UNIVERSITÉ CÔTE D'AZUR

Introduction à l'Intelligence Artificielle (L2 Portail Sciences et Technologies)

Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
andrea.tettamanzi@univ-cotedazur.fr

univ-cotedazur.fr

Séance 6
Apprentissage non supervisé (règles d’association, analyse formelle de concepts)

Plan pour cette séance

- Apprentissage non-supervisé
- Analyse de patrons fréquents
- Règles d’association
  - Algorithme Apriori
  - Algorithme FPGrowth
  - Mesures d’intérêt
- Analyse formelle de concepts

Apprentissage

- Apprentissage automatique :
  - « apprendre » à partir d’exemples
  - résoudre des tâches sans être explicitement programmés pour chacune
- Approches basés sur
  - Mathématiques et statistiques
  - Algorithmique et structures de données
  - Calcul intensif (de plus en plus)

Apprentissage non supervisé

Selon les informations disponibles durant la phase d'apprentissage, l'apprentissage est qualifié de :

- **Supervisé** – si les données sont étiquetées (c-à-d la réponse est connue pour chaque exemple)
- **Non supervisé** – pas d'étiquette (cas le plus général)
  - on cherche à déterminer la structure sous-jacente des données

Analyse de patrons fréquents

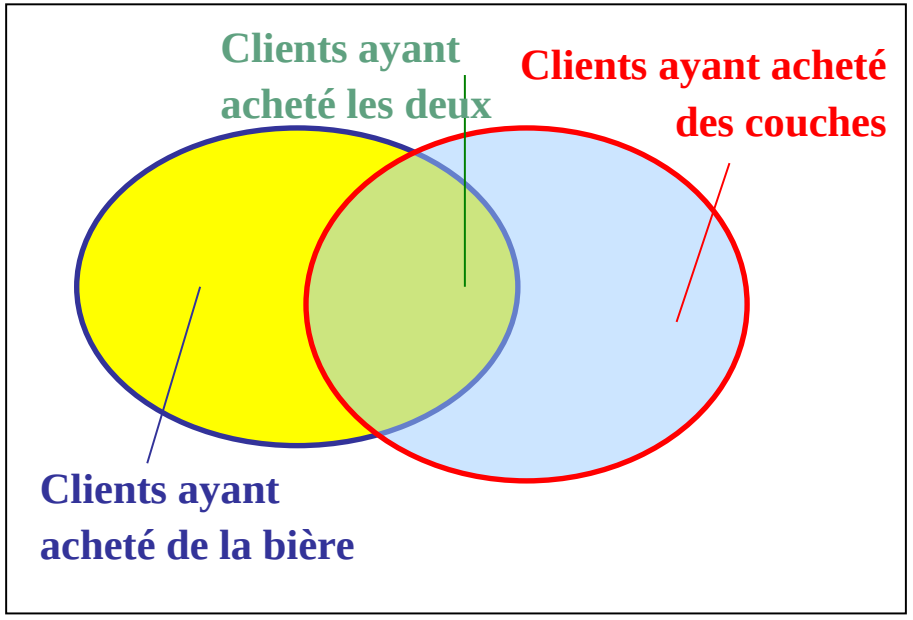
- Patron fréquent : un patron (ensembre d'articles, sous-séquences, sous-structures, couples attribut-valeur, etc.) qui apparaît fréquemment dans un jeu de données
- Proposé originairement par Agrawal, Imielinski et Swami [AIS93] dans le contexte de la fouille de données
- Motivation : trouver des régularités inhérentes dans les données
  - Quels produits sont souvent achetés ensemble— Bière et couches ?!
  - Qu'est-ce qu'on achète après avoir acheté un ordi ?
  - Quels types d'ADN sont sensibles à ce nouveau médicament ?
  - Pouvons-nous classifier automatiquement des pages Web ?
- Applications :
  - Analyse des données du panier, marketing croisé, conception de catalogue, analyse des campagnes de vente, analyse de logs Web (click stream), analyse des séquences ADN.

Pourquoi est-elle importante ?

- Expose les propriétés intrinsèques et importantes des jeux de données
- Constitue la base de nombreuses tâches essentielles d'exploration de données
  - Analyse d'association, de corrélation et de causalité
  - Motifs séquentiels, structurels (par exemple, sous-graphes)
  - Analyse des tendances en matière de données spatio-temporelles, multimédia, chronologiques et de flux
  - Classification : classification associative
  - Analyse de clusters : regroupement fréquent basé sur des modèles
  - Entreposage de données : hypercubes et gradient de cubes
  - Compression sémantique des données
  - Beaucoup d’applications

Concepts de base : patrons fréquents et règles d'association

Id Transaction	Articles achetés
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F



- Itemset X = $\{x_1, \ldots, x_k\}$
- Trouver toutes les règles $X \rightarrow Y$ ayant support et confiance $\geq$ minimum
  - support, *s*, probabilité que une transaction contienne $X \cup Y$
  - confiance, *c*, probabilité conditionnelle que une transaction contenant X contient aussi *Y*

