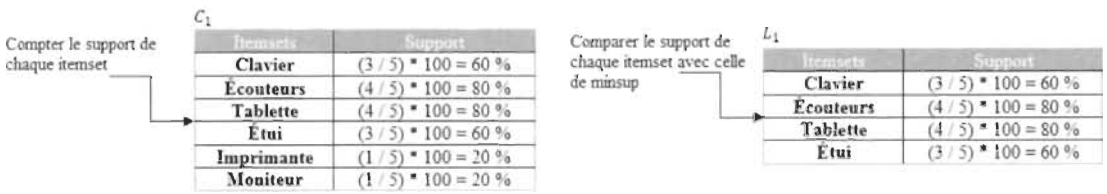

Figure 2 - Première itération de l'algorithme Apriori

2 Dans la deuxième itération (figure 3), l'algorithme génère l'ensemble des candidats 2-itemsets C_2 .Ensuite, il analyse les transactions dans le but de calculer le support de chaque candidat du C_2 . Enfin, il supprime les candidats dont le support est inférieur à minsup pour obtenir l'ensemble fréquent L_2.

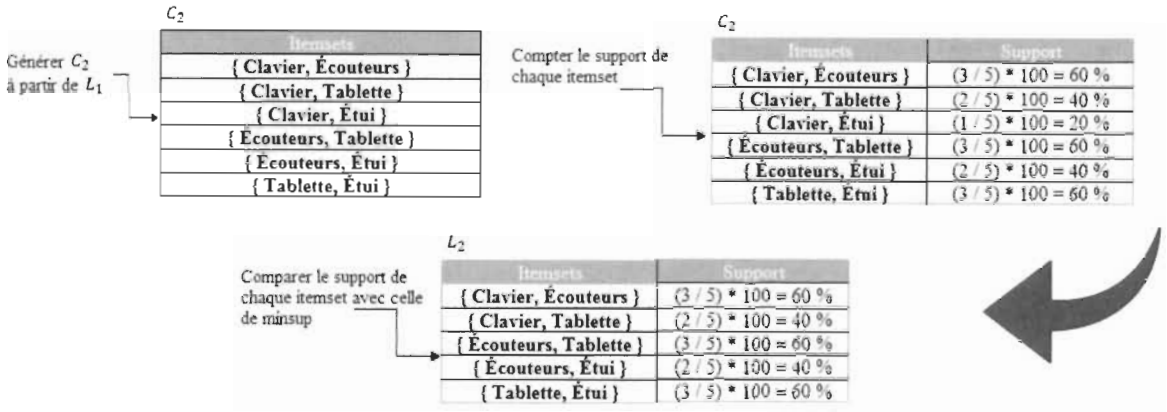

Figure 3 - Deuxième itération de l'algorithme Apriori

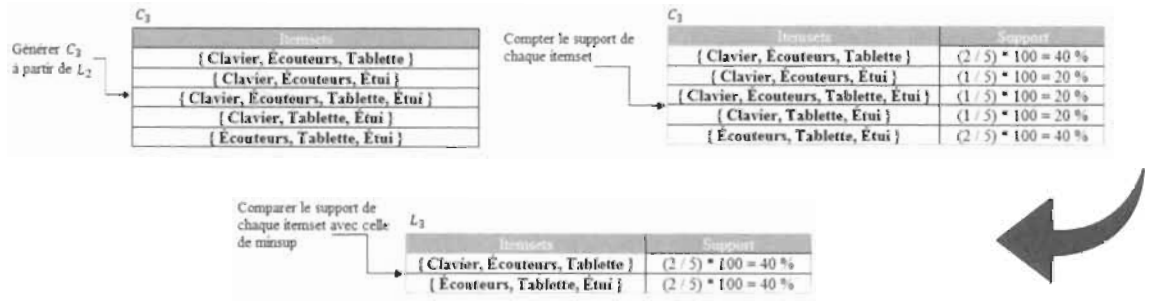

Figure 4 - Troisième itération de l'algorithme Apriori

3 Quant à la troisième itération (figure 4), l'algorithme réalise la génération de l'ensemble des candidats 3-itemsets. il scanne les transactions pour calculer l'occurrence de chaque itemset. L'ensemble fréquent L_2 est déterminé par l'élimination des candidats qui ne satisferont pas le minsup.

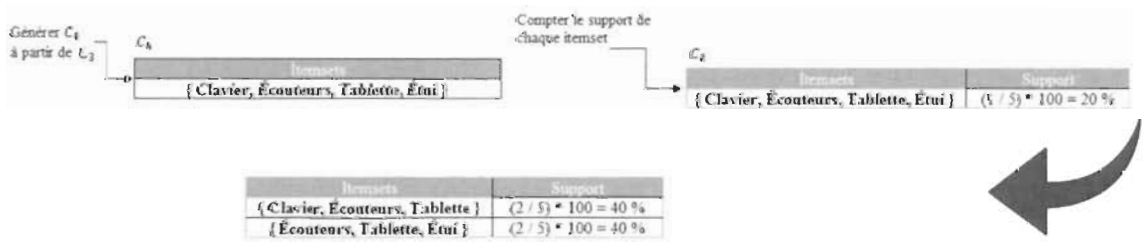

Figure 5 - Quatrième itération de l'algorithme Apriori

Enfin, L'algorithme calcule l'ensemble C_4. Le résultat est constitué d'un seul candidat : {Clavier, Écouteurs, Tablette, Étui} qui ne satisfait pas le minsup. Ainsi, l'algorithme se termine, à ce stade-là en affichant tous les Itemsets fréquents trouvés.

2.4.2. **Avantages et limites des règles d'association**

2.4.2.1. ***Avantages***

Les règles d'association représentent plusieurs avantages, parmi lesquels :

1. Leurs applications dans plusieurs domaines de la vie quotidienne, comme l'analyse du panier de la ménagère.
2. La découverte de connaissances utiles, cachées dans les grandes bases des données.
3. Leurs simplicités, efficacités et facilités de compréhension.
4. Leurs formalismes non-supervisés et généraux.
5. Leurs résultats clairs et faciles à interpréter.

2.4.2.2. ***Limites***

1. Le temps énorme consacré à la recherche des Itemsets fréquents. (Cherfi et Toussaint, 2002)
2. La grande quantité des règles d'association générées. (Gras et al. 2004)
3. La difficulté d'évaluer la qualité des règles d'associations par des indices statistiques ou par l'expert du domaine. (Diop et al, 2007)
4. La production des règles triviales et inutiles qui n'apportent pas de nouvelles informations. (Abdelali et Hicham 2003)

2.5. Les règles d'association maximales

Les règles d'association maximales s'inscrivent dans la continuité des règles d'association standards, cependant, elles se diffèrent dans la manière de calcul de support et de confiance.

Une règle d'association régulière $\boldsymbol{X \rightarrow Y}$ affirme que lorsqu'on voit $\boldsymbol{X}$ il faut aussi s'attendre à voir $\boldsymbol{Y}$ (avec une certaine confiance). Cependant, les règles d'association maximales $\boldsymbol{X \xrightarrow{max} Y}$ indiquent que lorsqu'on voit $\boldsymbol{X}$ tout seul on doit aussi s'attendre à voir $\boldsymbol{Y}$ seul.

2.5.1. **Taxonomie et catégorie**

Soit un ensemble d'items $\boldsymbol{I} = \{\boldsymbol{i_1}, \boldsymbol{i_2}, \boldsymbol{i_3}, \ldots, \boldsymbol{i_n}\}$, une taxonomie $\boldsymbol{T}$ de $\boldsymbol{I}$ qui représente les sous-ensembles disjoint d'items de $\boldsymbol{I}$, c'est-à-dire $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}, \ldots, \boldsymbol{t_n}\}$, chaque élément de $\boldsymbol{T}$ est appelé une catégorie d'items de $\boldsymbol{I}$. On notera $\boldsymbol{T(i)}$ la catégorie de $\boldsymbol{i}$, pour tout item de $\boldsymbol{I}$, si $\boldsymbol{X}$ est un ensemble d'itemsets qui sont tous d'une seule catégorie, alors on notera cette catégorie par $\boldsymbol{T(X)}$.

Transactions	*Items*
1	***Mexique, Maroc, Canada, Mines, Transport***
2	***Mexique, Maroc, Canada, Mines, Transport***
3	***Canada, Croissance***
4	***Canada, Emplois, Prix***
5	***Canada, Emplois, Prix***
6	***Canada, Croissance, Prix***
7	***Mexique, Mangue, Poivre***
8	***Mexique, Canada, Vente, Taxe***
9	***Mexique, Canada, Vente, Taxe***
10	***Mexique, Canada, Croissance***

Tableau 2.5 - Exemple de base de transactions

Nous pouvons extraire du tableau 2.5 une taxonomie $\boldsymbol{T}$ qui compte deux catégories : $\boldsymbol{T} = \{\boldsymbol{t_1} = \text{"pays"}, \boldsymbol{t_2} = \text{"}\boldsymbol{sujets}\text{"}\}$, où pays = {Mexique, Canada, Maroc} et sujets = {Mines, Transport, Croissance, Emplois, Prix, Mangue, Poivre, Vente, Taxe}

2.5.2. **M-support d'itemset**

Pour un ensemble de transactions D, le M-support de $\boldsymbol{X}$ dans D, notée par $\boldsymbol{S_D^{max}(X)}$, le nombre de transactions $\boldsymbol{t_a} \in \boldsymbol{D}$ qui M-supportent $\boldsymbol{X}$.
Considérons les données du tableau 2.5, prenons par exemple les deux itemsets $\{\boldsymbol{Mexique}, \boldsymbol{Canada}\}$ et $\{\boldsymbol{Canada}\}$.

$\mathbf{S_D^{max}(X) = M - supporte\{Mexique, Canada\} = 3}$. Dans l'ensemble de transactions D, l'itemset $\{\mathbf{Mexique}, \mathbf{Canada}\}$ apparait seul dans la catégorie $\mathbf{t_1 =}$ **pays**. Dans l'exemple ci-dessus, l'itemset $\{\mathbf{Mexique}, \mathbf{Canada}\}$ se trouve dans les transactions 8, 9 et 10.

Quant à $\mathbf{S_D^{max}(X) = M - supporte\{Canada\} = 4}$, on trouve l'itemset $\{\mathbf{Canada}\}$ dans les transactions 3, 4, 5 et 6 seul dans la catégorie $\mathbf{t_1 =}$ **pays** de l'ensemble de transactions D.

2.5.3. M-Support d'une règle d'association $X \xrightarrow{max} Y$

Le support maximal ou M-support d'une règle d'association maximale $\mathbf{S_D^{max}\left(X \xrightarrow{max} Y\right)}$ est le nombre de transactions dans **D** qui M-supporte **X** et qui supporte aussi **Y**.

$\mathbf{S_D^{max}\left(X \xrightarrow{max} Y\right)}$ est défini comme suit :

$$S_D^{max}\left(X \xrightarrow{max} Y\right) = |\{t: t\ M - support\ X\ et\ t\ support\ Y\}|$$

Prenons en considération les données du tableau 2.5, la règle M-association $\{\mathbf{Mexique, Canada}\} \xrightarrow{\mathbf{max}} \{\mathbf{Croissance}\}$.

$\mathbf{S_D(Y) = S_D(\{Croissance\}) = 3}$. On trouve Y dans 3 transactions 3, 6 et 10. Contrairement au M-support, on ne prend pas en considération le fait que Y apparait seul dans sa catégorie, qui est $\mathbf{t_2 = sujets}$.

$\mathbf{S_D^{max}(\{Mexique, Canada\}) = 3}$. On trouve l'itemset $\{\mathbf{Mexique, Canada}\}$ seul dans sa catégorie $\mathbf{t_1 = pays}$ dans les transactions 8, 9 et 10.

On définit le M-Support d'une règle d'association maximale par le nombre de transactions qui M-Supporte l'itemset $\{\mathbf{Mexique, Canada}\}$.

Donc, $\mathbf{S_D^{max}\left(\{Mexique, Canada\} \xrightarrow{max} \{Croissance\}\right) = 1}$, puisque seule la transaction 10 qui M-supporte l'itemset $\{\mathbf{Mexique, Canada}\}$ et supporte en même temps l'itemset $\{\mathbf{Croissance}\}$.

2.5.4. M-Confiance d'une règle d'association maximale

La confiance maximale représente le pourcentage de transactions qui sont conformes à la définition de la règle d'association maximale par rapport au nombre de transactions contenant au moins un élément de **T(Y)**, la M-Confiance de la règle est définie comme suit :

$$C_D^{max}\left(X \xrightarrow{max} Y\right) = \frac{S_D^{max}(X \xrightarrow{max} Y)}{|D(X, T(Y))|}$$

En se référant à notre exemple du tableau 2.5, la règle M-association $\{\mathbf{Mexique, Canada}\} \overset{\max}{\longrightarrow} \{\mathbf{Croissance}\}$.

Afin de mesurer la M-confiance de cette règle, on doit tout d'abord calculer la valeur de $\mathbf{D(X, T(Y))}$.

Les transactions qui M-supportent $\{\mathbf{Mexique, Canada}\}$ sont : 8, 9 et 10. Ces trois transactions contiennent au moins un élément de $\mathbf{T(Y)} = \{\mathbf{Croissance}\}$, ce qui résulte :

$$\boldsymbol{D\big(\{Mexique, Canada\}, T(\{Croissance\})\big) = 3}$$

$$\boldsymbol{C_D^{max}\left(X \overset{max}{\longrightarrow} Y\right) = \frac{S_D^{max}(\{Mexique,Canada\} \overset{max}{\longrightarrow} \{Croissance\})}{|D(\{Mexique,Canada\},T(\{Croissance\})|} = \frac{1}{3} = 0,33 = 33\%.}$$

2.5.5. **Avantages et inconvénients :**

Les règles d'association maximales représentent plusieurs avantages et inconvénients.

2.5.5.1. ***Les avantages***

1. La capacité d'extraire un ensemble maximal qui peut apparaitre seul ou maximal dans un ensemble de transactions.
2. Améliorer la qualité d'extraction des règles d'association.

2.5.5.2. ***Les limites***

3. La complexité computationnelle se manifeste dans le temps énorme consacré à l'extraction des règles maximales et l'explosion combinatoire.

2.6. Les règles d'association séquentielles

Une règle séquentielle résulte d'une règle d'association à laquelle on fait joindre une estampille temporelle. La complexité du processus de recherche des règles séquentielles exige différentes étapes. Dans la suite de cette section, on va présenter les notions de séquence, séquence fréquente et motif séquentiel.

2.6.1. **Notions et définitions :**

2.6.1.1. ***Séquence :***

Une séquence est une liste ordonnée, bien déterminée d'itemsets non-vides. Un ou plusieurs itemsets sont nommés éléments de séquence et utilisent le principe de précédence, c'est-à-dire, chaque élément de la liste est précédé par des éléments de précédentes transactions d'un client donné.

Le tableau 2.6 illustre un extrait d'un ensemble de transactions séquentielles. Le concept de séquence se manifeste dans le fait qu'un client s'acquiert d'un ensemble de produits et, par la suite, il procède à l'achat d'un ou plusieurs produits qu'il juge complémentaires.

Prenant par exemple le Client _ 1 qui dans la date du 13-07-2019 s'est procuré une tablette et des écouteurs, juste après quelques jours, il a ressenti la nécessité d'avoir un étui de protection pour sa tablette.

Clients	*13-07-2019*	*16-07-2019*
Client _ 1	Tablette, écouteurs	Étui de protection
Client _ 2	Téléviseur, Console de jeu	CD de jeux

Tableau 2.6 - Extrait d'un ensemble de transactions séquentielles

2.6.1.2. ***Fréquence d'une séquence***

On dit qu'une séquence est fréquente, si et seulement si le support de cette séquence est supérieur ou égal au support minimum, qui est fixé par l'utilisateur pour bien mesurer la pertinence d'une séquence.

2.6.1.3. ***Séquences fréquentes ou motifs séquentiels :***

Après la validation des séquences fréquentes, on passe à la recherche de celles qui ne sont pas incluses dans aucune autre séquence, c'est-à-dire celles qui ont une fréquence maximale qui seront appelées des motifs séquentiels. Ces motifs sont considérés comme étant une extension de la notion de règles d'associations intégrant diverses contraintes temporelles. L'extraction des motifs séquentiels est couramment associée à un intervalle de temps bien spécifié. Les règles d'association séquentielles mettent en évidence des associations inter-transactions à l'encontre des règles d'association

standards qui extraient des associations intra-transactions. Dans ce contexte, l'identification des objets est inévitable pour la poursuite de leurs comportements au fil du temps.

2.6.2. **Extraction des motifs séquentiels :**

L'extraction des motifs séquentiels est une tâche difficile vu que l'ordre des éléments doit être conservé dans une séquence, ce qui exige plus d'efforts computationnels. La contrainte de temporalité chronologique rend cette tâche plus précise dans les résultats, mais implique aussi une plus grande difficulté d'implémentation. La recherche de motifs séquentiels se base sur un format de données bien précis, qui relate des événements liés à différents acteurs. L'exemple du supermarché est le plus utilisé pour illustrer ce concept. Cela est, en partie, dû au fait que ce sont des besoins de type marketing qui ont apporté cette problématique. Ce format de données est illustré par le Tableau 2.7.

Le Format de données utilisé ci-dessous, permet de distinguer les achats de trois clients. Le but d'un algorithme d'extraction de motifs séquentiels sera alors de trouver des comportements fréquents dans les achats des clients. Le résultat attendu est donc une ou plusieurs séquences d'achats, représentant un comportement récurrent. Nous pouvons alors définir le comportement fréquent comme étant un comportement respecté par au moins n clients (n étant considéré comme un minimum de clients qui respectent ce comportement afin que celui-ci soit estimé fréquent).

Clients	20-06-2017	21-06-2017
Client_1	Tv	Support mural
Client_2	Pc, Clé USB	Anti-Virus
Client_3	Lait, Œuf, Farine	Huile

Tableau 2.7 - Format de données utilisé

2.6.3. **Propriétés des séquences fréquentes**

Il existe plusieurs propriétés principales qui sont des éléments déterminants pour l'extraction. Il faut noter que toutes ces propriétés sont déjà appliquées sur les règles d'association. Dans cette section, on ne détaillera que le support d'une séquence et la confiance d'une règle séquentielle.

2.6.3.1. *Support d'une séquence :*

Le support d'une séquence S est représenté par le pourcentage de clients qui la supportent.

$$S(\{ae\} \rightarrow \{bc\}) = S(\{ae\} \cup \{bc\})$$

Client	15-08-2017	30-08-2017
Client _ 1	Laptop, souris	Sac à dos
Client _ 2	Pc, USB	Anti-virus
Client _ 3	Pc, pochette, USB	Anti-virus, Lave écran
Client _ 4	Souris, Pc, USB	Tablette, Anti-virus
Client _ 5	Pc, USB	Souris

Tableau 2.8 - Ensemble de transactions

En considérant les données du tableau 2.8, on peut procéder au calcul du support de la règle $\{\mathbf{Pc,\ USB}\} \rightarrow \{\mathbf{Anti-virus}\}$ comme suit :

$$S(\{\mathbf{Pc,\ USB}\} \rightarrow \{\mathbf{Anti-virus}\}) = S(\{\mathbf{Pc,\ USB}\} \cup \{\mathbf{Anti-virus}\}) = {}^{3}/_{5} = 60\%$$

2.6.3.2. **Confiance d'une règle :**

La confiance d'une règle est une mesure de précision, c'est la probabilité qu'on achète un certain nombre d'articles A sachant qu'on a déjà acheté B, soit la probabilité conditionnelle p(A/B).

$$C(\{ae\} \rightarrow \{bc\}) = S(\{abce\})/S(\{ae\})$$

En se basant sur l'exemple ci-dessus, la confiance de $\{\mathbf{Pc,\ USB}\} \rightarrow \{\mathbf{Anti-virus}\}$ se traduit comme suit :

$$C(\{\boldsymbol{Pc,\ USB}\} \rightarrow \{\boldsymbol{Anti-virus}\}) = \frac{S(\{Pc,USB,Anti-virus\})}{S(\{Pc,USB\})} = {}^{3}/_{4} = 75\%$$

2.7. Conclusion

Dans le but de trouver de l'information cachée dans de grandes bases de données, on fait recours aux plusieurs méthodes d'extraction de règles d'association. L'importance

de ces outils réside dans la simplicité des résultats obtenus et le côté pratique qui est une étape primordiale dans la prise de décision.

