République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

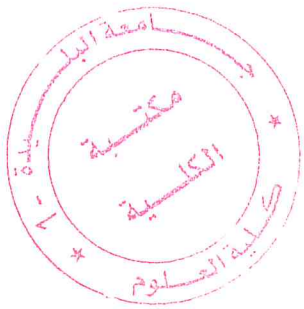
Faculté des sciences

Département d'informatique

Mémoire Présenté par :

BEDRANI houria

HADIDI fatima zohra



En vue d'obtenir le diplôme de Master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie logicielle

Thème

Extraction des itemsets contextuels fréquents

Soutenu le :

Mme FAREH MESSAOUDA	Président
Mr DERRAR HACENE	Examinateur
Mme ZAHRA FATMA ZOHRA	Promotrice

Promotion : 2017 / 2018

Remerciements

C'est avec l'aide de Dieu que ce travail a vu le jour,

Il n'aurait pu être achevé sans le soutien, les conseils et les encouragements de certaines personnes auxquelles nous tenons à exprimer ici nos sincères remerciements.

En premier lieu, nous exprimons toute notre gratitude pour notre Promotrice, Mlle. ZAHRA pour ces précieux conseils, sa disponibilité, la confiance qu'elle nous a toujours témoigné et la sollicitude dont elle nous a entouré, et ce tout au long de l'élaboration du présent travail. Nous n'oublions pas non plus nos Enseignants, qui nous ont transmis leur savoir tout au long du cycle d'études à l'UNIVERSITE DE BLIDA 1.

Nous adressons une pensée particulièrement affective à nos Amis qui ont rendu agréables nos longues années d'études.

Nous tenons enfin à remercier tous ceux qui ont collaborés de près ou de loin à l'élaboration de ce travail. Qu'ils acceptent nos humbles remerciements.

Dédicaces

Que ce travail témoigne de mes respects :

A mes parents :

Ma chère mère Hayat, pour ses sacrifices
Depuis qu'elle m'a mis au monde, qui m'a aidé surtout dans les Moments difficiles et elle m'a toujours soutenu et me guidé pour affronter les difficultés de la vie.
Qu'Allah te bénisse

A ma sœur :

Grâce à son tendre encouragement et son grand sacrifice, elle a pu créer le climat affectueux et propice à la poursuite de mes études.
Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers elle.
Je prie le bon Dieu de la bénir, de veiller sur elle, en espérant qu'elle sera toujours fiers de moi

A mon frère :

Qui est toujours été présents à mes cotés
Il va trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'il n'a cessé de me porter

A mes chéries amies :

Moufida, Amina.H, Amina.A, Asma, Dalel.....
Elles vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

A ma promotrice :

Mme. Zahra qui m'aide avec son grand encouragement pour compléter mon travail.
Enfin à toute ma famille et mes amis que j'ai n'ai pas cités, à tous ceux qui me connaissent.
Tous ceux que j'aime....

L'extraction des itemsets fréquents à partir des données constitue l'étape cœur de plusieurs méthodes de fouille de données. Cependant, la majorité des techniques d'extraction des itemsets fréquents ne prennent pas en considération le contexte de collecte des données. Par exemple, ce que le patient faisait au moment où les données ont été collectées dans le cadre de données médicales, la position et les coordonnées d'un capteur collectant des données environnementales. En effet, notre travail consiste à proposer une méthode d'extraction des itemsets contextuels fréquents. Et ce, en se basant sur des informations contextuelles préalablement collectées afin de prouver que ces dernières permettent d'augmenter la pertinence des connaissances découvertes via le processus d'extraction des itemsets fréquents.

En mettant en valeur les propriétés formelles de tels contextes, nous développons un algorithme efficace d'extraction d'itemsets contextuels fréquents. Les expérimentations effectuées sur un jeu de données ont montrés la qualité des résultats et l'efficacité de l'algorithme proposé.

Mots clés : Extraction des Itemsets Contextuels Fréquents, Informations Contextuels, Contexte.

Dédicaces

Que ce travail témoigne de mes respects :

A mes parents :

Ma chère mère KHEIRA, pour ses sacrifices

Depuis qu'elle m'a mis au monde, et qui m'a aidé surtout dans les Moments difficiles

Mon père MOHAMMED, qui m'a toujours soutenu et guidé pour affronter les difficultés de la vie,

Qu'Allah vous bénisse.

Mon mari : BRAHIM qui toujours me donner l'espoir pour résister et continuer ; je t'aime.

A mes sœurs :

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études.

Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.

Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi

A mes frères :

Qui ont toujours été présents à mes côtés

Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

A mon binôme FATIMA.Z pour sa bonne compagne et pour les meilleurs moments que nous avons passé ensemble

Mes chères amies Hayet, Kawther, Sarah.D, Fatiha, Ibtissem, Sarah.H.....

A ma promotrice Zahra qui m'aide avec son grand encouragement pour compléter mon travail.

Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

Enfin à tous mes amis que j'ai n'ai pas cités, à tous ceux qui me connaissent.

Tous ceux que j'aime....

Houria

Frequent itemsets mining from the data is the heart stage of many data mining methods. Many algorithms have been implemented to find these frequent itemsets. However, the majority of frequent itemsets extraction techniques do not consider the context of data collection. For example, what the patient was doing at when the data was collected as part of medical data, the position and coordinates of a sensor collecting environmental data. Indeed, our work consists of proposing a method for extracting frequent contextual itemsets based on the contextual information to prove that this contextual information during the frequent pattern mining makes it possible to increase the relevance of information that the extraction process can provide.

By highlighting the formal properties of these contexts, we develop an effective algorithm for extracting contextual items. Experiments performed on a dataset show the quality of the result and effectiveness of the proposed approach.

Keywords: Extraction of Frequent Contextual Itemsets, Contextual Information, Context.

ملخص

ان استخراج انماط المتكررة مهمة أساسية في العديد من المجالات استخراج البيانات. تم تنفيذ العديد من الخوارزميات للعثور على هذه الأنماط المتكررة. ومع ذلك ، غير ان معظم تقنيات استخراج مجموعات انماط المتكررة لا تأخذ في عين الاعتبار الجانب السياقي على سبيل المثال, ماذا كان يفعل المريض وعند جمع البيانات فى اطار البيانات الطبية او موقع والاحداثيات جهاز استشعار في اطار البيانات البيئية. في الواقع، يتكون عملنا من اقتراح طريقة لاستخراج أنماط سياقية متكررة تستند إلى المعلومات السياقية من أجل إثبات أن هذه المعلومات السياقية أثناء استخراج الأنماط المتكررة تجعل من الممكن زيادة اهمية البيانات التي يمكن أن توفرها عملية الاستخراج.

من خلال تسليط الضوء على الخصائص الرسمية لهذه السياقات ، نطور خوارزمية فعالة لاستخراج الأنماط السياقية. تظهر التجارب التي أجريت على مجموعة بيانات جودة النتيجة وفعالية النهج المقترح.

الكلمات المفتاحية: ، استخراج الأنماط السياقية المتكررة ، المعلومات السياقية ، السياق.

Figure 1.1 : Méthodes de data mining.....19

Figure 1.2 : L'algorithme Apriori. 22

Figure 1.3: L'algorithme générateur des candidats.....23

Figure 1.4 : L'algorithme Eclat.....26

Figure 1.5 : L'algorithme FP-Growth 27

Figure 1.6 : Structure d'une FP-tree.....31

Figure 1.7 : Construction FP-tree à partir de la 2ème transaction 32

Figure 1.8 : Etat final de la structure FP-tree 33

Figure 2.1 : Modélisation de la localisation de l'utilisateur en utilisant CC/PP. 41

Figure 2.2 : Exemple de représentation XML du contexte en utilisant l'ontologie CoOL.42

Figure 3. 1: Une hiérarchie du contexte.....47

Figure 3.2 : Contextual Information Graph (CIG).....48

Figure 3.3 : Une ontologie H.....48

Figure 3.4 : Une base contextuelle de séquences CB.....49

Figure 3.5 : Les motifs séquentiels dans des contextes minimaux de base contextuelle de la base contextuelle de séquences.....50

Figure 3.6 : Etapes d'extraction des motifs contextuels 51

Figure 3.7 : Exemple d'une base de connaissances $KB = (\mathcal{F}, \mathcal{H})$.....53

Figure 3.8 : Une base de données transactionnelle $\mathcal{T}_{KB,Animal}$ pour le contexte animal dans la base de connaissance KB représenté dans Figure 3.7.....54

Figure 4.1 : Algorithme Spécifique Contextuel –Apriori (SC_Apriori) 60

Figure 4.2 : Schéma global de l'approche proposé.....63

Figure 4.3 : Exemple sur la méthode du coude.....64

Figure 4.4 : Résultat en utilisant la méthode du coude. 65

Figure 4.5 : La représentation graphique des résultats.....68

Figure 4.6 : Prendre un jeu de données à 2 dimensions et le séparer en 3 groupes distincts.....69

Figure 5.1 : Temps d'exécution par rapport au nombre du contexte. 73

Figure 5.2 : temps d'exécution par rapport aux différents supports. 74

Figure 5.3 : Consommation de mémoire par rapport au nombre de contexte.....75

Figure 5.4 : Consommation de mémoire par rapport au seuil minimal 76

Figure 5.5 : les achats des clients et l'information contextuelle 76

Figure 5.6 : regroupement des clients par rapport aux contextes.....77

Figure 5.7 : Nombre des itemsets fréquents.....79

Figure 5.8 : Comparaison des résultats de context2 (tableau droite) avec context3 (tableau gauche) 80

Figure 5.9 : Temps d'exécution de SC_Apriori 81

Figure 5.10 : Consommation de mémoire SC_Apriori 82

Tableau 1. 1 : Une Base de données de 4 items..... 24

Tableau 1. 2 : Base de transactions..... 29

Tableau 1. 3 : Base transactions count..... 29

Tableau 1. 4 : Base transactions items ordonnes 30

Tableau 1. 5 : La comparaison entre les types des algorithmes..... 34

Tableau 2.1 : Vue de comparaison des modèles existantes de modélisation du contexte

.....43

Tableau 3.1 : Exemple de jeu de données cibles de la criminalité 51

Tableau 3.2 : Exemple de jeu de données cibles de la criminalité après l'expansion ... 52

Tableau 3.3 : La comparaison entre les approches 55

Tableau 4.1 : Une base contextuelle d'itemsets..... 58

Tableau 4.2 : Le score moyen de silhouette pour chaque nombre de clusters..... 66

ECD	Extraction de Connaissances à partir de Données
Apriori-Gen	Apriori_Générator
DHP	Direct Hashing and Pruning
FP-growth	Frequent Pattern growth
FP-Tree	Frequent Pattern tree
PrefixSpan	Prefix-projected Sequential pattern
UML	Unified Modeling Language
XML	eXtensible Markup Language
DTD	Document Type Definition
CC/PP	Composit Capability/Prefrence Profile
RDF	Resource Description Framework
CML	Context Modeling Language
OWL	Web Ontology Language
CONON	Context Ontology
CIG	Contextual Information Graph
CB	Base Contextuelle
SPM	Sequential Pattern Mining
GPM	Generalized Pattern Mining
CFP	Contextual Frequent Patterns
SC_Apriori	Spécifique Contextuel –Apriori
MinSup	Minimum Support
SSE	Sum of Squared Errors

Introduction générale... 15

Chapitre 1 L'extraction des itemset fréquents.....18

I. Introduction 18
II. Définition et objectifs du Data Mining.....18
III. Extraction des motifs fréquents 19
3.1 Définitions.....20
3.2 Les Algorithmes d'extractions des itemsets fréquents.....21
3.2.1 Les Algorithme de type « Tester-et-générer » 21
3.2.2. Les algorithmes de type «Diviser-pour-régner » 27
IV. Comparaison entre les types des algorithmes.....35
V. Conclusion.....35

Chapitre 2 Modélisation du contexte.....36

I. Introduction 37
II. Notion du contexte.....37
2.1. Définition du contexte.....37
2.2. Catégorisation du contexte.....38
2.3. Acquisition des informations du contexte.....38
III. Méthodes de modélisation de contexte.....39
3.1. Modèle clé-valeur 39
3.2. Modélisation orientée objet.....39
3.2.1 Modèle basé sur UML (Unified Modeling Language) 40
3.2.2. Modèle basé sur un langage de balise 41
3.2.2. Modèle basé sur CML (Context Modeling Language).....41
3.3 Modèles basés sur les ontologies 41
IV. Comparaison sur les modèles de représentation du contexte 43
V. Conclusion.....44

Chapitre 3 Extraction des motifs contextuels fréquents 45

I. Introduction 46

II. Approches d'extraction des motifs contextuels 46

2.1 Représentation du contexte 46

2.2.1 Hiérarchie du contexte 46

2.2.2 Contextual Information Graph (CIG) 47

2.2.2 Ontologie 48

2.2 Extraction des motifs contextuels fréquents 48

2.2.1 Extraction de motifs séquentiels contextuels 48

2.2.2 Extraction des règles d'association contextuelles 51

2.2.3 Extraction des graphes contextuels 52

2.2.4 Extraction des itemsets contextuels 53

III. Etude comparative 54

IV. Conclusion 56

Chapitre 4 Approche Proposée 57

I. Introduction 58

II. Itemsets contextuels fréquents 58

2.1 Base Contextuelle d'itemsets 58

III. Approche proposée 59

3.1 Extraction des itemsets contextuels spécifiques fréquents 60

3.2.1 Description de l'algorithme 60

3.2.2 Pseudo algorithme 60

3.2.3 Fonctionnement de l'algorithme SC_Apriori 60

3.2 Extraction des itemsets contextuels fréquents 62

3.2.1 Phase 1 : Extraction des contextes pertinents 63

3.2.1.1. Nombre de cluster k 63

3.2.1.2. Algorithme K-means 68

3.2.2. Phase 2 : Extraction des itemsets contextuels fréquents 70

IV. Conclusion 70

Chapitre 5 Tests & Validation 71

I. Introduction 72

II. Environnement de développement 72

2.1. L'environnement matériel 72

2.2. L'environnement logiciel 72

2.2.1 Python 72

2.2.2 Jupiter Notebook 73

2.2.3 Bibliothèques utilisées 73

III. Expérimentation et tests 73

3.1 Description des données 73

3.2 Extraction des itemsets fréquents spécifique 74

3.2.1 Tests le temps d'exécution 74

3.2.1.1. Par rapport au nombre du contexte 74

3.2.1.2. Par rapport au seuil minimal 75

3.2.2 Tests consommation du mémoire 75

3.2.2.1. Par rapport au nombre du contexte 76

3.2.1.2. Par rapport au seuil minimal 76

3.3 Extraction des itemsets contextuels fréquents 79

3.3.1 Extraction des contextes pertinents 79

3.3.2 La qualité des résultats 79

3.3.3 Extraction des itemsets contextuel 80

3.4 Extraction des itemsets spécifique fréquents en utilisant des contexte pertnient 80

3.4.1 Temps d'exécution 80

3.4.2 Consommation du mémoire 81

3.4.3 Discussion 82

IV. Conclusion 83

Conclusion générale et Perspectives **84**

I Conclusion 85

II Perspectives 85

Bibliographie **87**

Annexes **93**

Introduction Générale

I. Contexte et problématique

Les motifs fréquents sont une forme de connaissances extraites à partir des données. Leur but est de fournir à l'utilisateur des informations non triviales, implicites, présumées non connues et potentiellement utiles. Ils offrent ainsi à l'utilisateur une meilleure appréhension des données. Les motifs fréquents sont des itemsets, des sous-séquences, des sous arbres ou des sous-graphes apparaissant dans un jeu de données et vérifiant un seuil de support minimum fixé par l'utilisateur. Par exemple, un ensemble d'items, tels que lait et pain, qui apparaissent fréquemment ensembles dans des transactions d'un jeu de données est un itemset. Une sous-séquence, telle que acheter d'abord un PC puis un appareil photo numérique et ensuite une carte mémoire, qui apparaît fréquemment dans une base de transactions est un motif séquentiel.

La découverte de motifs joue un rôle clé dans la recherche d'associations, de corrélations et d'autres relations entre les données. De plus, la découverte de motifs peut aider l'indexation des données, la classification, le clustering et d'autres techniques de fouille de données. L'extraction de motifs fréquents est ainsi devenue une tâche importante de la fouille de données et un thème très étudié dans la communauté.

Néanmoins, la majorité des techniques d'extraction des motifs fréquents ne prennent pas en considération les informations contextuelles importantes. Par exemple, ce que le patient faisait au moment où les données ont été collectées dans le cadre de données médicales, la position et les coordonnées d'un capteur collectant des données environnementales. Ces éléments contextuels de données peuvent augmenter l'utilité des motifs extraits.

La question qui se pose, est " Comment intègre ces informations contextuels dans le processus d'extraction des connaissances de façon automatique?", plus précisément dans le cadre de l'extraction des motifs fréquents.

II. Objectifs

Dans ce travail on s'intéresse à l'extension du concept d'itemsets fréquents à les itemsets contextuels fréquents qui intègre le contexte. Par conséquent, l'objectif de ce travail est la proposition d'une méthode d'extraction des itemsets fréquents qui prend en considération les informations contextuelles lors de l'extraction des motifs fréquents. Cela

permet d'affiner et d'augmenter la pertinence des d'informations que le processus d'extraction peut fournir.

III. Organisation du mémoire

Le mémoire se partage en cinq chapitres

Chapitre 1 : « Extraction des itemsets fréquents »

Ce chapitre est consacré à la présentation des algorithmes les plus connue pour l'extraction des items fréquents.

Chapitre 2 : « Modélisation du contexte»

Dans ce chapitre, nous présenterons les méthodes existent pour modéliser les données contextuelles

Chapitre 3 : « Extraction des motifs contextuels fréquents »

Dans ce chapitre, nous présenterons les travaux et techniques passé qui se rapproches le plus de notre objectif

Chapitre 4 : « Approche proposée »

Ce chapitre, sera réservé pour présenter notre solution proposée et évaluation de notre application.

Chapitre 5 : « Test et Validation »

Le dernier chapitre concrétise et valide la méthode adoptée.

Chapitre 1

L'extraction des itemset fréquents

I. Introduction

L'extraction des itemsets fréquents est une étape intermédiaire, car les résultats de cette technique sont utilisés comme une entrée pour d'autres méthodes de découverte de connaissances. Néanmoins, ces résultats peuvent être aussi directement exploités et interprétés. Dans ce chapitre, nous allons présenter les algorithmes les plus connus dans la littérature pour l'extraction des itemsets fréquents.

Ce chapitre est organisé d'une manière suivante, dans la section 2, nous définissions c'est quoi le Data Mining et ses objectifs. Les algorithmes d'extraction des itemsets fréquents dans la section 3. Dans la section 4, nous avons fait une étude comparative des algorithmes présente dans la section précédant. Enfin, nous terminons notre chapitre par une petite conclusion.

II. Définition et objectifs du Data Mining

Le data mining, ou fouille de données, est l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse de bases de données informatiques (souvent grandes), de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des associations, des tendances inconnues ou cachées, des structures particulières restituant L'essentiel de l'information utile tout en réduisant la quantité de données [01].

La définition la plus communément admise de Data Mining est celle de [02]: « Le Data Mining est un processus non trivial qui consiste à identifier, dans des données, des schémas nouveaux, valides, potentiellement utiles et surtout compréhensibles et utilisables ».

En bref, la fouille de données est une étape centrale du processus d'ECD. Elle correspond à l'ensemble des méthodes et des techniques issues de spécialités différentes : statistiques, algorithmes génétiques, réseaux de neurones. Qui à partir de données permettent d'obtenir des connaissances exploitables. Les objectifs de la fouille de données peuvent être regroupés en cinq fonctions: classification, estimation, prédiction, optimisation, segmentation et explication (voir figure I.1).

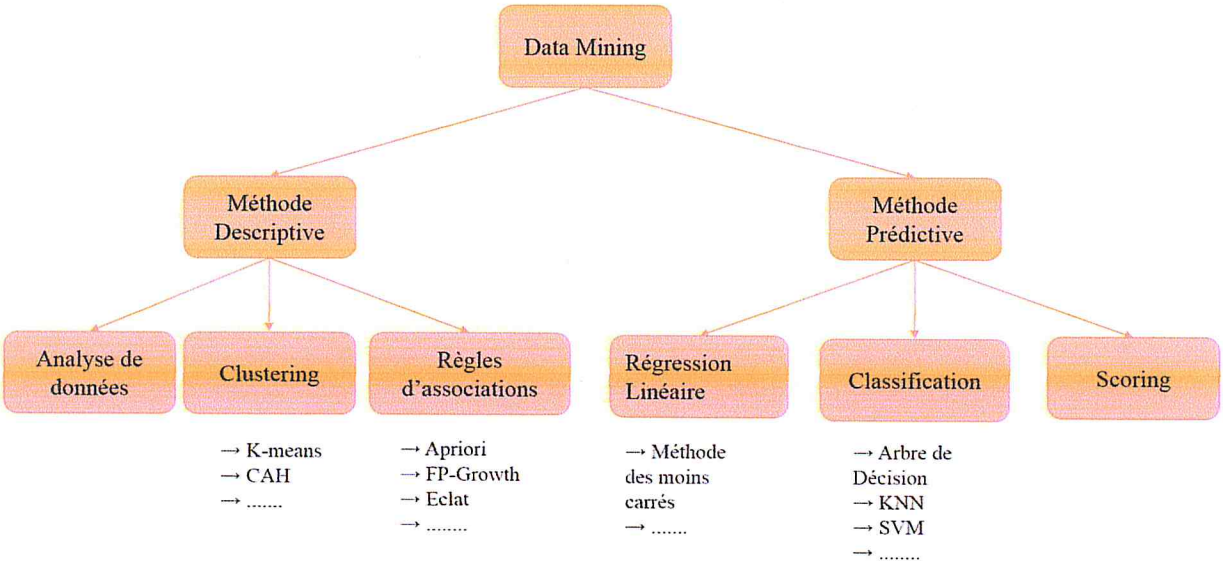


Figure 1.1: Méthodes de data mining.