Soit $sup_{min} = 50\%$, $conf_{min} = 50\%$

Pat. Freq. : {A:3, B:3, D:4, E:3, AD:3}

Règles d'association :

$A \rightarrow D$ (60%, 100%)

$D \rightarrow A$ (60%, 75%)

Patrons fermés et maximaux

- Un patron long contient un nombre combinatoire de sous-patrons, p.ex., les sous-patrons de {a_1, …, a_{100}} sont

$$\binom{1}{100} + \binom{2}{100} + \ldots + \binom{100}{100} = 2^{100} - 1 = 1.27 \cdot 10^{30}$$

- Solution: n'extraire que *patrons fermés et max-patrons*
- Un itemset X est fermé si X est *fréquent* et qu'il n'existe *aucun super-patron* Y ⊃ X, *ayant le même support* que X
- Un itemset X est un max-patron si X est fréquent et qu'il n'existe aucun super-patron fréquent Y ⊃ X
- Un patron fermé est une compression sans perte d'information de patrons fréquents
  - On réduit le nombre de patrons et de règles

Méthodes efficaces pour la fouille de patrons fréquents

- Propriété de fermeture descendante de patrons fréquents
  - Tout sousensemble d'un itemset fréquent est fréquent
  - Si **{bière, couches, cacahuètes}** est fréquent, alors **{bière, couches}** l'est aussi
  - En effet, toute transaction contenant {bière, couches, cachuètes} contient aussi {bière, couches}
- Méthodes efficaces : trois approches majeures
  - Apriori (Agrawal & Srikant@VLDB'94)
  - Frequent pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: une approche basée sur la génération et test de candidats

- Principe d'élagage d'Apriori : S'il y a un itemset qui est infréquent, ses super-ensembles ne devraient pas être générés/testés ! (Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94)
- Méthode :
  - Première passe : trouver les 1-itemsets (articles) fréquents
  - Générer itemsets candidats de taille (k + 1) à partir des itemsets fréquent de taille k
  - Tester ces candidats par rapport aux données
  - Terminer dès qu'aucun itemset candidat ne peut plus être généré

Exemple

$Sup_{min} = 2$

BD de transactions

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1e passe

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2e passe

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3e passe

L_3

Itemset	sup
{B, C, E}	2

Apriori : pseudocode

C_k: Itemsets candidats de taille k

L_k : Itemsets fréquents de taille k

L_1 = { articles fréquents };

pour (k = 1; $L_k \neq \varnothing$; k++) **:**

C_{k+1} = candidats générés de L_k;

pour toute transaction t dans BD:

incrementer le comptage des candidats dans C_{k+1} qui sont contenus en t

L_{k+1} = candidats dans C_{k+1} avec min_support

renvoyer $\cup_k L_k$;

Quelques détails importants

- Comment génère-t-on les candidats?
  - Étape 1 : faire une auto-jointure de L_k
  - Étape 2 : élagage
- Comment compte-t-on les supports des candidats ?
- Exemple de génération de candidats
  - L_3 = { *abc*, *abd*, *acd*, *ace*, *bcd* }
  - Auto-jointure : L_3 x L_3
    - *abcd*, de *abc* et *abd*
    - *acde*, de *acd* et *ace*
  - Élagage :
  - *acde* est supprimé, car *ade* n'est pas dans L_3
  - C_4 = { *abcd* }

Calcul du support des candidats

- Pourquoi ce calcul est-il problématique ?
  - Le nombre total des candidats peut être très important
  - Une transaction peut contenir beaucoup de candidats
- Méthode :
  - Le itemsets candidats sont stockés dans une structure de données appelée *arbre de hachage*
  - *Les feuilles* de l'arbre de hachage contiennent une liste d'itemsets avec leur nombre d'occurrences
  - *Les nœuds internes* contiennent une table de hachage
  - *La fonction Subset* trouve tous les candidats contenus dans une transaction

Goulet d'étranglement de la fouille de patrons fréquents

- Des passes multiples coûtent cher
- La fouille de patrons longs requiert plusieurs passes et génère plein de candidats
  - Pour trouver un itemset $i_1 i_2 \ldots i_{100}$
    - Nombre de passes : 100
    - Nombre de Candidats: $2^{100}-1 = 1.27*10^{30}$!
- Goulet d'étranglement : génération et test
- Peut-on éviter la génération de candidats ?