Dans ce chapitre on a présenté les différentes approches d'extraction des règles d'association, à savoir les règles d'association standards, maximales et séquentielles. Nous avons aussi décrit leurs fonctionnements ainsi que leurs propriétés et leurs opérations, tout en mettant en œuvre des exemples.

Quant au prochain chapitre, nous détaillerons les différentes méthodes de classification.

CHAPITRE 3 CLASSIFICATION

Ce chapitre présente les méthodes de classification, où la première partie expose la classification hiérarchique avec ses propriétés, tandis que la deuxième introduit la classification non hiérarchique dite de partitionnement.

3.1. Introduction

La classification automatique est un ensemble de méthodes conçu pour analyser des jeux de données que ce soit dans le cadre scientifique ou d'analyses stratégiques de la recherche. Ces méthodes s'appuient sur la combinaison de techniques dédiée à la classification supervisée ou non- supervisée.

On peut définir une méthode comme étant une marche rationnelle qui permet d'atteindre la connaissance d'un but. Vu l'intérêt de la classification, nombreux sont les chercheurs qui ont consacré énormément d'efforts et de savoir-faire pour mettre à jour différentes méthodes et les rendre disponibles aux utilisateurs. On peut distinguer plusieurs méthodes de classification à savoir :

- Les méthodes hiérarchiques (ascendantes ou descendantes).
- Les méthodes de partitionnement (K-means, K- médoïdes ...).
- Les méthodes neuronales (PMC, SOM, MULTISOM...).

C'est en optimisant un critère visant à regrouper les individus dans des classes les plus homogènes possibles, et, entre elles les plus distinctes possibles, que résulte une classification non-supervisée. En outre, la classification permet de mettre en évidence ces regroupements sans connaissance sur les données traitées.

La classification non-supervisée est un ensemble de méthodes ayant pour objectif la recherche d'une typologie ou segmentation existante, c'est-à-dire une partition ou répartition des individus en classes homogènes ou catégories.

Dans ce mémoire, nous faisons appel à la classification non-supervisée sur des données textuelles afin de pouvoir regrouper des objets ayant des similitudes en des groupes homogènes. À cet effet, on a eu recours aux méthodes de partitionnement (K-médoïdes) et aux méthodes hiérarchiques ascendantes (agglomératives) ou descendantes (divisives), tout en utilisant un environnement de développement gratuit RStudio (Présentation du RStudio, 2019).

Ces méthodes bien qu'elles soient différentes, se basent sur la notion de distance (dissimilarité ou similarité) afin de calculer la ressemblance entre 2 objets. Ces derniers peuvent se présenter sous différentes formes (vecteurs, neurones, ...) dépendamment des stratégies et des résultats voulus.

3.2. Représentation vectorielle

Un espace vectoriel est un ensemble de vecteurs dans un espace de p dimensions. Chaque vecteur représente un objet pour lequel on attribue une valeur de vérité (**VRAI, FAUX**). Faux (0) correspondant à l'absence de l'élément et Vrai (1) à sa présence. (Voir le tableau 3.1).

	Variable 1	**Variable 2**	**...**	**Variable p**
Objet 1	0	1	...	1
Objet 2	1	1	...	0
...	...	...	...	...
Objet n	1	0	...	1

Tableau 3.1- Représentation des données

3.3. Mesures de similarité et de dissimilarité

La notion de ressemblance a été très tôt perçue comme un concept clé en intelligence artificielle. Cette notion intervient dans plusieurs domaines comme la recherche d'informations, le raisonnement, l'apprentissage automatique. Elle est notamment utilisée dans l'analyse statistique de données, les sciences cognitives ou la psychométrie.

Afin de définir l'homogénéité d'un groupe d'objets, il est nécessaire de mesurer leurs ressemblances. En particulier, on peut distinguer deux types de données : celles qui sont dichotomiques ou binaires et celles qui sont quantitatives.

Dans ce mémoire, les données sont représentées sous forme dichotomique afin de calculer la ressemblance entre les objets. Et pour cela, il existe plusieurs mesures de ressemblance. Nous nous attarderons sur trois d'entre elles : similarité, dissimilarité et distance.

3.3.1. **Mesure de similarité**

Une mesure de similarité s entre deux objets x et y d'un ensemble O est une application $O \times O$ dans [0,1] qui satisfait les propriétés suivantes :

$$S(x,y) = S(y,x) \geq 0$$

$$S(x,x) = S(y,y) \geq S(x,y)$$

Ainsi, plus x et y se ressemblent, plus leur similarité est élevée.

3.3.1.1. ***Indice de similarité de Faith***

La première mesure que nous présentons est l'indice de Faith, qui a été proposé par Baroni-Urbani et Buser en 1976 (Baroni-Urbani et Buser 1976). Cette similarité pour les données binaires est supposée être comprise entre 0 (aucune similarité) et 1 (similarité complète).

L'indice de Faith se calcule comme suit :

$$\frac{\left(a + {}^{d}/_{2}\right)}{a+b+c+d} = S_F(x,y)$$

Où :

- a : paire de (VRAI, VRAI)
- b : paire de (FAUX, VRAI)
- c : paire de (VRAI, FAUX)
- d : paire de (FAUX, FAUX)

x \ y	1	0
1	a	c
0	b	d

3.3.1.2. ***Indice de similarité de Jaccard***

L'indice de Jaccard (Jaccard, 1901), permet de mesurer les similitudes entre les ensembles. Il est défini par la taille de l'intersection notée $|A \cap B|$ divisée par la taille de l'union $|A \cup B|$, pour deux classes quelconques A et B.

Cet indice se calcule comme suit :

$$S_J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Le cas particulier pour les objets x et y s'écrit :

$$S_J(x,y) = \frac{a}{a+b+c}$$

Exemple :

Soit deux objets x et y tels que (voir Tableau 3.2) :

- $x = (0, 1, 1, 0, 1)^T$
- $y = (1, 0, 0, 1, 1)^T$

x	0	1	1	0	1
y	1	0	0	1	1

Tableau 3.2 - Représentation des objets x et y

On en déduit que l'intersection contient un élément (a = 1) sur un total de cinq et que $b = 2$ et $c = 2$ (voir Figure 6) Donc : $S_J(x, y) = {}^1/_5 = 0.2$

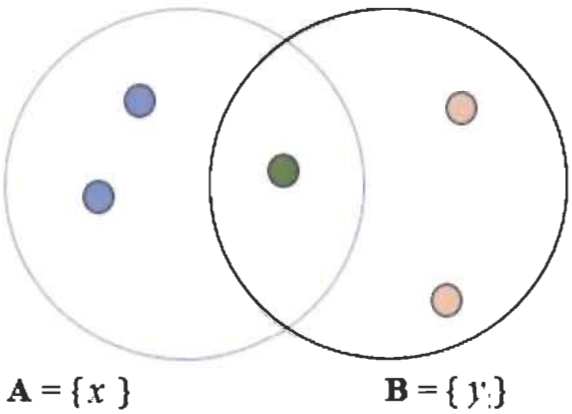

Figure 6 - Représentation graphique de l'indice de similarité de Jaccard

3.3.2. **Mesures de dissimilarité et de distance**

Une mesure de dissimilarité d entre deux objets x et y d'un ensemble O est une application de $O \times O$ dans $\mathbb{R}^+$ qui satisfait les propriétés suivantes :

a) $d(x, y) = d(y, x) \geq 0$

b) $d(x, y) = 0$ **si et seulement si :** $x = y$

Si de plus, on a c) pour $(x, y, z) \in O \times O \times O$, $d(x, y) \leq d(x, z) + d(z, y)$ (inégalité triangulaire) alors d définit une mesure de distance.

Contrairement à la similarité, moins x et y se ressemblent, plus leur dissimilarité ou leur distance s'élève.

3.3.2.1. *Distance euclidienne*

Soit :

$$x = (x_1, \dots, x_n) \in \mathbb{R}^n \qquad \text{et} \quad y = (y_1, \dots, y_n) \in \mathbb{R}^n$$

La distance euclidienne entre x et y est définie par :

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

3.3.2.2. *Distance Manhattan*

Soit :

$$x = (x_1, \dots, x_n) \in \mathbb{R}^n \qquad \text{et} \qquad y = (y_1, \dots, y_n) \in \mathbb{R}^n$$

La distance Manhattan est définie par :

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Dans ce mémoire, on a fait appel aux mesures de similarité pour extraire la ressemblance entre les objets.

3.4. Classification hiérarchique

Il existe de nombreuses méthodes visant à partitionner un ensemble de données en classes ou en sous-classes. La classification hiérarchique et la classification non hiérarchique sont deux d'entres elles. On s'intéresse dans cette section à la classification hiérarchique.

L'algorithme de classification hiérarchique est très intuitif avec une grande flexibilité puisqu'il permet de classifier les données de façon hiérarchique, ce qui facilite leurs interprétations.

3.4.1. **Choix d'un indice d'agrégation de classes**

Après avoir défini la distance (dissimilarité ou similarité) entre objets, on s'intéresse à la mesure de ressemblance entre classes d'objets.

3.4.1.1. *Méthode du plus proche voisin (saut minimal)*

L'agrégation D entre deux classes A et B est définie par les objets les plus proches appartenant à ces deux classes, tel qu'illustrée à la figure 7.

$$D(A,B) = \min\{d(x,y), x \in A, y \in B\}$$
$$= \max\{s(x,y), x \in A, y \in B\}$$

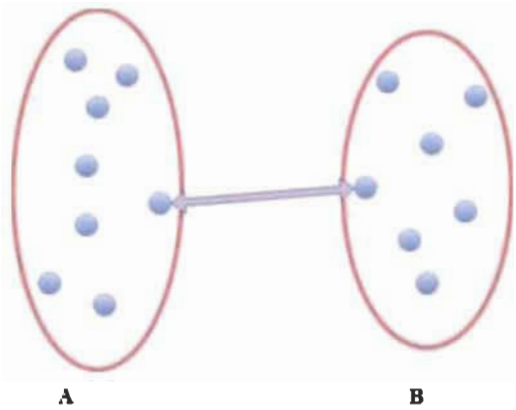

Figure 7 - Saut minimal

3.4.1.2. *Méthode du voisin le plus éloigné (saut maximal)*

L'agrégation D entre deux classes A et B est définie par les deux objets les plus éloignés, tel qu'illustrée à la figure 8.

$$D(A,B) = \max\{d(x,y), x \in A, y \in B\}$$
$$= \min\{s(x,y), x \in A, y \in B\}$$

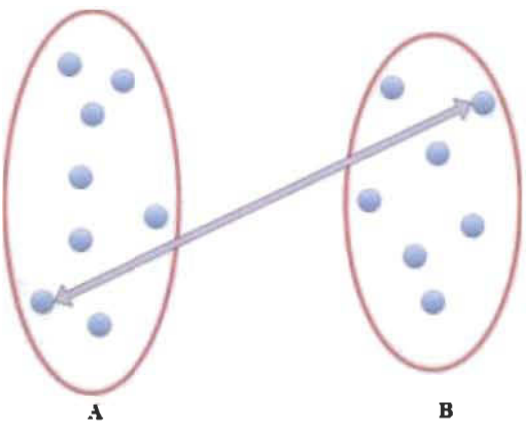

Figure 8 – Saut maximal

3.4.1.3. *Méthode du saut moyen*

L'agrégation D entre deux classes A et B est définie par la distance moyenne entre les objets de A et B, tel qu'illustrée à la figure 9.

$$D(A,B) = \frac{\sum_{x\in A}\sum_{y\in B} d(x,y)}{n_A\, n_B}$$

Où :

- n_A = nombre d'éléments ou effectif de A.
- n_B = nombre d'éléments ou effectif de B.

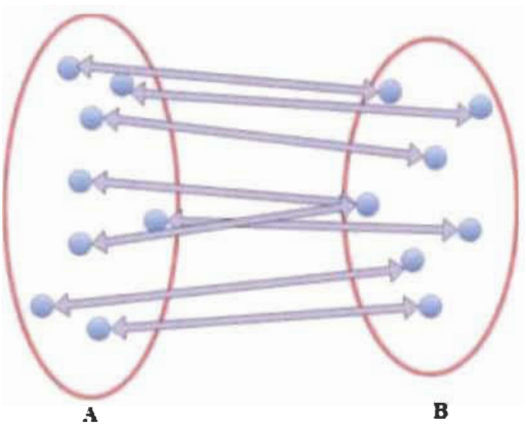

Figure 9 - Saut moyen

3.4.1.4. *Méthode Ward :*

Cette méthode est basée sur la perte d'inertie expliquée résultant de l'agrégation des classes A et B.

$$D(A,B) = \frac{n_A\, n_B}{n_A + n_B}\, d^2(g_A, g_B)$$

Où :

- g_A = Centre de gravité de la classe A.
- g_B = Centre de gravité de la classe B.

3.4.2. **Algorithme de Classification Ascendante Hiérarchique**

Dans le cas agglomératif de « Classification Ascendante Hiérarchique » (CAH), dans sa version de base, on commence par considérer que chaque objet est une classe. On cherche ensuite les deux classes les plus proches et on les réunit en une seule classe.

On recommence cette procédure jusqu'à l'obtention d'une seule classe qui contient tous les objets.

Dans la suite de ce mémoire, on ne s'intéresse qu'à la CAH avec la méthode de Ward. Ainsi, on se basera sur l'exemple classique « Iris de Fisher » qui représente des données de référence pour la classification supervisée et non supervisée. Proposées par le statisticien Ronald Aylmer Fisher en 1936 (Fisher, 1936), ces données comprennent 50 échantillons de chacune des trois espèces d'iris (Setosa, Versicolor et Virginica). Quatre caractéristiques ont été mesurées à partir de chaque échantillon : la longueur et la largeur des sépales et des pétales, en centimètres. Ainsi, les iris de 1 à 50 sont des « Setosa », de 51 à 100 des « Versicolor » et de 101 à 150 des « Virginica ».
La figure 10 donne une visualisation des espèces sur les 2 premiers axes factoriels obtenus à la suite d'une analyse en composantes principales (ACP).

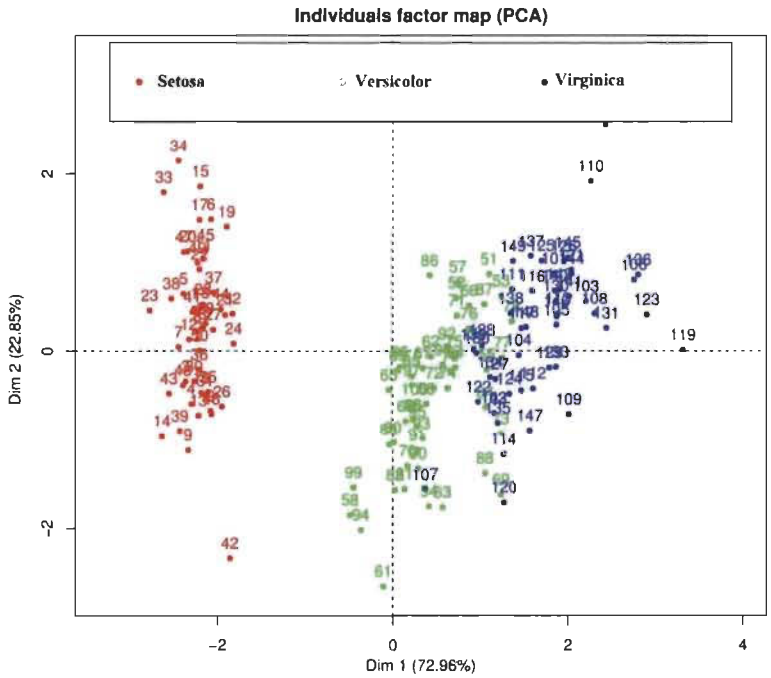

Figure 10 - Visualisation des espèces sur les 2 premiers axes factoriels

3.4.3. **Arbre de classification**

En CAH, l'arbre de classification dit dendrogramme résulte de la fusion pas à pas des objets à classifier. La longueur des branches représente la distance entre les objets ou

classes d'objets qu'elle relie. En coupant le dendrogramme à un certain niveau, qui correspond à une certaine distance, on peut voir le nombre de classes correspondants. La figure 11 représente un dendrogramme avec une coupure en trois classes d'un échantillent de 40 iris choisis aléatoirement.

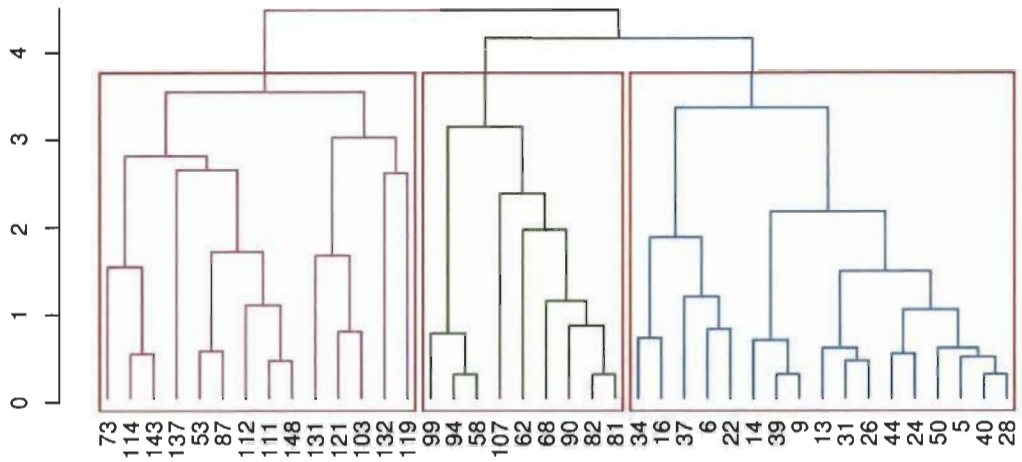

Figure 11 - Exemple de dendrogramme avec une coupure en trois classes

3.5. Classification non hiérarchique ou de partitionnement

Les méthodes de classification non hiérarchique cherchent à construire directement des partitions avec un nombre de classes fixe a priori. On s'intéresse plus particulièrement à la méthode des K-means et celle des K-médoïdes.

3.5.1. **Méthode des K-means**

K-means est une méthode qui a été développée par MacQueen en 1967 (MacQueen, 1967). Elle vise à partitionner un ensemble de données en K classes homogènes, K est le nombre de classes voulue ou fixé a priori.

3.5.1.1. ***Algorithme basique des K-means***

Dans sa version de base, l'algorithme des K-means s'énonce comme suit :

- **Entrée** : K le nombre de classes voulues ou fixé a priori
- **Début** :
  - Choisir aléatoirement les centres de classes
  - **Répéter**
    - Affecter chaque objet à la classe dont le centre est le plus proche
    - Recalculer le centre de chaque classe

- **Jusqu'à** (convergence)
- **Sortie :** Une partition des individus en K classes

Étant donné qu'elle est itérative, cette méthode converge vers une solution quel que soit son point de départ. Cependant, la partition finale obtenue dépend de la partition initiale.

3.5.2. **Méthode des K-médoïdes**

K-médoïdes ou le partitionnement autour des médoïdes est un algorithme de classification non hiérarchique qui est légèrement modifié par rapport à l'algorithme des K-means. En fait, c'est une variante des K-means. Il s'avère que le calcul des K-médoïdes est plus robuste au bruit que le calcul de K-means. (Jin et Han., 2011).
Le médoïde est l'élément le plus central de la classe, c'est à dire celui pour lequel la somme des distances aux autres éléments de la classe est la plus faible.
Nous obtenons ainsi l'algorithme des K-médoïdes, dans lequel le seul changement par rapport au K-means est le remplacement des centres de gravité par des médoïdes.

