Faculté d’électronique et d’Informatique

Département d’Informatique

Rapport de Projet

Master informatique visuelle

Module : Fouille des données (Data Mining)

THEME

Extraction des Règles d’Associations Distribuées à l’aide d’algorithme Close

Proposer par:

Mr NECIR Hamid

Réalisé par :

Mr Madadi Youcef

171731094320 / G2

Table des matières

[Problème proposé 1](#_Toc76765615)

[Objectifs 1](#_Toc76765616)

[Introduction 2](#_Toc76765617)

[Chapitre I : Définitions et les outils d’un système distribué de Data Mining 3](#_Toc76765618)

[1. Introduction 3](#_Toc76765619)

[2. Data Mining 3](#_Toc76765620)

[3. Les techniques de Data Mining 4](#_Toc76765621)

[3.1. Classification 4](#_Toc76765622)

[3.2. Clustering 5](#_Toc76765623)

[3.3. Régression 5](#_Toc76765624)

[3.4. Association 6](#_Toc76765625)

[4. Les règles d’association 6](#_Toc76765626)

[4.1. Les règles d’association fermées 6](#_Toc76765627)

[4.2. L’algorithme Close 6](#_Toc76765628)

[5. Data Mining Distribué 6](#_Toc76765629)

[5.1. Règles d’association distribuées 7](#_Toc76765630)

[5.2. La Distribution de données 7](#_Toc76765631)

[5.3. Formulation du problème d’extraction des règles d’associations distribuées 7](#_Toc76765632)

[Chapitre II : Solution de problème 8](#_Toc76765633)

[1. Introduction 8](#_Toc76765634)

[2. Extraction des règles d’associations distribuées 8](#_Toc76765635)

[2.1. Example : 8](#_Toc76765636)

[2.2. Optimization Algorithm: 8](#_Toc76765637)

[3. Conclusion 8](#_Toc76765638)

[Conclusion générale 9](#_Toc76765639)

# Problème proposé

Etant donnée le nombre croissant de grandes bases de données et de la complexité des algorithmes mis en œuvre pour leurs exploitations, la question d’optimisation préoccupe de plus en plus les chercheurs du domaine de la fouille de données.

L’une des techniques les plus utilisées pour extraire ces connaissances est la méthode des règles d’association.

Toutefois, la plupart des algorithmes d’extraction des itemsets fermés fréquents voient leurs performances se dégrader lorsque la taille des données augmente. Pour maintenir les performances de ces algorithmes, l’utilisation de méthodes et outils distribués apparait comme une solution naturelle.

Le projet demandé, vise à améliorer les performances de l’un des meilleurs algorithmes d’extraction des itemsets fermés fréquents à savoir l'algorithme Close.

Il s'agit de mettre en place une version de l’algorithme Close, dans le cas d'un contexte où les données seront partitionnées. Ceci permettra d'extraire des résultats locaux relatifs à chaque partition. Par la suite une seconde étape permettra de retrouver les résultats finaux à partir de ceux déjà extrait.

Dans notre cas, on considère ainsi n sources de données S1, S2,..., Sn. Soit D la taille de l'ensemble de nos données et di la taille de chaque source. Pour un itemset X donné, soit Sup(X) et Supi(X) le support de X dans Si respectivement.

Dans ce cas, Sup(X) est appelé le support « global » et Supi(X) le support « local » de l’itemset X dans Si. Pour un support minimum donné minsup, X est globalement fréquent si: Sup(X) >=minsup \* D.

De même, X est localement fréquent dans Si: si Supi(X) >= minsup \* di.

Ainsi, le problème d’extraction des règles d’association se réduit à trouver tous les itemsets globalement fréquents et de générer ensuite les règles d’associations correspondantes.

# Objectifs

Il est demandé d'implémenter l'algorithme Close avec une démarche plus performante et de rédiger un rapport expliquant votre démarche et les différents choix fait.

# Introduction

Les progrès de l'informatique et de la communication sur les réseaux câblés et sans fil ont donné naissance à de nombreux environnements informatiques distribués omniprésents. L'Internet, les intranets, les réseaux locaux et les réseaux sans fil en sont quelques exemples. Beaucoup de ces environnements comportent différentes sources distribuées de données volumineuses et de multiples nœuds de calcul. L'analyse et la surveillance de ces sources de données distribuées nécessitent une technologie d'exploration de données (Data Mining Technology) conçue pour les applications distribuées.