Fouille de patrons fréquents sans génération de candidats

- Faire pousser les patrons longs à partir des courts grâce aux articles localement fréquents
  - « abc » est un patron fréquent
  - Extraire transactions contenant abc: BD|abc
  - « d » est un article fréquent dans DB|abc → « abcd » est aussi un patron fréquent

FP-Tree

- On évite l’explosion combinatoire des candidats grâce à :
  - Une structure de donnée arborescente compacte
    - On évite les passes répétées sur la BD
  - Test restreint
    - Apriori : génération et test restreints
  - On utilise une recherche du genre « diviser pour mieux régner »
    - Apriori : construction du bas vers le haut

Algorithme FP-tree

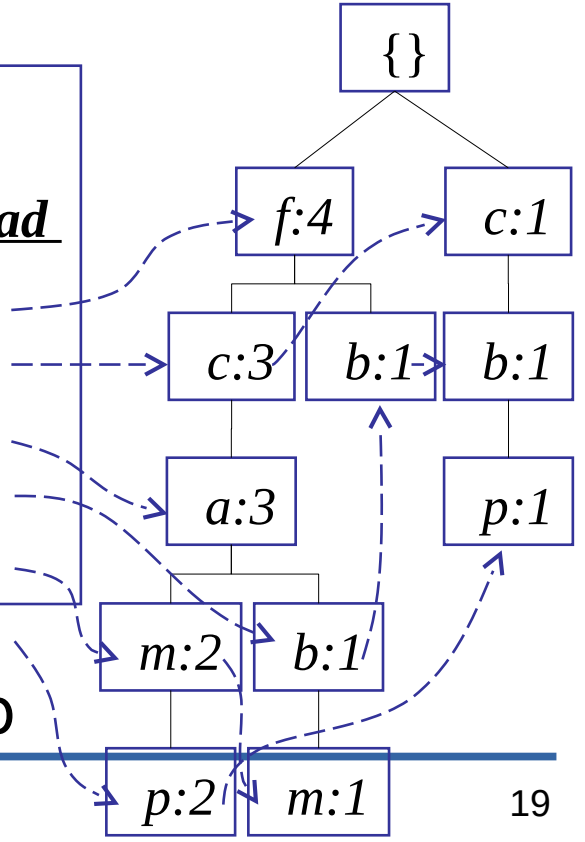
ID	Articles achetés	Articles fréquents (triés)
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

1. Faire une passe, trouver les articles fréquents (patrons de taille 1)
2. Trier les articles fréquents en ordre descendant, F-list
3. Faire autre passe, construire le FP-tree

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



F-list=f-c-a-b-m-p

Avantages de la structure FP-tree

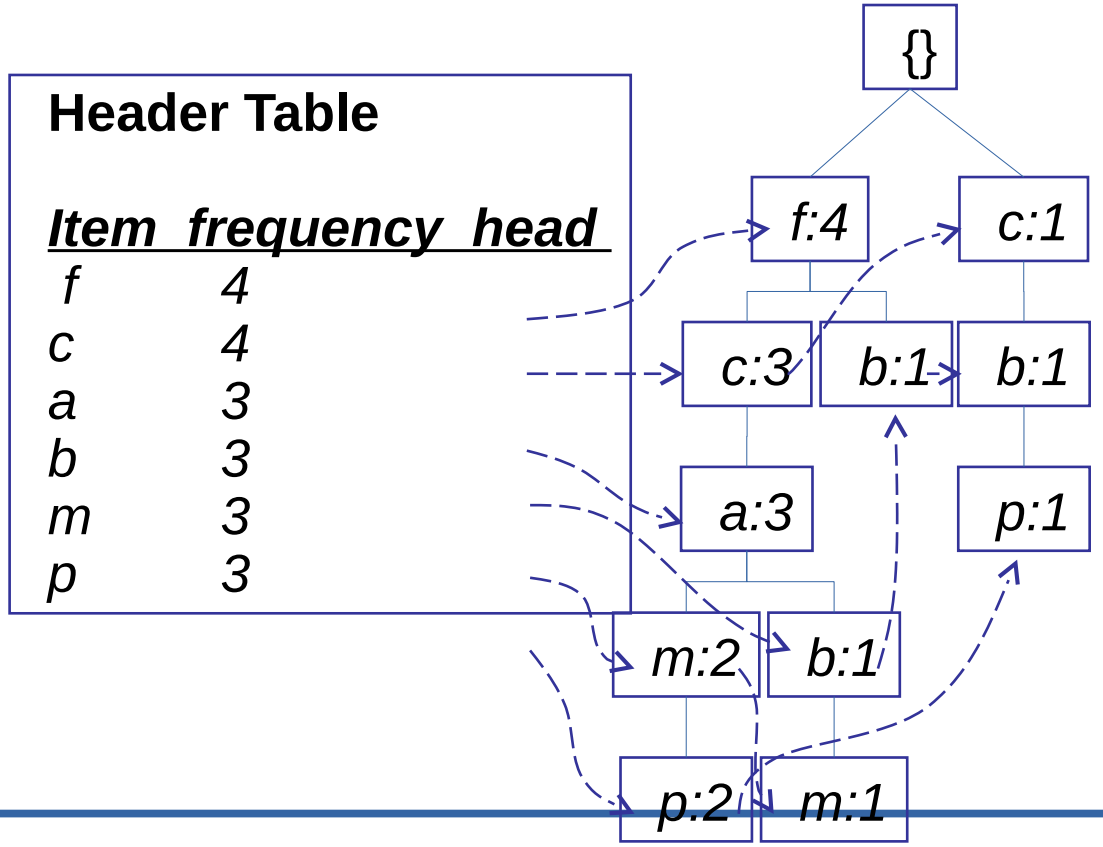
- Complétude
  - Préserve toute l’information pour la fouille de patrons fréquents
  - Ne coupe jamais un patron long d’aucune transaction
- Compacité
  - Réduit l’information non pertinente—plus de patrons non fréquents
  - Articles triés par fréquence descendante : plus ils apparaissent fréquemment, plus ils ont de chances d’être partagés
  - Jamais plus grand que la BD originale (sans compter les liens entre les nœuds et le champ comptage)
  - Pour certaines BD de benchmark, le taux de compression dépasse le 100