3.5.2.1. ***Algorithme des K-médoïdes :***

Dans sa version de base, l'algorithme des K-médoïdes s'énonce comme suit :

- **Entrée** : K le nombre de classes voulues ou K est fixé a priori.
- **Début** :
  - Choisir au hasard les médoïdes de classes
  - **Répéter**
    - Affecter chaque objet à la classe dont le médoïde est le plus proche
    - Recalculer les médoïdes de chaque classe à partir des objets regroupés
  - **Jusqu'à** (convergence)
- **Sortie :** Une partition des objets en K classes

3.6. Choix du nombre de classes

Le choix du nombre de classes n'est pas intuitif, spécialement quand il s'agit d'un jeu de données volumineux et qu'on n'a pas d'hypothèses sur les données.

3.6.1. **Choix d'un grand nombre de classes**

Un grand nombre de classes conduit à un partitionnement trop fragmenté de données, ce qui pourrait conduire à des classes peu ou pas pertinentes.

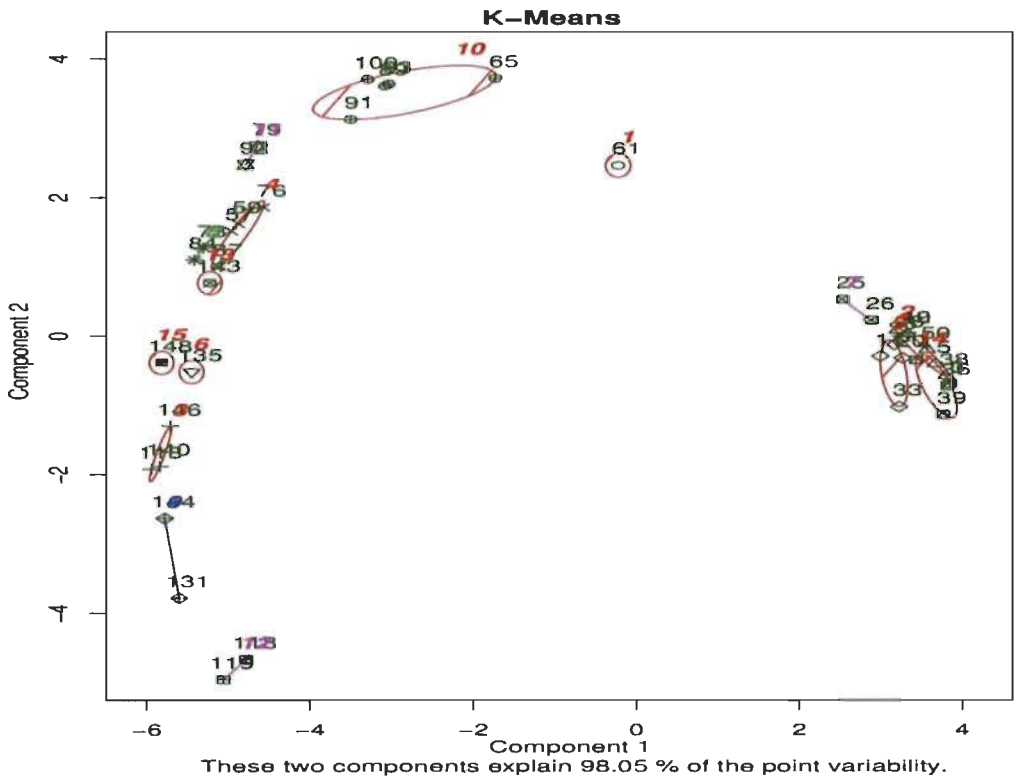

Figure 12 - Classification par les K-means des Iris de Fisher K = 15

La figure 12 illustre le résultat de la classification par les K-means sur échantillon prélevé au hasard de 40 objets des Iris de Fisher, avec un grand nombre de classes, K = 15.

On constate que dans ce cas, la méthode des K-means donne autant de classes demande, cependant, les classes qui contient plus d'un élément sont homogènes (appartiennent à la même espèce). Néanmoins, les classes sont non-équilibrés. Le tableau 3.3 détaille les résultats représentés dans la figure 12 en énumérant les objets de chaque classe. Pour rappel, les iris de 1 à 50 sont des « Setosa », de 51 à 100 des « Versicolor » et de 101 à 150 des « Virginica ».

Classes	Iris	Types
1	61	Versicolor
2	12, 40, 28, 38, 50, 5	Setosa
3	113, 140, 146	Virginica
4	87, 57, 59, 76	Versicolor
5	11, 33, 22, 20	Setosa
6	135	Virginica
7	25, 26	Setosa
8	73, 84	Versicolor
9	144, 131	Virginica
10	100, 89, 65, 91, 93	Versicolor
11	79, 92	Versicolor
12	119, 118	Virginica
13	143	Virginica
14	39, 3, 4, 36	Setosa
15	148	Virginica

Tableau 3.3 - Distribution des iris dans chaque classe K = 15

3.6.2. **Choix du petit nombre de K classes**

Un petit nombre de classes entrainera des classes générales contrairement au grand nombre de classes.

En guise d'illustration, on applique la méthode des K-means sur le même échantillon des Iris de Fisher, sauf que le nombre de classes est 2, donc K = 2, tel qu'illustre la figure 13.

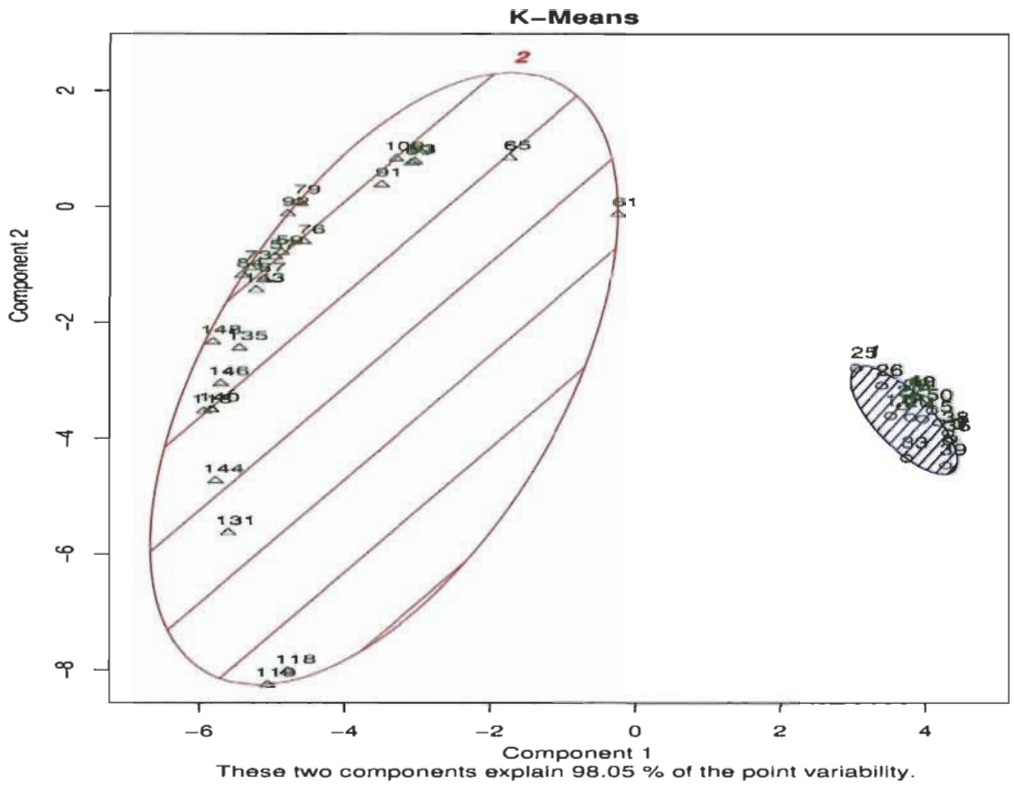

Figure 13 - Classification par les K-means des Iris de Fisher, K = 2

Lorsque K est petit, les résultats obtenus sont des classes de grande taille qui peuvent englober différents objets. Le tableau 3.4 détaille les résultats représentés dans la figure 13 en détaillant les objets de chaque classe.

Classes	Iris	Types
1	25, 39, 12, 11, 40, 28, 26, 3, 11, 4, 38, 50, 22, 36, 5, 20	Setosa
2	119, 144, 61, 113, 131, 100, 87, 143, 73, 140, 135, 89, 65, 57, 91, 79, 146, 59, 118, 84, 92, 76, 93, 148	Verginica, Versicolor

Tableau 3.4 - Distribution des iris dans chaque classe K = 2

3.6.3. **Choix optimal du nombre de classes**

Afin de choisir le nombre optimal K de classes, on fait appel aux méthodes les plus usuelles qui sont : « Eboulis » et « NbClust ».

3.6.3.1. *Méthode de l'éboulis*

C'est une méthode d'interprétation et de validation qui consiste à trouver le nombre approprié de classes dans un jeu de données.

Cette méthode prend les résultats des K-means avec différentes valeurs de K et calcule la variance des différentes classes. La variance est la somme des distances entre chaque centre d'une classe et les différents objets inclus dans la même classe.

À cet effet, la méthode de l'éboulis cherche à trouver un nombre K de classes de telle sorte que les classes retenues minimisent la distance entre leurs centres et les objets dans la même classe. Autant dire, la méthode minimise la variance intra-classes qui s'exprime comme suit :

$$V = \sum_{j=1}^{K} \sum_{x_i \in C_j} D^2(g_j, x_i)$$

Où :

- g_j : le centre de la classe C_j (le centroïde ou le centre de gravité)
- x_i : le $i^{ème}$ objet dans la classe ayant pour centre g_j.
- $D(g_j, x_i)$: la distance entre le centre de la classe et le point x_i.

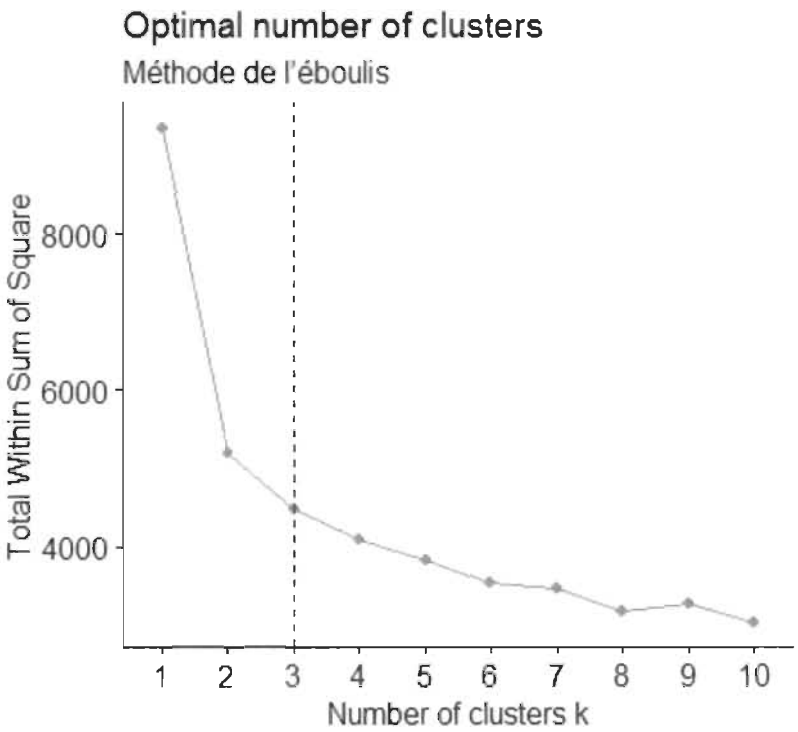

Figure 14 - Méthode de l'éboulis pour le nombre optimal K classes d'Iris de Fisher (150 espèces)

On remarque sur la figure 14 que le nombre optimal de classes est le point représentant le coude. Dans notre cas, le coude peut être représenté par K valant de 2 ou 3.

Généralement, le point du coude est représenté par le dernier point de chute significative de la courbe de variance. En effet, la chute entre 1 et 3 classes est significativement plus grande que celle entre 4 et 10 clusters.

Finalement, on constate que le choix se fera en fonction de la nature du jeu de données. Notre jeu de données contient 3 espèces d'iris (setosa, virginica et versicolor), ce que la méthode d'éboulis nous valide en proposant K = 3.

3.6.3.2. *Package NbClust*

NbClust est un package du logiciel R (Charrad et al., 2014). Ce dernier a été développé pour s'attaquer à des problèmes difficiles, tels que la qualité des classes, le degré d'adaptation du processus de classification à un ensemble de données spécifique et le nombre optimal de classes.

Il fournit une trentaine de critères d'arrêt qui détermine le nombre optimal de classes dans un ensemble de données. En outre, il offre une fonction qui effectue une classification hiérarchique et celle des K-means avec différentes mesures de ressemblance (similarité, dissimilarité), et différentes méthodes d'agrégation des classes.

Toute combinaison d'indices de validation et de méthodes de classification peut être demandée en faisant appel à une seule fonction NbClust, cela permet à l'utilisateur d'évaluer simultanément plusieurs méthodes de classification tout en faisant varier le nombre de classes et ainsi détermine le nombre optimal de classes tel qu'illustré la figure 15.

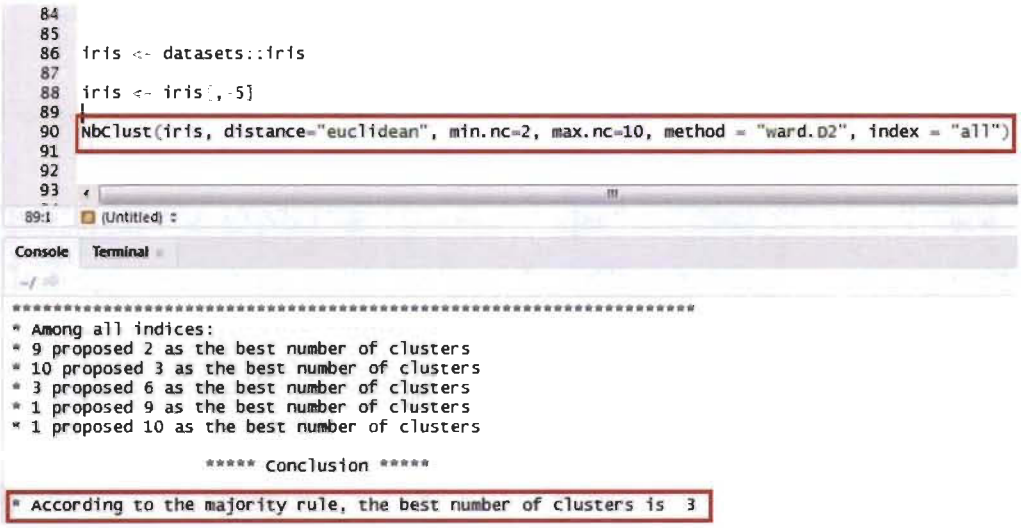

Figure 15 - Résultat proposé par le package NbClust d'Iris de Fisher (150 espèces)

On constate que NbClust propose également K = 3, en accord avec la majorité des critères à l'étude.

CHAPITRE 4 MÉTHODOLOGIE

4.1. Introduction

Le principal objectif de ce mémoire est d'explorer les règles d'association afin d'établir de nouveaux descripteurs du contenu d'un texte. Les résultats obtenus seront expérimentés tout en appliquant plusieurs classifieurs.

L'approche proposée s'inscrit dans la continuité des travaux de (Hilali, 2009 ; Descôteaux, 2014 ; Labiad, 2017). Elle se diffère toutefois par l'algorithme substantiel qui permet d'extraire des itemsets fréquents. De plus, nous classifions ces derniers afin de pouvoir regrouper ceux ayant des similitudes en des classes homogènes. L'hypothèse derrière notre approche est que lorsque les itemsets co-occurent fréquemment au sein d'un texte, alors on dit que ces derniers le représentent d'une façon plus pertinente. D'autant plus, il est possible de dégager les thèmes spécifiques traités dans ce document tout en considérant les itemsets fréquents.

Dans ce chapitre, nous allons voir en premier lieu les étapes de l'architecture globale de notre approche qu'on illustre dans la figure 16, ainsi que leurs différentes fonctionnalités. Par la suite, nous fournissons un exemple simple pour illustrer chacune de ces étapes en détail.

Le schéma non exhaustif ci-dessous interprète toutes les étapes essentielles de notre approche. D'abord, on commence par lire un texte brut comme entrée. Par la suite, on procède à sa segmentation, puis, on prépare nos segments tout en supprimant les éléments jugés non-pertinents pour la classification. L'étape d'extraction quant à elle vient par la suite afin d'extraire les itemsets fréquents pour finalement les classifier et les interpréter graphiquement.

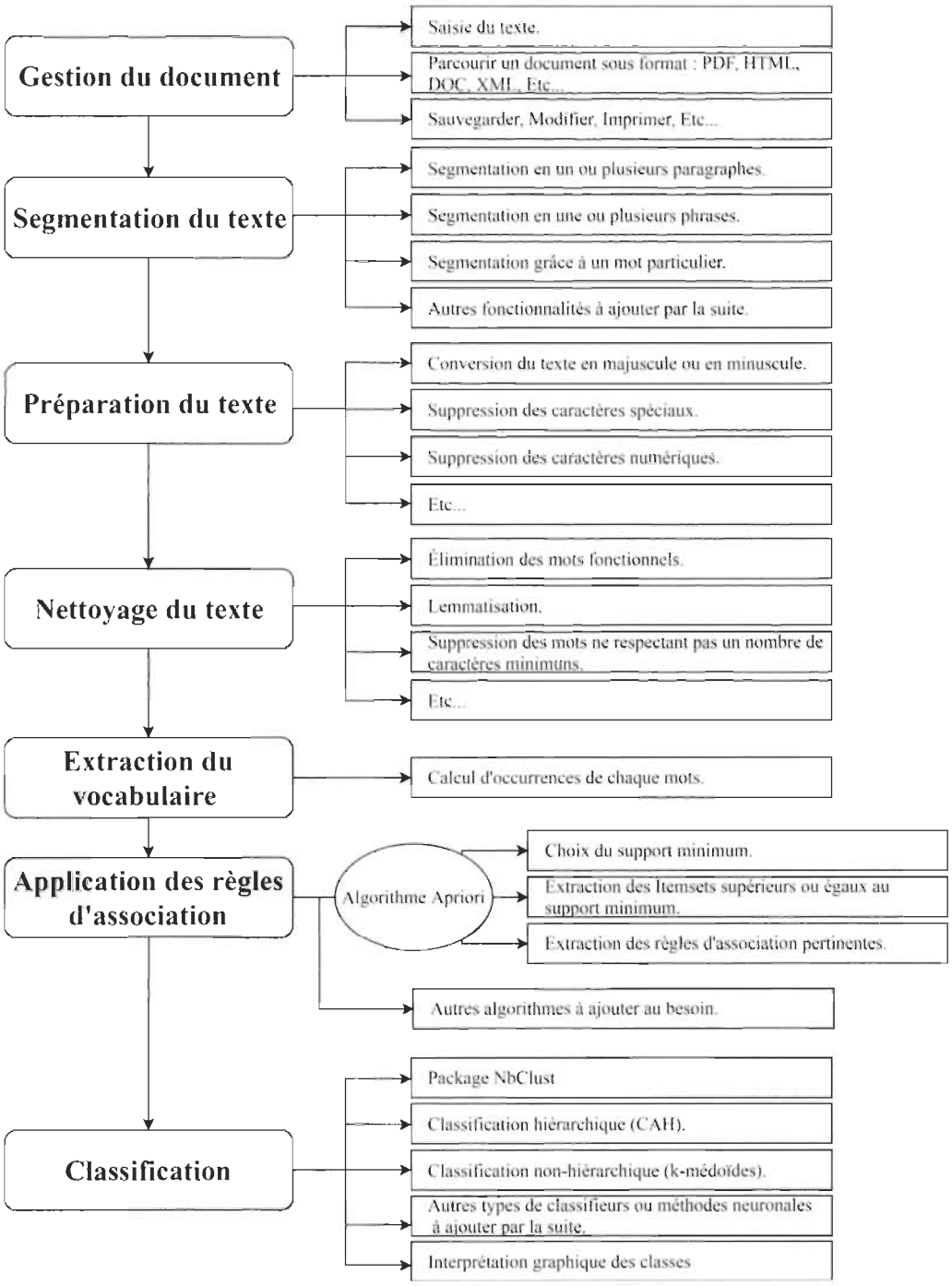

Figure 16 - Schéma global de notre méthodologie

4.2. Présentation du schéma

4.2.1. Gestion du document

Cette phase consiste à lire ou à saisir un ou plusieurs textes à analyser au moyen de l'application. Nous donnons la possibilité à l'utilisateur de saisir le texte, de lire des documents dans différents formats (PDF, HTML, DOC, XML, etc.) et de procéder en tout temps à l'édition manuelle du texte. En outre, l'utilisateur peut sauvegarder, ouvrir, imprimer, etc.