Dans le premier chapitre, nous nous intéressons à l'introduction de quelques définitions et outils nécessaires pour un système distribué de Data Mining. Puis, nous nous étalons dans le 2eme chapitre pour introduire la solution pour le problème proposé dans ce projet. Enfin, nous allons passer dans le troisième chapitre à la réalisation de notre solution à partir des études accomplies lors des deux premiers chapitres.

# Chapitre I : Définitions et les outils d’un système distribué de Data Mining

## Introduction

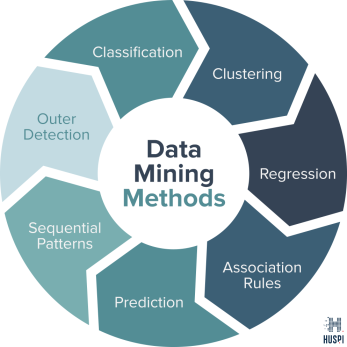
Dans ce chapitre, nous allons parler des définitions de Data Mining. Ensuite, nous allons présenter les outils des systèmes distribués de Data Mining.

## Data Mining

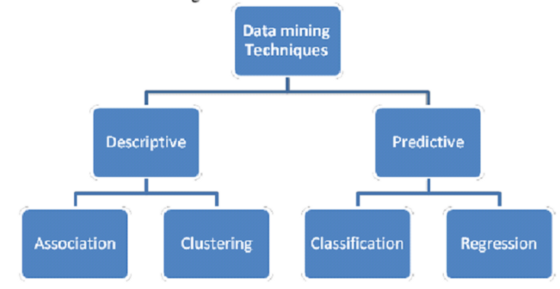
La fouille de données (Data Mining) vise à découvrir, dans les grandes quantités de données, les informations précieuses qui peuvent aider à comprendre les données ou à prédire le comportement des données futures.

Le datamining utilise, depuis sont apparition, plusieurs outils de statistiques et d’intelligence artificielle pour atteindre ses objectifs qui sont:

* Rechercher des régularités dans les données.
* Trouver des schémas fréquents, des associations, des corrélations parmi des ensembles d'articles dans des bases de données de transactions.
* Comprendre les habitudes d'achat des clients en trouvant des associations et corrélations entre les différents achats que les clients placent dans leur "panier" .

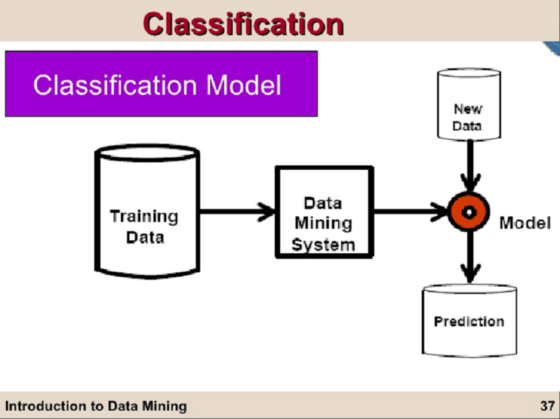


## Les techniques de Data Mining



### Classification

La technique de classification de datamining est utilisée pour classer les données dans différentes classes. Elle est basée sur l'apprentissage automatique. La classification est utilisée pour classer chaque élément dans un ensemble de données prédéfini de classes ou de groupes. Cette technique utilise des concepts mathématiques tels que les arbres de décision, la programmation linéaire, les réseaux neuronaux et les statistiques. Dans la classification, il y a un logiciel qui peut apprendre à classer les éléments de données dans des groupes. Les algorithmes utilisés pour la classification sont la régression logistique, Naive Bayes, K-Nearest Neighbor.



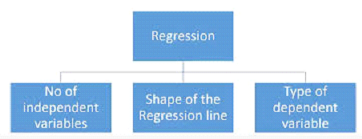
### Clustering

La technique de clustering de l'exploration de données est utilisée pour identifier les données qui sont semblables les unes aux autres. Cette technique aide à comprendre les similitudes et les différences entre les données. Cette technique crée des classes et des objets significatifs et place les objets dans chaque classe qui a des caractéristiques similaires. L'algorithme utilisé pour le clustering est l'algorithme de clustering hiérarchique, etc.