Partition des patrons et BD

- On peut partitionner les patrons fréquents en sous-ensembles suivant la F-list
  - F-list = f-c-a-b-m-p
  - Patrons qui contiennent p
  - Patrons qui contiennent m mais pas p
  - …
  - Patrons avec c mais ni a, ni b, m, p
  - Patron f
- Partition complète et non redondante

Trouver les patrons contenant p dans la BD p-conditionnelle

- On cherche un article p dans la header table du FP-tree
- On parcourt le FP-tree suivant le lien de chaque article fréquent *p*
- On accumule tous les *chemins-préfixes transformés* pour l'article *p* pour créer la BD des patrons contenant p

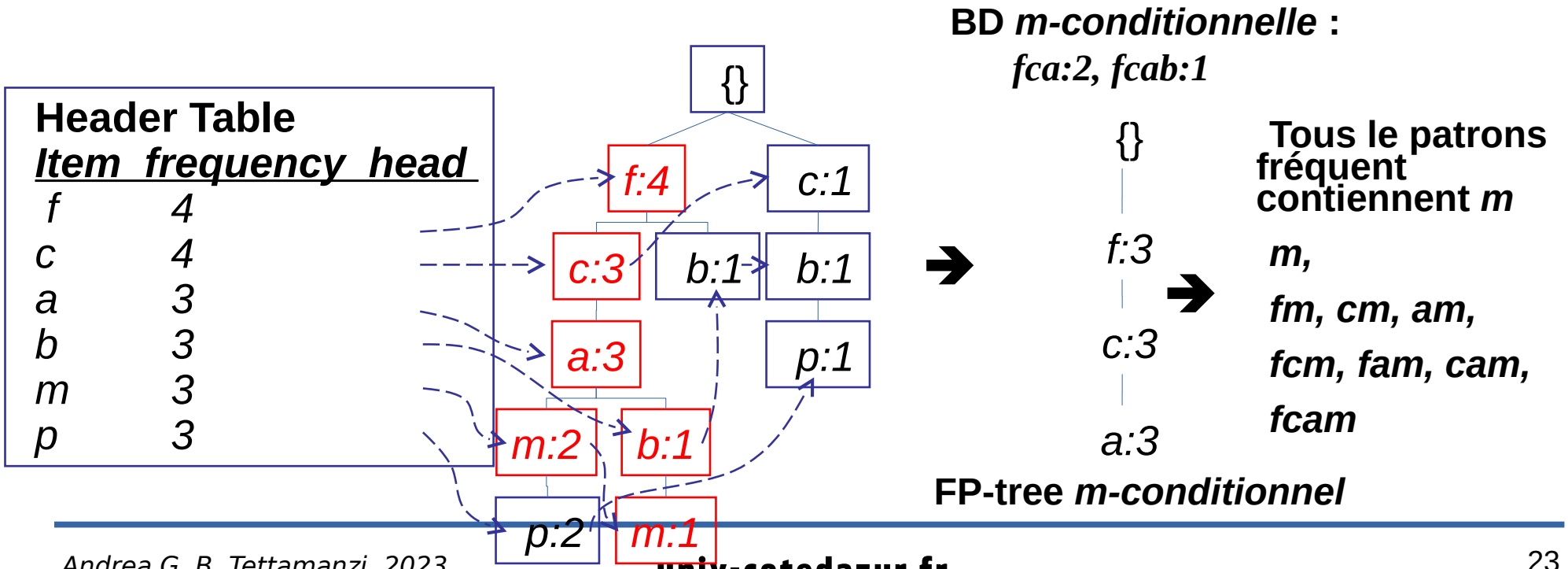


BD Conditionelles

art.	***BD conditionnelle***
c	***f:3***
a	***fc:3***
b	***fca:1, f:1, c:1***
m	***fca:2, fcab:1***
p	***fcam:2, cb:1***