Dans le cas d'un document comportant des balises (PDF, HTML, etc.), notre plateforme procèdera à les éliminer tout en ne gardant que le texte brut.

4.2.2. Segmentation

La segmentation est un processus qui permet le découpage du texte en représentation vectorielle.
Nous donnons la possibilité à l'utilisateur de choisir le type de la segmentation ainsi que le nombre d'éléments contenu dans chacun des segments relevés. Ces choix se déclinent comme suit :

- La segmentation en paragraphes.
- La segmentation en phrases.
- La segmentation basée sur un découpage utilisant un délimiteur donné.
- Autres fonctionnalités à ajouter par la suite.

4.2.3. Préparation du texte

Le texte brut comporte généralement des caractères qui peuvent dans certains cas ou en fonction de certains objectifs d'analyses se révéler non significatifs, telle la ponctuation, les caractères spéciaux ou numériques, etc... . Cette étape qui ne nécessite pas de base de connaissances préalablement a pour but de réduire la quantité d'informations à explorer et à éliminer ces caractères. Différentes fonctionnalités sont offertes. On en cite :

- Conversion du texte en majuscule ou en minuscule.
- Suppression des caractères spéciaux tels les symboles monétaires ($, €, £, ¥), la ponctuation (; : / \ |) etc.

- Suppression des caractères numériques.
- Etc.

4.2.4. **Nettoyage du texte**

Selon les objectifs de l'utilisateur, il est possible d'avoir recours à la réduction de la taille du texte en effectuant un certain nombre d'opérations qui élimineraient les unités lexicales jugées peu porteuses de significations pour des besoins d'extraction des itemsets.

La première fonctionnalité disponible dans notre plateforme permet la suppression des mots fonctionnels. Pour rappel, les mots fonctionnels sont généralement représentés par les déterminants, les conjonctions, les pronoms, etc..., ils sont souvent jugés non sémantiquement significatifs. Ils sont par ailleurs très fréquents et ralentissent l'analyse. Nous montrons dans le tableau 4.1 un extrait de la liste de mots fonctionnels dont nous disposons.

Mots fonctionnels
Le, la, les
De, dans, des
Avec, ainsi, pour
Au-dessus, au-dessous, au
Mais, par contre, cependant
Telle, cela, certain

Tableau 4.1 - Extrait de la liste des mots fonctionnels

La deuxième fonctionnalité consiste à remplacer les formes fléchies des unités lexicales par leurs formes canoniques. La lemmatisation est une étape importante à la réduction du lexique. Le tableau 4.2 illustre un extrait de la base de données utilisée pour des fins de lemmatisation.

Forme canonique	Forme fléchie
Être	Est, sont, fut, suis, sera
Analyser	Analysé, analysez, analyseront
Traiter	Traitâmes, traité, traiteront, traitaient
Aller	Vais, irez, allaient, vont
Parcourir	Parcours, parcourons, parcouraient
Ami	Amie, amis, amies

Tableau 4.2 - Extrait de la base de données de lemmatisation

4.2.5. **Extraction du vocabulaire**

Le processus d'extraction du vocabulaire consiste à construire une liste de mots sans redondance à partir des segments obtenus lors de l'étape de segmentation. De cette étape résulte une matrice à deux dimensions où les colonnes représentent les différents descripteurs uniques du texte et les lignes représentent les segments.

4.2.6. **Les règles d'association**

Cette étape vise à extraire les itemsets fréquents en faisant appel à différents algorithmes. Dans ce mémoire, nous utilisons une variante de l'algorithme Apriori qu'on nomme TM_Apriori. Notre choix est motivé par la volonté de dégager un nombre restreint d'itemsets fréquents.

TM_Apriori a la particularité de :

- **Décrémenter le seuil minimum** : Lors de la première itération, le support minimum est fixé à une valeur choisie par l'utilisateur. Lorsque le nombre d'itemsets fréquents extraits est inférieur à 10, alors le support minimum est diminué de 0.1.
- **Choisir un nombre d'items dans un itemset** : L'utilisateur peut fixer le nombre maximal des items dans un itemset.
- **Superset** : Un superset est un itemset défini par rapport à un autre itemset. Par exemple {a, b, c} est un superset de l'itemset {a, b}. Cette notion est utilisée afin d'éviter les redondances dans les itemsets fréquents.

La recherche de ces itemsets est effectuée de manière itérative. Le processus cesse lorsque le nombre des items dans les itemsets obtenus est supérieur ou égal à une valeur spécifiée par l'utilisateur ou quand le support minimum est inférieur à 0.1.

4.2.6.1. *Présentation de l'algorithme TM_Apriori*

Les notions utilisées dans le pseudo code sont présentées dans le tableau 4.3.

Données	Types	E/S	Assignation
D	Matrice d'entiers	Entrée	Base de connaissance
Min_Sup	Entier	Entrée	Seuil minimal de support
NB_Items	Entier	Entrée	Nombre maximal d'items dans un itemset
F_I	Matrice binaire	Sortie	Itemsets fréquents
C_I	Matrice d'entiers	Sortie	Candidat d'itemsets généré
Support	Réel	Sortie	Support = (Fréquence(itemsets)/nombre de transaction) * 100
T	Vecteur d'entrées	Sortie	Ensemble de transaction
//	--	--	Le texte qui vient après « // » est un commentaire

Tableau 4.3 – Variables de l'algorithme

Algorithme 1 : TM_Apriori

Entrée : D, Min_Sup, NB_Items

Sortie : C_I, F_I, Support, T-transaction (t ⊂ T)

Début

// Initialisation du support minimum à 40% et le nombre des items dans l'itemset à 3

Min_Sup ⟵ 40%

NB_Items ⟵ 3

// ici on a une boucle qui permet de parcourir toutes les transactions dans notre base de connaissances

Pour Chaque transaction t ∈ D **Faire**

    // Cette condition permet d'éliminer les items inférieurs au support minimum

    Si Support.t.items ≥ Min_Sup **Alors**

        F_I ⟵ C_I ∪ F_I

    Sinon

        Supprimer C_I.t.itemsets

    FSi

// Cette fonction permet de générer les combinaisons possibles sans redondance des items supérieurs au support minimum

TM_Apriori_Comb(F_I)

FPour

// Cette condition vérifie si le nombre des itemsets générés est inférieur à 10, si oui on diminue le support minimum de 0.1

Si Taille(C_I) ≤ 10 **Alors**

Min_Sup ⟵ Min_Sup – 0.1

FSi

Pour Chaque transaction t ∈ D **Faire**

// Cette condition vérifie si un itemset existe dans la transaction (t) et (t+1)

Si (t.itemsets = 1) et ((t+1).itemsets = 1) **Alors**

règle (t.itemsets) ⟶ ((t+1).itemsets) ⟵ 1

Sinon Si [(t.itemsets = 1) **et** ((t+1).itemsets = 0)

ou (t.itemsets = 0) **et** ((t+1).itemsets = 1)] **Alors**

règle (t.itemsets) ⟶ ((t+1).itemsets) ⟵ 0

FSi

FSi

// Cette condition permet d'éliminer les itemsets inférieurs au support minimum.

Si Support (t.itemsets) ≥ Min_Sup Alors

F_I ⟵ C_I ∪ F_I

Sinon

Supprimer C_I.t.itemsets

FSi

// Cette condition vérifie si le nombre des items dans un itemset est supérieur ou égal au NB_Items, ou si le support minimum est inférieur à 0.1. Si une des deux conditions est satisfaite, notre algorithme cesse l'itération immédiatement.

Si Taille (F_I) ≥ NB_Items **Ou** Min_Sup ≤ 0.1 **Alors**

Sortir de la boucle

FSi

// Cette fonction permet de générer les combinaisons possibles sans redondance des itemsets supérieurs au support minimum

TM_Apriori_Comb(F_I)

FPour

Retourner F_I

Fin

Figure 17 – Pseudo Code de l'algorithme TM_Apriori

L'algorithme « TM_Apriori », prend en entrée la liste des transactions «base de connaissances», le support minimum et le nombre des items dans un itemset. Il commence par calculer le support de chaque item et le comparer au support minimum, tout en ne gardant que ceux dont le support est supérieur au support minimum. Par la suite, l'algorithme « TM_Apriori_Comb » génère toutes les combinaisons possibles sans redondance et vérifie le nombre d'itemsets génèrés. Si ce nombre est inférieur à 10, il diminue le support minimum de 0.1 sinon il garde le support minimum initial.
L'algorithme commence à itérer toutes les transactions de notre base de connaissances et vérifie la ressemblance entre les itemsets pour créer les règles d'association entre ceux qui ont une forte ressemblance. Puis il ne garde que les itemsets dont le support est supérieur ou égal au support minimum. Finalement l'algorithme cesse de fonctionner quand le nombre des items dans un itemset est supérieur ou égal au NB_Items ou quand le support minimum est inférieur à 0.1.

Algorithme 2 : TM_Apriori_Comb

Entree : t.itemsets

Sortie : t.candidat, C_I

Début

Pour Chaque pairs d'itemsets **Faire**

t.canditat ⟵ t.itemsets ∪ (t+1).itemsets

C_I ⟵ C_I ∪ t.candidat

FPour

Retourner C_I

Fin

Figure 18 - Pseudo code de l'algorithme TM_Apriori_Comb

L'algorithme « TM_Apriori_Comb » prend en entrée une liste soit des items ou des itemsets, il permet de parcourir les éléments de la liste et de générer les combinaisons possibles sans redondance.

4.2.7. **Classification**

Dans la partie de la classification, on établit le degré de similitude entre les segments en comparant les itemsets fréquents utilisés pour décrire ces derniers. Plus le nombre d'itemsets partagés entre deux segments est grand, plus ils sont jugés comme étant similaires.

Notre approche adopte la méthode de classification K-médoïdes qui nécessite un choix de nombre de classes K. Le package NbClust nous propose un nombre optimal de classes à partir d'une matrice de similarité ou une matrice de distance. On fait recours à la méthode K-médoïdes pour classifier nos segments. Par la suite et pour valider le choix du nombre de classes, on fait appel à la classification ascendante hiérarchique (CAH).

Finalement, on illustre les résultats issus de la classification par des graphiques en deux dimensions obtenues par une ACP que nous exposons dans la partie de discussion des résultats.

4.3. Exemple à deux thématiques différentes

Dans l'exemple illustre dans le tableau 4.4, on évoque deux sujets différents : les cinq premières phrases parlent de la carrière du célèbre joueur du NBA Michael Jordan, tandis que les 5 dernières, abordent l'histoire de la multinationale Microsoft.
On va appliquer toutes les démarches de notre approche proposée dans la figure 16.

Michael Jordan est un ancien joueur de NBA.
Michael Jordan a contribué dans la popularité du NBA.
En 1991, Michael Jordan a remporté son premier titre de champion de la NBA.
Michael Jordan est l'un des meilleurs joueurs de la NBA.
Michael Jordan est devenu joueur milliardaire grâce à NBA.
Microsoft est une entreprise de technologie fondée par en 1975 Bill Gates.
Microsoft domine le marché international de technologie.
Microsoft est connue à l'échelle internationale dans la technologie spatiale.
L'innovation des technologies de systèmes d'exploitation est l'activité principale de Microsoft.
Microsoft investit dans des projets novateurs des nouvelles technologies.

Tableau 4.4 - Ensemble de phrases

4.3.1. **Segmentation**

Dans cette étape, on a opté pour la segmentation en une phrase, du fait que chaque phrase est considérée comme étant une transaction.

Le tableau 4.5 nous illustre les résultats obtenus.

Items
Michael Jordan est un ancien joueur de NBA.
Michael Jordan a contribué dans la popularité du NBA.
En 1991, Michael Jordan a remporté son premier titre de champion de la NBA.
Michael Jordan est l'un des meilleurs joueurs de la NBA.
Michael Jordan est devenu joueur milliardaire grâce à NBA.
Microsoft est une entreprise de technologie fondée par en 1975 Bill Gates.
Microsoft domine le marché international de technologie.
Microsoft est connue à l'échelle internationale dans la technologie spatiale.
L'innovation des technologies de systèmes d'exploitation est l'activité principale de Microsoft.
Microsoft investit dans des projets novateurs des nouvelles technologies.

Tableau 4.5 - Segmentation du texte en phrases

4.3.2. **Préparation du texte**

À ce stade, on procède comme suit :

- Conversion des chiffres en blanc.
- Conversion des caractères spéciaux.
- Conversion du texte en minuscule.

Le résultat est contenu dans le tableau 4.6 :

michael jordan est un ancien joueur de nba
michael jordan a contribué dans la popularité du nba
en michael jordan a remporté son premier titre de champion de la nba
michael jordan est un des meilleurs joueurs de la nba
michael jordan est devenu joueur milliardaire grâce à nba
microsoft est une entreprise de technologie fondée par en bill gates
microsoft domine le marché international de technologie
microsoft est connue à échelle internationale dans la technologie spatiale
innovation des technologies de systèmes exploitation est activité principale de microsoft
microsoft investit dans des projets novateurs des nouvelles technologies

Tableau 4.6 - Conversion du texte

On remarque que, par exemple, il y a eu une suppression du chiffre « 1975 » ainsi que le mot « L'innovation » est devenu « innovation ».

4.3.3. **Nettoyage du texte**

Quant à cette étape importante, on commence par éliminer les mots fonctionnels. Le tableau 4.7 nous résume les modifications apportées sur cet ensemble.

michael jordan ancien joueur nba
michael jordan contribué popularité nba
michael jordan remporté premier titre champion nba
michael jordan meilleurs joueurs nba
michael jordan devenu joueur milliardaire nba
microsoft entreprise technologie fondée bill gates
microsoft domine marché international technologie
microsoft connue échelle internationale technologie spatiale
innovation technologies systèmes exploitation activité principale microsoft
microsoft investit projets novateurs nouvelles technologies

Tableau 4.7 - Suppression des mots fonctionnels

On constate qu'il y a eu l'élimination des mots fonctionnels. Prenant l'exemple de la première phrase, on avait : « michael jordan est un ancien joueur de nba » qui est devenue : « michael jordan ancien joueur nba », résultant de la suppression de « est, un, de » considérés comme étant des mots fonctionnels.

4.3.4. **Lemmatisation**

L'étape de la lemmatisation a conduit aux résultats du tableau 4.8 :

michael jordan ancien joueur nba
michael jordan contribuer popularité nba
michael jordan remporter premier titre champion nba
michael jordan meilleur joueur nba
michael jordan devenir joueur milliardaire nba
microsoft entreprendre technologie fonder bill gates
microsoft dominer marché international technologie
microsoft connaître échelle international technologie spatial
innovation technologie système exploitation activité principal Microsoft
microsoft investir projet novateur nouvelle technologie

Tableau 4.8 - Lemmatisation

On en déduit que, par exemple : « microsoft investit projets novateurs nouvelles technologies » le verbe « investit » est converti à sa forme canonique « investir » ,

aussi le mot « projets » qui était au pluriel est rendu au singulier « projet ». En suivant le même processus, le résultat final de notre phrase est : « microsoft investir projet novateur nouvelle technologie ».

4.3.5. **Extraction du vocabulaire**

L'étape intitulée « extraction du vocabulaire » est celle où on crée une base de connaissances des mots uniques à partir de l'ensemble des transactions. Par la suite, on calcule le pourcentage de leurs occurrences.

L'extraction du vocabulaire de notre base de connaissances est détaillée dans le tableau 4.9.

Items	Pourcentage	Items	Pourcentage
michael	50 %	fonder	10 %
jordan	50 %	bill	10 %
ancien	10 %	gates	10 %
joueur	30 %	dominer	10 %
nba	50 %	marché	10 %
contribuer	10 %	international	20 %
popularité	10 %	connaître	10 %
remporter	10 %	échelle	10 %
premier	10 %	innovation	10 %
titre	10 %	système	10 %
champion	10 %	exploitation	10 %
meilleur	10 %	activité	10 %
devenir	10 %	principal	10 %
milliardaire	10 %	investir	10 %
microsoft	50 %	projet	10 %
entreprendre	10 %	novateurs	10 %
technologie	50 %	nouvelle	10 %
spatial	10 %		

Tableau 4.9 - Base de connaissances avec le pourcentage d'occurrences

Vu qu'on a segmenté notre texte en phrases, on se retrouve avec 10 transactions, dont 5 d'entre elles se trouve le mot « michael », ce qui nous conduit au calcul suivant :

$\{\text{michael}\} = \frac{5}{10} \times 100 = 50\%$, d'où 50% des transactions contiennent le mot « michael ».

On effectue le même calcul pour le reste des mots.

4.3.6. **Application des règles d'association**

Afin de dégager un nombre restreint d'itemsets fréquents, on a fait recours à notre algorithme TM_Apriori. Dans notre exemple, on fixe le support minimum à 30% et le nombre des items dans les itemsets obtenus à 4.

Items	Pourcentage
jordan	50 %
nba	50 %
michael	50 %
microsoft	50 %
technologie	50 %
joueur	30 %

Tableau 4.10 - Items avec un pourcentage ≥ 30 %

Étant donné que le support minimum est fixé à 30%, on ne garde que les items dont le pourcentage est supérieur ou égal à ce seuil (Voir tableau 4.10).

4.3.6.1. ***Itération 1***

Génération des combinaisons

On génère toutes les combinaisons possibles et cela sans redondance à partir des items du tableau 4.10. (Voir tableau 4.11).

jordan nba
jordan michael
jordan microsoft
jordan technologie
jordan joueur
nba michael
nba microsoft
nba technologie
nba joueur
michael microsoft
michael technologie
michael joueur
microsoft technologie
microsoft joueur
technologie joueur

Tableau 4.11 - Liste des itemsets possibles

Calcul du pourcentage des combinaisons possibles

Par la suite, on procède au calcul du pourcentage de ces combinaisons. Tant que le nombre des combinaisons extrait est supérieur à 10, on maintient la valeur du support minimum à 30%. (Voir tableau 4.12).

Itemsets	Pourcentage
jordan nba	50 %
jordan michael	50 %
jordan microsoft	0 %
jordan technologie	0 %
jordan joueur	30 %
nba michael	50 %
nba microsoft	0 %
nba technologie	0 %
nba joueur	30 %
michael microsoft	0 %
michael technologie	0 %
michael joueur	30 %
microsoft technologie	50 %
microsoft joueur	0 %
technologie joueur	0 %

Tableau 4.12 - Calcul du pourcentage des combinaisons

Élimination des combinaisons inférieures au support minimum

En employant le support minimum sur les données du tableau 4.12, ci-après le tableau 4.13 illustre les combinaisons respectant ce support.

Itemset	Pourcentage
jordan nba	50 %
jordan michael	50 %
jordan joueur	30 %
nba michael	50 %
nba joueur	30 %
michael joueur	30 %
microsoft technologie	50 %

Tableau 4.13 - Combinaisons avec un support $\geq$ 30%

4.3.6.2. *Itération 2*

Génération des combinaisons

On génère toutes les combinaisons possibles et cela sans redondance à partir des items du tableau 4.13. (Voir tableau 4.14).

jordan joueur michael
jordan joueur michael nba
jordan joueur microsoft technologie
jordan joueur nba
jordan michael microsoft technologie
jordan michael nba
jordan microsoft nba technologie
joueur michael microsoft technologie
joueur michael nba
joueur microsoft nba technologie
michael microsoft nba technologie

Tableau 4.14 - Liste des itemsets possibles

Calcul du pourcentage des combinaisons possibles

Par la suite, on procède au calcul du pourcentage de ces combinaisons. Tant que le nombre des combinaisons extrait est supérieur à 10, on maintient la valeur du support minimum à 30%. (Voir tableau 4.15).

