

# Extraction de règles d'association

Thierry Lecroq

Université de Rouen  
FRANCE

# Plan

- 1 Présentation
- 2 Extraction d'ensemble de mots fréquents
- 3 Génération des règles d'association

# Règles d'association

On cherche des relations entre les mots d'un corpus sous forme de règles :

**si** les mots  $x_1, \dots, x_m$  apparaissent dans un texte **alors** les mots  $x_{m+1}, \dots, x_n$  apparaissent aussi dans le texte

que l'on formalise en :

$$x_1 \cdots x_m \rightarrow x_{m+1} \cdots x_n$$

et que l'on appelle **règle d'association**.

Deux paramètres doivent être fournis :

- *minsup*, le pourcentage de textes où apparait la règle
- *minconf*, le pourcentage de fois où la règle est vérifiée

## Exemple

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

Avec  $minsup = 40\%$  et  $minconf = 50\%$ , les deux règles valides sont :  
 $D \rightarrow B$  et  $B \rightarrow D$ .

Avec  $minconf = 100\%$ , seule la règle  $D \rightarrow B$  est valide

# Mots fréquents

- Soit  $T$  un ensemble de textes
- On considère un texte  $t \in T$  comme étant un ensemble de mots
- $E$  est un ensemble de mots,  $support(E)$  = pourcentage de textes dans lesquels apparaissent tous les mots de  $E$

$$support(E) = \frac{card(\{t \in T \mid E \subset t\})}{card(T)}$$

- Un ensemble de mots  $E$  est fréquent si  $support(E) > minsup$

# Règles d'association

Une règle d'association  $r$  est une implication de la forme  $P_1 \rightarrow P_2$  entre deux ensembles de mots  $P_1$  et  $P_2$ ,  $P_1 \cap P_2 = \emptyset$ , telle que :

$$\text{support}(r) = \text{support}(P_1 \cup P_2)$$

$$\text{confiance}(r) = \frac{\text{support}(P_1 \cup P_2)}{\text{support}(P_1)}$$

- Une règle  $r$  est **valide** si  $\text{confiance}(r) > \text{minconf}$
- Une règle  $r$  est **totale** si  $\text{confiance}(r) = 1$  et **partielle** sinon

## Problème

Déterminer l'ensemble des règles dont le support est supérieur à  $minsup$  et la confiance à  $minconf$ , ce que l'on décompose en deux sous-problèmes :

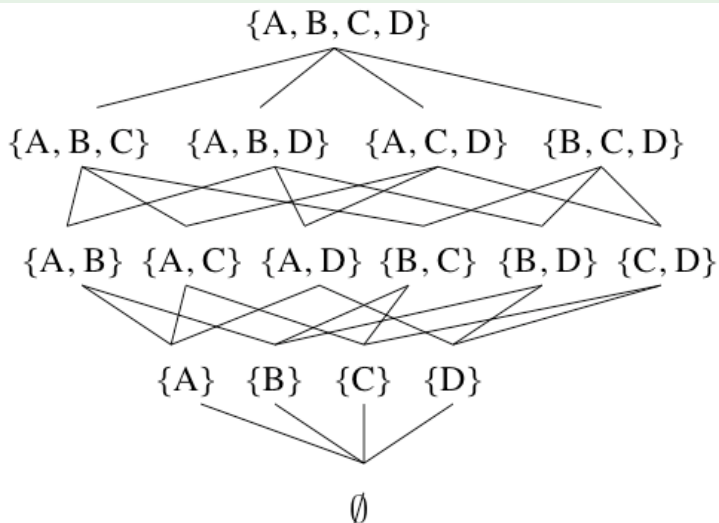
- 1 déterminer les ensembles  $E$  de mots fréquents ( $support(E) > minsup$ )
- 2 pour chacun de ces ensembles, générer toutes les règles  $r$  d'association valides ( $confiance(r) > minconf$ )

# Ordre, borne sup, borne inf

- Un ensemble  $E$  est ordonné s'il est muni d'une relation binaire  $\leq$  réflexive, antisymétrique et transitive ;
- Un majorant (*resp.* minorant) d'une partie  $A$  de  $E$  est un élément  $x$  tel que  $\forall y \in A, y \leq x$  (*resp.*  $\forall y \in A, y \geq x$ ) ;
- La **borne supérieure** (*resp.* borne inférieure) d'une partie  $A$  est le plus petit des majorants (*resp.* minorants). Elle n'existe pas forcément.