Parmi les méthodes de data mining, nous trouverons les règles d'associations et l'extraction des motifs fréquents.

III. Extraction des motifs fréquents

Les motifs fréquents [03] sont des ensembles d'items ou des sous séquences ou des sous structure qui apparaissent fréquemment dans un ensemble de donnée. Par exemple :

- Un ensemble d'items tel que le lait et le pain qui apparaissent souvent dans une base de transactions dans un supermarché, est un ensemble des itemsets fréquents.
- Une sous séquence telle qu'acheter premièrement un PC puis une caméra numérique ensuite une carte mémoire qui se produit souvent dans la base historique des achats, est une séquence d'itemsets fréquents.
- Les sous structures peuvent être des sous-graphes ou des sous arbres qui peuvent être combinés avec des ensembles ou des séquences d'items.

1) Définitions

- **Base de données :**

Soit O un ensemble fini d'objets, P un ensemble fini d'éléments ou items, et R une relation binaire entre ces deux ensembles. On appelle base de données ou contexte formel le triplet D = (O, P, R). La base de données D représente l'espace de travail [3].

- **Transaction :** Soit $T = \{T1, T2, T3, \dots, Tn\}, Ti \subseteq D$. On appelle Ti un ensemble de lignes contenant les occurrences de la base de données « D ». Tous les Ti sont appelés des transactions. L'exemple, du panier de la ménagère, les transactions sont les tickets de caisse (c'est-à-dire les achats effectués par les clients).
- **Motif** : c'est un ensemble d'items, séquence et arbres qui interviennent fréquemment ensemble dans une base de données. [03]
- **Item :** On appelle item toute variable Xi représentant une occurrence de D. [03]
- **Itemsets :** On appelle itemset, l'ensemble formé d'items. Exemple le singleton {X1} et la paire {X1, X2} sont des itemsets. Un itemset de taille k est noté k-itemset [03]
- **Une règle d'association :** Une règle d'association est constituée de deux Item I disjoints non vides liées par une relation de causalité I_1 et I_2, tel que I_1 est appelée la prémisse de la règle et I_2 le conséquent de la règle.
- **Séquence :** c'est une liste ordonné des éléments $S = < s_1, \dots, s_l >$, ou $l > 0$ et $s_i \subseteq I$ est un itemset pour tous $1 \leq i \leq l$.
- **Sous-séquence** : une séquence $S' = < s'_1, \dots, s'_l >$ est une sous-séquence de $S = < s_1, \dots, s_l >$, notée $S' \leq S$, si $\exists\ i_1 < i_2 < i_j < i_n$ tels que $s'_{i1} \subseteq s_{i1}, s'_{i2} \subseteq s_{i2}, \dots\dots, s'_{in} \subseteq s_{in}$.
- **Support :** On appelle support le pourcentage de T_i où apparait la règle d'association, comme présente dans l'équation (1).

$$\text{Support}(Xi \rightarrow Xj) = \frac{\text{Freq}(Xi \cup Xj)}{\text{Card}(T)} \dots\dots\dots\dots\dots (1)$$

Où Support $(Xi \rightarrow Xj)$ = Nombre de fois où X*i* et Xj apparaissent ensemble dans les transactions T.

- **Confiance** : On appelle confiance le pourcentage de fois où la règle est vérifiée, comme présente dans l'équation (2).

$$\text{Confiance}(Xi \rightarrow Xj) = \frac{\text{Freq}(Xi \cup Xj)}{\text{Freq}(Xi)} \dots\dots\dots\dots\dots (2)$$

2) Les Algorithmes d'extractions des itemsets fréquents

Dans cette section, nous allons présenter les lignes directrices des algorithmes les plus importants d'extraction des itemsets fréquents.

2.1. Les Algorithme de type « Tester-et-générer »

Les algorithmes reposant sur cette technique parcourent en largeur l'espace de recherche par niveau et considèrent un ensemble de motifs d'une taille donnée lors de chaque itération. A chaque niveau k, un ensemble de candidats de taille k est généré et les motifs fréquents sont retenus pour en générer d'autres au niveau suivant par jointure. Les supports des motifs candidats sont calcules et les candidats qui ont le support inferieur à minSupp sont élagués. Cet élagage est justifié par la propriété d'anti-monotonie du support.

Nous présentons dans cette section l'algorithme Apriori qui a été le premier algorithme par niveau propose pour l'extraction de règles d'association et qui constitue le principe sur lequel sont basées nos approches, ainsi que quelques variantes.

a) Algorithme Apriori

L'algorithme Apriori est un algorithme d'exploration de données conçu en 1994, par Rakesh Agrawal et Ramakrishna Sikrant [3] dans le domaine de l'apprentissage des règles d'association. Il sert à reconnaître des propriétés qui reviennent fréquemment dans un ensemble de données et d'en déduire une catégorisation.

Apriori détermine les règles d'association présentes dans un jeu de données, pour un seuil de support et un seuil de confiance fixés. Ces deux valeurs peuvent être fixées arbitrairement par l'utilisateur.

Un jeu de données, on dispose de Xi éléments (ex : lignes dans une table de données) connus aussi sous le nom de Transactions T. Chaque élément est décrit par un ensemble d'attributs Atti (ex : colonnes dans une ligne de données). Un attribut correspond aussi à un item. Un ensemble d'items est dit fréquent s'il correspond à un itemsets fréquent dans la base de transactions.

- **Fonctionnement de l'algorithme Apriori :**

L'algorithme Apriori est présent dans la Figure 1.1 en utilisant les notations suivantes :

- Ck : ensemble des k-itemset candidats dont on ne connait pas encore le support ;

- Fk : ensemble des k-itemset fréquents de taille k.

Les motifs fréquents sont calculés de façon itérative, dans l'ordre ascendant suivant leur taille. A chaque itération, la base de données est parcourue une fois et tous les motifs fréquents de taille k sont génères. La ligne 1 trouve tous les 1-itemset fréquents. L'algorithme alterne ensuite la génération des candidats et sélectionne parmi eux ceux étant fréquents dans m

*les lignes 3 à 15 : à l'itération k, l'ensemble Fk-1 des (k -1) - Itemset fréquents correspondant aux motifs de niveau (k -1) est utilisé pour générer l'ensemble Ck des k-motifs candidats.

Algorithm 1 Apriori ($\mathcal{DB}$, $minSupp$)
Entrées : $\mathcal{DB}$ base de données transactionnelle, $minSupp$ seuil du support minimum
Sortie : F ensemble de tous les motifs fréquents de $\mathcal{DB}$

1: $F_1 \leftarrow \{1\text{-motif fréquent}\}$
2: $k = 2$
3: while $F_{k-1} \neq \emptyset$ do
4: *//génération des candidats, voir algorithme 2*
5: $C_k \leftarrow Apriori - Gen(F_{k-1})$
6: *//calcul du support des candidats*
7: for $t \in \mathcal{DB}$ do
8: for all $c \in C_k$ do
9: if c est contenu dans t then
10: $count(c) + +$
11: *//incrémentation du nombre d'occurrence de c avec le compteur count*
12: end if
13: end for
14: *//sélection des motifs vérifiant la contrainte de support minSupp*
15: $F_k \leftarrow \{c \in C_k / supp(c) = \frac{count(c)}{|\mathcal{DB}|} \geq minSupp\}$
16: end for
17: $k = k + 1$
18: end while
19: $F \leftarrow \bigcup_k F_k$

Figure 1.2 : L'algorithme Apriori [3]

La procédure Apriori-Gen appelée en ligne 5 est présentée dans la Figure 1.3. Elle prend Fk-1 en entrée et génère Ck comme résultat. L'initialisation de Ck à l'ensemble vide est faite en ligne 1. Ensuite, une jointure est effectuée entre les éléments de Fk1 (lignes 2 à 6). Deux Itemset p et q de Fk-1 forment un motif c si et seulement s'ils ont (k-2) attributs (dans le

préfixe) en commun, ce qui est exprimée en utilisant l'ordre lexicographique[1] dans la condition de la ligne 4 de l'algorithme Apriori-Gen. Les étapes suivantes (lignes 7 à 11) assurent, après avoir génère un candidat de taille k à partir de deux (k -1) itemset fréquents, que tous les sous-ensembles du nouveau candidat sont fréquents.

Une fois que l'ensemble Ck des motifs candidats a été calculé, la base de transactions est parcourue afin de calculer le support de chaque candidat. Ainsi, parmi les candidats de Ck, ceux qui sont contenus dans la transaction t voient leur nombre d'occurrences incrémente dans la ligne 10. Par la suite, seuls ceux qui ont un support supérieur à minsup sont retenus.

Algorithm 2 Apriori-Gen ($F_k - 1$)

Entrée : $F_k - 1$
Sortie : C_k

1: $C_k = \emptyset$
2: for all $p \in F_{k-1}$ do
3: for all $q \in F_{k-1}$ do
4: if $p(1) = q(1), p(2) = q(2), \ldots, p(k-2) = q(k-2), p(k-1) < q(k-1)$ then
5: $c \leftarrow p \cup q(k-1)$
6: $C_k \leftarrow C_k \cup \{c\}$
7: end if
8: for all $s \subseteq c$ (avec s un $(k-1)$-motif) do
9: if $s \notin F_k - 1$ then
10: remove c from C_k
11: end if
12: end for
13: end for
14: end for
15: Retourner C_k

Figure 1.3 : L'algorithme générateur des candidats [3].

b) L'algorithme Eclat

Eclat [4] étudie des ensembles de transactions par classe d'équivalence d'items. On parle ici de placements respectivement horizontal et vertical de la base.

Cet algorithme repose sur le découpage de la base en classes d'équivalences et distribution de la charge de travail sur tous les processeurs. On considère que deux itemsets (ensemble d'items) sont dans la même classe d'équivalence s'ils désignent par les items qu'ils contiennent dans l'ordre lexicographique, possédant un préfixe commun. Par exemple, les

[1] Un ordre lexicographique est une relation d'ordre sur t_k, ou t est un ensemble totalement ordonne et k un entier. La relation d'ordre est définie de la façon suivante : $(x_1, x_2, .., x_k) \leq (y1, y2, \ldots, y_k)$, si et seulement s'il existe i tel que pour tout $j < i$, $x_j = y_j$ et $x_i < y_i$.

itemsets MERE et MEROIR sont dans la classe d'équivalence MER. Au lieu de transmettre des supports locaux ou des portions de base de données comme dans les principaux algorithmes dérivés d'Apriori.

Cet algorithme fonctionne en transmettant les listes de transactions correspondant à chaque classe d'équivalence au processeur qui s'occupe de celle-ci [4].

➢ **Etapes de l'algorithme**

- **Etape 01 (Phase d'initialisation) :** L'algorithme commence par scanner la base afin de construire les itemsets fréquents de *taille* 2. En effet, il est possible de générer ces ensembles avec peu de coût supplémentaire par rapport à la génération des itemsets fréquents de *taille* 1, profitant ainsi du gain obtenu en évitant de scanner la base 2 fois. Cependant, l'auteur de l'algorithme précise que si la base de données contient un grand nombre d'items, il est peut-être préférable de scanner la base deux fois afin d'éliminer les items infréquentés avant de générer les itemsets de taille 2 [4]. A ce stade, on dispose de l'ensemble des itemsets fréquents L3.

Exemple

Pour une base contenant 4 items, on rencontre les itemsets *AB, AC, BD* et *CD* chacun respectivement dans au moins une transaction de la base. Chaque processeur calcule le support local des itemsets puis effectue une réduction de somme des résultats des autres processeurs afin de construire les supports globaux de chaque itemsets de L2 [4].

Tableau 1.1 : Une Base de données de 4 items [4]

Items	A	B	C	D
A	_	_	_	_
B	**1**	_	_	_
C	**1**	**0**	_	_
D	**0**	**1**	**1**	_

- **Etape 02 (La phase de transformation)** : L'algorithme commence par partitionner L2 en classes d'équivalences qui seront redistribuées sur les processeurs avec une politique d'équilibrage de charge basée sur une heuristique. On effectue alors une transformation de la

base afin d'obtenir, non plus une liste d'items par transaction mais une liste de transactions par item (transformation verticale de la base).

- Partitionnement de L2 en classes d'équivalences.
- Calcul de la charge de travail pour chaque classe d'équivalence.

La mesure est effectuée en fonction du nombre d'éléments de classe d'équivalence. On considère toutes les paires à traiter par la suite. Ainsi, la charge est calculée par la valeur $C^2{}_8$. Par exemple, si dans la classe d'équivalence [A] on trouve les itemsets AB, AC et AD, la charge calculée sera $C_3^2 = 3$. Il s'agit alors de répartir les tâches à effectuer sur les processeurs en fonction d'une heuristique sur les charges calculées : On assigne toutes les classes d'équivalences par charge décroissantes sur le processeur qui a la charge la plus petite au moment de l'assignation.

Chaque processeur peut effectuer cette tâche indépendamment des autres puisqu'ils disposent alors de tous les supports globaux de L2. Phase de transformation verticale de la base. Chaque processeur scanne sa portion locale de la base afin de construire les listes de transactions correspondant aux classes d'équivalence de L2. Il faut alors transmettre respectivement les listes aux processeurs chargés de la classe d'équivalence correspondante. [4]

- **Etape 03 (La phase asynchrone):** Les processeurs effectuent concurremment la construction des itemsets de tailles croissantes par intersection des listes de transactions des éléments de chaque classe d'équivalence entre eux. Eliminant les itemsets de support insuffisant, on réduit rapidement le travail à effectuer en même temps qu'on augmente la taille des itemsets construits. C'est du moins ce qu'il se passe sur des données réelles.
- **Etape 04 (La phase de réduction finale) :** La dernière tâche de l'algorithme consiste en l'accumulation et la réunion des résultats de chaque processeur.

➢ **Récapitulatif de l'algorithme Eclat**

Algorithme : Eclat

Phase d'initialisation

- Scanne la partition locale de la base.
- Calcul des comptes locaux des itemsets de taille 2.
- Construction des comptes globaux de L_2

Phase de transformation

- Partitionnement de L_2 en classes d'équivalences.
- Distribution de L_2 sur les processeurs par classe d'équivalence.
- Transformation de la base locale en base verticale.
- Envoie des listes de transactions aux autres processeurs.
- L_2 **local** = listes des transactions des autres processeurs.

Phase asynchrone

- **Pour** chaque classe d'équivalence E_2 dans L_2 **local** *Construction*(E_2)

Phase final de réduction

- Regroupement des résultats et calcule des associations

Figure 1.4 : L'algorithme Eclat [4]

c) Variantes

Nous trouvons dans la littérature de nombreux algorithmes bases sur cette technique, permettant de générer tous les itemsets fréquents d'une base transactionnelle. Ces algorithmes peuvent être classes en trois approches principales. La première consiste à parcourir itérativement par niveau l'ensemble des itemsets. Cette approche inclut donc l'algorithme Apriori ainsi que, par exemple, AprioriTid [3], Partition [5], Sampling [6] ou l'algorithme DHP (Direct Hashing and Pruning) [7].

La seconde est basée sur l'extraction des itemsets fréquents maximaux. Parmi les algorithmes les plus efficaces bases sur cette approche, on peut citer Max-Miner [8], Pincer-Search [9], MaxClique et MaxEclat [10]. La dernière est basée sur l'extraction d'itemsets fréquents fermes ou un itemsets ferme est un ensemble maximal d'items communs à un ensemble d'objets. En ce qui concerne cette dernière approche, on peut citer Close et A-Close [11] qui sont aussi des algorithmes par niveau, Titanic [12] et Charm [13].

2.2. Les algorithmes de type «Diviser-pour-régner »

Les algorithmes essaient de diviser le contexte d'extraction en des sous-contextes et d'appliquer le processus de découverte des itemsets fréquent récursivement sur ces sous-contextes. Ce processus de découverte repose sur un élagage du contexte basé essentiellement sur l'imposition d'une métrique statistique et d'heuristiques introduites. L'exemple principal dans cette catégorie est l'algorithme FP-Growth (Frequent-Pattern Growth)

a) Algorithme FP-Growth

L'algorithme FP-Growth (Fréquent Pattern growth) consiste d'abord à compresser la base de données en une structure compacte appelée FP-Tree (Fréquent Pattern tree), puis à diviser la base de données ainsi compressée en sous projections de la base de données appelées bases conditionnelles. Chacune de ces projections est associée à un item fréquent. L'extraction des itemsets fréquents se fera sur chacune des projections séparément [14].

L'algorithme FP-Growth apporte ainsi une solution au problème de la fouille des itemset fréquents dans une grande base de données transactionnelle. En stockant l'ensemble des éléments fréquents de la base de transactions dans une structure compacte, on supprimer la nécessité de devoir scanner de façon répétée la base de transactions. De plus, en triant les éléments dans la structure compacte, on accélère la recherche des itemset.

Les différentes étapes de l'algorithme Fp-growth peuvent être résumées comme indiqué dans cette illustration.

➢ **Récapitulatif de l'algorithme Fp-growth**

Algorithme FP-growth

1. Balayer la base de transactions T une première fois
   - Créer L, la liste des items fréquents avec leur support
   - Trier L en ordre décroissant du support
2. Créer l'arbre N contenant une racine étiquetée « Null »
3. Procédure FP-growth(Fp-tree, *null*)
   A. Si FP-tree contient un seul chemin *P* alors
      - Pour chaque combinaison β de *P* faire
      - Générer l'itemset β U α de support = minimum des supports des nœuds de β
   B. Sinon pour chaque a_i dans l'indexe de FP-tree faire
      - Générer l'itemset $\beta = a_i$ U α de support = *ai.support*
      - Construire l'itemset condition de base de β
      - Construire FP-tree$_\beta$
      - Si FP-tree$_\beta \neq 0$
        - FP-growth(FP-tree$_\beta$, β)

Figure 1.5 : L'algorithme FP-Growth [14]

➢ **Structure d'une FP-Tree**

La structure d'un FP-Tree constitue deux éléments essentiels sont :

- **Une structure :** est sous forme d'un arbre avec une racine étiquetée **nulle** et d'un ensemble de nœuds préfixé par l'élément représente. Un nœud de l'arbre est composé par :
  - Nom-item : il s'agit de l'item que représente le nœud.
  - Count : le nombre d'occurrence de transaction où figure la portion de chemin jusqu'à ce nœud.
  - Node-link : il s'agit d'un lien inter-nœud vers les autres occurrences du même élément (ayant le même nom-item) figurant dans d'autres séquences de transactions. Cette valeur prend la valeur **nulle** s'il n'y a pas un tel nœud.
- **Un index :** est une table d'en-tête qui contient la liste des items fréquents et qui pointe sur la première occurrence de chaque élément. Chaque entrée dans cette table contient :
  - Nom-item : Le nom de l'élément.
  - Le pointeur tête de la séquence des nœuds ayant ce même nom-item.

➢ **Etapes de construction d'une FP-Tree**

La construction d'une structure FP-tree passe par 6 étapes principales. Les étapes 1 à 5 prépare la structure et insère les éléments qui doivent s'y trouver. La sixième étape consiste à valider les informations insérées dans les étapes précédentes :

- Calculer le support minimum
- Parcours de la base des transactions pour trouver la somme totale des différentes occurrences.
- Définir la priorité des éléments, puis trier les items en fonction de leur priorité.
- Création du nœud racine.
- Insertion des nœuds enfants.
- Validation

Exemple explicative de l'algorithme

- **Etape 1**

Considérons la base de transactions suivante. Supposons que le support minimum est défini à 50%.

Tableau 1.2 : Base de transactions [15]

TID	Items
1	f, a, c, d, g, i, m, p
2	a, b, c, f, l, m, o
3	b, f, h, j, o
4	b, c, k, s, p
5	a, f, c, e, l, p, m, n

Dans notre cas, pour obtenir un support de 50%, il faut le calculer ainsi :

$$Support\ minimum = (50/100 * 5 = 3)$$

Le résultat obtenu, **3** dans notre cas, constitue le support minimum et par conséquent tous les items da la base de transactions ayant un support inférieur à **3**occurrences minimums sera ignoré.

- **Etape 2**

Dans cette étape nous allons parcourir la base de transactions afin de calculer les fréquences des éléments qui s'y trouvent. Par la suite, une fois les différentes fréquences obtenues, seuls les éléments dont la fréquence est supérieure au support minimum défini dans l'étape 1 seront retenus, les autres seront ignorés. Dans notre cas, le tableau suivant représente les items retenus ainsi que leurs nombre d'occurrence respectif.

Tableau 1.3 : Base transactions count [15].

Item	Fréquence
f	4
a	3
c	4
b	3
m	3
p	3

La table ainsi obtenue constitue la table des entêtes (Headers Table) appelée aussi la table des pointeurs.

- **Etape 3**

Cette étape consiste à ordonner les différents éléments en fonction de leur poids. Il s'agit de les trier en fonction de leur nombre d'occurrences. Ce tri s'effectue en ordre décroissant, l'élément ayant comptabilisé le plus grand nombre d'occurrences est placé en tête et l'élément ayant comptabilisé le moins d'occurrences est placé en queue. Ce traitement sera effectué pour chacune des lignes de transactions contenues dans la base des transactions

Tableau 1.4 : Base transactions items ordonnes [15]

TID	Items	Items Fréquents Ordonnés
1	f, a, c, d, g, i, m, p	f, c, a, m, p
2	a, b, c, f, l, m, o	f, c, a, b, m
3	b, f, h, j, o	f, b
4	b, c, k, s, p	c, b, p
5	a, f, c, e, l, p, m, n	f, c, a, m, p

Dans notre cas, l'élément f ayant un support de 4 est placé en tête, l'élément p quant à lui à un support de 3 est par conséquent il se retrouve en dernière position. A la fin de cette étape, on peut considérer que tout est prêt pour commencer la construction la structure FP-tree avec la création et l'insertion des différents nœuds.