Itemsets	Pourcentage
jordan joueur michael	30 %
jordan joueur michael nba	30 %
jordan joueur microsoft technologie	0 %
jordan joueur nba	30 %
jordan michael microsoft technologie	0 %
jordan michael nba	50 %
jordan microsoft nba technologie	0 %
joueur michael microsoft technologie	0 %
joueur michael nba	30 %
joueur microsoft nba technologie	0 %
michael microsoft nba technologie	0 %

Tableau 4.15 - Calcul du pourcentage des combinaisons

Élimination des combinaisons inférieures au support minimum

En employant le support minimum sur les données du tableau 4.15, ci-après le tableau 4.16 illustre les combinaisons respectant ce support.

Itemsets	Pourcentage
jordan joueur michael	30 %
jordan joueur michael nba	30 %
jordan joueur nba	30 %
jordan michael nba	50 %
joueur michael nba	30 %

Tableau 4.16 - Combinaisons avec un support $\geq$ 30%

Notre algorithme cesse de générer de nouvelles combinaisons dès qu'on atteint le nombre fixé des items dans un itemset. Dans notre exemple, le nombre des items dans l'itemset « jordan joueur michael nba » est 4, ce qui valide notre contrainte. À ce stade-là, l'algorithme ne garde que les supersets et élimine leurs sous-ensembles.

Par exemple, l'itemset « jordan joueur michael nba » est un superset de l'itemset « joueur michael nba ». À cet effet, on ne garde que le superset. Par contre, l'itemset « microsoft technologie » n'est pas inclus dans le superset, ce qui implique qu'on garde les deux.

Le tableau 4.17 illustre les itemsets finaux:

Itemsets	Pourcentage
michael jordan joueur nba	30 %
microsoft technologie	50 %

Tableau 4.17 - Itemsets possibles

Finalement, une fois le processus est complété, une base de connaissances est générée sous forme d'une matrice binaire où les lignes représentent les transactions tandis que les colonnes représentent les itemsets générés. Leurs intersections représentent l'existence (codée 1) ou non (codée 0) de l'itemset dans chaque transaction (voir tableau 4.18).

Vecteur / Itemsets	Michael jordan joueur nba	microsoft technologie
michael jordan ancien joueur nba	1	0
michael jordan contribuer popularité nba	1	0
michael jordan remporter premier titre champion nba	1	0
michael jordan meilleur joueur nba	1	0
michael jordan devenir joueur milliardaire nba	1	0
microsoft entreprendre technologie fonder bill gates	0	1
microsoft dominer marché international technologie	0	1
microsoft connaître échelle international technologie spatial	0	1
innovation technologie système exploitation activité principal microsoft	0	1
microsoft investir projet novateurs nouvelle technologie	0	1

Tableau 4.18 - Matrice binaire de la base de connaissances

4.3.7. **Classification**

Afin de classifier les itemsets du tableau 4.18, on utilise dans un premier temps, la fonction « dist » du programme R avec la méthode de calcul « faith » qui nous permet de calculer la matrice de distance qu'on présente dans le tableau 4.19.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0									
2	0	0								
3	0	0	0							
4	0	0	0	0						
5	0	0	0	0	0					
6	1	1	1	1	1	0				
7	1	1	1	1	1	0	0			
8	1	1	1	1	1	0	0	0		
9	1	1	1	1	1	0	0	0	0	
10	1	1	1	1	1	0	0	0	0	0

Tableau 4.19 - Matrice de distance

Par la suite, on procède à la classification de notre matrice à l'aide de la méthode K-Médoïdes, où le nombre K de classes est donné par le package NbClust. Dans notre exemple, le résultat de NbClust nous propose K = 2 comme nombre optimal de classes. La figure 19 illustre la classification de notre matrice à l'aide de la méthode K-Médoïdes avec K = 2.

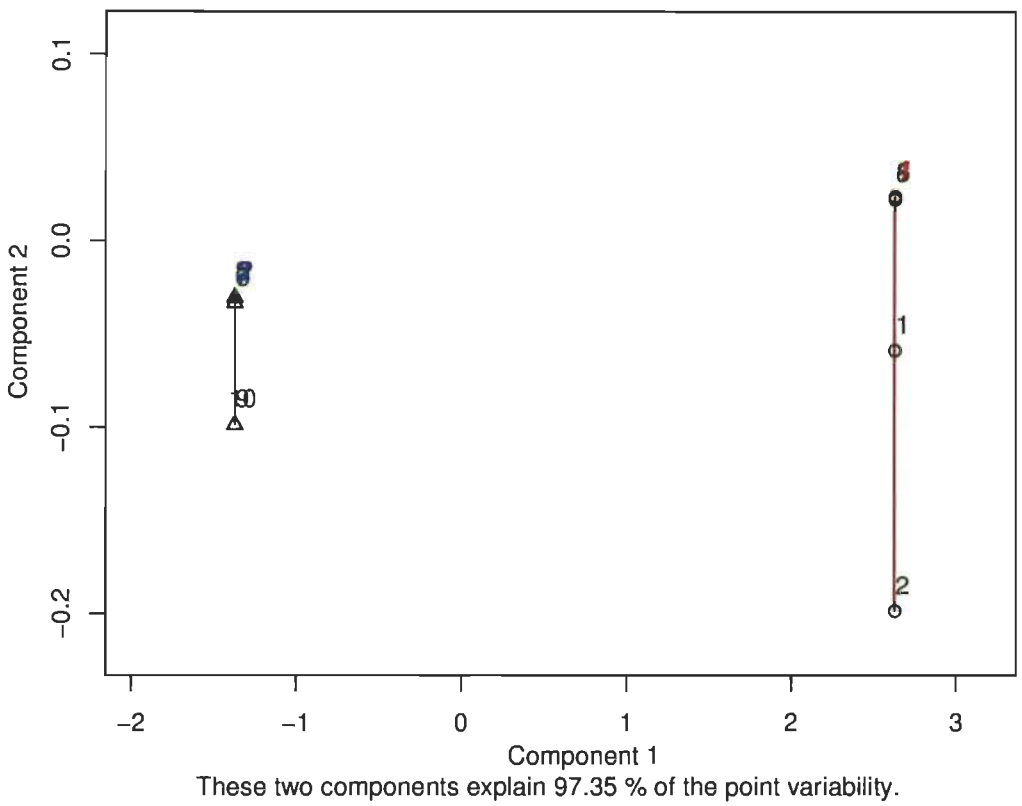

Figure 19 – Résultat de la classification des vecteurs en K-Médoïdes avec K = 2

Le tableau 4.20 détaille les résultats représentés dans la figure 19 en détaillant les objets de chaque classe.

Classes	Vecteurs
1	1, 2, 3, 4, 5
2	6, 7, 8, 9, 10

Tableau 4.20 - Distribution des vecteurs dans chaque classe K = 2

Afin de confirmer notre choix du nombre de classes K = 2, on utilise la classification ascendante hiérarchique (CAH). D'après la figure 20, on constate que le choix du K = 2 est le nombre de classes optimal vu qu'on trouve les mêmes groupes de segments dans le résultat du K-Médoïdes.

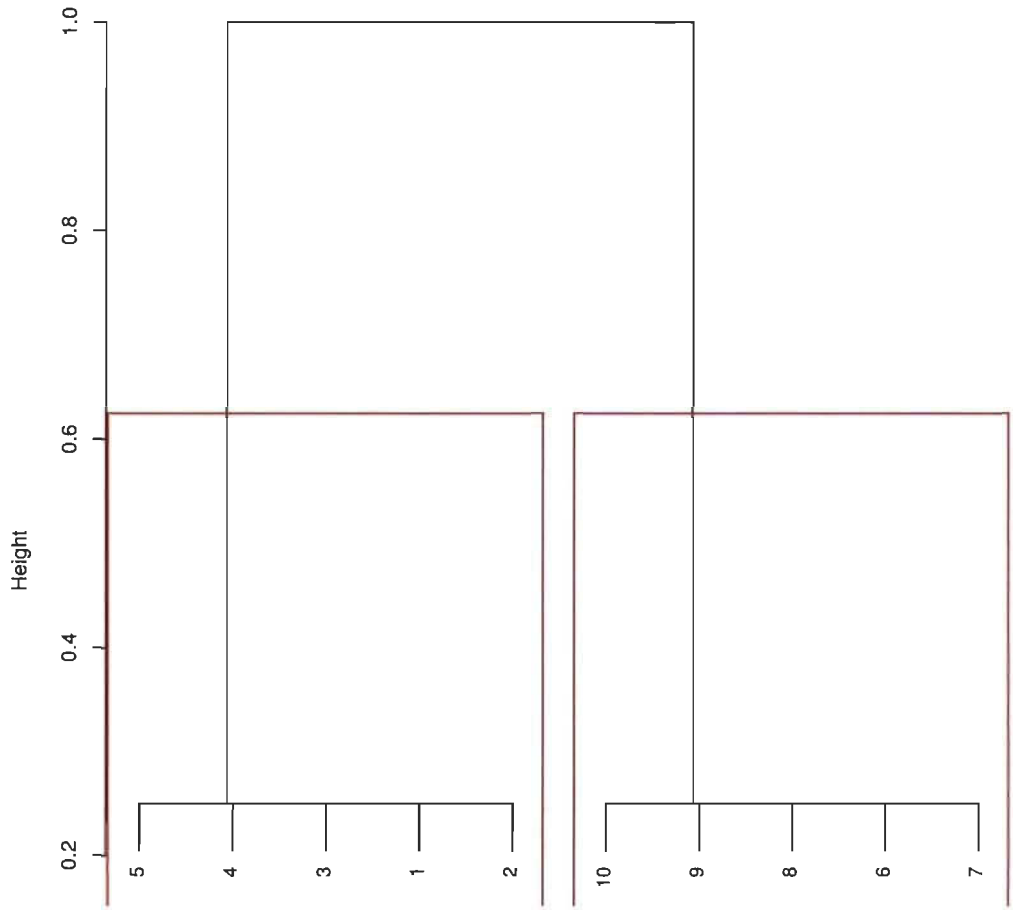

Figure 20 - Classification CAH des itemsets avec coupure en deux classes

De cette manière, notre approche a permis d'extraire des connaissances à partir d'un texte brut jusqu'à la classification. Finalement, c'est à l'expert du texte d'interpréter et de juger le résultat de la classification des itemsets.

Quant à notre exemple, on constate qu'on obtient une bonne classification puisque d'une part, l'ensemble de vecteurs {1, 2, 3, 4 et 5} appartenant à la même classe (voir figure 20) traitent le même sujet de NBA. D'autre part, l'ensemble de vecteurs {6, 7, 8, 9 et 10} ayant leur propre classe traitent le sujet de Microsoft.

4.4. Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes essentielles sur lesquelles se base notre approche dans le but d'explorer un texte brut et d'extraire des itemsets pour des fins de classification. Le chapitre qui suit élucidera la partie expérimentale de notre projet. Par la même occasion, on procèdera à analyser et interpréter les résultats obtenus.

CHAPITRE 5 EXPERIMENTATIONS ET DISCUSSIONS

5.1. Introduction

Dans cette section, nous discuterons deux différentes expérimentations sur des données textuelles en français. Le premier document possède des paragraphes bien adaptés à notre approche, tandis que le second simule une expérimentation réelle avec un nombre important de paragraphes dont les thématiques sont similaires.

Afin d'évaluer l'approche proposée, nous avons développé une plateforme en C# (Présentation du langage C#, 2019) capable d'importer des documents, de les prétraiter et d'extraire des itemsets fréquents à l'aide de la méthodologie présentée au chapitre 4. Pour la classification et la visualisation des résultats, on fera appel aux fonctionnalités de RStudio (Présentation du RStudio, 2019). Dans la suite de ce mémoire, notre plateforme prendra le nom de « IDETEX ».

5.2. Première expérimentation

Pour cette expérimentation, nous avons utilisé un document composé de 22 paragraphes et couvrant 3 sujets différents. Les 4 premiers paragraphes parlent de sport, en particulier « biographie de Michael Jordan », alors que les paragraphes allant de 5 à 8 portent sur le domaine informatique « Microsoft », les paragraphes 9 à 22, sont consacrés à la vie et à la contribution de « Wolfgang Amadeus Mozart » dans le domaine de la musique classique.

Etant donné la simplicité de ce document et notre connaissance préalable de sa structure qui ne contient que 3 thématiques différentes, on prévoyait que IDETEX identifie 3 classes bien séparées et homogènes.

Nous avons prétraité le document et extrait les itemsets tout en suivant les étapes qu'on a détaillées dans le chapitre de la méthodologie (voir Figure 16). Nous avons mesuré le pouvoir discriminant des itemsets fréquents.

Pour le besoin de l'expérimentation, nous avons fait une première expérimentation en utilisant comme descripteur du texte les itemsets fréquents, puis une deuxième avec les mots comme descripteurs et enfin nous avons comparé les résultats.

La figure 21 montre un extrait du document en question.

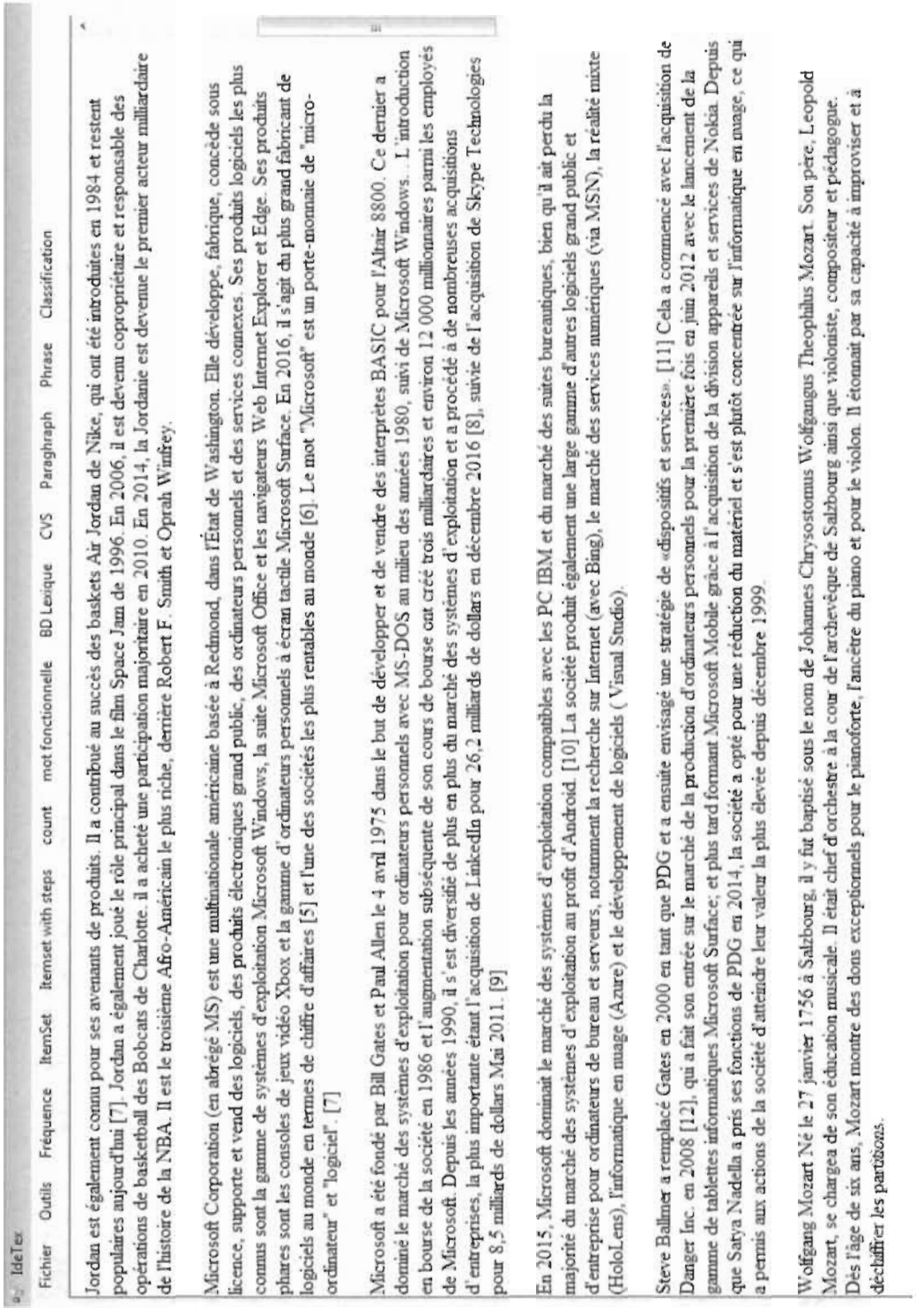

Figure 21 - Extrait du document de l'expérimentation

L'étape qui suit permet de segmenter notre document en paragraphes, ce qui est illustré dans la figure 22

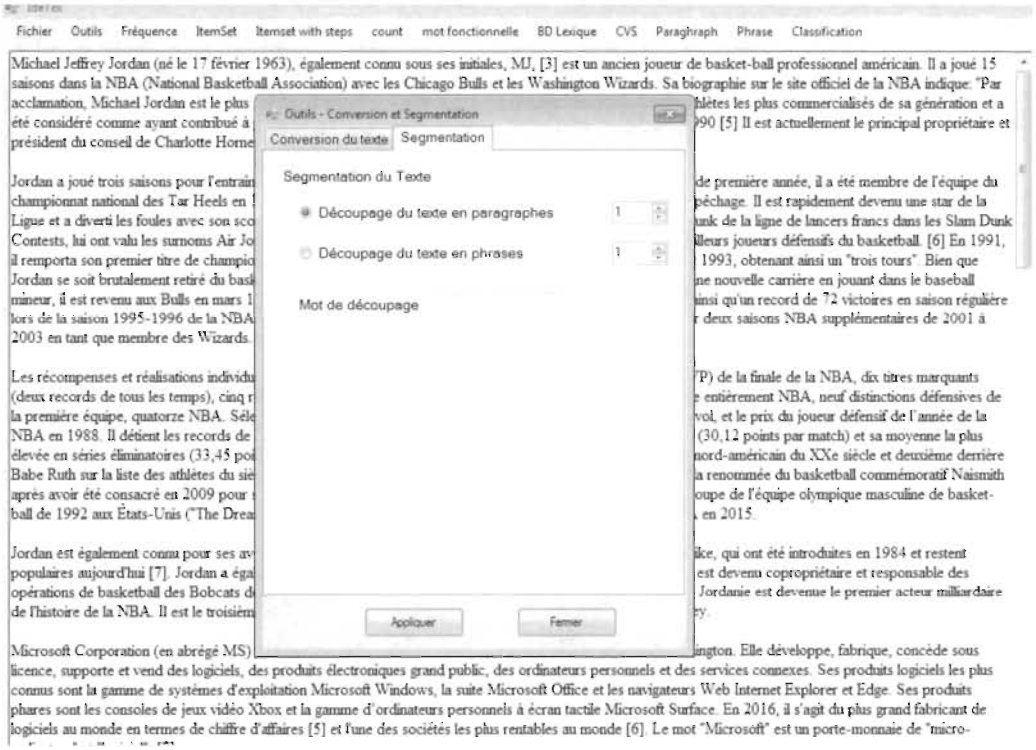

Figure 22 - Segmentation du texte

À ce stade, l'étape de nettoyage effectuée sur notre document comprend les éléments suivant (voir Figure 23) :

- Conversion des chiffres en blanc
- Conversion des caractères en minuscule
- Conversion des caractères spéciaux en blanc
- Suppression des mots fonctionnels
- Application de la lemmatisation

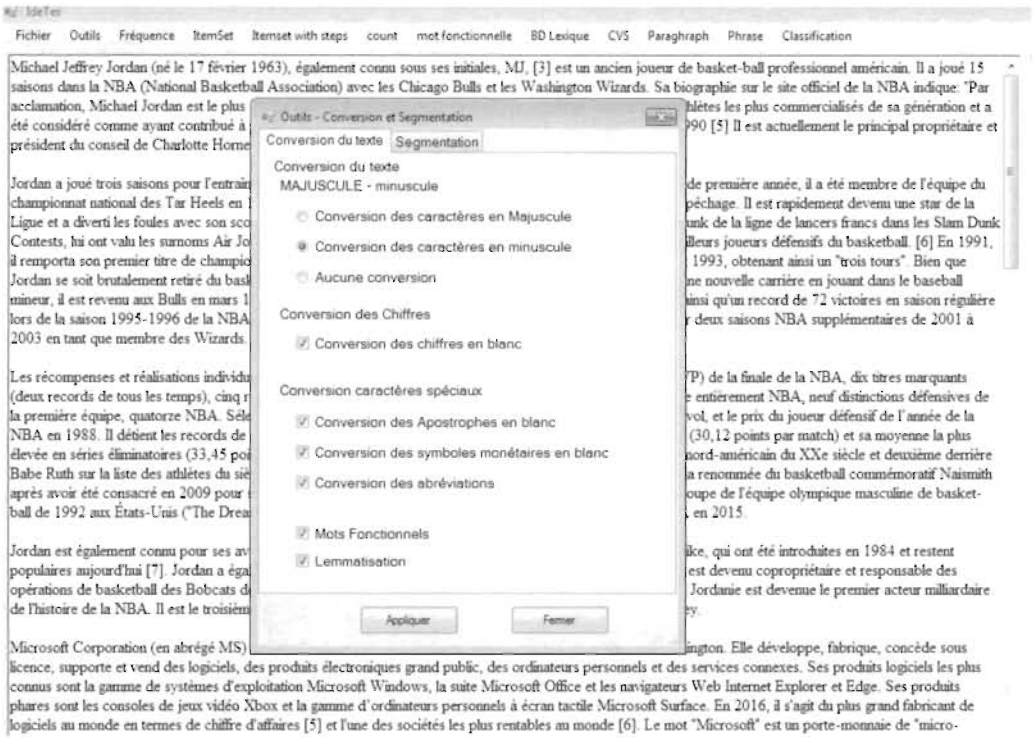

Figure 23 - Fonctionnalités de nettoyage du texte

5.2.1. **Itemsets fréquents comme descripteur du texte**

Une fois le document segmenté et nettoyé, l'étape de l'extraction des itemsets fréquents est effectuée. À la suite de plusieurs expérimentations sur le document, on conclut qu'en fixant le support minimum à 12% et le nombre d'items dans un itemset à 4, cela nous permet d'extraire des itemsets fréquents jugés pertinents. Nous avons également prévu un autre paramètre dans IDETEX. Néanmoins, pour les besoins de cette expérimentation, nous ne l'avons pas utilisé à savoir « Confiance » (voir Figure 24).