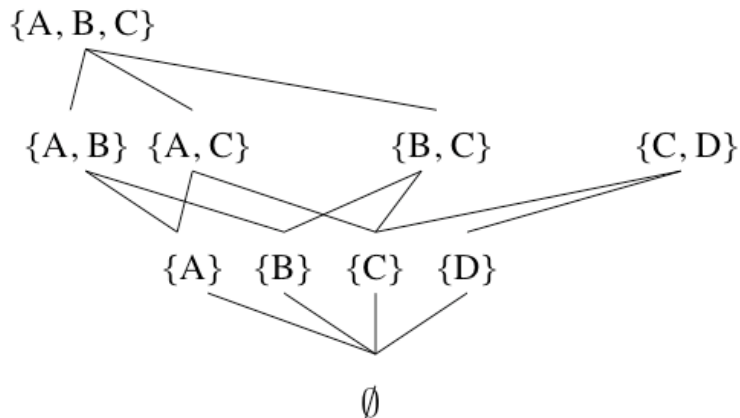
- Un **treillis** est un ensemble dans lequel chaque couple d'éléments possède une borne supérieure et une borne inférieure
- Si une seule des deux propriétés est vérifiée, on parle alors de **demi-treillis**
- L'ensemble des mots fréquents forme un demi-treillis :
  - ▶ l'intersection de deux ensembles de mots fréquents est fréquente
  - ▶ l'union de deux ensembles de mots non fréquents est non fréquente

## Exemple



# Demi-treillis

## Exemple



# Plan

- 1 Présentation
- 2 Extraction d'ensemble de mots fréquents
- 3 Génération des règles d'association

# Algorithme Apriori

## Schéma de l'algorithme

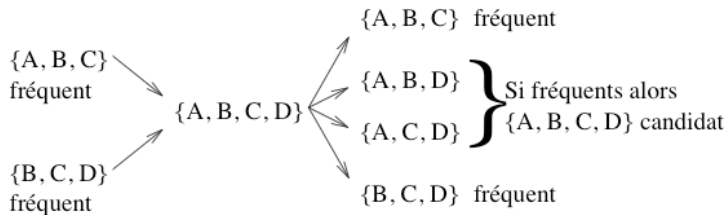
Parcours du treillis des ensembles de mots fréquents

- 1 les ensembles de cardinal  $k$  sont générés à partir de ceux de cardinal  $k - 1$
- 2 une fois les ensembles de cardinal  $k$  générés, on calcul leurs supports et on ne conserve que les fréquents

# Algorithme Apriori - Génération

- On fait l'union de tous les ensembles de mots n'ayant qu'un seul élément différent
- Seuls les ensembles dont tous les sous-ensembles sont fréquents sont conservés

## Exemple



# Algorithme Apriori - Génération

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

ensemble	support
$\{A\}$	1
$\{B\}$	4
$\{C\}$	2
$\{D\}$	2

$$F_1 = \{\{B\}, \{C\}, \{D\}\}$$

$$C_2 = \{\{B, C\}, \{B, D\}, \{C, D\}\}$$

# Algorithme Apriori - Génération

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

ensemble	support
$\{B, C\}$	1
$\{B, D\}$	2
$\{C, D\}$	0

$$F_2 = \{\{B, D\}\}$$

$$C_3 = \emptyset$$

$$F = \{\{B\}, \{C\}, \{D\}, \{B, D\}\}$$

# Algorithme Apriori - Génération

## APRIORI-GEN( $F$ )

- ▷ **Entrée** :  $F$  : ensembles de mots fréquents de cardinal  $k$
- 1 **Début**
- 2  $C \leftarrow \{c = f_1 \cup f_2 \text{ tels que } (f_1, f_2) \in F \times F, \text{ card}(c) = k + 1\}$
- 3 **pour chaque**  $c \in C$  **faire**
- 4     **pour chaque**  $s \subset c, \text{ card}(s) = k$  **faire**
- 5         **si**  $s \notin F$  **alors**
- 6              $C \leftarrow C \setminus \{c\}$
- 7 **Retourner**  $C$
- 8 **Fin**

