

# UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE



DATA MINING

---

## Rapport Algorithme Apriori

---

*Rédaction:*

MOULAI HASSINA SAFAA

Matricule : 201400007564

HOUACINE NAILA AZIZA

Matricule : 201400007594

M2 SII Groupe:3

*Professeur*

Mme. BABA ALI

December 24, 2018

# Contents

<b>1</b>	<b>Apriori</b>	<b>2</b>
1.1	Concepts de base pour apriori . . . . .	2
1.2	Principe de fonctionnement . . . . .	3
1.3	Pseudo-code . . . . .	4
1.4	Déroulement sur un exemple . . . . .	4

# Chapter 1

## Apriori

A priori est un algorithme fondamental proposé par R. Agrawal et R. Srikant en 1994 pour d'éléments fréquents pour les règles d'association booléennes [AS94b]. Le nom de l'algorithme est basé sur le fait que l'algorithme utilise connaissance préalable des éléments fréquents comme nous le verrons plus tard. c'est un algorithme facile à comprendre et très utilisé .

### 1.1 Concepts de base pour apriori

Tout d'abord avant de plonger directement dans le principe de fonctionnement globale de l'algorithme on a à définir quel que notion :

#### **un itemset**

un item set (ensemble d'items) est un ensemble comportant des items ou des éléments qui se produisent ensemble . par exemple : un itemset de transactions  $T=(T1=\text{lait,café},T2=\text{yaourt,crème glacée},T3=\text{couche bébé....})$

#### **le support**

supp (X) d'un jeu d'éléments X est le rapport entre les transactions dans lesquelles un jeu d'éléments apparaît et le nombre total de transactions.

#### **frequent itemset**

Un ensemble d'éléments fréquent est un ensemble d'éléments dont le support est supérieure à une prise en charge minimale spécifiée par l'utilisateur (notée  $L_k$ , où k est la taille de l'ensemble d'éléments)

#### **candidate itemset**

Un groupe d'éléments candidat est un groupe d'éléments potentiellement fréquent (noté  $C_k$ , où k est la taille du groupe d'éléments)

## Apriori propriété

chaque subset (sous-ensemble d'item) d'un fréquent itemset doit être fréquent (condition du support minimale vérifié).

## Opération JOIN (jointure)

Pour trouver le  $L_k$  (frequent itemset de items), on utilise un ensemble de candidate itemset qui sont générés grâce à la jointure de  $L_{k-1}$  avec  $L_{k-1}$  (produit cartésien).

## 1.2 Principe de fonctionnement

Apriori emploie un approche itérative tel que chaque k-itemsets est utilisé pour explorer les (k+1)-itemsets .

les différentes étapes :

1. Exploration de la base de données pour avoir le support de chaque 1-itemset (ensemble d'un seul item).
2. Comparer le support(fréquence) avec le **min\_supp** .
3. Supprimer les 1-itemsets ayant un support inférieur au **min\_supp** générer alors  $L_1$ .
4. Faire une jointure de  $L_{k-1}$  avec  $L_{k-1}$  pour générer les ensembles de candidate k-itemsets.
5. Verifier la propriété APRIORI pour élaguer les k-itemsets qui ne sont pas fréquents.
6. Exploration de la base de données pour avoir le support de chaque candidate k-itemset vérifiant la propriété apriori.
7. Comparer le support de chaque candidate k-itemset avec **min\_supp**
8. Garder que l'ensemble des k-itemsets vérifiant la condition de **min\_supp** et on aura ainsi  $L_k$
9. si  $L_k$  est vide alors pour chaque frequent itemset 1 générer les subsets non vide de 1, et pour chaque s subsets non vide de 1, écrire la règle "s implique(1-s)" si la confiance C de la règle "s implique 1-s" satisfait le **support min de confiance**
10. sinon aller à 4