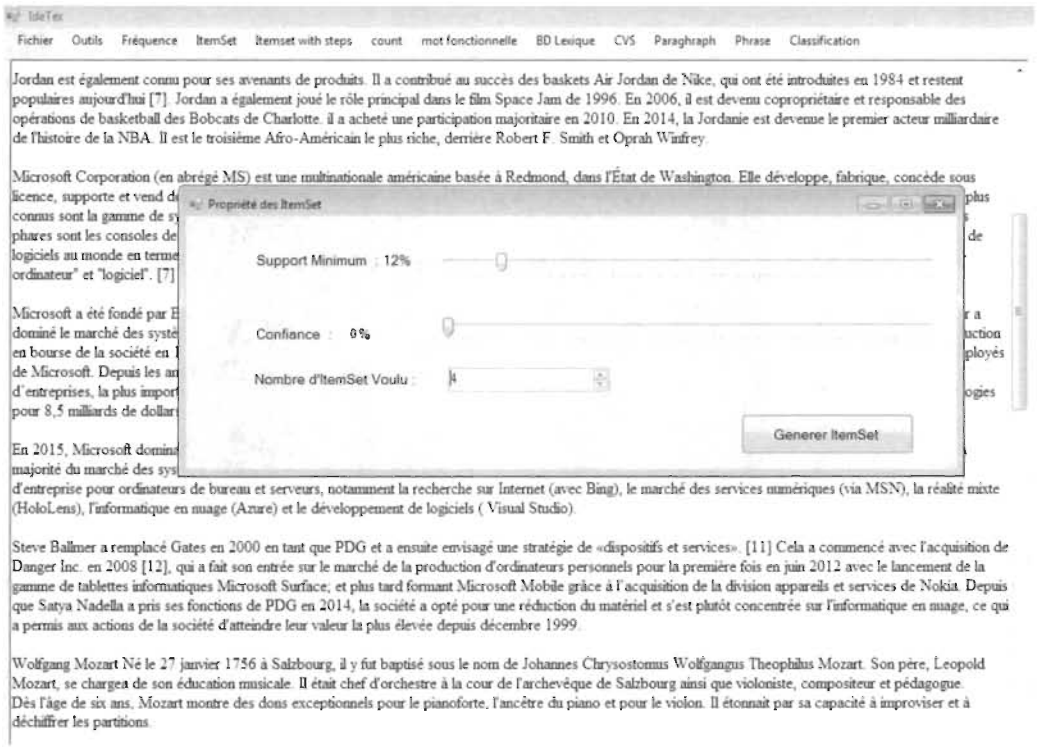

Figure 24 - Choix des paramètres

Le résultat de l'extraction des itemsets fréquents est une matrice binaire dont un extrait est présenté à la figure 25, les lignes représentent les segments tandis que les colonnes contiennent les itemsets. Au total IDETEX a pu générer 66 itemsets fréquents qui décrivent notre document.

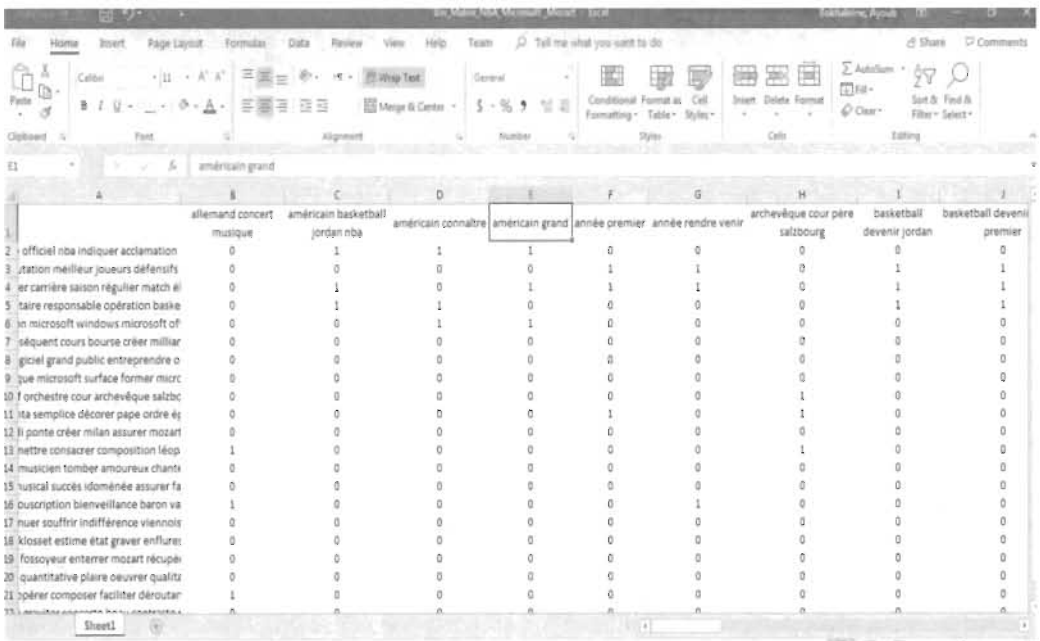

Figure 25 - Matrice binaire des Itemsets

Afin de classifier nos itemsets, on a utilisé le Package NbClust. Ce dernier a proposé un nombre optimal de 3 classes, ce qui est conforme à la structure de notre document qui contient bel et bien 3 thématiques différentes (Voir figure 26).

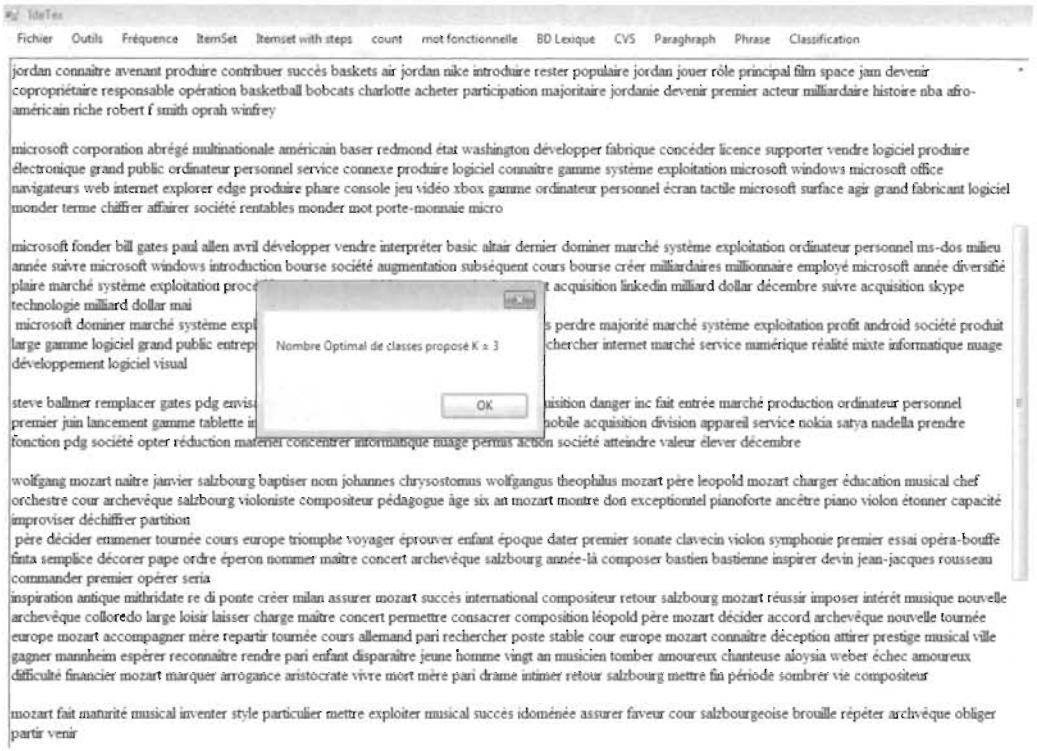

Figure 26 - Choix Optimal du nombre de classes du package NbClust

À l'étape de classification, nous avons utilisé K-Médoïdes. La figure 27 illustre les classes obtenues. On remarque que la classe 1 regroupe uniquement les paragraphes 1 à 4, qui couvrent le sujet de « Michael Jordan ». La classe 2 contient les paragraphes 5 à 8 relatifs à l'informatique « Microsoft ». Finalement, la classe 3 contient les paragraphes 9 à 22, qui traitent le sujet de musique « Mozart ». La classification ascendante hiérarchique confirme les résultats obtenus (Voir figure 28).

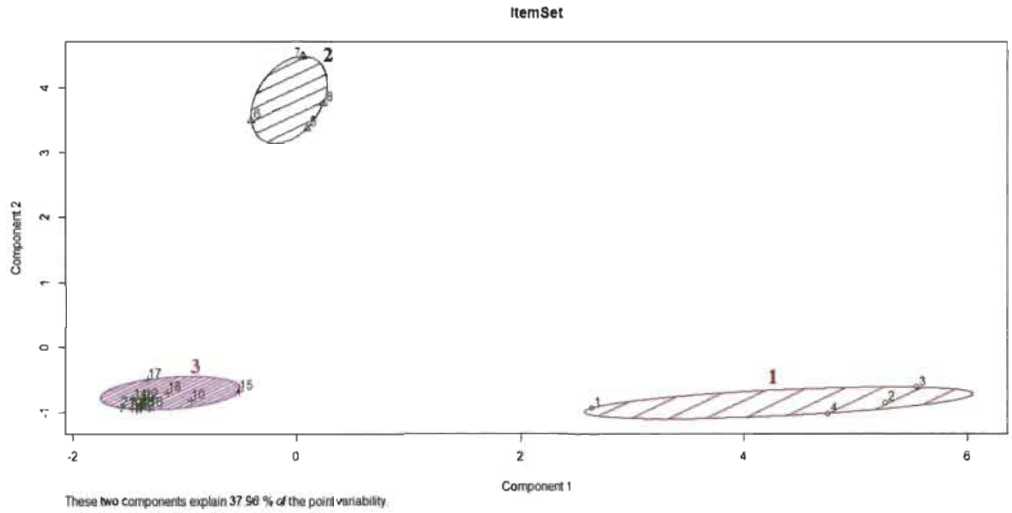

Figure 27 - Classification des Itemsets avec K-Médoïdes

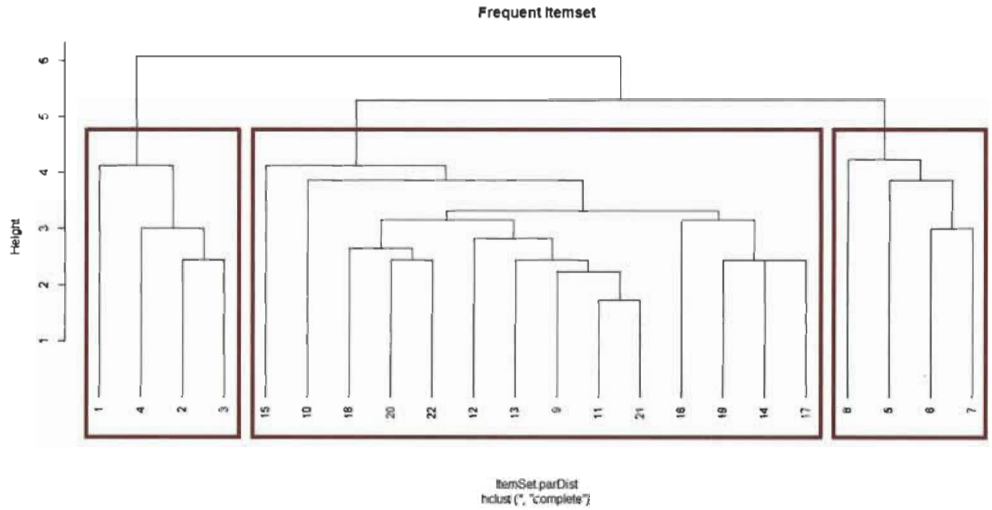

Figure 28 - Classification des Itemsets avec CAH

5.2.2. **Mots comme descripteurs**

Nous utiliserons le même document traité et nettoyé pour extraire les mots. IDETEX a pu générer 659 mots comme descripteurs du texte. La classification par K-Méloïdés avec K = 3 (Figure 29) donne des classes hétérogènes. En effet, la classe 1 qui contient la majorité des paragraphes combine les sujets qui traitent aussi bien le sport, l'informatique et la musique. Ce même constat a été observé par la CAH tel que spécifié dans la figure 30.

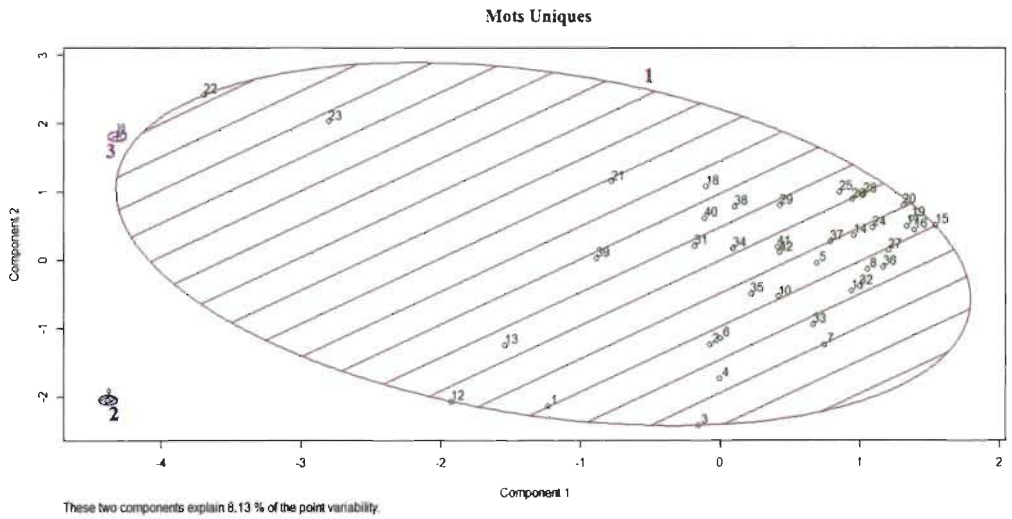

Figure 29 - Classification des mots avec K-Médoïdes

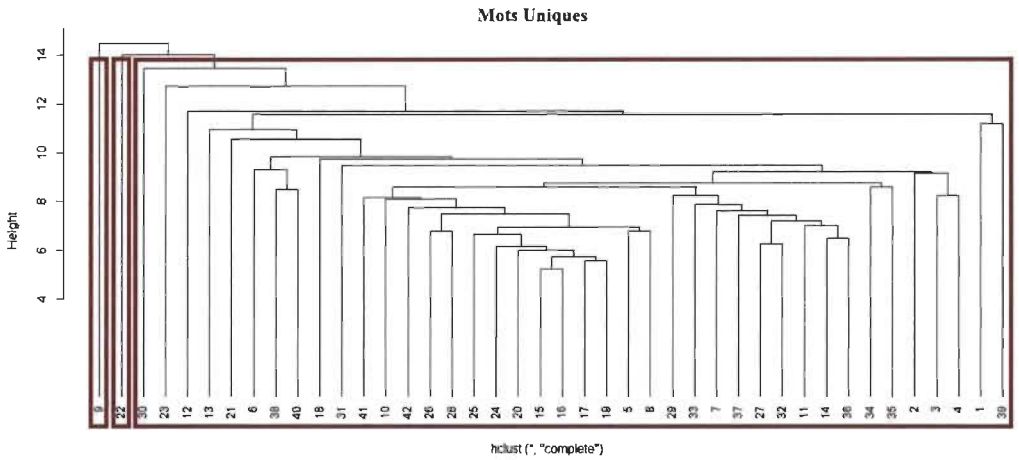

Figure 30 - Classification des mots uniques avec CAH

L'utilisation des itemsets fréquents comme descripteurs du texte permet de réduire significativement la taille de la matrice à classifier (66 itemsets fréquents comparés à 659 mots uniques). Le résultat donne une classification homogène, vu qu'on sait au préalable que notre document contient 3 thématiques différentes.

5.3. Deuxième expérimentation

La seconde expérimentation concerne un document plus volumineux et complexe dont le but est de démontrer la pertinence de l'utilisation d'itemsets fréquents comme descripteurs.

L'expérimentation a été réalisée sur le livre « La civilisation des Arabes » écrit en français (Gustave Le Bon, 1884). Le document comporte 6 chapitres. La segmentation de ces chapitres est effectuée en découpant le texte en 6 chapitres :

- Chapitre 1 et 2 contiennent les segments de 1 à 6 qui couvrent respectivement les sujets « L'arabie » et « Les arabes »
- Chapitre 3 contient les segments 7 à 16 qui traitent le sujet « Les Arabes avant Mahomet »
- Chapitre 4 contient les segments 17 à 27 qui racontent l'histoire de « Mahomet. Naissance de l'empire arabe »
- Chapitre 5 contient les segments 28 à 32 qui parlent du « Le Coran »
- Chapitre 6 contient les segments 33 à 42 qui parlent de « Les conquêtes des Arabes »

En utilisant notre méthodologie mentionnée dans le chapitre 4, nous avons prétraité le document en suivant la même procédure appliquée dans la première expérimentation. Nous avons aussi considéré les mêmes paramètres d'extraction des itemsets où le support minimum est égal à 12% et le nombre des items dans un itemsets à 4. IDETEX a généré 43 itemsets. La grande similarité thématique fait de ce document une expérimentation complexe. La discrimination des classes indépendantes n'est pas un processus facile à définir même si notre approche montre une énorme capacité à déterminer des classes homogènes, comme le montre la figure 31.