### Régression

La régression est une technique d'exploration de données qui analyse la relation entre les variables et crée des modèles prédictifs. La technique de régression permet d'analyser et de prédire les résultats sur la base de données connues précédemment en appliquant des formules. La régression est très utile pour trouver des informations sur la base d'informations connues existantes.



### Association

L'association est une technique qui permet de trouver le lien entre deux ou plusieurs éléments. Cette technique peut créer un modèle caché dans un ensemble de données. Ce modèle est découvert sur la base de la relation entre les éléments qui sont dans la même transaction. Cette technique est utilisée dans l'analyse du panier de la ménagère pour identifier les produits que les clients achètent fréquemment ensemble.



## Les règles d’association

Sa principale application est « l’analyse du panier de la ménagère ». Elle a été introduite en 1993. Cette technique s’applique aux problèmes d’exploitation des données contenant des articles bien définis, qui se regroupent entre eux de manière intéressante.

Elle consiste à analyser des bases de données de transactions composées d’une liste d’articles achetés, afin d’identifier les groupes d’articles les plus fréquemment vendus ensemble.

### Les règles d’association fermées

Les règles d'association fermées sont une technique qui permet de trouver le lien entre deux ou plusieurs éléments avec un niveau de confiance élevé (une règle forte). Cette technique permet de créer un modèle solide dans un ensemble de données.

### L’algorithme Close

L’algorithme Close est un algorithme itératif proposé en 1998 par Pasquier et al. Il fait de l’extraction des itemsets fermés fréquents par niveau de façon que, durant chaque itération k de l'algorithme, un ensemble de k-générateurs candidats est considéré. Chaque élément de cet ensemble est constitué de trois éléments : le k-générateur candidat, sa fermeture, et leur support. À la fin de l'itération k, l'algorithme stocke un ensemble contenant les k-générateurs fréquents, leurs fermetures, qui sont des itemsets fermés fréquents, et leurs supports.

## Data Mining Distribué

L’application du Data Mining sur différente architectures homogènes ou hétérogènes est appelée Data Mining distribué. Pour ce faire, on utilise différents types de programmation parallèle ou distribuée. Parmi les différentes techniques utilisées dans le Data Mining distribué, une attention particulière est portée aux règles d’association distribuées.

### Règles d’association distribuées

La technique de l’extraction des règles d’association est celle qui a attiré le plus d’attention des chercheurs en fouille de données et celle pour laquelle beaucoup de travaux ont été effectués. Elle permet l’expression des associations entre les items d’une base de données en faisant la découverte des règles intelligibles et exploitables dans un ensemble de données volumineux.

### La Distribution de données

Le schéma d’une base de données relationnelle fournit toutes les informations concernant les relations stockées. Ces dernières sont élémentaires pour identifier leurs dépendances mutuelles qui elles-mêmes déterminent l’algorithme du data mining à appliquer. Il existe deux types de distribution de données : homogène et hétérogène.

La distribution homogène se fait pas un découpage horizontal de la base de données. Les instances contiennent le même ensemble d’attributs et sont distribuées entre les sites. Ce modèle se produit généralement dans la même organisation ou à travers les domaines similaires.

La distribution hétérogène est un découpage vertical de la base de données. Les attributs peuvent contenir des tables avec différents schémas et sont distribués entre les sites. Dans ce modèle, on peut relier les tables par différentes clés de jointure.

### Formulation du problème d’extraction des règles d’associations distribuées

Soit I l’ensemble d'items et DB la base de données des transactions, où chaque transaction a un identificateur unique (tid) et contient un ensemble d'items appelé itemset. Un itemset avec K items est appelé un k-itemset. Le support d'un itemset X, noté sup(X), est le nombre de transactions dans lesquelles cet itemset apparaît comme un sous-ensemble. Un itemset est fréquent (ou large) si son support est supérieur à la valeur d’un support minimum (minsup) indiqué par l'utilisateur. Une règle d'association est une expression A → B, où A et B sont des itemsets. Le support de la règle est la probabilité qu’une transaction contient A et B en même temps, noté par Sup (A∩B). La confiance de la règle est la probabilité conditionnelle qu’une transaction contient B étant donné qu'elle contient A, noté par Sup(A∩B) /Sup (A). Une règle est fréquente si son support est supérieur à minsup et solide si sa confiance est supérieure à la confiance minimale indiquée par l’utilisateur (minconf). Dans le cas de bases de données distribuées, la base de données DB est partitionnée en {DB1, DB2, ..., DBn} et distribuée à travers les n sites S1, S2,... Sn. Soit D la taille de la base de données DB et di la taille de la partition DBi. Pour un itemset X donné, soit Sup(X) et Supi(X) le support de X dans DB et DBi respectivement. Dans ce cas, Sup(X) est appelé le support « global » et Supi(X) le support « local » de l’itemset X dans le site Si. Pour un support minimum donné minsup, X est globalement fréquent si Sup(X)  minsup \* D. de même, X est localement fréquent dans le site Si si Supi(X)  minsup \* di. Ainsi, le problème d’extraction des règles d’association dans la base de données distribuée se réduit à trouver tous les itemsets globalement fréquents et de générer ensuite les règles d’associations correspondantes.