# Algorithme Apriori

## APRIORI( $T, \text{minsup}$ )

- ▷ **Entrée** :  $T$  : corpus,  $\text{minsup}$  : entier
- ▷ **Sortie** :  $\cup_k F_k$
- 1 **Début**
- 2    $C_1 \leftarrow \{\text{singletons}\}$
- 3    $k \leftarrow 1$
- 4   **tantque**  $C_k \neq \emptyset$  **faire**
- 5     **pour chaque**  $c \in C_k$  **faire**
- 6      **pour chaque**  $t \in T$  **faire**
- 7       **si**  $c \subset t$  **alors**
- 8           $\text{support}(c) \leftarrow \text{support}(c) + 1$
- 9           $F_k \leftarrow \{c \in C_k \mid \text{support}(c) \geq \text{minsup}\}$
- 10        $k \leftarrow k + 1$
- 11        $C_k \leftarrow \text{APRIORI-GEN}(F_{k-1})$
- 12   **Retourner**  $\cup_k F_k$
- 13 **Fin**

# Algorithme Close

- repose sur l'extraction de **générateurs** d'ensemble de mots **fermés** fréquents
- le nombre d'ensembles de mots fermés fréquents est généralement bien inférieur au nombre d'ensembles de mots fréquents

# Fermeture

La fermeture d'un ensemble de mots  $A$  est un ensemble de mots  $B$  tel que  $B$  apparaît dans les mêmes textes que  $A$ .

Pour la calculer on utilise deux fonctions :

- $f$  : associe à un ensemble de mots les textes où il apparaît
- $g$  : associe à un ensemble de textes les mots qu'ils ont en commun

Soit  $A$  un ensemble de mots :

$$\text{fermeture}(A) = g \circ f(A)$$

## Exemple

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

- $f(\{D\}) = \{1, 5\}$
- $g(\{1, 5\}) = \{B, D\}$
- $fermeture(\{D\}) = \{B, D\}$
- $\{D\}$  est un générateur de  $\{B, D\}$

# Algorithme close

- 1 Initialisation de l'ensemble des générateurs avec l'ensemble des singletons formés par les mots du corpus
- 2 Calcul de la fermeture des générateurs de niveau  $k$  et de leur support
- 3 Ajout des fermetures des générateurs à l'ensemble des ensembles de mots fermés fréquents
- 4 Génération des générateurs de niveau  $k + 1$

# Calcul fermeture/support

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	$\emptyset$	0
{B}	$\emptyset$	0
{C}	$\emptyset$	0
{D}	$\emptyset$	0

# Calcul fermeture/support

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{A, B, D}	1
{C}	$\emptyset$	0
{D}	{A, B, D}	1

# Calcul fermeture/support

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	2
{C}	$\emptyset$	0
{D}	{A, B, D}	1

# Calcul fermeture/support

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	2
{C}	{C}	1
{D}	{A, B, D}	1

# Calcul fermeture/support

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	3
{C}	{C}	2
{D}	{A, B, D}	1

# Calcul fermeture/support

Exemple avec  $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	4
{C}	{C}	2
{D}	{B, D}	2

# Calcul fermeture/support

Exemple avec  $minsup = 2/5$

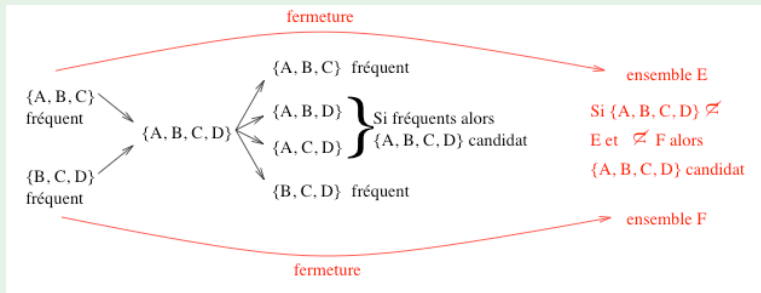
		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1/5
{B}	{B}	4/5
{C}	{C}	2/5
{D}	{B, D}	2/5