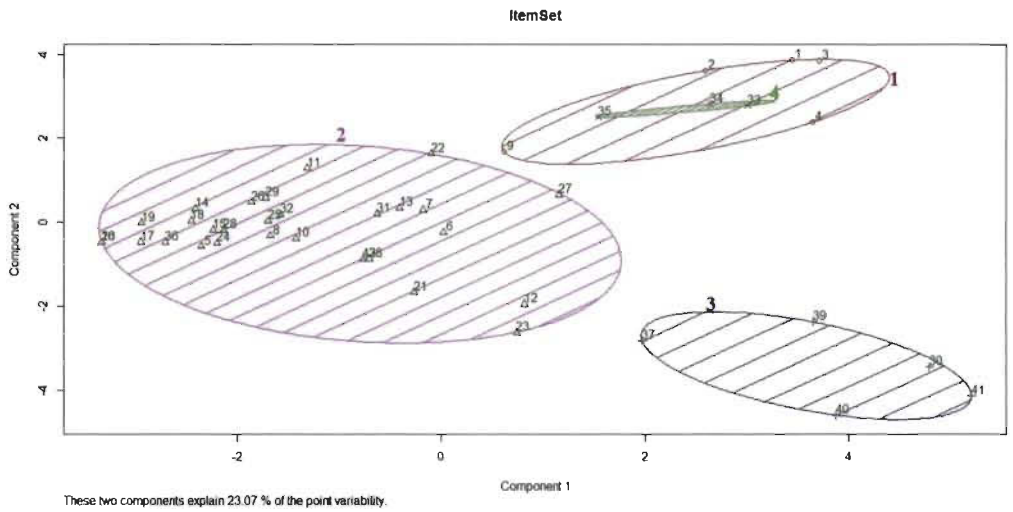

Figure 31 - Classification des Itemsets avec K-Médoïdes

La classe 1 contient les segments 1, 2, 3 et 4 qui traitent les sujets « L'arabie » et « L'arabe ». On constate que le segment 9 qui se positionne tout près de la classe 2 dont le sujet est « les arabes avant mahomet » a des points en commun avec les segments de la classe 1, ce qui justifie sa présence dans cette dernière.

Étant donné que notre représentation graphique est faite sur un plan à deux dimensions, la classe 4 semble en fait être incluse dans la classe 1. Toutefois, après vérification de nos données d'entrée, nous nous rendons compte que les segments 33, 34 et 35 qui traitent le sujet « Les conquêtes des Arabes » sont des aspects plus précis de la grande classe 1.

La classe 2 contient la plus grande majorité des segments qui traitent tous les sujets suivants : « Les Arabes avant Mahomet », « Mahomet. Naissance de l'empire arabe » et « le Coran » en plus de quelques segments qui proviennent d'autres chapitres.
La classe 3 englobe les segments du chapitre 6 qui, quant à eux parlent de « Les Conquêtes des Arabes ».

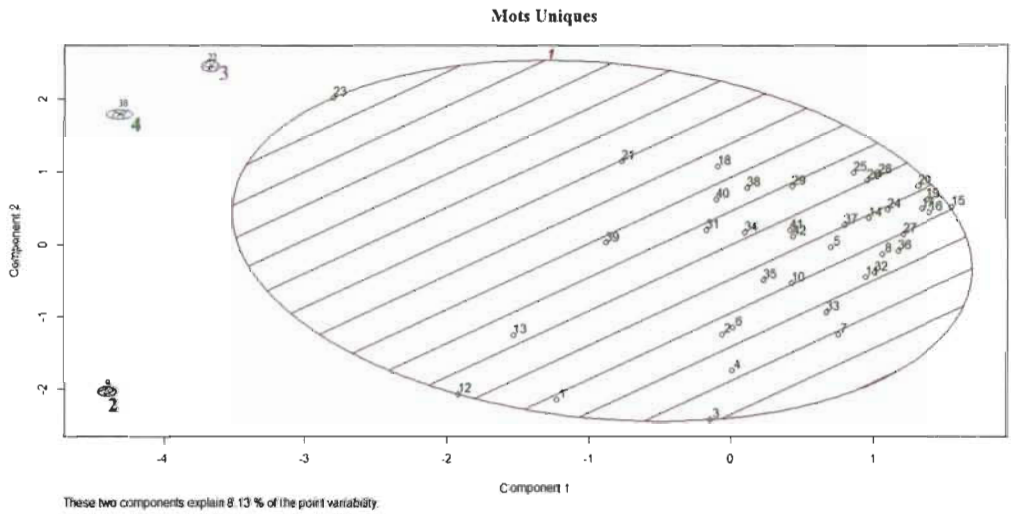

Figure 32 - Classification des mots uniques avec K-Médoïdes

La figure 32 montre d'une part l'inconvénient de classifier les données en ne se basant que sur les mots uniques d'autre part, elle prouve la pertinence d'utiliser les itemsets fréquents comme descripteurs du texte.

CHAPITRE 6 CONCLUSION ET PERSPECTIVES

Nous nous sommes intéressés dans ce travail à l'amélioration de la qualité des résultats issus d'un processus de classification. Traditionnellement celui-ci repose sur l'utilisation des mots, de N-Gram de caractères comme descripteur d'un document textuel. Bien que les résultats aient été encouragent, des limitations présentent. Nous citerons, entre autres, la taille du lexique et les différents aménagements pour la diminuer. Notre défi dans ce travail était de montrer que l'utilisation des itemsets fréquents permettait de contourner ce type de problématique. De plus, nous avons considéré des classifieurs non-supervisés à savoir la méthode des K-Médoïdes et la classification ascendante hiérarchique.

Nos expérimentations ont confirmé l'importance d'utiliser les règles d'association pour des fins d'extraction de l'information pertinente et de qualité que l'on peut retrouver dans les données textuelles. Les résultats obtenus, lorsqu'on utilise les itemsets comme descripteurs du texte, montrent une classification plus homogène que dans le cas où les mots sont les descripteurs.

Bien que les résultats obtenus soient bons, on constate que le choix des seuils (support minimum, nombre d'items dans un itemset) est une étape extrêmement importante. Lors de l'extraction des itemsets, le choix du support minimum ne devrait pas être arbitraire vu qu'il joue un rôle important sur la qualité des itemsets extraits.

Les perspectives à court terme de ce travail seront consacrées à poursuivre notre réflexion et, plus particulièrement, à développer :

- Une intégration d'un outil qui sera en mesure de proposer un seuil minimal optimal.
- D'autres méthodes de classification pour pouvoir comparer les résultats obtenus.
- Une généralisation de notre approche afin de l'élargir à d'autres textes et d'autres langues.

RÉFÉRENCES

1. Abdelali, M. and O. Hicham (2003). « Création de règles d'association ». Caen, Ensicaen.
2. Nouasria A. (2016). « Extraction d'associations lexicales fortes dans les commentaires ». Mémoire de maitrise en informatique, université du Québec à Trois-rivières..
3. Agrawal, R., Imielinski, T., et Swami, A. (1993). « Mining association rules between sets of items in large databases », SIGMOD Conference, Vol 22, N 2, pp. 207-216. doi : 10.1145 / 170035.170072
4. Baroni-Urbani, C, Buser, M.W. 1976 « Similarity of Binary Data », Systematic Biology. Vol 25, version 3, pp. 251-259. doi : https://doi.org/10.2307/2412493
5. Biskri I, Achouri A., Rompré L., Descôteaux S, Bensaber Boucif A., (2013) « ComputerAssisted Reading : Getting Help from Text Classification and Maximal Association Rules », Journal of Advances in Information Technologie.
6. Blanchard, J., (2005). « Un système de visualisation pour l 'extraction, l 'évaluation, et l'exploration interactives des règles d 'association ». École Polytechnique de l'Université de Nantes.
7. Charrad M, Ghazzali N, Boiteau V, Niknafs A, 2014. « NbClust : An R Package for Determining the Relevant Number of Clusters in a Data Set », Journal of statistical software, Volume 61, Issue 6. doi : 10.18637/jss.v061.i06.
8. Cherfi, H. and Y. Toussaint (2002). « Adéquation d'indices statistiques à l'interprétation de règles d'association ». Actes des 6eme Journées internationales d'Analyse statistique des Données Textuelles. Saint-Malo.1.
9. Descôteaux, S., (2014). « Les règles d'association maximale au service de l'interprétation des résultats de la classification ». Mémoire de maitrise en informatique, université du québec à trois-rivières, trois-rivières, p. 174
10. Diop, C. T., M. Lo, et al. (2007). « Intégration de règles d'association pour améliorer la recherche d'informations XML ». Quatrième conférence francophone en Recherche d'Information et Applications. École Nationale Supérieure des Mines de Saint-Étienne.
11. Fisher R. A. (1936). « The use of multiple measurements in taxonomic problems ». Annals of Eugenics. Vol 7, version 2, pp.179-188. doi : 10.1111/j.1469-1809.1936.tb02137.
12. Gras, R., Couturier, R., Bernadet, M., Blanchard, J., Briand, H., Guillet, F., Kuntz, P., Lehn, R., et Peter, P. (2004). « Quelques critères pour une mesure de qualité de règles

d'association - un exemple : l'intensité d'implication ». Revue des Nouvelles Technologies de l'Information.

13. Gustave Le Bon, (1884). « La civilisation des Arabes ».
14. Hájek, P., Havel, I., et Chytil, M. (1966). « The GUHA method of automatic hypotheses determination » Vol 1, issue 4, pp. 293-308. doi : 10.1007 / BF02345483
15. Han, J., et Kamber. M. (2006) « data mining: concepts and techniques ». p. 761.
16. Hilali, H., (2009). « Application de la classification textuelle pour l'extraction des règles d'association maximales ». Mémoire de maitrise en informatique, université du Québec à Trois-rivières.
17. Jaccard, P. 1901. « Étude comparative de la distribution florale dans une portion des Alpes et des Jura ». Bulletin de la Société Vaudoise des Sciences Naturelles, 37 : pp. 241-272. doi : 10.5169/seals-266440
18. Jin X., Han J. (2011). « K-Medoids Clustering », Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston. doi : https://doi.org/10.1007/978-0-387-30164-8
19. Labiad, A., (2017). « Sélection des mots clés basée sur la classification et l'extraction des règles d'association ». Mémoire de maitrise en informatique, université du Québec à Trois-rivières.
20. MacQueen, J. (1967) « Some methods for classification and analysis of multivariate observations ». Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol 1, pp. 281-297, University of California Press, Berkeley. doi : https://projecteuclid.org/euclid.bsmsp/1200512992
21. Nombré, C.I.,Brou, K., Kimou, K., (2016). « ALOA2i:Optimisation d'extraction des k-itemsets frequents (pour $K \leq 2$) ». Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées, INRIA. hal-01386948.

WEBOGRAPHIE

22. Présentation du RStudio, 2019, « http://ncss-tech.github.io/stats_for_soil_survey/chapters/1_introduction/1_introduction.html#3_rstudio:_an_integrated_development_environment_(ide)_for_r »
23. Présentation du langage C#, 2019, « https://docs.microsoft.com/fr-fr/dotnet/csharp/tour-of-csharp/ »

ANNEXE Frequent Itemsets as Descriptors of Textual Records

(Ayoub Bokhabrine, Ismail Biskri et Nadia Ghazzali, 11th International Conference on Computational Collective Intelligence, Springer, Cham, 2019)

Frequent Itemsets as Descriptors of Textual Records

Ayoub Bokhabrine[1], Ismaïl Biskri[2] and Nadia Ghazzali[3]

[1] Université du Québec à Trois-Rivières – ayoub.bokhabrine@uqtr.ca
[2] Université du Québec à Trois-Rivières – ismail.biskri@uqtr.ca
[3] Université du Québec à Trois-Rivières – nadia.ghazzali@uqtr.ca

Abstract

The analysis of numerical data, whether structured, semi-structured, or raw, is of paramount importance in many sectors of economic, scientific, or simply social activity. the process of extraction of association rules is based on the lexical quality of the text and on the minimum support set by the user. In this paper, we propose to use frequent itemsets as descriptors and classifying them by using K-Medoids algorithm and Hierarchical cluster. We present how they can be identified and used to define a level of similarity between several segments. The experiments conducted demonstrate the potential of the proposed approach for defining similarity between segments.

Keywords: Clustering, Frequent Itemsets, Descriptor, Segment, Text, K-Medoids, Ascending hierarchical cluster.

Introduction

The digitization of documents facilitated the dissemination of information. As soon as an event occurs multiple articles are written and broadcast on different digital platforms. Several textual documents distributed on the web are composed of only a few hundred words. It is by consulting various documents that a rich description can be obtained. Different documents may address the same subject and each of these documents may contain additional information. However, the quantity of data available and their lack of structure limit our ability to capture this information, hence the need to use tools that facilitate access to information. Automatic classification is one of the strategies applied to the problem of organizing information. A classificatory process applied to textual documents, whether automated or not, organizes documents so that those who share similarities are clustered together. The resulting organization can be used to guide, for example, information retrieval, knowledge extraction, summary help, etc.

Several automatic classifiers have been published. Comparing these classifiers to determine their performance is a complex task and, above all, subjective. A classifier can perform with a particular set of data and generate noisy classes with another set. The relevance of a classification is assessed according to the homogeneity of the resulting classes. This criterion is however relative. The evaluation of a cluster is based on interveners' research objectives and their knowledge of the subject area. The quality sought for an automated classification system is to be able to target the relevant information within the targeted segments and determine how this information can be used to establish a level of similarity between these segments.

The numerical classification is based on the identification and evaluation of descriptors that differentiate one class from another. the choice of a descriptor instead of another is to take a position on the nature of the results generated. It influences the classifier's behavior because of the presence or absence of a descriptor is an index to target the class to which a document belongs.

For textual classification, the word is often used as a discriminating descriptor (McCallum and Nigam, 1998). When several words appear at comparable frequencies in two segments then these segments are considered to be similar. However, it is common for segments to share a large number of words, even if these segments deal with different subjects. The mere presence of these words, therefore, is sparsely informative and its utility in establishing the level of similarity between segments is limited. Nevertheless, the relationship between these words and others can highlight specific peculiarities of certain segments. These relationships can be used to establish the level of similarity between segments.

Association rules

Association rule mining is a technique to uncover the relationship between various items, elements, or various variables in a very large database. It is also at a basic level, involves the use of machine learning models to analyze data for patterns, or co-occurrence, in a database. It identifies frequent if-then associations, which are called association rules. For example, customers who buy items X and Y also purchase item Z, in this case the association rule takes the following form: $(X, Y) \rightarrow Z$ Where the set (X, Y) is the antecedent, and Z: the consequent. Since then, the approach has been transposed to other domains, the association rules can be applied to various domains in that the concept of transaction can be defined.

Let T be a set of transactions such as: $T = \{t1, t2, t3..., tn\}$, the elements that make up the transactions $ti \in T$ are called items. An item is a datum whose nature depends on the area covered. For example, the items may correspond to descriptors extracted from a music (Rompré et al., 2017), to descriptors extracted from an image (Alghamdi et al., 2014) or simply to words extracted from a text (Zaïane and Antoine, 2002). Thus, a transaction can be defined simply as a subset of descriptors.

Let $I = \{i1, i2, i3, ..., id\}$ be a set of distinct d items, each subset that can be generated from items $ii \in T$ is called an itemsets. For a set I of size d, the number of possible itemsets is $2d$ (Tan et al., 2002). The number of potential itemsets is exponential, depending on the size of I. The objective to be reached during the process of extracting association rules is to discover hidden relationships, there is no index to target the items to consider. Thus, the search space is equivalent to all possible itemsets. Although it is theoretically possible to create $2d$ itemsets from a set of size d, in practice several combinations appear little or just not in transactions. Therefore, these combinations can be ignored. The support is a measure that allows to target the itemsets to ignore. The support of an itemsets X represents the percentage of transactions of T that contain X. It is denoted S(X) and given by the equation (1.1) where n equals to the total number of transactions contained in T and σ(X) to the support count. The support count of an itemsets X represents the number of transactions of T that contain X. It is given by the equation (1.2)

$$S(X) = \sigma(X)/n \quad (1.1)$$

$$\sigma(X) = |\{t_i \mid X \subseteq t_i,\ t_i \in T\}| \quad (1.2)$$

An itemsets X is considered frequent when its support is greater than or equal to a predetermined threshold. Let X and Y be two frequent itemsets such that $X \cap Y = \emptyset$, an association rule denoted $X \rightarrow Y$ expresses a co-occurrence relation between these itemsets. By convention, the first term is called the antecedent while the second is called the consequent. An association rule is considered quality according to a measure m and a previously fixed threshold. Thus, an association rule $X \rightarrow Y$ is judged of quality if $(X \rightarrow Y) \geq$ threshold.

The quantity of rules generated, their relevance and utility are highly dependent on the measures and minimum thresholds set. The evaluation of the interest measures of the association rules has been the subject of several studies (Le Bras et al., 2010, Geng and Hamilton, 2006 and Tan et al., 2002). Even if there are several variants, the extraction of association rules is usually done using the Appriori algorithm (Agrawal and Srikant, 1994) or FP-Growth (Han et al., 2000). Other algorithms are presented in (Fournier-Viger et al., 2017). The two main difficulties in extracting association rules are memory management and the computational effort required to search for frequent itemsets. Controlling the number of items to consider is the best way to deal with these difficulties. For two decades, several studies have focused on the application of association rules for classification purposes (Liu et al., 1998, Zaïane and Antoine, 2002, Bahri and Lallich, 2010). The different classifiers that result from this work produce results that are able to compete with those obtained using other approaches such as decision trees (Mittal et al., 2017).

Segments are considered as transactions while descriptors (Itemset, frequency of appearance of itemset, etc.). Let a set of descriptors Segments = {itemset1, itemsets2, itemset 3, ..., itemset l}, then a set of segments can be represented as follows:

$$Segment_1 = \{itemset_{10}, itemset_{21}, itemset_{16}, itemset_{20}, itemset_{18}\}$$

$$Segment_2 = \{itemset_{21}, itemset_{17}, itemset_{10}, itemset_{20}, itemset_{19}\}$$

$$Segment_3 = \{itemset_9, itemset_5, itemset_8, itemset_2, itemset_7\}$$

This segment set allows us to construct a binary matrix where the segments are considered as vectors and the itemsets as a descriptor, their intersection represents either the existence of the itemsets as described by 1 or in the opposite case it is considered 0.

The binary matrix is generated based on the results of the completion of the process of itemsets extraction, following that, the binary matrix is utilized as the input of the classifier. Thus, these classifiers attempt to encounter the similarity between segments and finally cluster them in separated classes.

Clustering methods

Unsupervised classification or Clustering as a technique for discovering subgroups within observations is utilized broadly in applications like market segmentation wherein, we attempt and discover some structure in the data. In our case we used unsupervised clustering method with textual data, our experimentation is based on two clustering methods the **Hierarchical method** and a variant of **K-means** named **K-medoids** clustering.

Dividing around Medoids, the K-medoids algorithm is a partitioning algorithm which is slightly changed from the K-means algorithm. They both try to limit the squared-error yet the K-medoids calculation is robust to noise than K-means calculation. In K-means calculation, they pick means as the centroids however in the K-medoids, data points are chosen to be the medoids. A medoid can be characterized as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal. The distinction between K-means and K-medoids is similar to the contrast among mean and median: where mean shows the average value of all data items gathered, while median demonstrates the incentive around that which all data items are equally distributed around it. The fundamental thought of this algorithm is to initially process the K representative objects which are called as medoids. After discovering the set of medoids, each object of the dataset is appointed to the nearest medoid. The figure 1 represent and explain the K-medoids algorithm.

Input: Dissimilarity matrix $D = \{d_{ij}\}_{m \times m}$, number of clusters k

Output: Partition $C = \{C_1, \ldots, C_K\}$.