Des BD conditionnelles de patrons aux FP-trees conditionnels

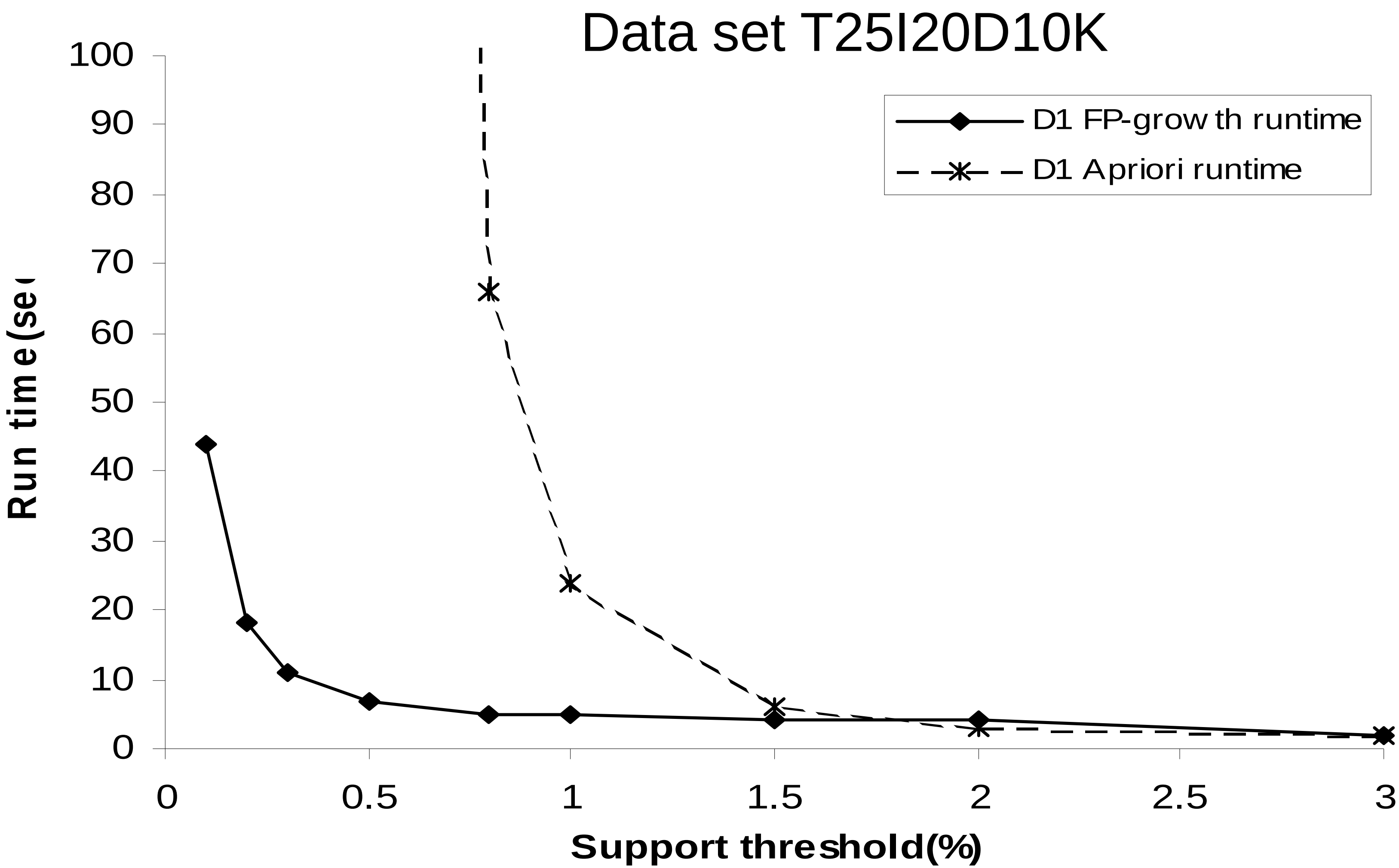
- Pour chaque BD de patrons
  - Accumuler les comptages de chaque article dans la BD
  - Construire le FP-tree des articles fréquents de la BD de patrons



Fouille de patrons fréquents avec FP-Trees

- Idée : « culture » de patrons fréquents
  - On fait pousser les patrons fréquents en partitionnant récursivement les patrons et la BD
- Méthode
  - Pour chaque article fréquent, on construit sa BD conditionnelle de patrons, puis son FP-tree conditionnel
  - On répète cette procédure sur chaque FP-tree conditionnel nouvellement créé
  - Jusqu'à ce que le FP-tree résultant ne soit vide, ou qu'il ne contienne qu'un chemin—ce chemin va générer toutes les combinaisons de ses sous-chemins, qui sont toutes des patrons fréquents

FP-Growth vs. Apriori: Passage à l'échelle par rapport au seuil de support



Pourquoi FP-Growth gagne-t-il ?