### 1.3 Pseudo-code

---

**Algorithm 1** APRIORI
 

---

**Input:** D, base de données de transactions, support minimum

**Output:** L, itemset fréquent dans D

```

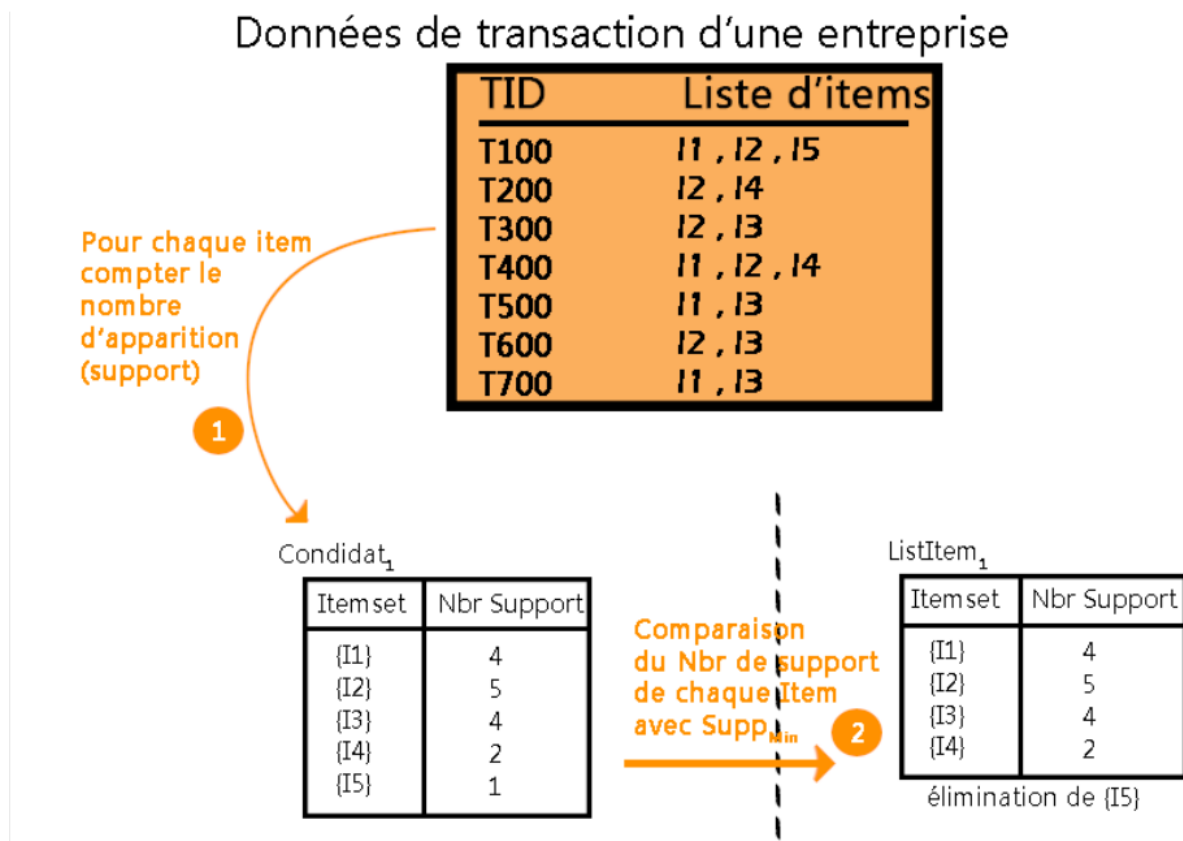
1 L1 = find frequent 1-itemsets(D);
2 for  $k \leftarrow 2; L_{k-1} \neq \emptyset; k++$  do
3    $C_k \leftarrow \text{apriori\_gen}(L_{k-1})$ 
4   for transaction  $t \in D$  do
5     scan D for counts
6      $C_t = \text{subset}(C_k, t)$ ; générer les subsets de t qui sont candidats
7     for candidate  $c \in C_t$  do
8       c.count++;
9        $L_k = \{c \in C_k, c.\text{count} > \text{support minimum}\}$ 
9 return  $L = \bigcup_k L_k$ 
  
```

---

### 1.4 Déroulement sur un exemple

Nous prenons comme exemple applicatif les données de transaction d'une entreprise.

Nous y appliquons l'algorithme apriori pour retrouver les motifs fréquents, comme suit:



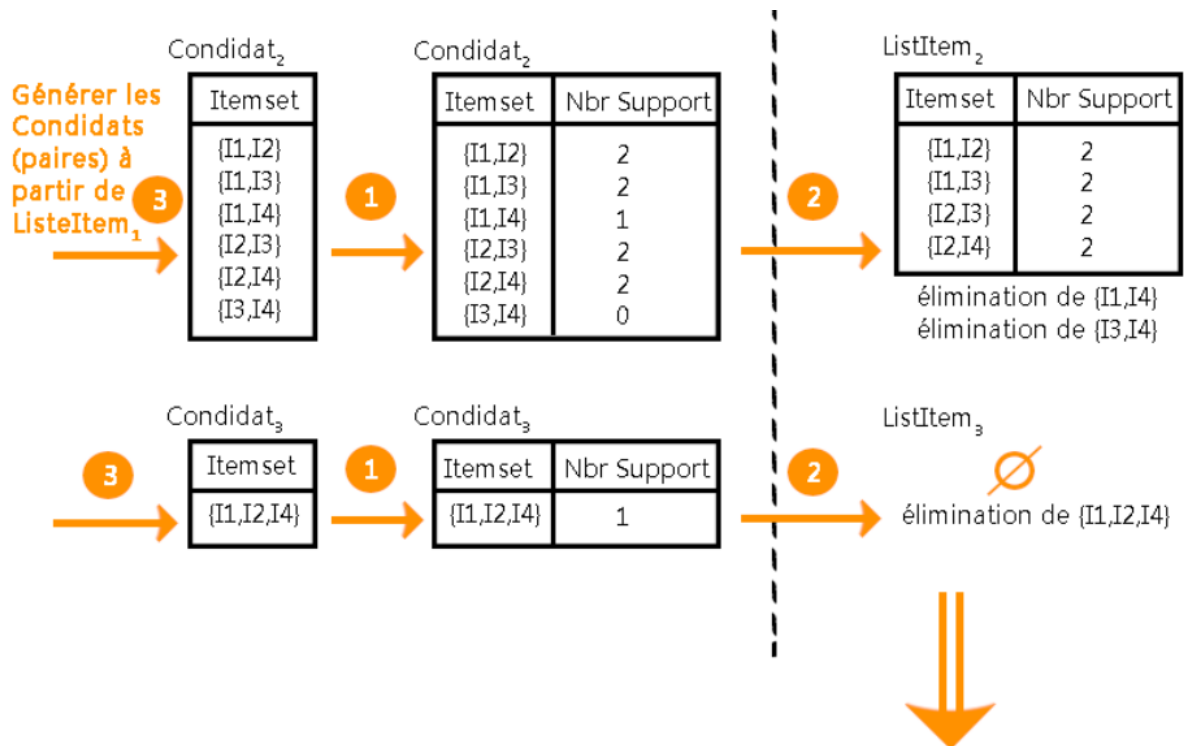


Figure 1.1: Déroulement de l'algorithme Apriori sur un exemple.