- On ajoute {B}, {C} et {B, D} à l'ensemble des ensembles de mots fréquents
- On conserve {B}, {C} et {D} pour calculer les générateurs de niveau supérieur

# Algorithme close

Les générateurs de niveau  $k + 1$  sont obtenus de la même manière que dans l'algorithme Apriori, mais ceux appartenant à la fermeture d'un générateur de niveau  $k$  sont supprimés.



- À partir de  $\{\{B\}, \{C\}, \{D\}\}$ , on génère les ensembles  $\{BC\}, \{BD\}, \{CD\}$
- $\{B, D\} \subset f(\{D\})$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
$\{BC\}$	$\{BC\}$	1/5
$\{CD\}$	$\emptyset$	0/5

Pas de nouvel ensemble de mots fréquents

# Algorithme fermeture

## FERMETURE( $E, T$ )

▷ **Entrée** :  $E$  : ensemble d'ensembles de mots,  $T$  : corpus

- 1 **Début**
- 2   **pour chaque**  $t \in T$  **faire**
- 3     **pour chaque**  $e \in E$  **faire**
- 4       **si**  $e \in T$  **alors**
- 5           $G \leftarrow G \cup e$
- 6     **pour chaque**  $e \in G$  **faire**
- 7       **si**  $\text{fermeture}(e) = \emptyset$  **alors**
- 8           $\text{fermeture}(e) \leftarrow G$
- 9       **sinon**  $\text{fermeture}(e) \leftarrow \text{fermeture}(e) \cap G$
- 10        $\text{support}(e) \leftarrow \text{support}(e) + 1$
- 11 **Fin**

# Algorithme génération

## CLOSE-GEN( $F, k$ )

- ▷ **Entrée** :  $F$  : ensemble d'ensembles de mots fréquents de cardinal  $k$
- 1 **Début**
- 2  $C \leftarrow \{c = f_1 \cup f_2, (f_1, f_2) \in F \times F, \text{card}(c) = k + 1\}$
- 3 **pour chaque**  $c \in C$  **faire**
- 4     **si**  $c \not\subset \text{fermeture}(f_1)$  **et**  $c \not\subset \text{fermeture}(f_2)$  **alors**
- 5         **pour chaque**  $s \subset c$  **faire**
- 6             **si**  $s \notin F$  **alors**
- 7                  $C \leftarrow C \setminus \{c\}$
- 8 **Retourner**  $C$
- 9 **Fin**

# Algorithme close

## CLOSE( $T, \text{minsup}$ )

- ▷ **Entrée** :  $T$  : corpus,  $\text{minsup}$  : entier
- 1 **Début**
- 2  $G_1 \leftarrow$  ensemble de mots de cardinal 1
- 3  $k \leftarrow 1$
- 4 **tantque**  $G_k \neq \emptyset$  **faire**
- 5      $C_k \leftarrow \text{FERMETURE}(G_k, T)$
- 6     **pour chaque**  $c \in C_k$  **faire**
- 7         **si**  $\text{support}(c) \geq \text{minsup}$  **alors**
- 8              $F_k \leftarrow F_k \cup \{c\}$
- 9              $\text{ferm}_k \leftarrow \text{ferm}_k \cup \text{ferm}(\{c\})$
- 10      $G_{k+1} \leftarrow \text{CLOSE-GEN}(F, k)$
- 11 **Retourner**  $\cup_k \text{ferm}_k$
- 12 **Fin**

# Plan

- 1 Présentation
- 2 Extraction d'ensemble de mots fréquents
- 3 Génération des règles d'association

# Génération des règles d'association

$e \in \{\text{ensemble de mots fréquents}\}, \text{card}(e) \geq 2$

- générer les sous-ensembles  $h$  de  $e$ ,  $h \neq \emptyset, h \neq e$
- calculer  $c = \text{support}(e) / \text{support}(e - h)$  ;
- si  $c \geq \text{minconf}$  alors générer la règle  $(e - h) \rightarrow h$ .