- **Etape 4**

A partir du résultat obtenu lors de l'étape précédente, nous commençons la construction de la structure FP-tree. Tout d'abord l'élément 'Racine' de l'arbre est créé. Cet élément racine ne contiendra aucun élément. Il contiendra uniquement des liens vers ses éléments enfants.

On commence par parcourir chaque élément de la transaction. Puis pour chaque élément de la transaction on vérifie l'existence d'un nœud correspondant, s'il n'existe pas, le nœud est créé, dans le cas contraire le nombre d'occurrence est incrémenté. Puis pour chaque élément créé on va établir un lien depuis la table des entêtes vers l'élément inséré dans l'arbre.

A ce stade l'arbre est encore vide, donc la procédure de vérification de l'existence d'un nœud en particulier indiquera qu'il n'existe pas et par conséquent il faudra le créer.

La première transaction est composée des éléments (f, c, a, m, p) triés par ordre décroissant en fonction de leur poids. L'élément f étant le premier de la liste, un nœud correspondant est inséré à partir de l'élément racine de l'arbre. Le nœud f contient le compte de 1 car c'est la première fois que l'on insère cet élément. Un lien est établi entre l'élément racine de l'arbre et l'élément f puis un autre lien est établi à partir de la table des entêtes.

On poursuit de même avec les éléments suivants (c, a, m, p). Ainsi on obtient la structure illustrée dans la figure au-dessus.

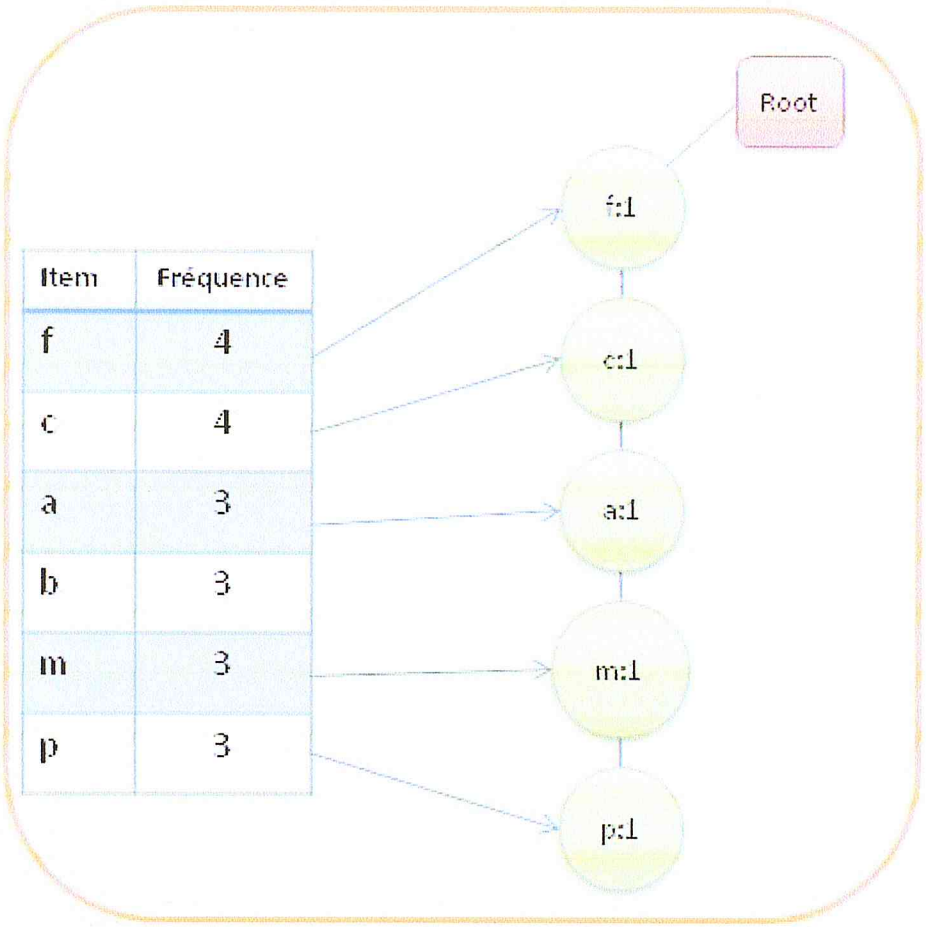


Figure 1.6 : Structure d'une FP-tree [15].

- **Etape 5 :**

La construction se poursuit avec la deuxième transaction qui est composée des éléments (f, c, a, b, m). Cette fois-ci l'arbre contient des éléments et par conséquent pour chaque élément trouvé son nombre d'occurrences est incrémenté de 1.

La transaction contient l'élément ***f***, l'arbre aussi. Par conséquent le nombre d'occurrence de ***f*** passe à ***2***. Il en est de même pour les éléments ***c***, et ***a***. Nous arrivons à l'élément ***b***. Il n'existe pas d'éléments ***b*** correspondant, donc un nouveau nœud est créé à partir de notre position actuelle, c.à.d. le nœud ***a*** et un nouveau lien est créé à partir de ***a*** vers ***b*** puis un lien à partir de la table des entêtes vers le nouveau élément inséré. Concernât l'élément ***m*** restant, étant donné qu'il n'existe pas de nœud correspondant à partir de notre

position (nœud ***b***) un nouveau nœud est créé et initialisé avec une valeur de ***1***, puis en plus du lien créé à partir de l'élément ***b***, un lien est créé à partir du nœud ***m*** déjà existant.

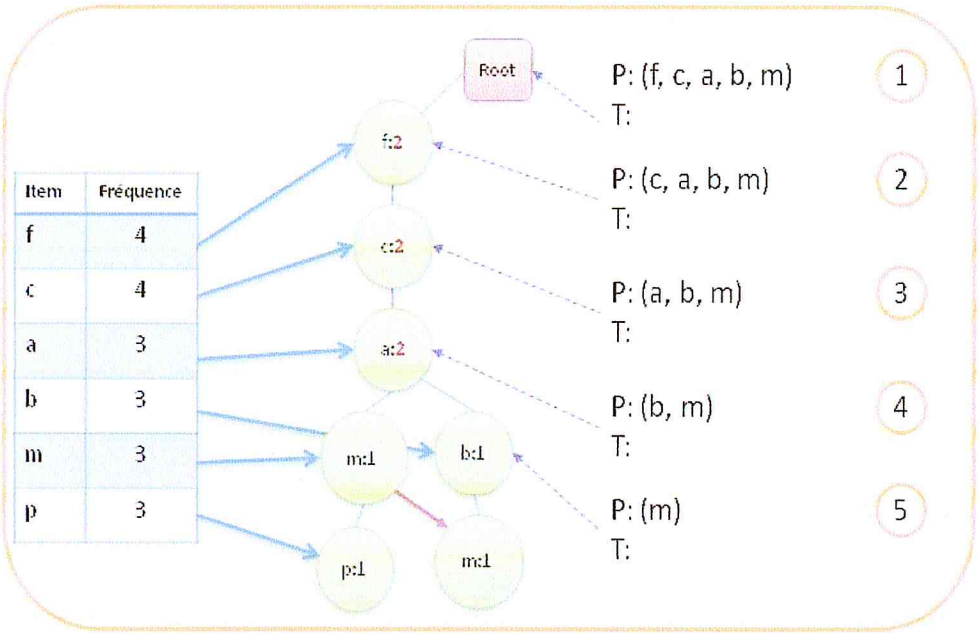


Figure 1.7 : Construction FP-tree à partir de la 2ème transaction [15].

A la fin du traitement de toutes les transactions de la base, on obtient une structure finale de FP-tree.

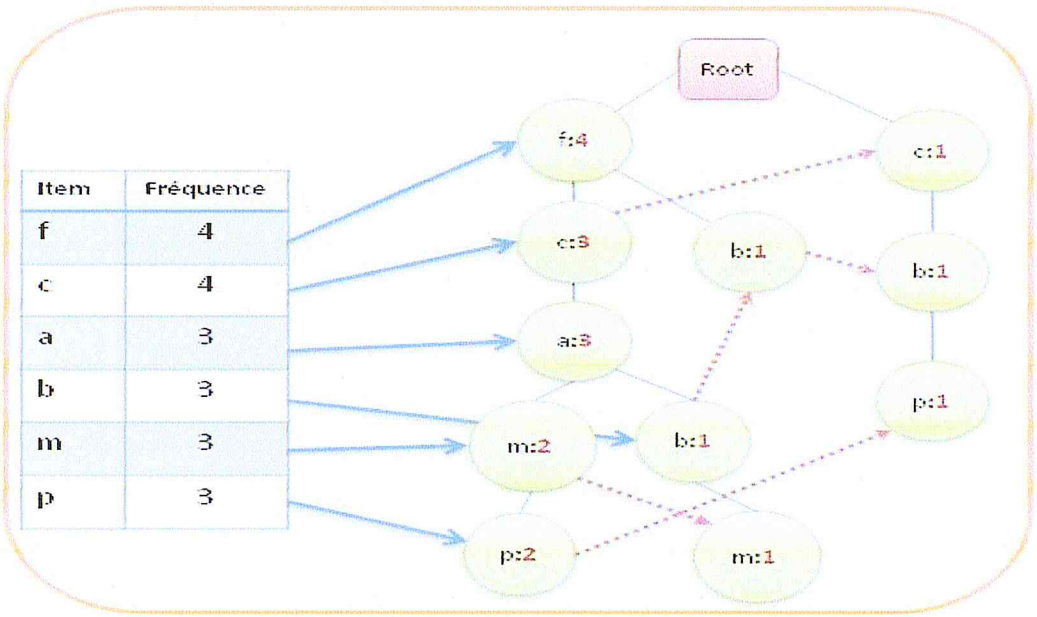


Figure 1.8 : Etat final de la structure FP-tree [15].

- **Etape 6**

Cette dernière étape consiste à valider les informations de l'arbre. Comment savoir si elles sont correctes? La réponse à cette question est très simple, il suffit de comparer les informations obtenues à partir des différents nœuds de l'arbre avec les informations de la table des entêtes. Pour cela, il faut compter, et additionner si besoin, toutes les occurrences d'un élément dans l'arbre et comparer le résultat obtenus à celui stocké dans la table des entêtes. Ainsi après avoir compté tous les éléments de la structure, on obtient le résultat suivant : (f:4, c:4, a:3, b:3, m:3, p:3) ce qui est conforme à la tables des entêtes.

b) Variantes

On trouve dans la littérature de nombreux algorithmes bases sur cette technique, permettant de générer tous les itemsets fréquents d'une base transactionnelle. L'algorithme FELINE [16] qui exploite la structure de données CATS, pour l'extraction des itemsets fermés, H-Mine [17] qui exploite des modèles fréquents dans les grandes bases de données. PrefixSpan (Prefix-projected Sequential pattern) [18] qui explore la projection de préfixe dans l'exploration de modèles séquentiels, LCM [19] est un algorithme efficace pour énumération des ensembles d'itemsets fermés fréquents.

IV. Comparaison entre les types des algorithmes

Le tableau 1.4 présente une étude comparative entre les deux types d'algorithme pour extraction des itemsets présents précédemment en fonction des critères suivants :

- Méthode utilisé
- Temps d'exécution
- Utilisation de mémoire
- Capacité

Tableau 1.5 : La comparaison entre les types des algorithmes [20].

	Algorithme de type «Tester-et-Générer»	**Algorithme de type « Diviser-pour-régner»**
Méthode Utilises	Générer des singletons, des paires, des triples et à chaque transaction, la méthode de calcul est itérative.	Insérer les items triés par fréquence dans une arborescence d'itemsets
Temps d'exécution	La génération des candidats est extrêmement lente. L'exécution augmente exponentiellement selon le nombre des différents items.	L'exécution augmente linéairement, selon le nombre des transactions et des items
Utilisation de mémoire	Stocker des singletons, des paires, des triplés.	Stocke une version compacte de la base de données.
Capacité	La génération des candidats est très parallélisable.	Les données sont très inter dépendant, chaque nœud à besoin de la racine

Les deux types d'algorithmes sont utilisés pour l'extraction des itemsets fréquents de la base de données. Les algorithmes de type «Tester-et-Générer» fonctionne bien avec une grande base de données par contre pour les algorithmes de type « Diviser-pour-régner».

D'après le tableau 1.5 nous voyons que :

- Les algorithmes de type «Tester-et-Générer» effectue plusieurs scans pour générer des candidats cela nécessite de grandes espace mémoire et la procédure est lente.
- Les algorithmes de type « Diviser-pour-régner» scanne la base de données seulement deux fois. Ils nécessitent moins de mémoire grâce à sa structure compacte et le temps de la procédure est moins comparé à les algorithmes de type «Tester-et-Générer».

V. Conclusion

Dans ce chapitre nous avons vu les algorithmes d'extractions d'itemsets fréquents à partir de données, nous avons aussi expliqué leur fonctionnement. L'efficacité de ces algorithmes d'extraction n'est plus un obstacle pour des données parfaites, mais encore il est nécessaire de développer des méthodes quand il s'agit de données qui prennent en considération le contexte de collecte de ces derniers. C'est ce que nous allons développer dans les prochains chapitres.

Chapitre II

Modélisation du contexte

I. Introduction

La modélisation du contexte est la première étape dans le processus d'extraction d'itemsets contextuels. Donc dans ce chapitre, nous explicitons les différents modèles existants pour représenter un contexte.

Ce chapitre est organisé de la manière suivante. Dans la section 2, nous définissions c'est quoi le contexte d'une manière générale. Nous décrivons les méthodes existantes pour modéliser un contexte dans la section 3. Une étude comparative entre ces méthodes dans la section 4. Enfin, nous mettons une petite conclusion pour terminer ce chapitre.

II. Notion du contexte

2.1. Définition du contexte

Le contexte étant une notion complexe et abstraite, il est difficile de trouver une définition détaillée. Les définitions générales sont les plus nombreuses ; nous listons ici les principales [21]. Parmi les premiers à essayer de définir le contexte, nous trouvons Schilit et Theimer [22], pour lesquels le contexte est constitué de la localisation de l'utilisateur, ainsi que des identités et des états des personnes et des objets qui l'entourent. Brown et al. [23] ajoutent à cette définition des données telles que l'identité de l'utilisateur, son orientation ou la température. Ryan et al. [24] ajoutent la notion de temps.

Pascoe [25] introduit un élément important : l'intérêt. En effet, il définit le contexte comme un sous-ensemble d'états physiques et conceptuels qui ont un certain intérêt pour une entité donnée. Cette notion d'intérêt ou de pertinence est reprise par Abowd, Dey et al. [26] dans leur définition, qui est communément acceptée :

« Le contexte couvre toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité est une personne, un endroit ou un objet que l'on considère pertinent par rapport à l'interaction entre un utilisateur et une application, y compris l'utilisateur et l'application eux-mêmes. » [26]

Winograd [27] reprend cette définition pour la détailler, car il considère que, malgré le fait qu'elle couvre tous les travaux existants, elle est trop générale : tout élément peut être considéré comme faisant partie du contexte. En premier lieu, il précise que le contexte est un ensemble d'informations. Cet ensemble est structuré et partagé, et il peut évoluer dans le temps. En deuxième lieu, selon lui, l'appartenance d'une information au contexte ne dépend

pas de ses propriétés inhérentes, mais de la manière dont elle est utilisée. Une information fait partie du contexte seulement si le système dépend d'elle, d'une façon ou d'une autre.

Malgré le grand nombre de définitions existantes et les similarités (la plupart font références à la localisation et l'environnement), le mot contexte reste toujours général.

2.2 Catégorisation du contexte

Plusieurs chercheurs ont proposé des catégorisations selon différentes approches. Mais il y a une autre catégorisation proposée par les mêmes auteurs qui fondée sur les valeurs que peut prendre une information contextuelle, qui sont : [28]

- **Le contexte continu :** dans cette catégorie, les valeurs varient continuellement. Un élément d'un contexte continu est fonction de différents paramètres et sa valeur est calculée en se servant d'une formule. Exemple : les informations GPS ;
- **Le contexte énumératif :** les valeurs d'un composant du contexte sont un ensemble discret de valeurs ;
- **Information contextuelle d'état** : les éléments de cette catégorie peuvent prendre deux valeurs opposées. Par exemple : la lumière dans une pièce peut être allumée ou éteinte. Les valeurs de ces éléments sont obtenues à partir d'un calcul de prédicat ;
- **Le contexte descriptif** : il est basé sur les descriptions des éléments du contexte. Il existe bien d'autres catégorisations qu'ont été proposées que celles qui ont été présentées ici, mais aucune d'elles ne se veut exhaustive. De nouveaux regroupements seront effectués à mesure que de nouvelles caractéristiques des informations contextuelles seront découvertes. Il n'en demeure pas moins que ces efforts de classification sont louables et permettent aux développeurs de l'informatique diffuse de manipuler plus efficacement les informations contextuelles.

2.3 Acquisition des informations du contexte

Les informations de contexte sont des informations collectées à partir de plusieurs sources hétérogènes. De ce fait, elles ont des caractéristiques variables. Chaque contexte observable peut être statique ou dynamique.

L'acquisition des informations de contexte peut se faire selon plusieurs méthodes [29]:

- **Acquisition par profil** : cette méthode consiste à récupérer des informations soit à travers une interface graphique soit par l'intermédiaire d'un fichier de profil.
- **Acquisition par sonde** : cette méthode consiste à utiliser des sondes (capteurs) pour récupérer les informations de contexte.
- **Acquisition par dérivation :** la dérivation ou l'interprétation du contexte consiste à utiliser un ou plusieurs observables pour déduire ou calculer à la volée un contexte de plus haut niveau en utilisant des méthodes d'interprétation.

III. Méthodes de modélisation de contexte

Dans cette section, nous allons présenter les différentes méthodes de modélisation du contexte qui existent dans la littérature.

3.1. Modèle clé-valeur

Le modèle clés- valeurs [30] utilise la paire clés-valeurs pour définir la liste des attributs et leurs valeurs décrivant les informations de contexte. Par exemple, nous avons un contexte qu'est définit par l'utilisateur 'x' qui est localisé dans un emplacement 'y' à un temps 't' est modélisé comme suit :

- Name = 'contexte',
- User = 'docteur Kamran',
- Localisation ='Hôpital Khalil Amrane',
- Temps = 'lundi 28 mars 2012 16 :41 :29' ".

- **Avantages et inconvénients**

La modélisation clés-valeurs utilise des structures de données simples à gérer. Par contre, cela ne permet pas une description complète du contexte, ni l'expression des relations qui peuvent exister entre les informations de contexte. Par exemple, un changement de la valeur d'une propriété (ex. la bande passante réseau) peut affecter les valeurs d'autres propriétés (ex. reste de la puissance de la batterie).

3.2. Modélisation orientée objet

Henriksen et al. [31] ont proposé un ensemble de concepts de modélisation basés sur une approche orientée objet. Dans cette approche, les informations de contexte sont regroupées en un ensemble d'entités. Chaque entité représente un objet conceptuel ou

physique tel qu'une personne, un dispositif ou un réseau. Les propriétés des entités telles que le nom d'une personne sont représentées par des attributs. Les entités sont liées à leurs attributs à travers des associations. Voici quelques exemples de modèles orientés objet.

3.2.1. Modèle basé sur UML (Unified Modeling Language)

La généricité du langage UML en fait un langage approprié pour modéliser le contexte Sheng et Benatallah [32] ont proposé un méta-modèle basé sur une extension d'UML qui permet de modéliser le contexte auquel des services web sont sensibles. Ce langage est appelé ContextUML.

3.2.2. Modèle basé sur un langage de balise

Le point commun entre toutes les méthodes qui utilisent un langage de balises est la structure hiérarchique des données modélisées. Cette modélisation consiste en un ensemble de balises avec des attributs. Dans XML (eXtensible Markup Language), les balises sont définies dans une DTD (Document Type Definition) et dans CC/PP (Composit Capability/Prefrence Profile) [33] qui est la proposition du W3C pour la représentation de profils comme vue dans la figure 2.1. C'est un cadriciel basé sur RDF (Resource Description Framework) [34] pour stocker des paires clé-valeur avec les balises appropriées. CC/PP permet de décrire les capacités d'un dispositif ainsi que les préférences de l'utilisateur en utilisant une structure de profils. Le vocabulaire offert par CC/PP n'est pas riche et a besoin d'être étendu car il est restreint à la description de profil. De plus, il ne permet pas la description de relations et de contraintes complexes entre les informations de contexte.

Les travaux d'Indulska et al. [35] ont étendu le vocabulaire de CC/PP pour pouvoir décrire la localisation, les caractéristiques du réseau et les dépendances d'une application. Ainsi, cette modélisation peut être utilisée pour décrire les contextes observables associés à une application sensible au contexte, mais Indulska et al. ont conclu que malgré cette extension, cette modélisation reste non intuitive et difficile à utiliser pour décrire des informations complexes [36].

La figure 2.1 montre un exemple CC/PP utilisé pour la modélisation de la localisation de l'utilisateur.

[LocationProfile [PhysicalLocation [Country, State, City, Suburb]]
[LogicalLocation [IPAddress]]
[GeodeticLocation [Longitude, Latitude, Altitude]]
[Orientation [Heading, Pitch]]]

Figure 2.1 : Modélisation de la localisation de l'utilisateur en utilisant CC/PP [37]

3.2.3. Modèle basé sur CML (Context Modeling Language)

Afin de pouvoir modéliser les caractéristiques des informations de contexte et leurs propriétés, Henricksen et al. [38] ont proposé un langage orienté objet appelé CML (Context Modeling Language) qui permet de modéliser le contexte auquel une application est sensible d'une manière formelle. Un outil graphique assiste le concepteur d'applications dans la tâche de description du contexte auquel son application est sensible. Il lui offre un moyen de décrire les caractéristiques des informations de contexte (capturé, statique, dérivé ou information de profile) et les dépendances entre ces informations. CML permet aussi de spécifier la qualité de chaque information observée et sa validité temporelle.