1. Select the subset $k \subset \{1, \ldots, m\}$. Its elements are pointers to examples (prototypes)
2. **While (not** termination condition) **do**
3. Assign objects to cluster using the rule

$$C_i = \begin{cases} \arg\min d_{ij} \; if \; i \notin k \\ j \in k, \quad i = 1, \ldots, m \\ i \quad otherwise \end{cases}$$

4. Update the examples, that is

$$j_r^* = \arg min_{t: C_t = r} \sum_{t': C_{t'} = r} d_{tt'} \; r = 1, \ldots, k$$

5. **End while**
6. If the index value remained unchanged after testing m objects – Stop. Otherwise return to step 2.

Fig. 1. Explanation of the K-medoids algorithm.

The Hierarchical clustering is an algorithm that cluster similar itemsets into classes. The endpoint is a set of clusters, where each cluster is distinguished from each other, and the objects within each cluster are extensively similar.

Hierarchical clustering commences by treating each itemsets as a discrete cluster. Then, it executes repeatedly the two stages: (1) recognize the two clusters that are closest together, and (2) consolidate the two most similar clusters. This last unless all the clusters are merged together. And the result is represented as a dendrogram graph.

Methodology

Our approach exploits frequent itemsets to describe documents. However, it does not require a training phase, nor a ready database as word embedding. Frequent itemsets is extracted from each of the segments and compared. The degree of similarity between two segments is a function of the number of frequent itemsets they share. The assumption behind this approach is when words co-occur frequently within sentences that make up a text, then these words are representative of that text. Thus, considering a few frequent itemsets, it is possible to identify the specific themes covered in the documents. The proposed approach has 5 steps:

The first step is segmenting the documents to prepare them for the extraction of frequent itemsets. The documents are treated as sets of transactions where the sentences or subsections constitute the transactions, and the words are the items. The number of different words likely to appear in a set of textual documents is theoretically depending on the vocabulary size and the writing language of documents. The number of words that forms French is estimated more than 500 000 by the Quebec Office of the French Language. Considering this fact, it is possible to generate $2^{500\,000}$ itemsets from 500 000 words, it is necessary to impose certain input text conditions to control the number of words. The diversity of a lexicon increasing with the size of a text, we must limit the input texts to a few thousand words.

The second step is dedicated to the reduction of the number of items and therefore of the search space during the extraction of frequent itemsets. Some words deemed not very informative are removed from the transactions. A list of 502 stop words is used. Numbers and punctuation characters are also deleted. in addition, the lemmatization process is applied to unify the lexicon of the text, so the inflected forms of a word can be analyzed as a single item.

The third step is to extract frequent itemsets. This step is performed using the Apriori algorithm. An effort is made to identify a small number of frequent itemsets. The search for frequent itemsets is done iteratively. During the first iteration, the minimum support is set to a high value. When the number of frequently itemsets is less than 10, then the minimum support is decreased by 0.1. The process stops when the number of items in the itemsets obtained is greater than a value specified by the user (e.g. number of items = 3) or the minimum support is less than 0.1.

The fourth step is to establishes the degree of similarity between the segments. The frequent itemsets used to describe the segments are compared. The greater the number of itemsets shared by two segments, the most likely are judged similar.

The last step consists of clustering the similarity matrix using both Hierarchical method and K-medoids clustering. The outputs of those clusters are plotted to visualize the quality of the clustering and assisting the user to analyze the results.

Experimentation and discussion

In this section, we will be discussing two different experimentation writing in French language, the first experimentation is well adapted document that possesses few paragraphs, however, the second one simulates a real-world experimentation that covers a complex document.

In order to evaluate the proposed approach, we developed an application in C# capable to import documents, pre-processing them and extract itemsets using our above-mentioned methodology, then using RStudio for clustering and visualizing the results.

A well-adapted experimentation

In this experimentation, we worked on a small document consists of 22 paragraphs and covered 3 various subjects where paragraphs {1 to 4} converse on sport particularly "the biography of Michael Jordan" while paragraphs {5 to 8} cover IT subject on "Microsoft company", and the last paragraphs {9 to 22} contain only music subject such as "Wolfgang Amadeus Mozart". Using this simple data as an input in our approach with their 3 different thematics that specific to each subject provides assistance in receiving the 3 expected clusters.

During our experiments, we pre-processed the document and extract the itemsets following the steps mentioned earlier in the methodology section. Thus, we measured the discriminating power of frequent

itemsets. We compared the clustering produced when the descriptors are the frequent itemsets versus the clustering produced when the words are the descriptors.

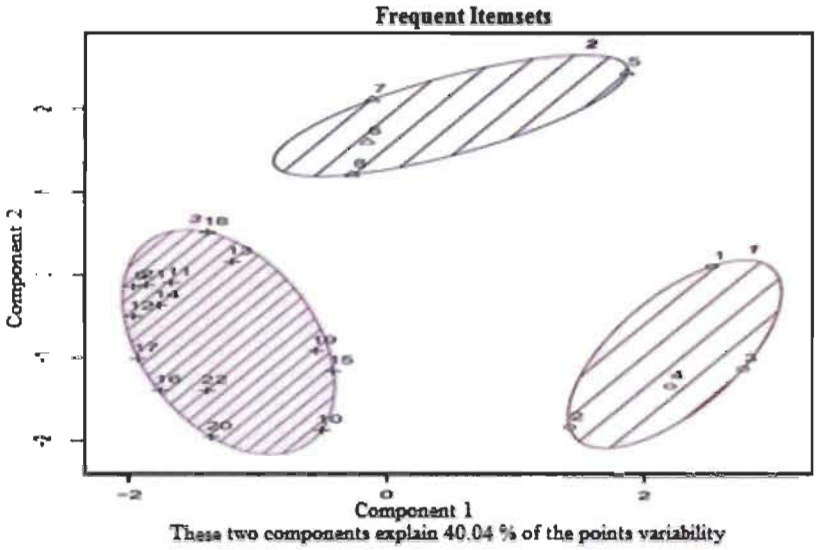

Fig. 2. K-Medoids clustering using Frequent Itemsets with K = 3.

Figure 2 illustrates the accuracy obtained by considering frequent itemsets using the K-medoid algorithm, herewith cluster 1 combines only the paragraphs {1 to 4} which covers sport, while cluster 2 combines the paragraphs {5 to 8} which is IT subject and the last cluster 3 combines the paragraphs {9 to 22} which is Music. These results are validated by the ascending hierarchical clustering (see Figure 3). We acknowledge the utilization of frequent itemsets as descriptors can be used to describe more precisely the content of this document.

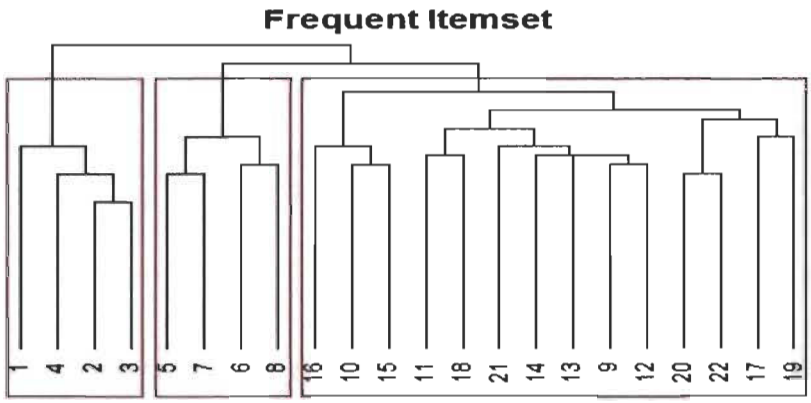

Fig. 3. Ascending hierarchical clustering using Frequent Itemsets.

On the flip side Figure 4 shows a heterogeneous clustering by considering only unique words as descriptors. We imply that various paragraphs {1,4} "Sport" and {5,6,7 and 8} "IT" are combined in the same cluster 1 along with "Music". Almost the same results are confirmed by the ascending hierarchical clustering shown in Figure 5. While using only unique words as descriptors, we notice that paragraphs dealing with subjects other than Sport are included into other paragraphs dealing with IT or Music. It should be noted when frequent itemsets are considered, the similarity classes generated converged into homogeneous clusters.

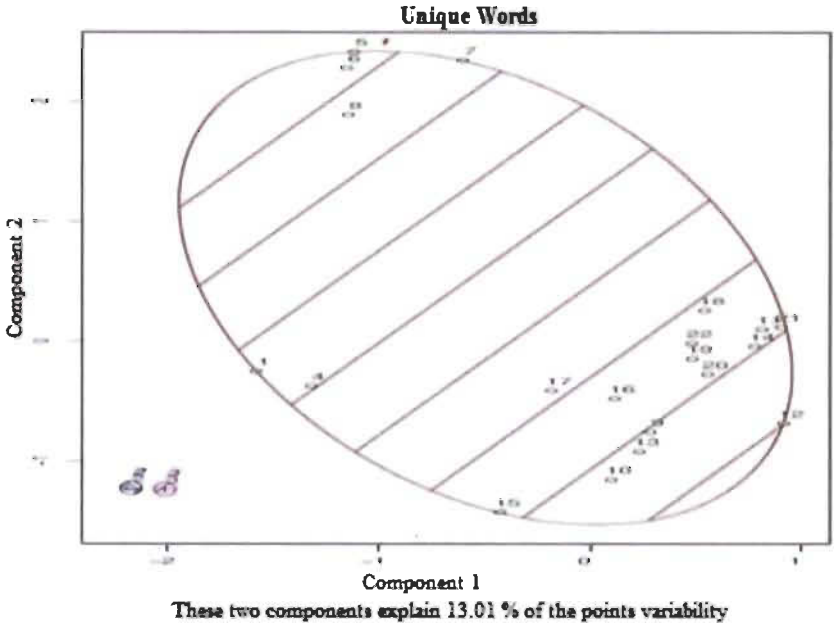

Fig. 4. K-Medoids clustering using Unique Words with K = 3.

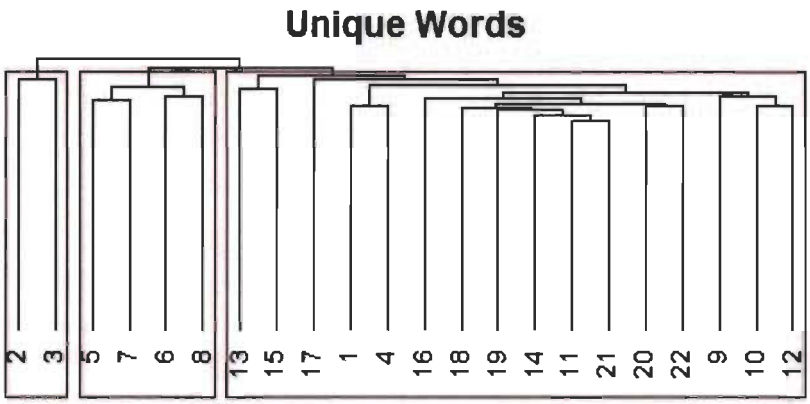

Fig. 5. Ascending hierarchical clustering using Unique Words.

Real-world experimentation

The purpose of this second experiment is to demonstrate the relevance of utilization of frequent itemsets as a descriptor of document, the experiment was performed on the book « La civilisation des Arabes » writing in French language. The document contains 6 chapters grouped into 4 parts: part 1 {chapter 1 and chapter 2} covers "L'arabie" and "Les arabes", part 2 {chapter 3} discuss "Les Arabes avant Mahomet", part 3 {chapter 4} narrate "Mahomet. Naissance de l'empire arabe", and finally part 4 {chapter 5 and chapter 6} talks about "Le Coran" and "Les conquêtes des Arabes". Using our above mention methodology, we pre-processed the document, and run various experimentations with different minimum support value ending up with satisfying result of 12% as a minimum support, the itemset extraction method gives us 43 itemsets.

The high similarity of thematic makes this document a complex experimentation, the discrimination of independent classes is not an easy process to define, even though our approach shows a huge capacity to determine homogeneous classes as shown in Figure 6.

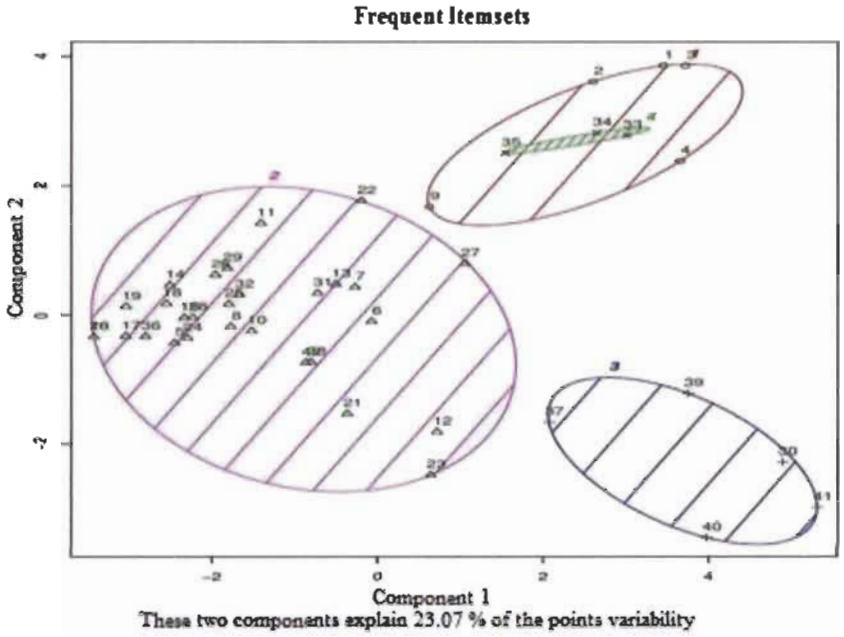

Fig. 6. K-Medoids clustering using Frequent Itemsets with K = 4.

Taking into consideration that our graphic representation is plotted in a two-dimension, actually the cluster 4 seems to be included into cluster 1, however, after checking our input data we figure out that segments of the cluster 4 are a specification of the big cluster 1 which both are treating the same subject just the cluster 1 is covering general topic and the cluster 4 more specified ideas in the same topic.

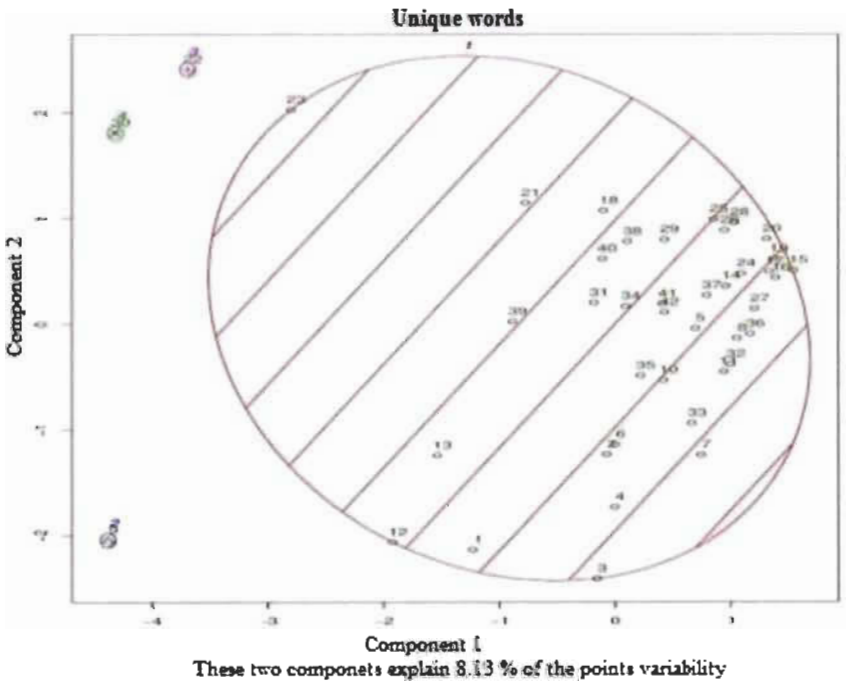

Fig. 7. K-Medoids clustering using Unique Words for relatively big document with K = 4.

The Figure 7 demonstrates the lack of clustering the data based on unique words as a descriptor, on the flip side proving the power of using frequent itemsets as a data descriptor for clustering.

Conclusion and perspectives

We proposed an unsupervised approach to establishing relationships between textual records. The proposed methodology depends on the utilization of frequent itemsets. These descriptors express the co-occurrence of words within the sentences that make up a content. Frequent itemsets tend to be more discriminating than unique words. In this way, they can improve the description of a class. One of the upsides of the proposed strategy is that the outcomes delivered are easy to interpret. The experiments carried out suggest that frequent itemsets, as defined, are sufficiently informative to be used to establish coherent links between segments. Despite the good results obtained, we note that the choice of thresholds (minimum support) remains a critical and decisive step, for this, and in a perspective of improving our approach, we propose:

Develop a function using the R package «NbClust» for a dynamic choice and precise thresholds.

Utilize our extracted itemsets with various classification and/or clustering tools to figure out whether we get the same or better improvement.

To generalize our prototype application in order to test our approach on different languages.

References

1. Agrawal, R., Imielinski T., Swami, A.: Mining association rules between sets of items in large databases, pp. 207-216. SIGMOD Conference (1993). doi: 10.1145/170036.170072
2. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules, pp. 487-499. 20th International Conference on Very Large Database (1994).
3. Alghamdi, R. A., Taileb, M., Ameen, M.: A new multimodal fusion method based on association rules mining for image retrieval. In: 17th IEEE Mediterranean Electrotechnical Conference "MELECON", pp. 493-499. IEEE Press, Beirut, Lebanon (2014). doi: 10.1109/MELCON.2014.6820584
4. Huy, T.N., Shao, H., Tong, B., Suzuki, E.: A feature-free and parameter-light multi-task clustering framework. In Knowledge and Information Systems, vol. 36, pp. 20, 17, 42. Springer, Verlag (2013). doi: https://doi.org/10.1007/s10115-012-0550-5
5. Bahri, E., et Lallich, S.: Proposition d'une méthode de classification associative adaptative. 10eme journées Francophones d'Extraction et Gestion des Connaissances, EGC 2010, vol. RNTI-E-19, pp. 501-512. EGC 2010 (2010)
6. Fournier-Viger, P., Lin, J.C.W., Vo, B., Chi, T.T., Zhang, J., Le, H.B.: A survey of itemsets mining. In Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. (2017). doi: 10.1002/widm.1207
7. Geng, L., Hamilton, H. J. : Interestingness measures for data mining A survey. In ACM Computing Surveys (CSUR), vol. 38, no 3, pp. 9. ACM, New York (2006). doi: 10.1145/1132960.1132963
8. Han, J., Pei, J.,Yin, Y. : Mining frequent patterns without candidate generation. In ACM sigmod record. Vol. 29, No. 2, pp. 1-12. ACM, New York (2000). doi: 10.1145/342009.335372
9. Le Bras, Y., Meyer, P., Lenca, P., et Lallich, S. : Mesure de la robustesse de règles d'association. QDC 2010.
10. Liu, B., Hsu, W., Ma, Y. : Integrating classification and association rule mining. In Knowledge Discovery and Data Mining, pp. 80–86. American Association for Artificial Intelligence Press, New York (1998)
11. McCallum, A., Nigam, K. : A comparison of event models for naive bayes text classification. In AAAI workshop on learning for text categorization, vol. 752, pp. 41-48. American Association for Artificial Intelligence Press, New York (1998)
12. Mittal, K., Aggarwal, G., Mahajan, P.: A comparative study of association rule mining techniques and predictive mining approaches for association classification. vol 8, no 9. International Journal of Advanced Research in Computer Science (2017). doi: 10.26483/ijarcs.v8i9.4984
13. Rompré, L, Biskri, I., Meunier, J-G. : Using Association Rules Mining for Retrieving Genre-Specific Music Files, In Proc. of FLAIRS 2017, pp. 706-711 (2017).
14. Tan, P. N., Kumar, V., Srivastava, J. : Selecting the right interestingness measure for association patterns. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 32-41. ACM, New York (2002). doi: 10.1145/775047.775053
15. Zaïane, O. R., et Antonie, M. L. : Classifying text documents by associating terms with text categories. In Australian computer Science communications. vol. 24, No. 2, pp. 215-222. IEEE Computer Society Press, Los Alamitos (2002)