- Diviser pour mieux régner :
  - Il décompose la tâche de fouille et la BD suivant les patrons fréquents obtenus jusque là
  - Cela conduit à une recherche ciblée de BD plus petites
- Autres facteurs
  - Pas de génération de candidats, pas de test
  - Compression de la BD : structure FP-tree
  - Pas de passes répétées de toute la BD
  - Opérations basiques—contage des articles localement fréquents et construction de sous-FP-trees—pas de requête pour chercher des patrons

Fouille de patrons fréquents fermés : CLOSET

- Flist: liste de tous les articles fréquents triés par support croissant
  - Flist: d-a-f-e-c
- On divise l'espace de recherche
  - Patrons contenant d
  - Patrons contenant d, mais pas a, etc.
- On trouve les patrons fréq. fermés récursivement
  - Chaque transaction contenant d contient cfa aussi
    → cfad est un patron fréquent fermé

Min_sup=2

TID	Articles
10	a, c, d, e, f
20	a, b, e
30	c, e, f
40	a, c, d, f
50	c, e, f

Mesures d'intérêt : Corrélations (Lift)

- *Jouer football* ⇒ *manger céréales* [40%, 66.7%] est fourvoyant
  - Le % général des étudiants qui mangent des céréales est 75% > 66.7%.
- *Jouer football* ⇒ *ne pas manger céréales* [20%, 33.3%] est plus précis, bien qu'avec un moindre support et une moindre confiance
- Mesure d'événements dépendants/corrélés : lift

$$lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)}$$

	Football	Pas de football	Total
Céréales	2000	1750	3750
Pas de céréales	1000	250	1250
Total	3000	2000	5000

$$lift(F,C) = \frac{2000/5000}{3000/5000 \cdot 3750/5000} = 0.89$$

$$lift(F,\neg C) = \frac{1000/5000}{3000/5000 \cdot 1250/5000} = 1.33$$

Quelles mesures utiliser ?

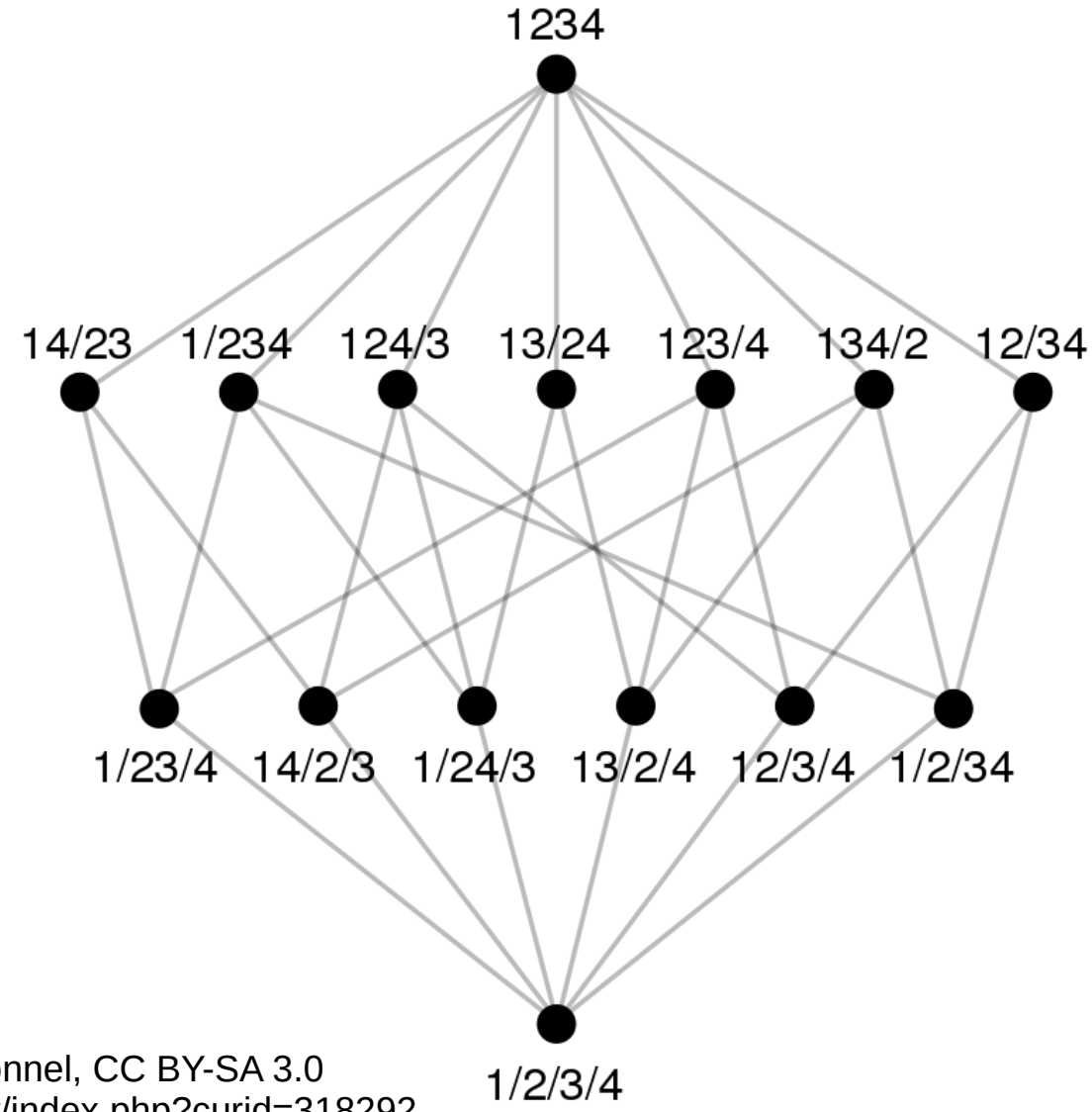
- **lift** et χ^2 ne sont pas de bonnes mesures de corrélation pour de grandes BD de transactions
- **all-conf** ou **coherence** pourraient marcher mieux
- **all-conf** et **coherence** ont toutes les deux la propriété de fermeture descendante
- On peut en dériver des algorithmes de fouille performants