- **Avantages et inconvénients**

Cette modélisation ne permet pas de décrire la validité temporelle des informations ni les relations qui peuvent exister entre les informations de contexte, mais elle est très efficace pour raisonner sur le contexte et déduire des actions de réaction si une situation pertinente est détectée. L'approche basée sur la logique peut être utilisée dans l'informatique sensible au contexte afin d'intégrer et d'interpréter les données collectées.

3.3. Modèles basés sur les ontologies

Les ontologies représentent une autre solution pour modéliser le contexte [39]. Patricia [40] a justifié l'utilisation des otologies par trois arguments :

i. Une ontologie permet de partager les connaissances dans un system distribué.
ii. Une ontologie comprend des sémantiques déclaratives permettant d'élaborer des raisonnements sur les informations contextuelles.
iii. Avec une représentation explicite d'une ontologie commune, l'interopérabilité des applications et des terminaux est assurée.

Voici quelques travaux qui basée sur le modèles ontologie :

- Sachin et al ont proposé un context-aware data mining framwork, où le contexte sera représenté dans une ontologie. [41]
- Xiao Hang et al ont proposé une ontologie de contexte encodé en OWL (CONON) pour la modélisation de contexte. CONON (Context Ontology) fournit une ontologie de contexte supérieure qui capture les concepts généraux sur le contexte de base et offre également une extensibilité pour l'ajout d'une ontologie spécifique à un domaine de manière hiérarchique. Leurs études dans cet article montrent qu'un modèle de contexte basé sur une ontologie est réalisable et nécessaire pour prendre en charge la modélisation de contexte. [42]

La figure 2.2 représente un exemple de langage de description de contexte utilisant les ontologies. Il considère qu'une information du contexte a un certain aspect (ou représentation) et représente une certain entité. Cet exemple décrit la position géographique (information de contexte) représenté par les coordonnées de Gauss-Krueger (aspect ou représentation) relative à un téléphone mobile (entité). [43]

```
<instance xmlns=http://demo.heywow.com/schema/cool
      xmlns:a=http://demo.heywow.com/schema/aspects
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <contextInformation>
   <entity system="urn:phonenumber">+49-179-1234567</entity>
   <characterizedBy>
    <aspect name="GaussKruegerCoordinate">
     <observedState xsi:type="a:o2GaussKruegerType">367032533074</observedState>
     <units>10m</units>
    </aspect>
    <certaintyOfObserver>90</certaintyOfObserver>
   </characterizedBy>
  </contextInformation>
</instance>
```

Figure2.2: Exemple de représentation XML du contexte en utilisant l'ontologie CoOL [43].

- **Avantages et inconvénients**

L'avantage de ces approches réside dans les caractéristiques même de l'ontologie, qui offrent non seulement le moyen de faire des descriptions sémantiques, mais aussi de publier les données décrites à travers le réseau. Cependant ces modèles restent souvent complexes à implémenter dans des cas réels.

IV. Comparaison sur les modèles de représentation du contexte

Le tableau 2.1 présente une étude comparative entre les différentes méthodes de modélisation de contexte en fonction des critères suivants :

i. Les Caractéristiques.
ii. Les relations et dépendance.
iii. Réutilisation et extension du modèle.
iv. Mise en œuvre

Tableau2.1: Vue de comparaison des modèles existantes de modélisation du contexte [21].

Modèle\ Critères	Caractéristiques	Relations et dépendance	Réutilisation et extension du modèle	Mise En Œuvre
Paires d'Attribut/ Valeur	Simplicité d'utilisation, pauvreté d'expression	Non	Non	Facile
Modélisation orientée objet	Utilisation d'un méta-modèle	Oui	Oui	Facile
Ontologies	Utilisation d'un moteur d'inférence	Oui	Oui	Difficile

Après avoir étudié ces différentes approches on a aboutie le résultat suivant :

- Les approches clés-valeurs sont caractérisées par une pauvreté d'expression et la simplicité des données qu'elles représentent. Elles sont basées sur une description par un tuple (clé/valeur) et elles sont caractérisées par leurs facilités de mise en œuvre. elles ne permettent pas la description des relations existantes entre les informations contextuelles.
- Les approches orientées objets sont caractérisées par leurs possibilité de réutilisation du fait qu'elles utilisent des modèles formels pour la description

de contexte et qu'elles permettent la description d'un méta-modèle qui peut être réutilisé par plusieurs applications.

- Les approches orientée ontologies quant à elle bien qu'elles sont caractérisées par leurs difficulté de mise en œuvre. son utilisation de moteur d'inférence lui permet de décrire des relations entre les informations du contexte et un raisonnement sur ces dernières.

Chacune des approches de modélisation présentée dans le tableau peut fournir une solution efficace pour un domaine particulier, et / ou pour un type particulier de raisonnement, mais aucun d'eux ne peut simultanément satisfaire à toutes les exigences de la modélisation des informations contextuelles.

Pour notre contribution, nous avons choisis d'utiliser le modèle paires d'attribut/ valeur pour modéliser les informations contextuelles, puisque il est le modèle le plus utilise et plus il est le plus facile à implémenter.

V. Conclusion

Dans ce chapitre, nous avons présenté les différents modèles existants pour représenter ou bien modéliser un contexte.

Notre perspective dans le prochain chapitre est de présenter un état de l'art sur l'extraction des motifs contextuels fréquents à travers l'étude des approches qu'ont introduisant la notion des contextes pour l'extraction des motifs fréquents.

Chapitre III

Extraction des motifs contextuels fréquents

I. Introduction

Nous nous mettons dans un cadre de découverte des connaissances où nous nous intéressons à l'extraction des motifs fréquents à partir de données. Ces données contiennent des informations contextuelles.

Dans chapitre nous présentons un l'état de l'art sur l'extraction des motifs contextuels fréquents. Le chapitre est organisé de la manière suivante. La section 2, nous allons présenter les approches connexes à notre approche. Ensuite, nous faisons une étude comparative entre ces approches dans la section 3. Dans la section 4, nous citons les avantages et les inconvénients de chaque approche. Enfin, nous mettons une petite conclusion pour terminer notre chapitre.

II. Approches d'extraction des motifs contextuels

La majorité des techniques d'extraction des motifs fréquents existent ne prennent pas en considération les informations contextuelles importantes. Par exemple, ce que le patient faisait au moment où les données ont été collectées dans le cadre de données médicales, la position et les coordonnées d'un capteur collectant des données environnementales.

Dans cette section, nous présentons les approches qui sont basées sur l'hypothèse que les données contextuelles supplémentaires pourraient augmenter l'utilité des motifs. Avant d'expliciter les approches existantes d'extraction de motifs contextuels, il est utile de définir en premier c'est quoi un motif contextuel par rapport à ces approches.

- **Motif Contextuel :** sont des motifs qui sont trouvés fréquents en fonction de leur contexte [44] [46].

2.1. Représentation du contexte

2.1.1. Hiérarchie du contexte [44]

A notre connaissance, Rabatel et al. sont les premiers à explorer l'utilisation des informations contextuelles pour l'extraction des motifs fréquents [44]. Dans ce travail, les informations contextuelles sont représentées par hiérarchie du contexte. Cette hiérarchie est un graphe orienté acyclique, notée par $H = (V_H , E_H)$, tel que :

- V_H : est un ensemble des sommets aussi nommée contexte,
- E_H : est l'ensemble des arêtes dirigée entre les contextes.

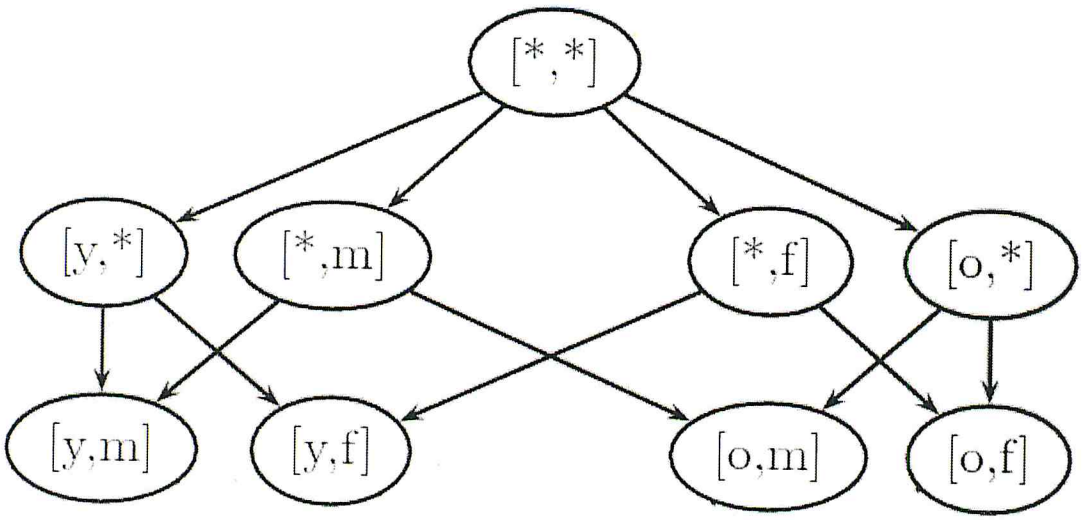


Figure 3.1 : une hiérarchie du contexte [44]

2.1.2. Contextual Information Graph (CIG) [45]

Dans [45], les auteurs ont proposé un cadre général pour encoder les informations contextuelles provenant de multiples sources sous forme des graphes via un algorithme d'encodage appelé CIG pour Contextual Information Graph. Ils assument que toutes les informations contextuelles sont des valeurs correspondants à des caractéristiques discrètes, ou ils ont été convertis en valeurs discrètes selon certain critères.

Graphe d'informations contextuelles (notée CIG) est un graphe orienté bipartie, dénoté par $G = (U, V, E)$, tel que:

- U décrit l'ensemble des informations contextuelles,
- V décrit l'ensemble d'identités
- E est l'ensemble de liens existant entre U et V, de tels sorts où l est la fonction de nommage des sommets de G.

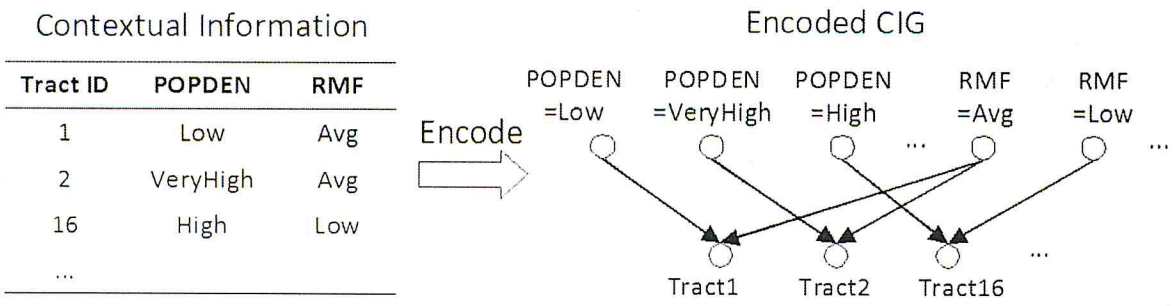


Figure 3.2 : Contextual Information Graph (CIG) [45]

2.1.3. Ontologie [46]

Dans [46], les informations contextuelles sont représentées par une ontologie qu'est composée uniquement d'une hiérarchie des classes. L'ontologie est considérée comme un graphique acyclique dirigé, dénoté par $H = (V_H, E_H)$, tel que V_H représente un ensemble de sommets qu'ils sont appelés contextes et $E_H \subseteq V_H \times V_H$ est un ensemble d'arêtes dirigées entre les contextes.

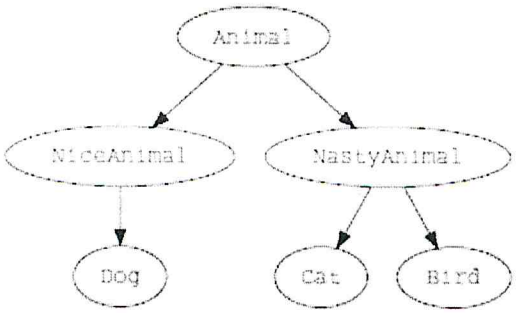


Figure 3.3 : Une ontologie H [46]

2.2. Extraction des motifs contextuels fréquents

Dans cette partie, nous allons présenter les algorithmes d'extraction des motifs contextuels de chaque approche.

2.2.1 Extraction de motifs séquentiels contextuels

Dans [44], Rabatel et al. ont proposé un algorithme pour l'extraction des motifs contextuels fréquent dans le cas des motifs séquentiels, notée par Contextuel SPM. Leur algorithme est basé sur l'algorithme de PrefixSpan [47] qu'est dédié pour résoudre le problème d'extraction des motifs séquentiels traditionnel.

Leur approche se compose de deux étapes. A partir d'une base de données de séquences contextuelles, ils extraient toutes les séquences fréquentes dans au moins un contexte minimal. Ensuite, à partir de l'ensemble des séquences obtenues de l'étape précédant, ils génèrent l'ensemble des motifs séquentiels contextuels.

L'algorithme proposé prend en entrée une base contextuelle de séquences CB, un seuil de support minimum, et une hiérarchie de contextes H (comme vue dans la figure 3.4), il retourne les motifs séquentiels contextuels de CB. La figure ci-dessus représente une base contextuelle de séquences utilise dans cet algorithme [44].

id	Age	Gender	Sequence
s_1	young	male	$\langle(ad)(b)\rangle$
s_2	young	male	$\langle(ab)(b)\rangle$
s_3	young	male	$\langle(a)(a)(b)\rangle$
s_4	young	male	$\langle(c)(a)(bc)\rangle$
s_5	young	male	$\langle(d)(ab)(bcd)\rangle$
s_6	young	female	$\langle(b)(a)\rangle$
s_7	young	female	$\langle(a)(b)(a)\rangle$
s_8	young	female	$\langle(d)(a)(bc)\rangle$
s_9	old	male	$\langle(ab)(a)(bd)\rangle$
s_{10}	old	male	$\langle(bcd)\rangle$
s_{11}	old	male	$\langle(bd)(a)\rangle$
s_{12}	old	female	$\langle(e)(bcd)(a)\rangle$
s_{13}	old	female	$\langle(bde)\rangle$
s_{14}	old	female	$\langle(b)(a)(e)\rangle$

Figure 3.4 : une base contextuelle de séquences CB [44]

a) Extraction des motifs séquentiels fréquents dans des contextes minimaux

Le premier objectif de l'algorithme est d'extraire des séquences fréquentes dans des contextes minimaux, et pour chaque d'eux, l'ensemble des contextes minimaux correspondant où il est fréquent. Cette partie réalisée en utilisant le principe de l'algorithme PrefixSpan : en prenant pour préfixe une séquence s, l'algorithme construit et parcourt la base projetée correspondante afin de trouver les items i qui peuvent être assemblés pour former un nouveau motif séquentiel s'. Ensuite, une base de données projetée s' est construite et le processus se poursuit.

b) Générer des motifs séquentiels contextuels

Après l'extraction des motifs séquentiels avec l'ensemble des contextes minimaux où une séquence est fréquente. A partir de cet ensemble, ils génèrent des motifs séquentiels contextuels générés par une séquence s, c'est-à-dire, l'ensemble de (c, s) où c est un contexte tel que s est spécifique à c. Ceci est effectué par une analyse ascendante de la hiérarchie de contexte (c'est-à-dire, le parcours de la hiérarchie à partir des feuilles à la racine), afin de collecter les contextes les plus généraux étant des sous-ensembles de $\mathcal{F}$, où $\mathcal{F}$ est un ensemble de contextes minimaux, c'est-à-dire où une séquence est spécifique.

Exemple

sequence	$[y, m]$	$[y, f]$	$[o, m]$	$[o, f]$
$\langle(a)\rangle$	**5/5**	**3/3**	**2/3**	**2/3**
$\langle(b)\rangle$	**5/5**	**3/3**	**3/3**	**3/3**
$\langle(d)\rangle$	2/5	1/3	**3/3**	**2/3**
$\langle(e)\rangle$	0/5	0/3	0/3	**3/3**
$\langle(a)(b)\rangle$	**5/5**	**2/3**	1/3	0/3
$\langle(b)(a)\rangle$	0/5	**2/3**	**2/3**	**2/3**
$\langle(bd)\rangle$	1/5	0/3	**3/3**	**2/3**

Figure 3.5 : Les motifs séquentiels dans des contextes minimaux de base contextuelle de la base contextuelle de séquences [44]

La figure 3.5 montre les séquences qu'ils sont fréquentes dans au moins un contexte minimal, ainsi que leur support pour chaque contexte minimal sous la forme $(sup_c(s)/|B(c)|)$, où B(c) est la base contextuelle de séquences et s est une séquence. Lorsque le support est affiché en gras, la séquence est fréquente dans le contexte minimal correspondant.

Par exemple, $s = \langle(a)(b)\rangle$ est fréquent dans [y, *] et dans ses descendants [y, m] et [y, f], c'est-à-dire s est général dans [y, *]. De plus, s n'est pas général dans [*, *] car s n'est pas fréquents dans toutes ces descendants. En conséquence, s est spécifique dans [y, *] et $([y,*], \langle(a)(b)\rangle)$ est un motif séquentiel contextuel.

2.2.2. Extraction des règles d'association contextuelles

Dans l'article [45], Dong et al. ont proposé que l'extraction des motifs contextuels s'effectue en deux étapes suivantes :

1. Construire un graphe des informations contextuelles CIG (illustre dans la section 2.1.2).
2. Appliquer l'algorithme d'extraction des motifs à partir d'une source unique pour découvrir des motifs contextuels. Ce dernier est basé sur l'algorithme d'extraction des motifs généralises existant appelé GPM pour Generalized Pattern Mining [48, 49, 50, 51, 52]

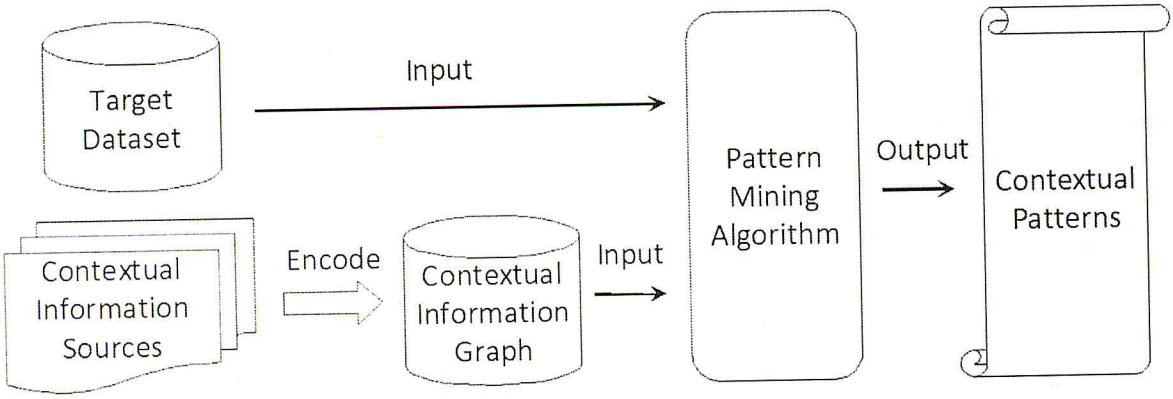


Figure 3.6 : Etapes d'extraction des motifs contextuels [45]

L'algorithme d'extraction des règles d'association contextuelles est basé sur l'algorithme d'extraction des règles d'association généralisée [51]. Comme dans [51], première les transactions sont développées avec les informations contextuelles liées aux items en utilisant les requêtes de KI path [45] sur le CIG, où un item peut être un prédicat ou une identité. Les items dupliqués trouvent en cours de développement seront éliminé. Par exemple, après l'expansion le tableau 3 deviendra le tableau 3.1.

Tableau 3.1 : Exemple de jeu de données cibles de la criminalité [45].

tid	Items
4	Wednesday, Drugs, within (Tract26), close_to (Trent Av)
251	Thursday, Robbery, within (Tract16), close_to (Garland Av)
266	Friday, Vehicle Theft, within(Tract1), close_to (Freya St), Close_to (Wellesley Av)

Tableau 3.2 : Exemple de jeu de données cibles de la criminalité après l'expansion [45].

tid	Items
4	Wednesday, Drugs, within (Tract26), close_to (Trent Av),within (RMF=High), within (POPDEN=Low), close_to (Road)
251	Thursday, Robbery, within(Tract16), close_to (Garland Av), Within (RMF=Low), within (POPDEN=High), close_to(Road)
266	Friday, Vehicle Theft, within (Tract1), close_to (Freya St), close_to (Wellesley Av), within (RMF=Avg), within (POPDEN=Low), close_to(Road)

Ensuite, les itemsets fréquents sont énumérés et élagués. Toutes itemsets contenant un item ▷ les autres seront élagués pour empêcher de générer des itemsets comme {within (Tract26), within (POPDEN=Low)} et {close_to (Trent Av), close_to(Road)}. Ces itemsets indiquent la corrélation entre un item et son contexte, qui peuvent être fréquents mais fournissent des connaissances redondantes déjà représentées dans la CIG.

2.2.3. Extraction des graphes contextuels

Tout d'abord, Dong et al. [45] définissent l'isomorphisme du sous-graphe contextuel en étendant conceptuellement l'isomorphisme du sous-graphe généralisé [49, 50]. Ils dénotent l'ensemble des sommets d'un graphe g par V(g) et l'ensemble des arêtes par E (g).

Un graphe g est un sous graphe contextuel d'un autre graphe g' s'il existe un isomorphisme de sous-graphe contextuel de g à g'.

L'algorithme de l'extraction des graphes contextuels est similaire à l'algorithme d'extraction des règles d'associations contextuelles, ici les graphes sur-généralisés [9, 3] doivent être élagués. Si un motif graphique gp est un sous-graphe contextuel d'un motif graphique gp' (tel que $gp' \neq gp$) et que leurs supports sont identiques, donc gp est un motif sur-généralisé.