# Chapitre II : Solution de problème

## Introduction

Pour trouver les itemsets fermées globalement fréquent, le système distribué commence d’abord par trouver les règles d’associations de la base de données locale et donc trouvera les itemsets fermées localement fréquents. L’intersection de toutes les règles d’association trouvées par le système distribué donnera les règles d’association de la base de données globale et donc les itemsets fermées globalement fréquent. Plus simplement, si nous avons A-->B (B est la fermeture locale de A), il faut que l’item B soit la fermeture locale de A dans toutes les bases de données locales pour qu'on puisse dire que B est une fermeture globale de A.

## Extraction des règles d’associations distribuées

Pour extraire les règles d'associations globales, il suffisait d'obtenir l'intersection des règles d'associations locales extraites de chaque base de données.

### Example :

Nous avons 3 bases de données {Db1, Db2, Db3} contient :

* LCAR1 = {A->B, C->A, D->E}
* LCAR2 = {A->E, C->A, D->E}
* LCAR3 = {B->AC, C->AD, E->A}

LCAR1 **Ω** LCAR2 = {C->A, D->E}

**GCAR = LCAR1 Ω LCAR2 Ω LCAR3 = {C->A}**

**LCARi** : Local closed Association rules of Dbi (Règlement de l'Association fermée locale de Dbi)

**GCAR :** Global closed Association rules (Règlement de l'Association fermée Global)

### Optimization Algorithm:

|  |
| --- |
| **For each Dbi :**  **If GCAR.isEmpty() :**  **GCAR = Receive LCARi**  **Else**  **GCAR = { GCAR Ω Receive LCARi}**  **If GCAR.isEmpty():**  **# Algorithm ended and there is no Global closed Association rules**  **End for**  **# GCAR contains the Global closed Association rules** |

## Conclusion

L'exploration de données à l'aide de bases de données distribuées permettra un traitement des données plus optimal et minimisera les ressources consommées.

# Chapitre III : Implémentation

## Création des Base de donne

### Création d'une classe « DATABASE » capable de gérer la gestion des bases de données et les algorithmes que nous leur appliquons.

* • DataFrame : un attribut qui gère une matrice (base de données) de manière encadrée de la bibliothèque de Panda
* • Dimension : un attribut qui contient la dimension de la base de données
* • Columns : un attribut qui contient les colonnes de la base de données
* • CalculeSup : une méthode qui calcule les supports de chaque élément de la base de données
* • Intersection : une méthode qui crée une combinaison entre deux éléments (A et B deviennent AB)
* • isClose : une méthode qui vérifie si l'élément B est proche de A
* • CloseItems : une méthode qui vérifie les éléments fermés et extrait les règles d'association (algorithme de fermeture)

### Création d'une classe « DATABASEMANAGER » qui gère la base de données globale

## Création des Règles d’association

### Création d’une classe « AssociationRule » qui gère la création d’une règle d’association

### Création d’une classe « AssociationRulesManager» qui gère les liste des règle d’association

## Lancer les algorithms de close dans des system distribue

Dans notre cas, nous utilisons des threads pour déjeuner le calcul dans plusieurs bases de données.

## Récupérez le GCAR

Chaque fois que nous rejoignons un thread, nous appliquons une itération de l'algorithme d'optimisation (Receive LCARi) et à partir de celui-ci, nous pouvons construire le résultat final qui est le GCAR si nous avons trouvé une règle d'association fermée commune.

## Implémentation :

La mise en œuvre se fait dans le système Google Colab :  
<https://colab.research.google.com/drive/17N8UUfSPxWQYc3hbL9JiCPH_cJfgTjsC?usp=sharing>