symbol	measure	range	formula
ϕ	ϕ-coefficient	-1...1	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	-1 ...1	$\frac{P(A,B)P(\overline{A},\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B})+P(A,\overline{B})P(\overline{A},B)}$
Y	Yule's Y	-1 ...1	$\frac{\sqrt{P(A,B)P(\overline{A},\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}}$
k	Cohen's	-1 ...1	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
PS	Piatetsky-Shapiro's	-0.25 ...0.25	$P(A,B)-P(A)P(B)$
F	Certainty factor	-1 ...1	$\max(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)})$
AV	added value	-0.5 ...1	$\max(P(B\|A)-P(B), P(A\|B)-P(A))$
K	Klosgen's Q	-0.33 ...0.38	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$
g	Goodman-kruskal's	0 ...1	$\frac{\Sigma_j \max_k P(A_j,B_k)+\Sigma_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-max_k P(B_k)}$
M	Mutual Information	0 ...1	$\frac{\Sigma_i\Sigma_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_J)}}{\min(-\Sigma_i P(A_i)\log P(A_i)\log P(A_i), -\Sigma_i P(B_i)\log P(B_i)\log P(B_i))}$
J	J-Measure	0 ...1	$\max(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}))$
			$P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})$
G	Gini index	0 ...1	$\max(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A}[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]-P(B)^2-P(\overline{B})^2,$
			$P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B}[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]-P(A)^2-P(\overline{A})^2)$
s	support	0 ...1	$P(A,B)$
c	confidence	0 ...1	$max(P(B\|A), P(A\|B))$
L	Laplace	0 ...1	$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$
IS	Cosine	0 ...1	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
γ	coherence(Jaccard)	0 ...1	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
α	all_confidence	0 ...1	$\frac{P(A,B)}{\max(P(A),P(B))}$
o	odds ratio	0 ...∞	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(\overline{A},B)P(A,\overline{B})}$
V	Conviction	0.5 ...∞	$\max(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})})$
λ	lift	0 ...∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 ...∞	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
χ^2	χ^2	0 ...∞	$\Sigma_i\frac{(P(A_i)-E_i)^2}{E_i}$

Analyse Formelle de Concepts

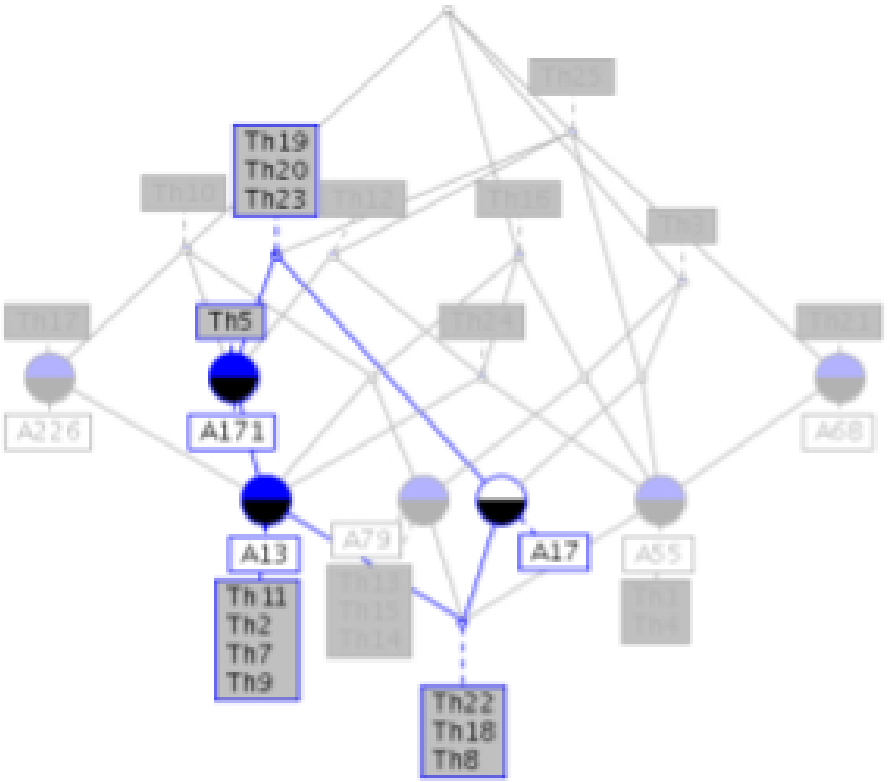
- Étude des concepts lorsqu'ils sont décrits formellement
- Introduite par Rudolf Wille en 1982 en tant qu'application de la théorie des treillis (voir treillis de Galois)
- Elle dispose également d'une solide base philosophique
- Un concept peut être défini par :
  - son intension : ensemble des attributs partagés par les instances du concept
  - son extension : ensemble des instance du concept (c-à-d les objets qui appartiennent au concept)