L'approche proposé, c'est-à-dire, " l'algorithme d'encodage CIG + extraction des motifs ", découvre des motifs qu'ils sont plus prédictifs, perspicaces et intéressants que les algorithmes traditionnels d'extraction des motifs ne peuvent pas trouver.

2.2.4. Extraction des itemsets contextuels

Dans [46], les auteurs ont proposé un algorithme pour extraire des itemsets contextuels fréquent dans de le cadre Linked Open Data. Cet algorithme visé à découvrir des itemsets dont la propriété d'être fréquent dépend du contexte.

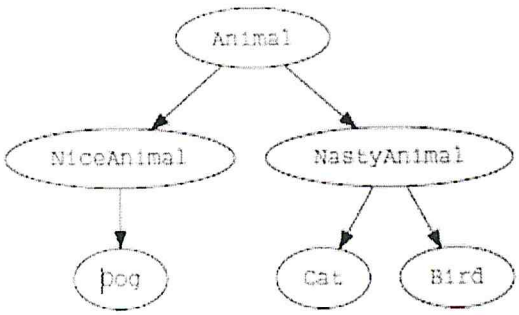
L'algorithme requit une base de connaissances, notée par KB (comme vue dans la figure 3.7), qu'est composé d'une ontologie et un ensemble des faits, noté par $\mathcal{F}$, cet ensemble est défini en utilisant le triple de RDF sous la forme :

$$(\ Sujet, Propriété, Objet\ )$$

Chaque élément du triple est défini en fonction de l'ontologie qu'elle désigne la hiérarchie de contexte, notée par $\mathcal{H}$ (présente dans la section 2.1.3).

Subject	Predicate	Object
Bill	eats	Tweety
Tweety	hates	Bill
Bill	playsWith	Boule
Tom	eats	Tweety
Tom	hates	Bill
Garfield	hates	Bill
Garfield	eats	Tweety
Garfield	hates	Boule

(a) A fact base $\mathcal{F}$.



(b) An ontology $\mathcal{H}$.

Figure 3.7 : Exemple d'une base de connaissances $KB = (\mathcal{F}, \mathcal{H})$[46].

Basé sur cette base de connaissance, les autres construisent une base de données transactionnelle de tel sort pour chaque transaction correspond à l'ensemble de prédicats et d'objets des sujets d'une classe donnée. Plus précisément, étant donné une base de connaissances KB et $c \in V_{\mathcal{H}}$, une base transactionnelle pour c par rapport à KB, noté par $\mathcal{T}_{KB,c}$, est l'ensemble des transactions sous la forme $\mathcal{T} = (\text{tid}, I_c)$ où $I_c = \{(\text{pred}, \text{obj}) | (s, \text{pred}, \text{obj}) \in \mathcal{F}$ et c est la classe de s$\}$.

tid	I_{Animal}
$Bill$	$\{(eats, Tweety), (playsWith, Boule)\}$
$Tweety$	$\{(hates, Bill)\}$
Tom	$\{(eats, Tweety), (hates, Bill)\}$
$Garfield$	$\{(hates, Bill), (eats, Tweety), (hates, Boule)\}$

Figure 3.8 : une base de données transactionnelle $\mathcal{T}_{KB,Animal}$ pour le contexte animal dans la base de connaissance KB représenté dans Figure 3.7 [46]

L'algorithme proposé est inspiré de celui proposé dans [44]. Il est noté par CFPs pour Contextual Frequent Patterns, il est utilisé pour extraire des itemsets dans une base de données *DBpedia* [53]. L'algorithme s'effectue en étapes suivantes :

i. Les itemsets fréquents sont extraits de chaque contexte minimal, cette étape est une implémentation de l'algorithme Apriori [54]
ii. Les fichiers de sortie de l'étape précédente sont lus et les itemsets sont indexés par l'ensemble des contextes minimaux où ils sont fréquents.
iii. Génération des itemsets contextuels fréquents.

III. Etude comparative

Le tableau 3.2 présente une étude comparative entre les différentes approches d'extraction des motifs contextuels présentées précédemment en fonction des critères suivants :

- Représentation du contexte
- Type de motif
- Représentation sémantique
- Sources

Tableau 3.3 : Comparaison entre les approches

	Rabatel et al. [44]	**Dong et al. [45]**	**Poncelet et al. [46]**
Représentation du contexte	Graphe orienté acyclique	Graphe orienté bipartie	Ontologie
Type de motif	Les motifs séquentiels	Les règles d'association, les graphe et motifs séquentiels	Les itemsets
Représentation sémantique	No	No	Oui
Source	Source unique	Multi-sources	Source unique

On remarque qu'il y'a peu de travaux dans la littérature qui font référence à l'utilisation des informations contextuelles lors de l'extraction des motifs fréquents. Trois travaux significatifs ont été présentés et étudiés dans ce chapitre.

Le premier travail présente l'algorithme « Contextuel SPM » de Rabatel et al. [45]. Cet algorithme ne vise pas à extraire des motifs séquentiels qui sont spécifiques à une classe, mais à extraire tous les motifs séquentiels dans une base de données et décrive s'il existe certains des contextes plus ou moins généraux où un motif séquentiel est spécifique en manipulant une hiérarchie de contexte.

L'algorithme CPM de Dong et al [46] s'adresse à impliquer des informations contextuelles en utilisant de façon non intrusive les méthodes d'extraction de motifs à source unique existantes. C.-à-d., en deux-étapes « L'encodage CIG + Extraction des motifs ». L'encodage CIG (Graphe d'informations contextuelles) fonctionne avec différents types de méthodes d'extraction des motifs (itemset/règles d'associations, séquence et graphique).

L'algorithme CFP de Poncelet et al. [46] permet de caractériser un nœud de la taxonomie (contexte) fournissant des indices sur la façon dont les connaissances extraites peuvent être organisées pour les analyses. Puisque les nœuds de taxonomie qu'ils exploitent sont des classes qu'ils auraient pu présenter leur contribution directement à partir d'un point de vue de classe RDF. Ce choix aurait signifié que le lien conceptuel avec la notion de contexte dans le cadre de data mining était perdu. Ils ont donc préféré conserver la terminologie du contexte. L'efficacité de l'algorithme dépend de la forme de l'ontologie. En raison de la nature de leur algorithme et la méthode d'élagage, ils obtenaient des meilleurs résultats si l'ontologie n'est pas grand mais avec une grande hauteur. Un autre problème lors de l'examen des itemsets en tant que des règles est de savoir comment gérer leur nombre et comment les ordonnées afin d'être validés par un expert.

IV. Conclusion

Dans ce chapitre, Nous venons de faire un tour de littérature où nous avons présenté les approches qu'ont introduisant la notion des contextes pour l'extraction des motifs fréquents, notre perspective dans le prochain chapitre est de proposer une autre méthode pour l'extraction des itemsets contextuels.

Chapitre IV

Approche Proposée

Chapitre IV

I. Introduction

Dans ce chapitre, nous allons présenter notre approche pour extraire des itemsets contextuels fréquents. Notre chapitre sera organisé de la manière suivante. La section 2, nous définissions les notions liées aux itemsets contextuels fréquents. Dans la section 3, nous allons présenter les deux approches proposées et décrivions l'algorithme proposé. Enfin, nous mettons une petite conclusion pour termine ce chapitre.

II. Itemsets contextuels fréquents

Dans cette section, nous proposons une description formelle de la notion de contexte et définissions les notions nécessaires pour appréhender les itemsets contextuels fréquents.

2.1. Base Contextuelle d'itemsets

Dans cette partie, nous allons donner la notion de base contextuelle d'itemsets, dans laquelle chaque itemset est associé à diverses informations contextuelles.

- **Base contextuelle d'itemset :** une base contextuelle d'itemsets, notons par B(c), est définie comme suivant (ID, C_1, … … , C_n, I), où ID est un identificateur, pour $1 \leq i \leq n$ C_i représente un ensemble de toutes les valeurs possibles d'une donnée contextuelle et I est dans le domaine des itemsets[1]. Un tuple $t \in B(c)$ est noté $< i, c_1, \dots \dots, c_n >$.

Exemple

Tableau 4.1 : Une base contextuelle d'itemsets

cid	Client-genre	Client-ville	Client-âge	Itemsets
10	Femme	Boston	Age moyenne	{b, d, c, a,}
20	Homme	Chicago	Jeune	{b, f, c, e, g}
30	Homme	Chicago	Age moyenne	{a, h, b, f}
40	Femme	New York	Agé	{b, c, e, h, f}

La même chose pour Client-ville = {Boston, Chicago, New York} et Client-âge = {Age moyenne, Jeune, Agé}.

Un tuple de cette base est, par exemple, <{b}, Femme, Boston, Age moyenne> signifiant que l'item {b} a été enregistré pour un client qui est une femme d'un âge moyenne de Boston.

- **Contexte :** un contexte c dans une base contextuelles d'itemset est noté $[c_1, \dots\dots, c_n]$,où pour $1 \leq i \leq n, c_i \in C_i$.

Exemple: en considérant l'exemple précédant, un contexte dans ce cas pourra être « *une femme d'un âge moyenne de Boston.*» i.e.., $c = [\text{Femme, Boston, Age moyenne}]$ ou « *un jeune homme de Chicago.*», i.e.., $c = [\text{Homme , Chicago , Jeune}]$.

- **Itemset contextuel :** soit $x =< id, i, c_1, \dots\dots, c_n >$ un itemset contextuel dans B(c). où id est l'identificateur de l'itemset, i est l'ensemble des items et c_1, ..., c_n représente le contexte de x.
- **Itemset contextuel fréquent :** soit un contexte c et un itemset i, i est un itemset contextuel fréquent, si et seulement si i est fréquent dans B(c), c'est-à-dire, si $\text{Freq}_{B(c)}(i) \geq \sigma$, où σ est le seuil minimum.
- **Itemset contextuel spécifique fréquent :** soit x et y deux contextes, un itemset i est dit un itemset contextuel spécifique fréquents si et seulement s'il est fréquent dans x mais pas dans y.

  Exemple

  Nous prenons la même base de données (Tableau 4.1) et soit les deux contextes x et y tel que :

  x est un contexte « *homme et femme de ville Chicago et quelle que soit leur âges.*» et y est « *femme de n'importe quelle ville Chicago et quelle que soit leur âges.*».

  D'après le tableau 4.1, nous voyons que l'itemset {g} est fréquents seulement dans le contexte x et pas dans y. donc {g} est appelé itemset contextuel spécifique fréquent.

III. Approche proposée

3.1. Extraction des itemsets contextuels spécifiques fréquents

3.2.1. Description de l'algorithme

Cet algorithme est inspiré de l'algorithme Apriori, avec nos propres contributions qui sont la proposition d'une structure efficace, pour pouvoir extraire des itemsets fréquents dans chaque contexte. Il est nommé Spécifique Contextuel –Apriori (SC_Apriori).

3.2.2. Pseudo algorithme

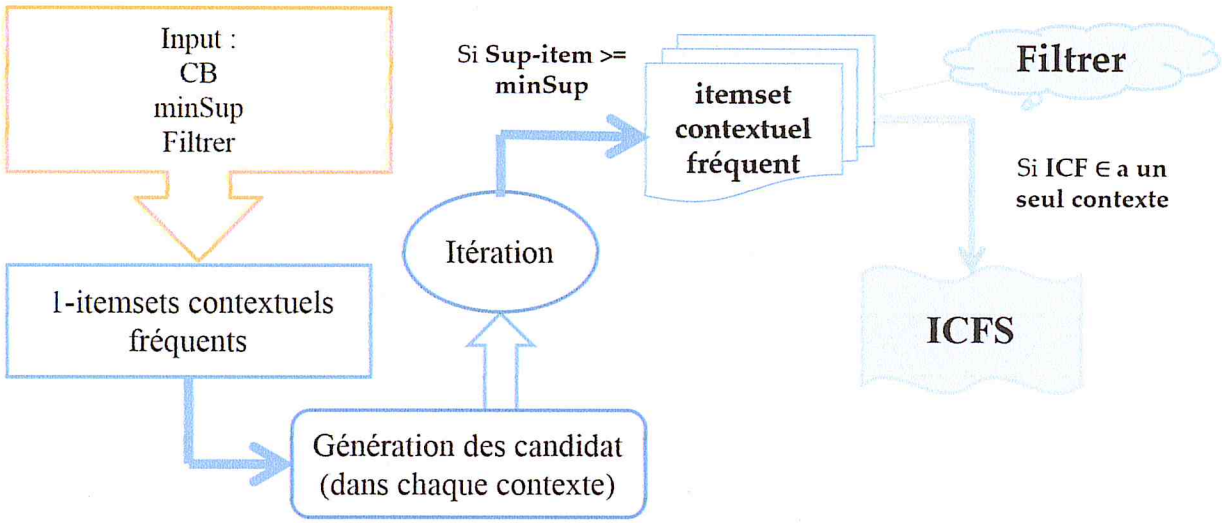


Figure 4.1 : Algorithme Spécifique Contextuel –Apriori (SC_Apriori)

3.2.3. Fonctionnement de l'algorithme SC_Apriori

L'algorithme est présenté dans la récapitulatif1 au-dessous. En utilisant les notions suivantes :

- L_k : ensemble des k-itemsets contextuels candidats et leur contextes dont on ne connait pas encore le support
- C_k : ensemble des k-itemsets contextuels fréquents de taille k

Les itemsets contextuels fréquents sont calculés de façon Les fréquents sont calculés de façon itérative, dans l'ordre ascendant suivant leur taille. A chaque itération, chaque contexte est parcoure une fois et tous les itemsets contextuels de taille k sont générer.

La ligne 1, trouve tous les 1-itemsets contextuels fréquents. L'algorithme alterne ensuite la génération des candidates pour les 1-itemsets qui sont fréquents dans le même contexte.

Par exemple:

1-itemsets$_1$ = {a}, {b}, {e} fréquents dans contexte1 et 1-itemsets$_2$ = {c}, {b}, {e} fréquents dans contexte2. Donc après la génération des candidate nous n'allons pas trouver un itemsets de taille 2 {a, c} puisque les deux itemsets sont pas fréquents dans le même contexte par contre nous allons obtenu les candidate suivante : {a, b}, {a, e}, {c, b}, {c, e}.

La ligne 3 : à l'itération k, l'ensemble C_{k-1} des (k-1)-Itemsets contextuels fréquents correspondant aux itemsets contextuels de niveau (k -1) est utilisé pour générer l'ensemble L_k des k-itemsets contextuels candidats et on calculer pour chaque candidate leur support.

La ligne 4 : nous faisons un filtrage si l'utilisateur veut avoir juste des itemsets contextuels fréquents spécifiques, c'est-à-dire, pour chaque itemsets contextuels fréquents trouvé nous allons vérifier s'il appartient à un seul contexte si est vraie donc nous l'ajouter à l'ensemble des itemsets fréquents spécifique avec son contexte.

- **Récapitulatif1 de l'algorithme**

Algorithme : **SC_ Apriori**

Entrée : CB base de données contextuelle transactionnelle, minsup seuil de support minimum, Filtre est initiales à Vraie
Sortie : ensemble des itemsets contextuels fréquents spécifique ICFS
1 : ICFS = ∅ ;
2 : $C_1 = \{i \mid i \in Ic \wedge sup\{(i)\} \geq minsup\}$; // C1 : 1-itemsets contextuels fréquents
3 : k =2
4 : Tant que $C_k \neq \emptyset$ Faire
L_k = CandidateGeneration (C_{k-1}) ; //L_k : candidate de k-itemsets contextuels
Combine paire de (k-1) - itemsets contextuels qui sont fréquents dans le même contexte.
Calculer le support de chaque candidat $X \in L_k$
$C_k = \{X \mid X \in L_k \wedge sup(X) \geq minsup\}$; // C_k = k-itemsets contextuels fréquents
k=k+1 ;
5: If Filtre = Vraie faire

Pour chaque Ic ∈ $\bigcup_k C_k$ Faire

Si Ic appartient à un seul contexte c :

ICFS ← ICFS ∪ (Ic, c)

Return ICFS

Fin Si

6 : Return C= $\bigcup_k C_k$;

Cette approche soulève cependant une difficulté qu'est les contextes à fouiller sont nombreux. Nous pouvons tomber dans le cas où deux contextes ou plusieurs ont les mêmes itemsets fréquents.

Afin de surmonter cette difficulté, nous avons proposé une autre approche où nous utilisons le clustering pour nous puisons obtenir les contextes les plus pertinents et utiles. Le centroïde qu'est le centre de cluster pour notre cas va représenter le contexte pertinent.

3.2. Extraction des itemsets contextuels fréquents

Cette approche est basée sur deux phases qui sont :

- **Phase 1** : consiste à extraire les contextes pertinents en utilisant l'algorithme de clustering Kmeans
- **Phase 2** : consiste à extraire des itemsets contextuels fréquents en utilisant l'algorithme Apriori où bien SC_Apriori si nous voulons avoir les itemsets contextuels qu'ils sont spécifique à un contexte pertinent comme illustre dans la figure au-dessous.

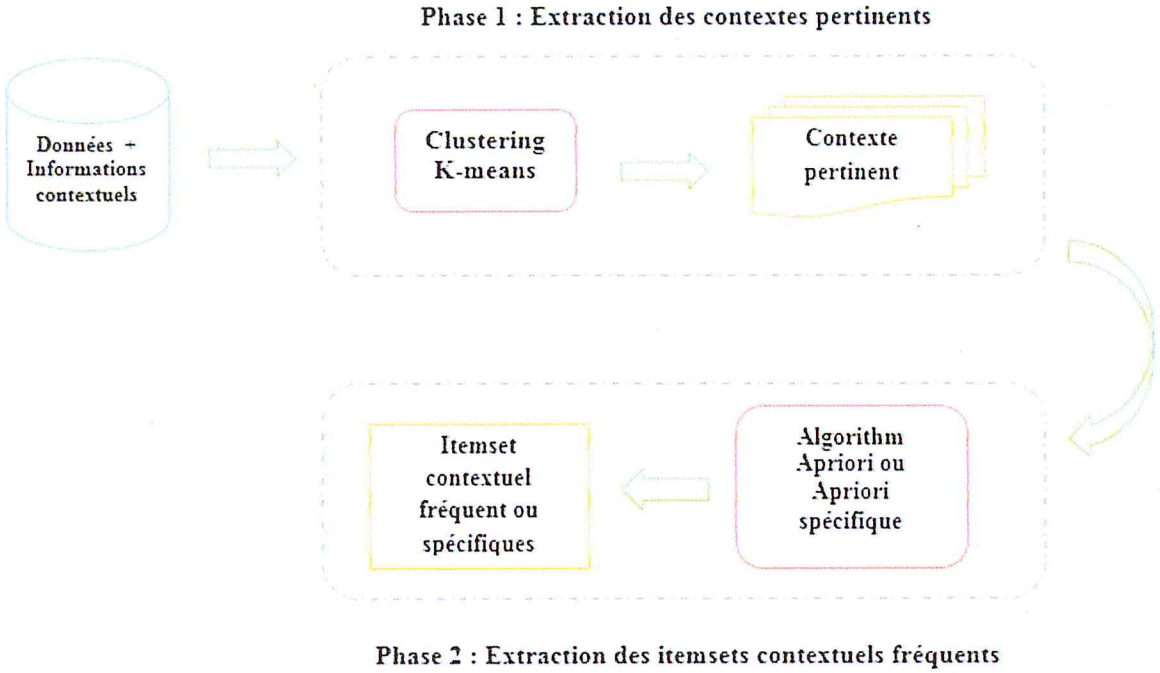


Figure 4.2 : schéma global de l'approche proposée

3.2.1. Phase 1 : Extraction des contextes pertinents

Pour extraire les contextes pertinents en utilise le clustering. Clustering est considéré comme la technique la plus importante de l'apprentissage non supervisé. Pour cette phase nous allons implémenter le package K-means [55]

3.2.1.1. Nombre de cluster k

La détermination du nombre optimal de clusters dans un ensemble de données est un problème fondamental dans le partitionnement des clusters. Donc, nous avons essais trois méthodes qui nous aidons à sélectionner le nombre de cluster.

a) Méthode du coude

L'idée de la méthode du coude (Elbow method en anglais) est d'exécuter l'algorithme de clustering k-means sur une data pour une plage de valeurs de cluster k (par exemple k = [2, 20]), et pour chaque valeur de k, calculer la somme des erreurs au carré (sum of squared errors en anglais), noté par SSE.

SSE est défini comme la somme de la distance carrée entre chaque membre du cluster et so_n centre :

$$SSE = \sum_{i=1}^{k} \sum_{x \in ci} dist(x, c_i)^2 \quad \ldots\ldots\ldots\ldots (1)$$

Ensuite, tracez un diagramme en lignes du SSE pour chaque valeur de k. Nous verrons que l'erreur diminue à mesure que k s'augmente. En effet, lorsque le nombre de clusters s'augmente, elles doivent être plus petites, de sorte que la distorsion est également moindre L'idée de la méthode du coude est de choisir le k auquel SSE diminue brusquement. Cela produit un "effet de coude" dans le diagramme, comme il est illustre dans la figure 4.3.

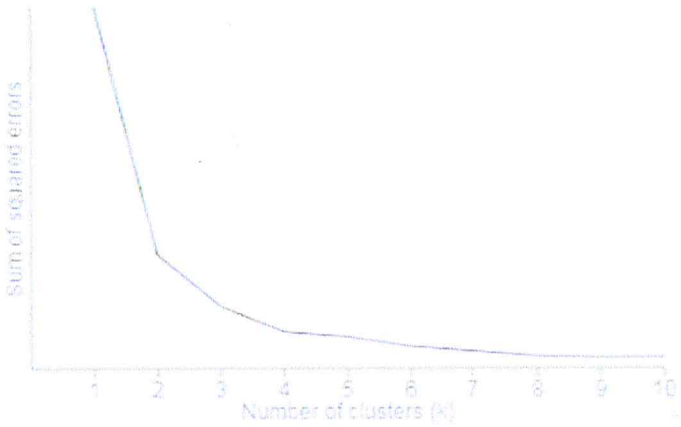


Figure 4.3: Exemple sur la méthode du coude [56]

Exemple

Dans cet exemple, nous avons utilisé un jeu de données[2] généré. Dans ce jeu de donnée, le nombre de transaction est égale à 600, le nombre de l'item est égal à 100 et nombre de contexte associé à chaque transaction est égal à 10. Le résultat obtenu est illustre dans la figure suivant.