## ensemble mots fréquents

$$e = \{ABC\}$$

sous-ensemble $h$	règle $(e - h) \rightarrow h$
$\{A\}$	$BC \rightarrow A$
$\{B\}$	$AC \rightarrow B$
$\{C\}$	$AB \rightarrow C$
$\{AB\}$	$C \rightarrow AB$
$\{AC\}$	$B \rightarrow AC$
$\{BC\}$	$A \rightarrow BC$

## Réduction du nombre de règles

$$\begin{aligned} \text{support}(A) &\geq \text{support}(A, B) \\ \text{support}(A, B, C)/\text{support}(A) &\leq \text{support}(A, B, C)/\text{support}(A, B) \\ \text{confiance}(A \rightarrow BC) &\leq \text{confiance}(AB \rightarrow C) \end{aligned}$$

$$A \rightarrow BC \text{ valide} \Rightarrow \left. \begin{array}{l} AB \rightarrow C \\ AC \rightarrow B \end{array} \right\} \text{valides}$$

$$AB \rightarrow C \text{ non valide} \Rightarrow \left. \begin{array}{l} A \rightarrow BC \\ B \rightarrow AC \end{array} \right\} \text{non valides}$$

- Générer l'ensemble  $H$  des singletons de  $e$
- Pour chaque  $h \in H$ , calculer  $c = \text{support}(e) / \text{support}(e - h)$
- Si  $c \geq \text{minconf}$  alors ajouter la règle  $r : (e - h) \rightarrow h$
- Sinon enlever  $h$  de l'ensemble  $H$
- Générer un nouvel ensemble  $H$  d'ensembles de cardinal 2

## ensemble de mots fréquents

$\{ABC\}$

$H = \{\{A\}, \{B\}, \{C\}\}$

sous-ensemble	règle
$\{A\}$	$BC \rightarrow A$
$\{B\}$	$AC \rightarrow B$
$\{C\}$	$AB \rightarrow C$

Si  $BC \rightarrow A$  non valide,  $H = \{\{B, C\}\}$

sous-ensemble	règle
$\{BC\}$	$A \rightarrow BC$

# Génération des règles d'association

## GEN-RÈGLES( $E, \text{minsup}, \text{minconf}$ )

- ▷ **Entrée** :  $E$  : ensemble d'ensembles de mots,  $\text{minsup}, \text{minconf}$  : entiers
- 1 **Début**
  - 2   **pour chaque**  $e \in E, \text{card}(e) \geq 2$  **faire**
  - 3      $m \leftarrow 1$
  - 4      $H \leftarrow \{\text{singletons sous-ensemble de } e\}$
  - 5     **tantque**  $m \leq \text{card}(e)$  **faire**
  - 6       **pour chaque**  $h \in H$  **faire**
  - 7           $\text{confiance}(r) \leftarrow \text{support}(e) / \text{support}(e - h)$
  - 8          **si**  $\text{confiance}(r) \geq \text{minconf}$  **alors**
  - 9             $R \leftarrow R \cup \{(e - h) \rightarrow h\}$
  - 10         **sinon**  $H \leftarrow H \setminus \{h\}$
  - 11          $H \leftarrow \text{APRIORI-GEN}(H)$
  - 12          $m \leftarrow m + 1$
  - 13     **Retourner**  $R$
  - 14 **Fin**

# Références



R. Agrawal et R. Srikant.

Fast algorithms for mining association rules in large databases.

In *proceedings of the 20th international conference on Very Large Data Bases (VLDB'94)*, pages 478–499, 1994.



N. Pasquier Y. Bastide R. Taouil et L. Lakhal.

Pruning closed itemset lattices for association rules.

In *Actes des 14<sup>e</sup> journées Bases de Données Avancées (BDA'98)*, pages 177–196, 1998.

# Remerciements

Guillaume Duhamel