Treilli



Par User:ed_g2s — Travail personnel, CC BY-SA 3.0
https://commons.wikimedia.org/w/index.php?curid=318292

Treillis de Galois

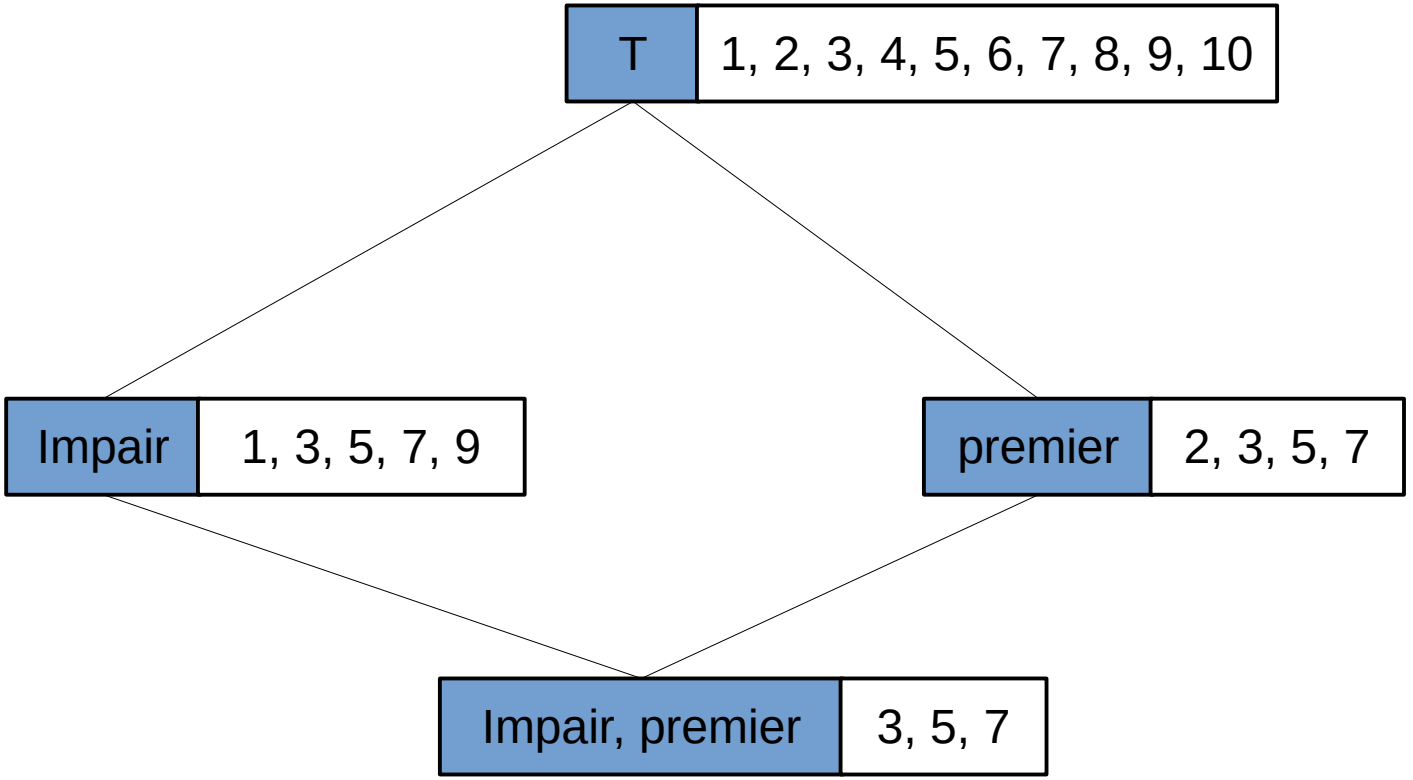


Par Idéalités — Travail personnel, CC BY-SA 4.0
https://commons.wikimedia.org/w/index.php?curid=74651384

Exemple

nombre	composé	pair	impair	premier	carré
1			x		x
2		x		x	
3			x	x	
4	x	x			x
5			x	x	
6	x	x			
7			x	x	
8	x	x			
9	x		x		x
10	x	x			

Exemple


T
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Impair
1, 3, 5, 7, 9
premier
2, 3, 5, 7
Impair, premier
3, 5, 7