[2] https://archive.ics.uci.edu/ml/datasets.html

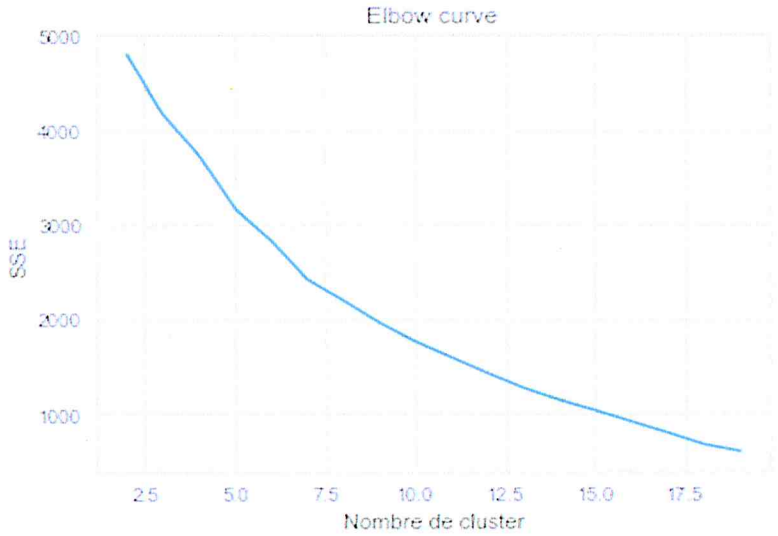


Figure 4.4: résultat en utilisant la méthode du coude

D'après la figure 4.4, nous remarquons que le diagramme de coude du jeu de donnés ne comporte pas de coude clair. Au lieu de cela, nous voyons une courbe assez lisse, et on ne sait pas quelle est la meilleure valeur de k à choisir. Dans de tels cas, nous avons essayé d'autres méthodes différentes pour déterminer le k optimal, tel que le calcul des scores de silhouette et gap static

b) Silhouette Analyses

Silhouette [57] est un graphe montrant comment chaque observation appartient plus ou moins à son cluster. Supposons que n observations aient été réparties en k cluster par un quelconque algorithme. Soit a(i) la moyenne des dissimilarités (ou distances) de l'observation i avec toutes les autres observations au sein d'un même cluster. Plus a(i) est petit meilleur est l'assignation de i à son cluster, a(i) est la dissimilarités moyenne de i à cet cluster. Soit b(i) la plus faible moyenne des dissimilarités (ou distances) de l'observation i à chaque autre cluster dont ne fait pas i partie. Le cluster avec cette plus faible dissimilarité moyenne est appelé cluster voisine de i car c'est le meilleure cluster suivante pour l'observation i.

La silhouette de l'*i*ème observation est alors donnée par :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad \ldots\ldots\ldots\ldots\ldots (2)$$

Exemple

Nous utilisons le même jeu de données utilisant dans la méthode de coude. Le tableau suivant présenter les valeurs de score moyen de silhouette pour nombre maximum de clusters égale à 20.

Tableau 4.2 : le score moyen de silhouette pour chaque nombre de clusters

Nombre de clusters (k)	Le score moyen de silhouette
2	0.41640777192683104
3	0.42916675596960624
4	0.4085636317494554
5	0.4473372036115915
6	0.4716590246215364

D'après le tableau 4.2, nous voyons que pour k = 6, le score est le plus élevée donc d'après la définition de la méthode silhouette, nous déduisions que k=6 est le nombre optimale de clusters.

c) La statistique du gap

- **Concept**

La statistique du gap a été publiée par Tibshirani et al. [58]. L'approche peut être appliquée pour n'importe quelle méthode de clustering (exemple : k-means et hiérarchique clustering). La statistique du gap compare la variation totale intracluster pour des différentes valeurs de k avec leurs valeurs attendues sous une distribution de référence nulle des données, c'est-à-dire une distribution sans clustering évident.

Le jeu de données de référence est généré à l'aide de simulations Monte Carlo du processus d'échantillonnage. C'est, pour chaque variable (x_i) dans l'ensemble de données, nous calculons sa plage $[min(x_i), max(x_i)]$ et générer des valeurs pour les n points uniformément de l'intervalle min à max.

Pour les données observées et les données de référence, la variation totale d'intracluster est calculée en utilisant différentes valeurs de *k*. La statistique du gap pour un *k* donné est définie comme suit:

$$\mathrm{Gap_n(k)} = \mathrm{E_n^*}\log(\mathrm{W_k}) - \log(\mathrm{W_k}) \ldots\ldots\ldots\ldots \quad (3)$$

Tel que :

- $\mathrm{E_n}^*$: dénote l'attente sous une taille d'échantillon n de la distribution de référence $\mathrm{E_n}^*$ est défini par bootstrapping (B) en générant des copies des jeux de données de référence B et en calculant la moyennelog$(\mathrm{W_k^*})$.

La statistique du gap mesure l'écart de la valeur de $\mathrm{W_k}$ observée par rapport à sa valeur attendue sous une hypothèse nulle. L'estimation des clusters optimaux $(\hat{\mathrm{k}})$ sera la valeur qui maximise $\mathrm{Gap_n(k)}$, c'est-à-dire, la valeur qui donne la plus grande statistique d'écart. Cela signifie que la structure de clustering est loin de la distribution uniforme des points.

Un écart type, noté par $\mathrm{sd_k}$ de $\log(\mathrm{W_k^*})$ est également calculé afin de définir l'erreur type noté par $\mathrm{s_k}$ de la simulation comme suit :

$$\mathrm{s_k} = \mathrm{sd_k} \times \sqrt{1 + 1/\mathrm{B}} \ldots\ldots\ldots\ldots\ldots (4)$$

Enfin, une approche plus robuste consiste à choisir le nombre optimal de cluster K comme le plus petit k. C'est-à-dire que nous choisissons la plus petite valeur de k de sorte que la statistique de l'écart se situe à moins d'un écart-type de l'écart à k + 1 comme présentée dans l'équation suivante.

$$\mathrm{Gap(k)} \geq \mathrm{Gap(k+1)} - \mathrm{s_{k+1}} \ldots\ldots\ldots\ldots (5)$$

- **Algorithme**

L'algorithme comprend les étapes suivantes:

- Cluster les données observées en faisant varier le nombre de clusters pour $\mathrm{k} = 1, \ldots\ldots\ldots, \mathrm{k_{max}}$ et calculer le $\mathrm{W_k}$ correspondant.
- Générer des ensembles de données de référence B et regrouper chacun d'eux avec un nombre de cluster $\mathrm{k} = 1, \ldots\ldots\ldots, \mathrm{k_{max}}$. Calculer les statistiques d'écart estimées présentées dans l'équation (3).
- Laisser $\overline{\mathrm{w}} = (1/\mathrm{B})\sum_{\mathrm{b}}\log(\mathrm{W_{bk}^*})$, calculer l'écart type comme suite :

$$sd(k) = \sqrt{(1/B)\sum_b \left(\log(W^*_{bk}) - \overline{w}\right)^2} \ldots\ldots (6)$$

Ensuite, on définit l'erreur type s_k comme présente dans l'équation (4).

- Choisissez le nombre de clusters comme le plus petit k comme présente dans l'équation (5).

Exemple

Nous utilisons le même jeu de données utilisant dans la méthode de coude. Les résultats obtenus en utilisant la méthode de statistique du gap pour le nombre de cluster k = [2, 20] est illustre dans la figure suivant.

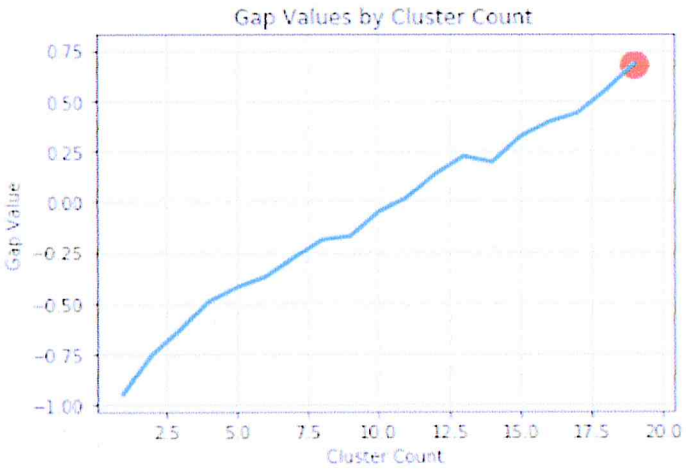


Figure 4.5: la représentation graphique des résultats

D'après la figure 4.5, nous voyons que le nombre optimal de cluster est comprise dans la plage [17.5 ,20] comme est indiqué par le point rouge.

Après avoir obtenir le nombre optimal de cluster, nous allons maintenant applique l'algorithme de k-means. Dans la partie suivante, nous présenterons brièvement les concepts de cet algorithme.

3.2.1.2. Algorithme K-means

K-means [59] est l'un des algorithmes de clustering populaires et simples. Le but de cet algorithme est de trouver des clusters dans les données fournies. La figure suivante illustre un exemple de clustering utilisant le k-means. La figure suivante illustre un exemple de clustering utilisant k-means. Tel que, le diagramme situé à gauche montre un jeu de données avant le regroupement et à droite le jeu de données après le clustering.

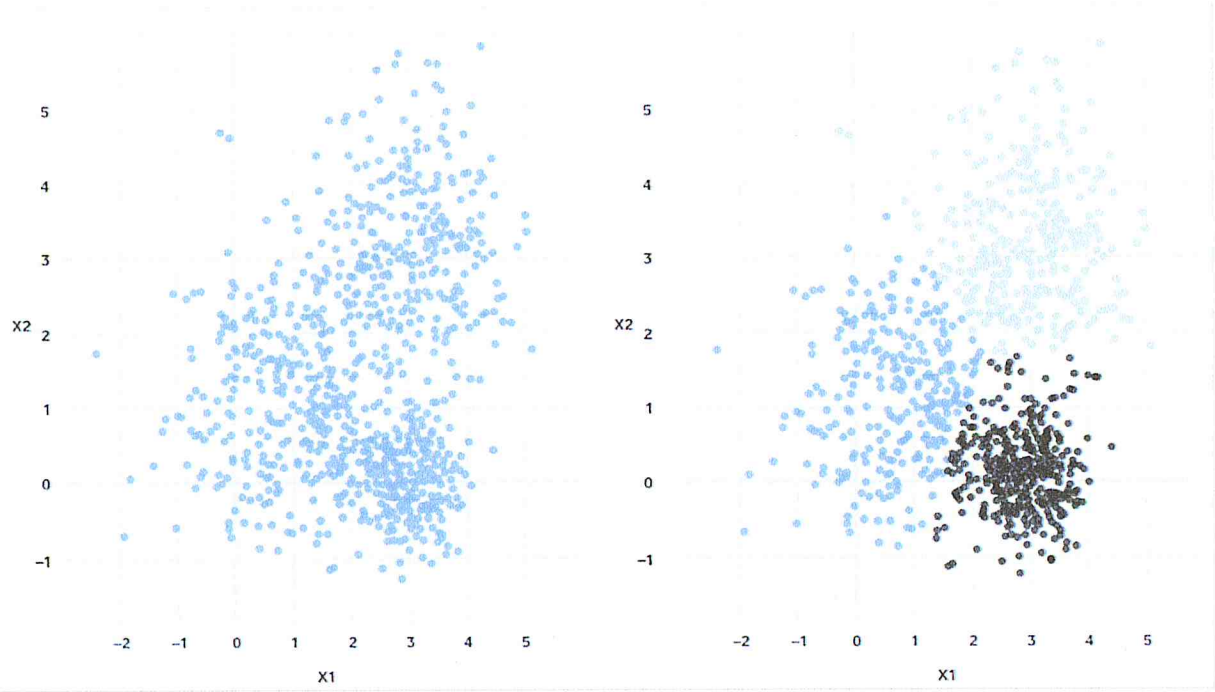


Figure 4.6 : Prendre un jeu de données à 2 dimensions et le séparer en 3 groupes distincts [60]

a) Algorithme

L'algorithme comprend les étapes suivantes:

✓ **Etape 1 :** choisir aléatoirement k centres de clusters, nommé centroïdes. Supposons que ce sont $c_1, c_2, \dots \dots, c_k$ et on peut dire que :

$$\mathbf{C} = c_1, c_2, \dots \dots, c_k$$

Où C est l'ensemble de tous les centroïdes.

✓ **Etape 2** : dans cette étape, on assigne chaque variable, noté par xi, dans l'ensemble de données au centre le plus proche. Cela se fait en calculant la distance euclidienne entre le point dans l'ensemble de données et chaque centroïde.

$$\underset{c_i \in C}{\operatorname{argmin}} \operatorname{dist}(c_i, x)^2 \dots \dots \dots \quad (7)$$

Où dist(.) est la distance euclidienne.

✓ **Etape 3 :** consiste à trouver un nouveau centroïde en prenant la moyenne de tous les points attribués à ce cluster.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i \ldots\ldots\ldots (8)$$

Ou S_i est l'ensemble de tous les points assignés à l'iéme cluster.

- ✓ **Etape 4 :** dans cette étape, répétez les étapes 2 et 3 jusqu'à ce qu'aucune des affectations de cluster ne change. Cela signifie que jusqu'à ce que les clusters restent stables.

3.2.2. Phase 2 : Extraction des itemsets contextuels fréquents

Dans cette phase, nous avons le choix c'est nous voulons voire juste les itemsets contextuels spécifiques donc nous allons utiliser l'algorithme de SC_Apriori sinon en utilisant l'algorithme Apriori qui va nous retourne un ensemble des itemsets contextuels fréquents pour chaque contexte et pour cela nous pouvons obtenir un itemset fréquent qu'il est fréquent dans un ou plusieurs contextes.

IV. Conclusion

Dans ce chapitre, nous avons présenté les deux approches proposées pour l'extraction des itemsets contextuels fréquents. Le chapitre suivante est consacré à démontre l'efficacité de ces méthodes grâce au résultat obtenu.

Chapitre V

Test et Validation

I. Introduction

Après avoir décrit notre solution, nous aborderons dans ce chapitre la partie des expérimentations de notre approche.

Ce chapitre sera organisé du la manière suivante. Dans la section 2, nous allons présenterons l'environnement de travail et l'outil de développement utilisé. Ensuite, nous présenterons notre jeu d'essais et tests et des exemples pour mieux comprendre les approches proposées dans la section 3. Enfin, nous mettons une conclusion pour ce chapitre.

II. Environnement de développement

2.1. L'environnement matériel

Nous avons utilisé :

- Un ordinateur portable Lenovo avec les caractéristiques suivantes :
  - 4 GO RAM
  - 4096 MBytes DDR3
  - Intel (R) Core (TM) i3-2350M CPU @ 2.30 GHz
  - Système d'exploitation : Windows 7 de 64-bit

2.2. L'environnement logiciel

2.2.1. Python

Python est un langage de programmation dynamique de haut niveau, interprété et polyvalent qui met l'accent sur la lisibilité du code. La syntaxe en Python aide les programmeurs à coder en moins d'étapes que Java ou C+++. Le langage a été fondé en 1991 par le développeur Guido Van. Le Python est largement utilisé dans les grandes organisations en raison de ses multiples paradigmes de programmation. Ils impliquent généralement une programmation fonctionnelle impérative et orientée objet. Il dispose d'une bibliothèque standard complète et volumineuse qui dispose d'une gestion automatique de la mémoire et de fonctions dynamiques [61].

Python est accompagné d'un grand nombre de bibliothèques intégrées. Par exemple scikit-learn pour l'exploration de données, l'analyse de données.

2.2.2. Jupiter Notebook

Jupyter Notebook est une application web open-source qui vous permet de créer et de partager des documents contenant du code, des équations, des visualisations et du texte explicatif, Jupyter Notebook est utilisé pour [62]:

- Nettoyage et transformation de données.
- Simulation numérique.
- Modélisation statistique.
- Visualisation de données.

2.2.3. Bibliothèques utilisées

- **Pandas**

Pandas est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles [63].

- **Scikit-learn**

Scikit-learn est une bibliothèque libre Python dédiée à l'apprentissage automatique. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques de Python [64].

- **Mlxtend**

Mlxtend est une bibliothèque d'apprentissage automatique pour python qui contient une implémentation de l'algorithme apriori [65].

III. Expérimentation et tests

3.1. Description des données

Notre test est fait avec le jeu de donnée de Wiki [66]. Dans ce jeu de donnée, le nombre est égale à 912 transactions, le nombre des items est égale à 215. Les informations contextuelles associées aux transactions contiennent 30 contextes (la description de ce jeu de

donnée se trouvée dans les annexes). Le contexte dans ce jeu de donnée représente le profil d'enseignant.

3.2. Extraction des itemsets fréquents spécifique

Dans cette partie, nous avons effectué des tests pour l'évaluation de la performance de l'algorithme SC_ Apriori pour extraire les itemsets fréquents spécifique pour chaque contexte de jeu de données Wiki.

3.2.1. Tests le temps d'exécution

Les tests de temps d'exécution par rapport au nombre du contexte et au seuil minimale, nous ne faisons pas un test pour le nombre de transaction puisque le nombre de transaction diffère de contexte à un autre.

3.2.1.1. Par rapport au nombre du contexte

Ce test est fait pour évaluer la performance de l'approche par rapport au nombre de contexte et un minsup = 0.05 pour trouves tous les itemsets possibles. Le temps d'exécution est donné en milliseconde (ms).

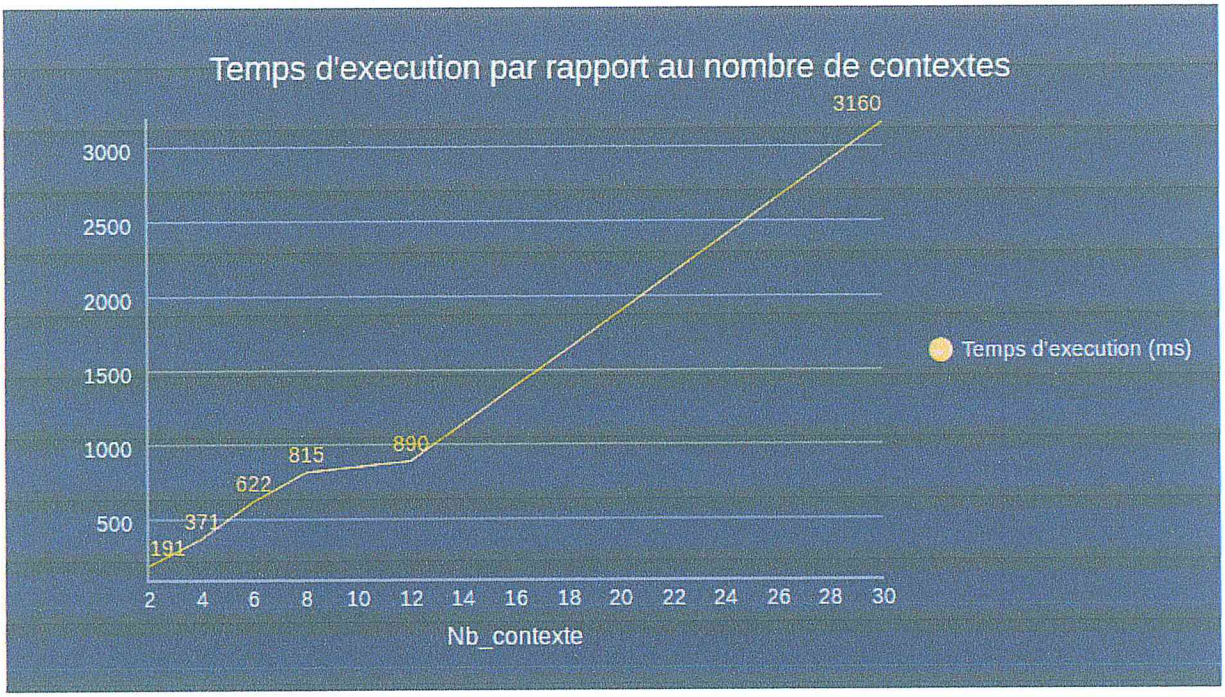


Figure 5.1 : Temps d'exécution par rapport au nombre du contexte

Interprétation :

On remarque que le temps d'exécution augmente quand le nombre de contextes augmente. Avec 12 contextes, notre approche prend presque 890 ms pour terminer le calculer. La raison pour cela est que dans notre approche nous travaillons avec des contextes où le nombre de transaction se diffère

3.2.1.2. Par rapport au seuil minimal

Ce test est fait avec le même jeu de donnée utiliser précédemment, Ce test est fait pour évaluer la performance de l'approche avec des supportes différents, de 0.05 (5%) jusqu'à 0.5 (50%). Le temps d'exécution est donné en seconde (s).

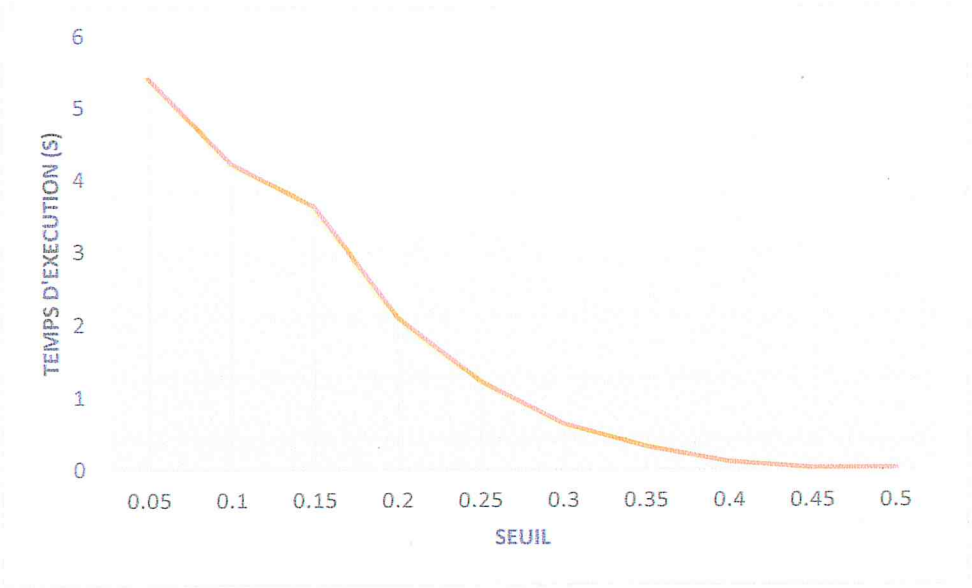


Figure 5.2: temps d'exécution par rapport aux différents supports

Interprétation

On remarque que le temps d'exécution augmente quand le support diminue. La raison pour cela est que chaque contexte contient 215 items si on diminuer le support l'espace de recherche augment.

3.2.2. Tests consommation du mémoire

Nous testons la consommation du mémoire par rapport au nombre du contexte et seuil minimale.

3.2.2.1. Par rapport au nombre du contexte

Ce test est fait pour évaluer la performance de l'approche par rapport au nombre de contexte et un minsup = 0.05 pour trouves tous les itemsets possibles.

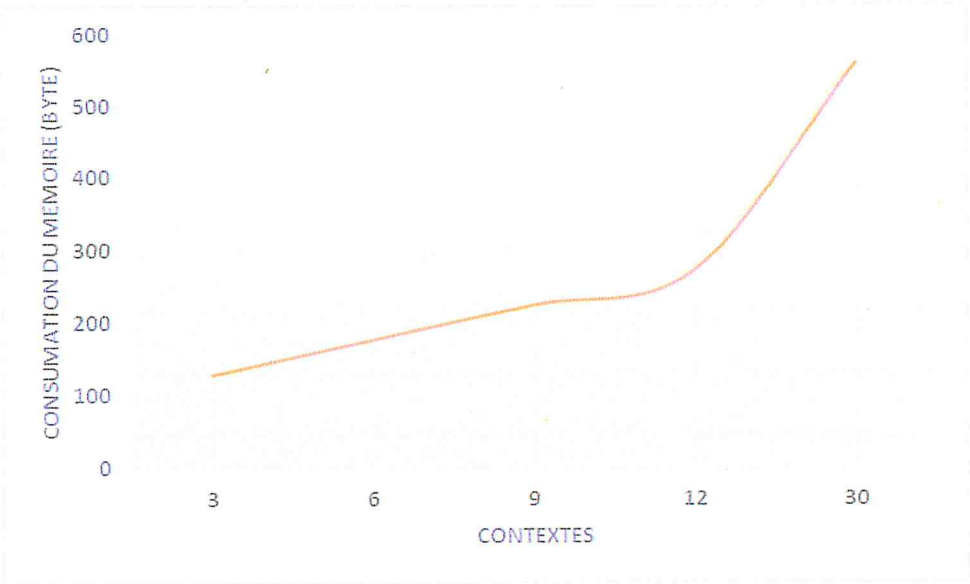


Figure 5.3 : Consommation de mémoire par rapport au nombre de contexte

Interprétation

On remarque que consommation de mémoire s'augmente lorsque le nombre de contexte s'augment.

3.2.2.2. Par rapport au seuil minimal

Ce test est fait pour évaluer la performance de l'approche avec des supportes différents, de 0.1 (10%) jusqu'à 0.5 (50%). Le nombre de contexte est égal à 30

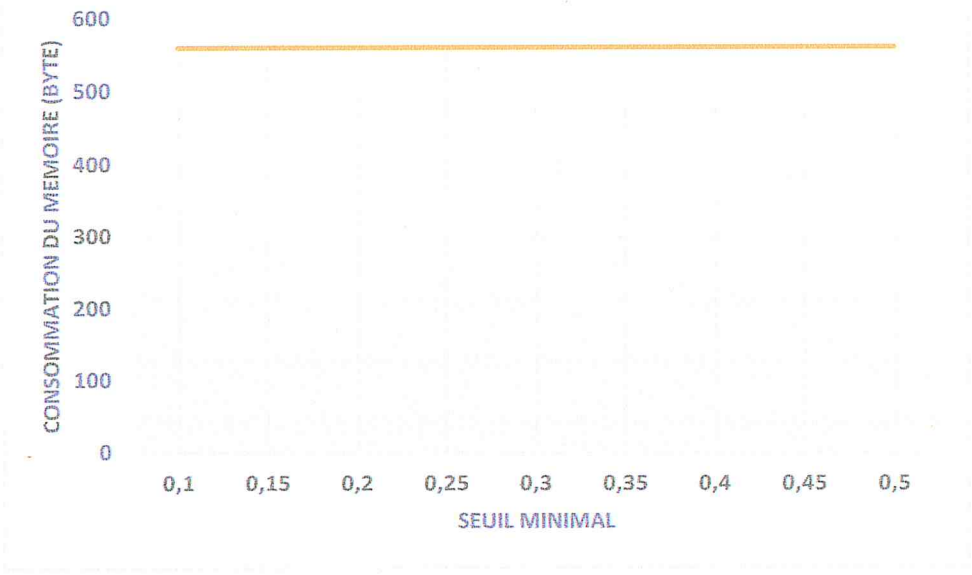


Figure 5.4 : Consommation de mémoire par rapport au seuil minimal

Interprétation

On remarque que consommation de mémoire ne s'augment pas lorsque le support se diminue pas comme le temps d'exécution. Donc nous conclurons que le nombre de contexte est le seul qui puisse affecter la consommation du mémoire

Exemple illustratif

Prenons une base de données qui décrit les achats effectués par des différents clients et leurs informations sont représentées de la façon suivante :

	Client_Genre	Client_Ville	Married	N_Children	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
client1	FEMALE	INNER_CITY	YES	YES	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0
client2	MALE	TOWN	YES	NO	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0
client3	FEMALE	INNER_CITY	YES	YES	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0
client4	FEMALE	TOWN	YES	YES	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0
client5	FEMALE	RURAL	YES	NO	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0
client6	FEMALE	TOWN	YES	YES	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0
client7	FEMALE	RURAL	YES	NO	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0
client8	MALE	TOWN	YES	NO	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
client9	FEMALE	SUBURBAN	YES	YES	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0
client10	MALE	TOWN	YES	NO	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0

Figure 5.5 : les achats des clients et l'information contextuelle

Où le client1 a acheté l'article1 et article3, le client2 a achète l'article2 et l'article6 et ainsi de suite.

Ensuite quand nous perrons en considération les informations contextuelles, nous aurons chaque client va appartient à un contexte comme illustre dans la figure suivante :

	Client_Genre	Client_Ville	Married	N_Children	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
client1	FEMALE	INNER_CITY	YES	YES	1	0	1	0	0	0	1	0	1	0	0	0	0	0	1
client3	FEMALE	INNER_CITY	YES	YES	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0
client5	FEMALE	RURAL	YES	NO	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0
client7	FEMALE	RURAL	YES	NO	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0
client9	FEMALE	SUBURBAN	YES	YES	1	0	0	0	1	0	1	0	1	0	0	1	0	0	0
client4	FEMALE	TOWN	YES	YES	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0
client6	FEMALE	TOWN	YES	YES	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0
client2	MALE	TOWN	YES	NO	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0
client8	MALE	TOWN	YES	NO	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0

Figure 5.6 : regroupement des clients par rapport aux contextes

Où le client1 et client3 appertisant au contexte « une femme mariée et a d'enfants de Inner city», le client2 et client8 appertisant au contexte « un homme mariée et n'a pas d'enfants de Town» et ainsi de suite.

Nous supposons que le minsup = 0.5, si nous appliquons l'algorithme SC_Apriori nous obtenions le résultat suivants :

- Article3, Article11 et Article15 sont fréquent seulement dans le contexte « une femme mariée et a d'enfants de Inner city»
- Article4 et Article14 sont fréquent seulement dans le contexte « une femme mariée et n'a pas d'enfants de Rural»
- Article2 et Aritcle10 sont fréquent seulement dans le contexte « un homme mariée et n'a pas d'enfants de Town»

L'algorithme SC_Apriori permet d'extraire des itemset fréquents qui sont spécifique à un contexte donnée. Ces itemsets pourront aidée l'expert à adapter sa stratégie au type du client et prendre des décisions adéquates.

3.3. Extraction des itemsets contextuels fréquents

Dans cette partie, nous avons effectué des tests par rapport à la qualité des résultats en utilisant l'algorithme apriori pour extraire les itemsets qui sont fréquents dans chaque contextes pertinents. Nous travaillons avec le même jeu de données utilisé précédemment.

3.3.1. Extraction des contextes pertinents

Dans cette phase, nous utilisions k-means sur les contextes afin d'obtenir les contextes les plus pertinentes. Nous avons utilisé la méthode de silhouette pour obtenir le nombre optimale de cluster k qu'est égale à 6. Pour plus de détails sur la méthode consulter le chapitre 4.

Nous avons obtenu les résultats suivants :

- **Contexte1** : « *les enseignants qu'ont l'âge entre [20, 40[, ils sont des professeurs associés de l'université UOC et ils n'ont pas leur doctorats.*»
- **Contexte2** : « *les enseignants qu'ont l'âge entre [40, 60[, ils ont leur doctorats dans le domaine arts et sciences humaines.*»
- **Contexte3** : « *les enseignants qu'ont l'âge entre [30, 60[, ils sont des professeurs associés de l'université UPF.*»
- **Contexte4 :** « *les enseignants qu'ont l'âge entre [20, 50[, ils sont des professeurs associés de l'université UOC. Ils n'ont pas un compte d'utilisateur dans Wikipédia.*»
- **Contexte5 :** « *les enseignants qu'ont l'âge entre [30, 50[, ils travaillent dans l'université UOC.*»
- **Contexte6** : « *les enseignants qu'ont l'âge entre [20, 60[, ils travaillent dans l'université UPF.*»

3.3.2. La qualité des résultats

Nous avons travaillé avec un minsup qu'est égale à 0.3, nous comparons par rapport aux nombres des itemsets fréquents trouvée dans une base contextuelle d'itemsets et la base de transaction traditionnelle, c'est-à-dire, sans considère les contextes.

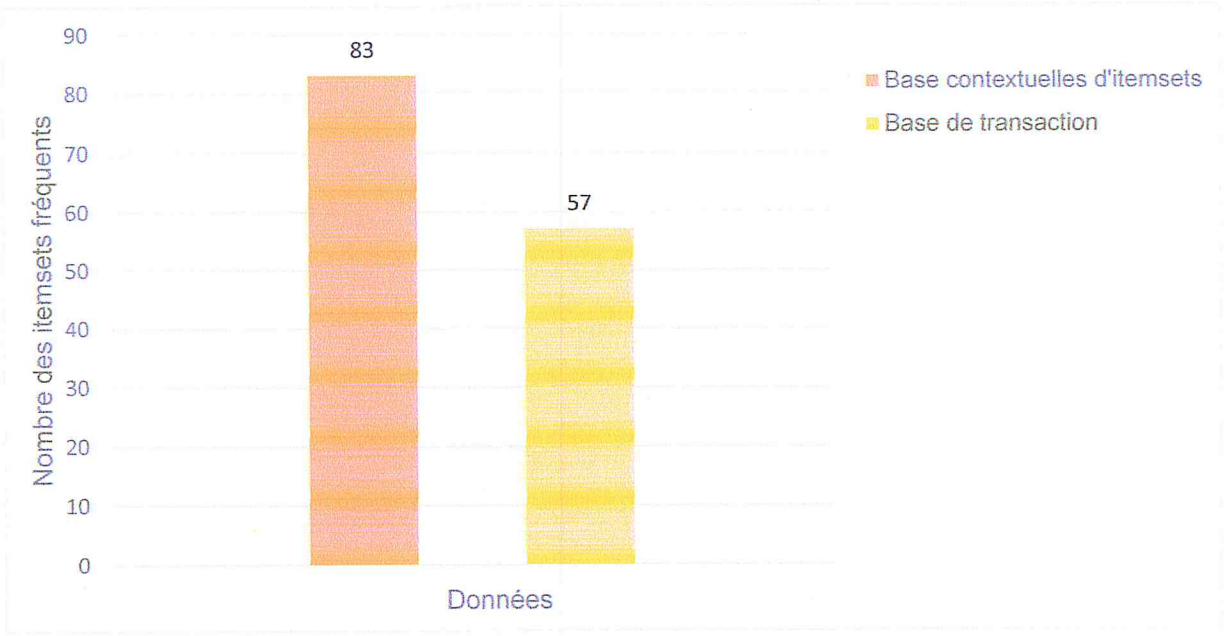


Figure 5.7 : Nombre des itemsets fréquents

Interprétation

Nous remarquons que si nous considérons les informations contextuelles dans le processus d'extractions des itemsets fréquents, cela à permettre d'augmente la pertinence des informations que le processus d'extraction à fournir.

3.3.3. Extraction des itemsets contextuel

Si nous utilisons l'algorithme Apriori dans cette phase, nous allons obtenu toutes les itemsets fréquents dans un contexte pertinent et ce dernier peut être fréquents dans un autre contexte où pas. Pour mieux comprendre, nous allons utiliser une figure qui illustre une partie de résultat obtenu de cette phase pour deux contextes qui sont contexte2 et contexte3 obtenu de la phase précédant.

	support	itemsets
0	0.360656	{'PU1_D'}
1	0.442623	{'PU2_D'}
2	0.327869	{'PU3_D'}
3	0.344262	{'PEU1_E'}
4	0.475410	{'PEU1_F'}
5	0.426230	{'PEU2_E'}
6	0.327869	{'PEU2_F'}
7	0.344262	{'PEU3_D'}
8	0.377049	{'ENJ1_E'}
9	0.426230	{'ENJ2_E'}

	support	itemsets
0	0.358209	{'PU1_D'}
1	0.378109	{'PU2_D'}
2	0.313433	{'PU3_D'}
3	0.348259	{'PEU1_E'}
4	0.572139	{'PEU1_F'}
5	0.417910	{'PEU2_E'}
6	0.368159	{'PEU2_F'}
7	0.383085	{'PEU3_D'}
8	0.412935	{'ENJ1_E'}
9	0.472637	{'ENJ2_E'}

Figure 5.8 : Comparaison des resultats de context2 (tableau droite) avec context3 (tableau gauche)

D'après la figure 5.8, nous remarrquons que les deux contextes ont les mémes itemsets fréquents. Ces itemsets, nous permttons de conclure par exemple, pour les deux contextes tous les enseignants sont d'accord que l'utilisation de Wikipedia stimule la curiosité et le divertit ({ENJ2_E}, {ENJ1_E}).

3.4. Extraction des itemsets spécifique fréquents en utilisant des contexte pertinent

Dans cette section, nous allons effectuer des tests pour l'évaluation de la performance de l'algorithme SC_ Apriori pour extraire les itemsets fréquents spécifique en utilisant des contextes pertinents.

3.4.1. Temps d'exécution

Nous avons teste avec un minsup = 0.05 pour trouves tous les itemsets possibles pour les contextes pertinent. Le résultat est illustre dans la figure au-dessous.

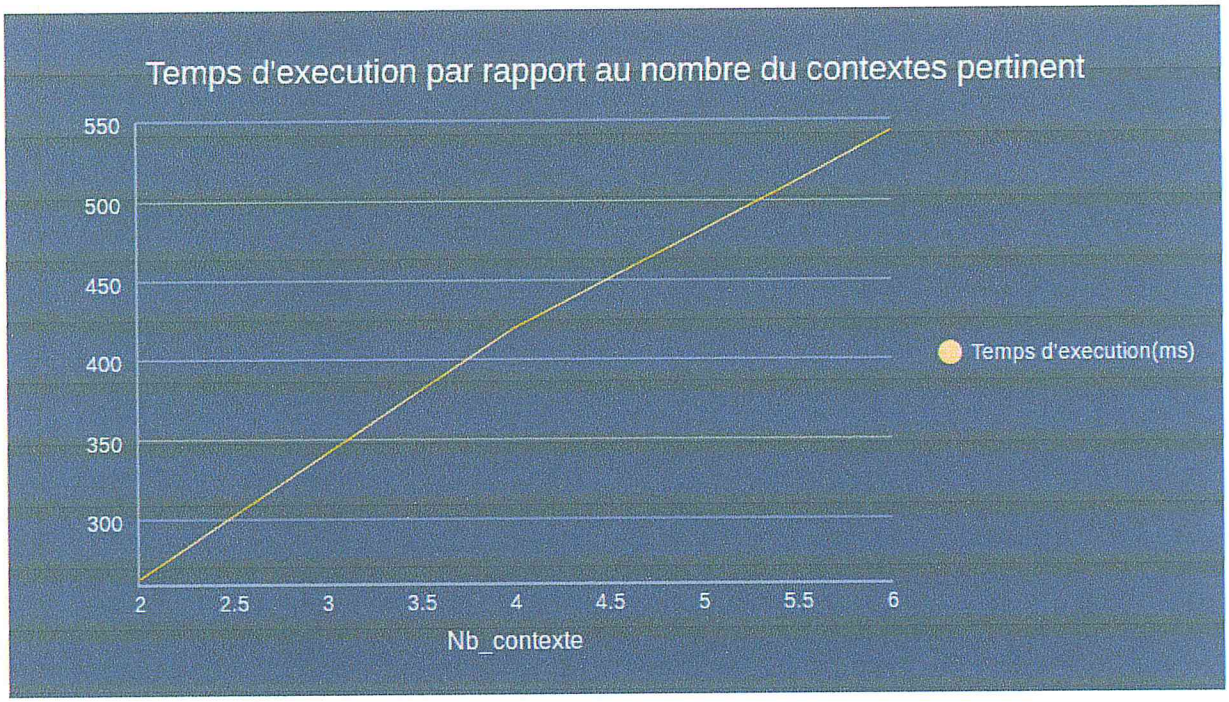


Figure 5.9 : Temps d'exécution de SC_Apriori par rapport au nombre de contextes pertinent

Interprétation :

On remarque que le temps d'exécution augmente quand le nombre de contextes augmente.

3.4.2. Consommation de mémoire

Nous testons avec un minsup =0.05 (5%) par rapport au nombre de contextes pertinents. Les résultats sont illustres dans la figure au-dessous.

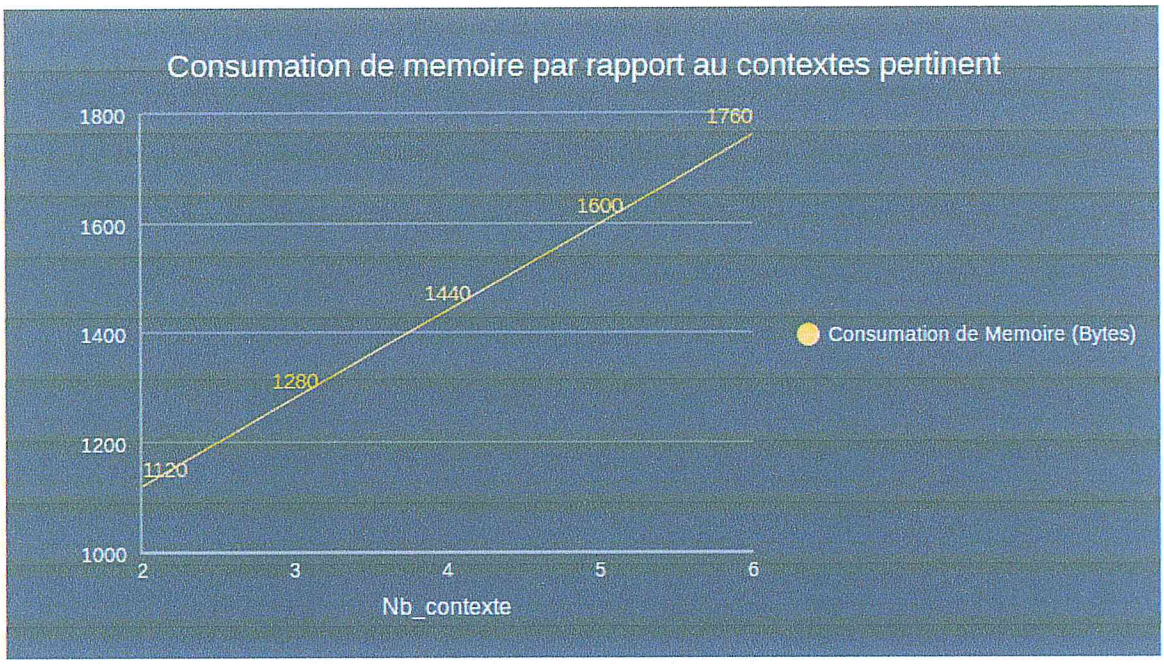


Figure 5.10 : Consommation de mémoire SC_Apriori par rapport au nombre de contexte pertinent

Interprétation

On remarque que consommation de mémoire s'augmente lorsque le nombre de contexte s'augment.

Par rapport au seuil minimal, quel que soit le seuil choisi la consommation de mémoire ne change pas.

3.4.3. Discussion

Nous utilisons cet algorithme juste pour répondre au besoin spécifique, c'est-à-dire, si nous prenons le jeu de données Wiki si un expert vaut avoir l'avis des enseignants qui appariant à un contexte particulier il peut utiliser l'algorithme de SC_Apriori mais si il vaut savoir s'il existe des avis en commun entre des enseignants qui appariant aux différents contextes dans ce cas il préférable d'utiliser apriori.

Nous allons utiliser le même exemple illustratif pour mieux comprendre cette approche et voir la différence entre les deux approches proposées.

Exemple Illustratif

Nous prenons la même base de données et nous allons appliquer le Kmeans pour avoir les contextes les plus pertinents. Les contextes pertinent qui nous avons obtenir sont comme suivante :

- Contexte 1 = « *femmes mariées de n'importe quelle ville et elles peuvent avoir d'enfants où no* »
- Contexte 2 = « *hommes et femmes sont mariée de ville Town et peuvent avoir d'enfants où no* »
- Contexte 3 = « *hommes et femmes mariés de n'importe quelle ville et ils n'ont pas d'enfants* »

Après avoir obtenir les contextes les plus pertinents, nous allons faire l'extraction des itemsets contextuels fréquents avec un minsup = 0.2.

- **Résultat donnée par Apriori :**
  - Contexte 1 = {Article1, Article7, Article9, Article3, Article6, Article8}
  - Contexte 2 = {Article1, Article2, Article6, Article7, Article9, Article10, Article13}
  - Contexte 3 = {Article1, Article2, Article4, Article6, Article7, Article8, Article14}

D'après Apriori, nous voyons que l'Article1, 6 et 7 sont fréquents dans toutes les contextes cela nous permettons de conclure que ces articles sont achats par tous les clients.

- **Résultat donnée par SC_Apriori :**
  - Contexte 1 = {Article3}
  - Contexte 2 = {Article10, Article13}
  - Contexte 3 = {Article4, Article14}

IV. Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel et l'environnement logiciel, les bibliothèques utilisées pour implémenter notre approche. On a effectué des tests pour vérifier la qualité des résultats de notre approche et son l'efficacité.

Conclusion Générale

I. Conclusion

L'extraction des itemsets à partir de données en prennent en considération les informations contextuelles est une technique qui utilise une variété d'outils d'analyse de données pour découvrir. Ces connaissances découvertes sont très intéressantes, elles sont utilisées dans différents domaines pour faire des prédictions valides et / ou prendre des décisions.

Pour comprendre et cerner la problématique de l'extraction d'itemsets contextuels fréquents, nous avons consacré le premier chapitre à l'étude des différents algorithmes classiques d'extraction des itemsets fréquents. Dans deuxième chapitre, on a présenté les modèles existants pour représentes les informations contextuelles avec une comparaison selon des critères bien précis. Dans le troisième chapitre, on a introduit les algorithmes d'extraction des itemsets contextuels fréquents. Le Quatrièmes chapitre présente la solution proposée, l'algorithme SC-apriori qu'est inspiré de l'algorithme Apriori. Le dernier chapitre a été consacré aux tests et validation de la solution proposée.

Ce travail nous a permis d'approfondir nos connaissances sur la notion d'extraction d'itemsets fréquents et la notation de contexte.

II. Perspectives

Les perspectives de ce travail préliminaire sont nombreuses. Nous terminons notre travail en dégagent quelques perspectives :

- Il est probablement intéressant de pouvoir créer des méthodes parallèles distribuées pour extraire des itemsets contextuelles fréquents à partir de données contextuelles dans le contexte de Big Data.
- Développer une approche s'adapte aux différents types de modèle du contexte puisque l'approche proposée traite juste des contextes qui sont modélisés en utilisant le modèle clé et valeur.
- Les travaux futurs incluront des expérimentations sur différents jeux de données réels, ainsi on étudiera également comment les résultats obtenus peuvent être exploités pour la classification.
- Travailler sur l'extraction d'autres types de motifs en intégrant les données contextuelles, telles que les motifs séquentiels, les sous graphes, les motifs spatiotemporels, ...etc.

- Intégration de l'aspect contextuel dans l'extraction des motifs à partir de données incertaines.

Bibliographie

[1] M. Kantardzic, "Data Mining–Concepts, Models, Methods, and Algorithms," IEEE Press, Piscataway, NJ, USA, 2003. M. Kantardzic, "Data Mining–Concepts, Models, Methods, and Algorithms," IEEE Press, Piscataway, NJ, USA, 2003.

[2] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth P, "From data mining to knowledge discovery in databases, advices in knowledge discovery and data mining," MIT Press, vol. 1, pp 1–36, 1998.

[3] R.Agrawal, T.Imieliński, A.Swami, "Mining association rules between sets of items in large databases", IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, 1993.

[4] Zaki, MJ. Scalable Algorithms for Association Mining. IEEE Trans. Knowl. Data Eng., 2000, 12(3):372-390, 2000.

[5] Savasere, A., Omiecinski, E., & Navathe, S "An efficient algorithm for mining association rules in large databases". Proc. of the Int. Conf. VLDB (pp. 432-444), 1995.

[6] Toivonen, H "Sampling large databases for association rules". Proc. of the Int. Conf. on VLDB (pp. 134{145), 1994.

[7] Park, J. S., Chen, M.-S., & Yu, P "An efficient hash based algorithm for mining association's rules". Proc. of the ACM Int. Conf. of SIGMOD (pp. 175-186), 1995.

[8] Bayardo, R. J "Efficiently mining long patterns from databases. Proc. of the ACM

Int. Conf. of SIGMOD (pp. 85-93), 1998.

[9] Lin, D., & Kedem, Z. M "PINCER-SEARCH: A new algorithm for discovering the maximum frequent sets". Proc. of the Int. Conf. on Extending Database Technology (pp.105-119), 1998.

[10] Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W "New algorithms for fast

Discovery of association rules". Proc. of the Int. Conf. on KDD (pp. 283-286), 1997

[11] Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. "Discovering frequent closed itemsets for association rules". Proc. of the Int. Conf. on Database Theory (pp. 398-416), 1999b.

[12] Stumme, G., Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L "Computing iceberg concept lattices with titanic". Data and Knowledge Engineering, 42, 189-222, 2002.

[13] Zaki, M. J., & Hsiao, C.-J "CHARM: An efficient algorithm for closed itemset Mining". SIAM Int. Conf. on Data Mining (pp. 33-43), 2002.

[14] Han, J, Pei, J, Ying, Y, Mao and R. "Mining frequent patterns without candidate generation: a frequent-pattern tree approach". Data Min. Knowl. Discov, 2004, 8(1):53-87.

[15]http://blog.khaledtannir.net/2012/07/lalgorithme-fp-growth-les-bases13/#.W6fKJHszbDd

[16] W. Cheung, O.R. Zaiane "Incremental Mining of Frequent Patterns without Candidate Generation or Support Constraint", Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS 2003), Hong Kong, China, 2003.

[17] Pei, J, Han, J, Lu, H, Nishio, S, Tang, S, Yang, D, "H-mine: Hyper-structure mining of frequent patterns in large databases". In: Proc. 2001 IEEE Intern. Conf. Data Mining, San Jose, USA, 29 November - 2 December, 2001:441-448.

[18] J.Pei, J.Han, B.Mortazavi-Asl, H.Pinto "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern". Intelligent Database Systems Research Lab, School of Computing Science, Simon Fraser University Burnaby, B.C., Canada V5A 1S6.

[19] Schlegel B, Karnagel T, Kiefer T, Lehner W "Scalable frequent itemset mining on many-core processors". In: Proc. 9th Intern. Workshop Data Management on New Hardware, New York, USA, 24 June, 2013: paper 3.

[20] Fournier Viger, Philippe & Lin, Chun-Wei & Vo, Bay & Truong, Tin & zhang, Ji & Le, Bac. (2017). A Survey of Itemset Mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.

[21] T. Chaari : Adaptation d'applications pervasives dans des environnements multi-contextes. Thèse de doctorat, INSA de Lyon, septembre 2007.

[22] B. Schilit et M. Theimer: Disseminating Active Map Information to Mobile Hosts. IEEE Network, 8(5):22-32, 1994.

[23] P. J. Brown, J. D. Bovey and X. Chen: Context-aware Applications: from the Laboratory to the Marketplace. IEEE Personal Communications, 4(5):58-64, October 1997.

[24] N. S. Ryan, J. Pascoe et D. R. Morse: Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant.In V. Gaffney, M. van Leusen et S. Exxon, éditeurs : Computer Applications in Archaeology 1997, British Archaeological Reports, Oxford, October 1998. Tempus Reparatum.

[25] J. Pascoe: Adding generic contextual capabilities to wearable computers. In Wearable Computers, 1998. Digest ofPapers. Second International Symposium on, pages 92-99, octobre 1998.

[26] D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith et P. Steggles : Towards a Better Understanding of Context and Context-Awareness. In HUC '99 : Proceedings of the 1st international symposium on Handheld andUbiquitous Computing, pages 304-307, London, UK, 1999.

[27] T.Winograd : Architectures for context. Human-Computer Interaction, 16 :401-419, 2001.

[28] Aroua, Adel; Bouarar, Rahima; La modélisation hybride des connaissances contextuelles dans un environnement ubiquitaire, 2013

[29] N. Belhanafi-Behllouli, Thèse de Doctorat de l'Institut National des Télécommunications, Ajout de mécanismes de réactivité au contexte dans les intergiciels pour composants dans le cadre d'utilisateurs nomades, Evry-France, 27 Novembre 2006.

[30] T.Chaari. Adaptation d'applications pervasives dans des environnements multi-contextes. Thèse de Doctorat en informatique, Institut national des sciences appliquées de Lyon, 2007

[31] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. Journal of Pervasive and Mobile Computing, volume 2(1) : pages 37-64, Elsevier, 2006.

[32] Q. Z. Sheng and B. Benatallah. ContextUML : A UML-Based Modeling Language for Model- Driven Development of Context-Aware Web Services. In The 4th International Conference on Mobile Business (ICMB'05), IEEE Computer Society. Sydney, Australia, July 11-13 2005.

[33] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran. Composite Capability/Preference Profile (CC/PP): Structure and vocabularies 1.0. Technical report, W3C recommandation, 15 january 2004.

[34] F. Manola and E. Miller. RDF Primer. Technical report, W3C recommandation, 10 February 2004.

[35] J. Indulska, R. Robinson, A. Rakotonirainy, and K. Henricksen. Experiences in Using CC/PP in Context-Aware Systems. In 4th international Conference on Mobile Data Management, pages 247-261, London, UK, 2003. Springer-Verlag.

[36] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In Pervasive 2002, pages 167-180, Zurich, Switzerland, 2002.

[37] Induska,J ; Robinson,R ;Rakotonirainy,A ; and Henricksen,K,2003. Experiences In Using CC/PP In Context-Aware Systems. In Proceeding of the 4th International Conference on Mobile Data Management.January 21-24, 2003, Malbourne, Australia. (Lecture notes In computer science, 2003, Vol.2574, pp.247_261)

[38] Patricia Dockhorn Costa. Towards a services platform for context-aware applications. Master thesis. En schede, The Netherlands: University of Twente, august 2003.

[39] Harry Chen, Tim Finin, and Anupam Joshi. 2003. An ontology for context-aware pervasive computing environments. Knowl. Eng. Rev. 18, 3 (September 2003), 197-207.

[40] Costa, P.D. (2003). Towards a Services Platform for Context-Aware Applications.

[41] Sachin Singh, Pravin Vajirkar, and Yugyung Lee. Context-aware Data Mining using Ontologies School of Computing and Engineering,.University of Missouri–Kansas City,. Kansas City, MO 64110 USA. {sbs7vc, ppv22e, leeyu}@umkc.edu.

[42] Xiao Hang Wang, Da Qing Zhang, Tao Gu, Hung Keng Pung . Ontology Based Context Modeling and Reasoning using OWL

[43] Strang,T; et al.. Service Interoperability on Context Level in Ubiquitous Computing Environments. International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR 2003w), January 2003, L'Aquila/Italy.

[44] J. Rabatel, S. Bringay, and P. Poncelet. Contextual sequential pattern mining. In (ICDMW), ICDM, pages 981-988. IEEE, 2010.

[45] W. Dong, W. Fan, L. Shi, C. Zhou, and X. Yan, "A general framework to encode heterogeneous information sources for contextual pattern mining," Proc. 21st ACM Int. Conf. Inf. Knowl. Manag. - CIKM '12, p. 65, 2012

[46] Julien Rabatel , Madalina Croitoru , Dino Ienco , Pascal Poncelet, Contextual itemset mining in DBpedia, Proceedings of the 1st International Conference on Linked Data for Knowledge Discovery, p.22-31, September 19, 2014, Nancy, France

[47] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by patterngrowth: the PrefixSpan approach. IEEE Transactions on Knowledge and Data Engineering, 16(11):1424–1440, 2004.

[48] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In VLDB, pages 420-431, 1995.

[49] A. Cakmak and G. Ozsoyoglu. Taxonomy-superimposed graph mining. In EDBT, pages 217-228, 2008.

[50] A. Inokuchi. Mining generalized substructures from a set of labeled graphs. In ICDM, pages 415-418, 2004.

[51] R. Srikant and R. Agrawal. Mining generalized association rules. In VLDB, pages 407-419, 1995.

[52] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In EDBT, pages 1-17. Springer, 1996.

[53] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal, 2014.

[54] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In VLDB, volume 1215, pages 487-499, 1994.

[55] https://github.com/tofti/python-kmeans

[56] https://bl.ocks.org/rpgove/0060ff3b656618e9136b

[57] Peter J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987), no 0, 53 – 65.

[58] Robert Tibshirani, Guenther Walther et Trevor Hastie, Estimating the number of clusters in a data set via the gap statistic, Journal of the Royal Statistical Society : Series B (Statistical Methodology) 63 (2001), no 2, 411–423.

[59] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297 , 1967.

[60]https://medium.com/square-corner-blog/so-you-have-some-clusters-now-what-abfd297a575b.

[61] https://docs.python.org/3/

[62] http://jupyter.org/index.html.

[63] http://pandas.pydata.org/pandas-docs/version/0.23/.

[64] http://scikit-learn.org/stable/documentation.html.

[65] http://rasbt.github.io/mlxtend/.

[66] Meseguer, A., Aibar, E., Lladós, J., Minguillón, J., Lerga, M. (2015). "Factors that influence the teaching use of Wikipedia in Higher Education". JASIST, Journal of the Association for Information Science and Technology. ISSN: 2330-1635. doi: 10.1002/asi.23488. https://archive.ics.uci.edu/ml/datasets.html

Annexes

1. Description de data set

1.1 Wikipédia

- **Information sur les données (Wikipédia)**

Recherches en cours sur les perceptions des professeurs d'université et les pratiques d'utilisation de Wikipédia comme ressource pédagogique. Sur la base d'un modèle d'acceptation technologique, les relations entre les constructions internes et externes du modèle sont analysées. La perception de l'opinion de collègues sur Wikipédia et la qualité perçue des informations sur Wikipédia jouent un rôle central dans le modèle obtenu.

- **Information d'attribut**

Attribut	Information
AGE	Numeric
GENDER	0=Male; 1=Female
DOMAIN	1=Arts & Humanities; 2=Sciences; 3=Health Sciences; 4=Engineering & Architecture; 5=Law & Politics
PhD	0=No; 1=Yes
YEARSEXP (years of university teaching experience)	Numeric
UNIVERSITY	1=UOC; 2=UPF
UOC_POSITION (academic position of UOC members)	1=Professor; 2=Associate; 3=Assistant; 4=Lecturer; 5=Instructor; 6=Adjunct
OTHER (main job in another university for part-time members)	1=Yes; 2=No
OTHER_POSITION (work as part-time in another university and UPF members)	1=Professor; 2=Associate; 3=Assistant; 4=Lecturer; 5=Instructor; 6=Adjunct
USERWIKI (Wikipedia registered user)	0 = No; 1 = Yes

- Les éléments de sondage suivants sont l'échelle de Likert (A-E) allant de fortement en désaccord / jamais (A) à fortement en accord / toujours (E)

Utilité perçue

PU1	L'utilisation de Wikipédia facilite le développement de nouvelles compétences pour les étudiants
PU2	L'utilisation de Wikipédia améliore l'apprentissage des étudiants
PU3	Wikipédia est utile pour l'enseignement

Facilité d'utilisation perçue

PEU1	Wikipédia est facile à utiliser
PEU2	Il est facile de trouver dans Wikipédia les informations que vous recherchez
PEU3	Il est facile d'ajouter ou de modifier des informations dans Wikipédia

Plaisir perçu

ENJ1	L'utilisation de Wikipédia stimule la curiosité
ENJ2	L'utilisation de Wikipédia est divertissante

Qualité

QU1	Les articles dans Wikipédia sont fiables
QU2	les articles de Wikipédia sont mis à jour
QU3	Les articles dans Wikipédia sont complets
QU4	Dans mon domaine de compétence, la qualité de Wikipédia est inférieure à celle des autres ressources pédagogiques.
QU5	Je fais confiance au système de rédaction de Wikipédia

Visibilité

VIS1	Wikipédia améliore la visibilité du travail des étudiants
VIS2	Il est facile d'avoir un enregistrement des contributions faites dans Wikipédia
VIS3	Je cite Wikipédia dans mes articles académiques

Image sociale

IM1	L'utilisation de Wikipédia est bien considérée par ses collègues
IM2	En milieu universitaire, le partage de ressources éducatives libres est apprécié
IM3	Mes collègues utilisent Wikipédia

Attitude de partage

SA1	Il est important de partager du contenu académique sur des plateformes ouvertes
SA2	Il est important de publier les résultats de la recherche sur d'autres supports que des revues ou des livres universitaires
SA3	Il est important que les étudiants se familiarisent avec les environnements de collaboration en ligne

Comportement d'utilisation

USE1	J'utilise Wikipédia pour développer mon matériel pédagogique
USE2	J'utilise Wikipédia comme plate-forme pour développer des activités pédagogiques avec les étudiants
USE3	Je recommande à mes étudiants d'utiliser Wikipédia
USE4	Je recommande à mes collègues d'utiliser Wikipédia
USE5	Je reconnais que mes étudiants utilisent Wikipédia dans mes cours

Profil 2.0

PF1	Je contribue aux blogs
PF2	Je participe activement aux réseaux sociaux
PF3	Je publie du contenu académique sur des plateformes ouvertes

Pertinence de l'emploi

JR1	Mon université encourage l'utilisation d'environnements collaboratifs ouverts sur Internet
JR2	Mon université considère l'utilisation d'environnements collaboratifs ouverts sur Internet comme un mérite pédagogique

Intention comportementale

BI1	À l'avenir, je recommanderai l'utilisation de Wikipédia à mes collègues et aux étudiants.
BI2	À l'avenir, j'utiliserai Wikipédia dans mon activité d'enseignement

Des incitations

INC1	Pour concevoir des activités éducatives utilisant Wikipédia, il serait utile: un guide des meilleures pratiques
INC2	Pour concevoir des activités éducatives à l'aide de Wikipédia, il serait utile d'obtenir les instructions d'un collègue.
INC3	Pour concevoir des activités éducatives avec Wikipédia, il serait utile de suivre une formation spécifique.
INC4	Pour concevoir des activités éducatives sur Wikipédia, il serait utile de: mieux reconnaître les institutions

Expérience

EXP1	Je consulte Wikipédia pour les problèmes liés à mon domaine de compétence
EXP2	Je consulte Wikipédia pour d'autres questions académiques
EXP3	Je consulte Wikipédia pour des problèmes personnels
EXP4	Je contribue à Wikipédia (éditions, révisions, améliorations d'articles ...)
EXP5	J'utilise des wikis pour travailler avec mes étudiants

2. Installation python et ses bibliothèques

- **Python**

Installer Python depuis le site : https://www.python.org/ .

- **Pandas**

Bibliothèque de manipulation des données pour installer pandas il faut taper la commande suivant dans le terminale : pip install pandas.

- **Numpy**

Bibliothèque pour le calcul scientifique, pour installer Numpy il faut taper : pip install numpy.

- **Jupyter Notebook**

Pour installer Jupyter à l'aide du gestionnaire de paquets de Python, il faut taper les deux commandes suivantes :

- python3 -m pip install --upgrade pip
- python3 -m pip install jupyter

- **Apyori**

Implémentation de l'algorithme Apriori avec Python, pour l'installer il faut taper : pip install apyori.

- **Scikit-learn**

Bibliothèque d'apprentissage automatique, pour l'installer il faut taper : pip install scikit-learn où bien pip install -U scikit-learn.

1. Manipulation de données avec Pandas

- **Lire fichier csv**


```
import pandas as pd
data = pd.read_csv('dataset.csv')
data.head()
```

- **Sauvegarder en fichier csv**

```
df.to_csv('clustering_kmeans2.csv', sep=',',index=False)
```

- **Génération des données**

```
df = pd.DataFrame(np.random.randint(2, size=(1000, 110)),
                  columns=['C {}'.format(i) for i in range (110)],
                  index=['Client {}'.format(i) for i in range(1000)])
df.head()
```

2. Méthodes de calculer le nombre optimal de clusters

- **Méthode Elbow**

```
def ElbowSSE (data, maxClusters = 10):
    res = list()
    for n in range (2, maxClusters):
        kmeans = KMeans(n_clusters=n)
        kmeans.fit(data)
        res.append(np.average(np.min(cdist(data, kmeans.cluster_centers_, 'euclidean'), axis=1)))

    plt.plot(range (2, maxClusters), res)
    plt.title('elbow curve')
    plt.show()
```

- **Méthode Silhouette**

```
def silhouette_Methode (data, maxClusters = 15) :

    for n_clusters in range (2,  maxClusters):

        clusterer = KMeans(n_clusters=n_clusters, random_state=10)
        cluster_labels = clusterer.fit_predict(data)
        silhouette_avg = silhouette_score(data, cluster_labels)
        print("For n_clusters =", n_clusters,
          "The average silhouette_score is :", silhouette_avg)
        sample_silhouette_values = silhouette_samples(data, cluster_labels)
        y_lower = 10

    for i in range(n_clusters):

        ith_cluster_silhouette_values = \
            sample_silhouette_values[cluster_labels == i]

        ith_cluster_silhouette_values.sort()

        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i
```