

Extraction de Motifs :
Les règles d'association et les motifs séquentiels

HMIN326

Pascal Poncelet
LIRMM
Pascal.Poncelet@lirmm.fr
http://www.lirmm.fr/~poncelet

Plan

- Contexte général
- Règles d'association
- Motifs séquentiels
- Applications : Web Mining, Text Mining
- Conclusions

Le processus de KDD

Databases Datawarehouse DataMart Web → Données cibles → Pré-traitement et nettoyage → Données transformées → Motifs / Modèles → visualisation

3

Recherche de motifs fréquents

- Qu'est ce qu'un motif fréquent ?
 - Un motif (ensemble d'items, séquences, arbres, ...) qui intervient fréquemment ensemble dans une base de données [AIS93]
- Les motifs fréquents : une forme importante de régularité
 - Quels produits sont souvent achetés ensemble ?
 - Quelles sont les conséquences d'un ouragan ?
 - Quel est le prochain achat après un PC?



4

Recherche de motifs fréquents

- **Analyse des associations**
 - Panier de la ménagère, cross marketing, conception de catalogue, analyse de textes
 - Corrélation ou analyse de causalité
- **Clustering et Classification**
 - Classification basée sur les associations
- **Analyse de séquences**
 - Web Mining, détection de tendances, analyses ADN
 - Périodicité partielle, associations temporelles/cycliques



5

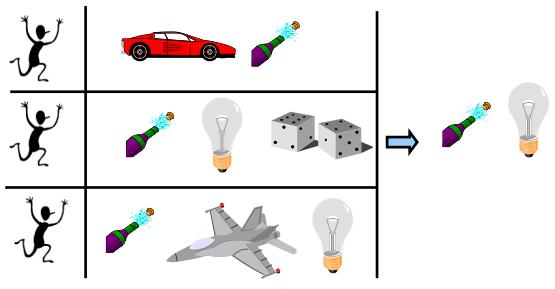
« Panier de la ménagère »

- **Recherche d'associations**
 - recherche de corrélations entre attributs (items)
 - caractéristiques : « panier de la ménagère »
 - de très grandes données
 - limitations : données binaires
- **Recherche de motifs séquentiels**
 - recherche de corrélations entre attributs (items) mais en prenant en compte le temps entre items => comportement



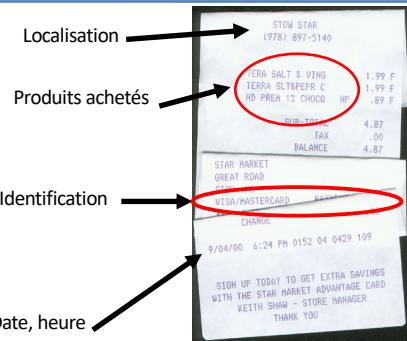
6

Recherche de règles d'association



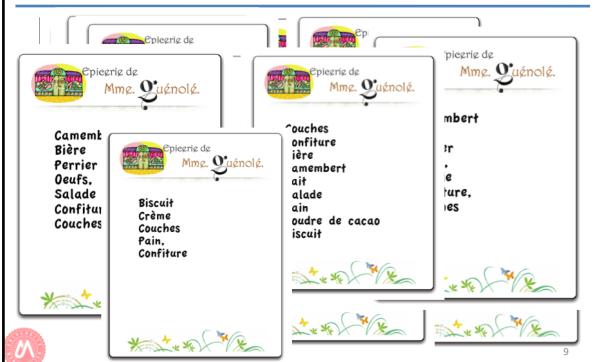
7

Panier de la ménagère



8

Aidons Mme Guénolé



9

La légende

Stories – Beer and Diapers

♦ Diapers and Beer. Most famous example of market basket analysis for the last few years. If you buy diapers, you tend to buy beer.

- T. Blischok headed Terradata's Industry Consulting group.
- K. Heath ran self joins in SQL (1990), trying to find two itemsets that have baby items, which are particularly profitable.
- Found this pattern in their data of 50 stores/90 day period.
- Unlikely to be significant, but it's a nice example that explains associations well.

Ronny Kohavi - ICML 1998

10

Recherche de règles d'association

- Règles de la forme

ANTECEDENT → CONSEQUENT [Support, Confiance]
(support et confiance sont des mesures d'intérêt définies par l'utilisateur)

- Achat(x, « Beurre ») ET Achat(x, « Pain ») → Achat(x, « Lait ») [70%, 80%]
- Achat(x, « Bière ») ET Achat(x, « Gâteaux ») → Achat(x, « Couches ») [30%, 80%]
- Achat(x, « Caviar ») → Achat(x, « Champagne ») [10%, 90%]

11

Panier de la ménagère

Identification

Localisation

LIVRE D'ARCHITECTURE
CONTENANT LES PRINCIPES GÉNÉRAUX DE CET ART.
LES PLANS, LES PROJETS & LES MODELES
DE QUATRE-VOISINS DES BÂTIMENTS FAITS EN FRANCE
Par J. Jon...
PARIS
L'IMPRIMERIE DE M. BOISSY
1750

Premier paragraphe

« Livre d'architecture contenant les principes généraux ... »

Position # Date

Mots # Produits

12

Interprétation

- $R : X \rightarrow Y (A\%, B\%)$
 - **Support :** portée de la règle
Proportion de paniers contenant tous les attributs
 $A\%$ des clients ont acheté les 2 articles X et Y
 - **Confiance :**
Proportion de paniers contenant le conséquent parmi ceux qui contiennent l'antécédent
 $B\%$ des clients qui ont acheté X ont aussi acheté Y
 - Beurre, Pain → Lait [70%, 80%]
 - Bière, Gâteaux → Couches [30%, 80%]
 - Caviar → Champagne [10%, 90%]

13

Utilisation des règles d'association

Bière, ... → Couches

- **Couches** comme conséquent
déterminer ce qu'il faut faire pour augmenter les ventes
- **Bière** comme antécédent
quel produit serait affecté si on n'arrête de vendre de la bière
- **Bière** comme antécédent et **Couche** comme conséquent
quels produits devraient être vendus avec la Bière pour promouvoir la vente de couches

14

Définitions des ensembles fréquents

- Soit un ensemble $I = \{I_1, I_2, \dots, I_m\}$ d'items, une transaction T est définie comme les sous-ensembles d'items dans I ($\subseteq I$).
 - $I = \{\text{Bière, Café, Couche, Gâteaux, Moutarde, Saucisse, ...}\}$
 - $T_1 = \{\text{Café, Moutarde, Saucisse}\}$
- Une transaction n'a pas de dupliques
- Soit une base de données D un ensemble de n transactions et chaque transaction est nommée par un identifiant (TID).
 - $D = \{\{T_1, \{\text{Café, Moutarde, Saucisse}\}\}, \{T_2, \{\text{Bière, Café, Gâteaux}\}\}, \dots\}$

15

Une base de données

- Une représentation de la base de données D

Client	Pizza	Lait	Sucre	Pommes	Café
1	1	0	0	0	0
2	0	1	1	0	0
3	1	0	0	1	1
4	0	1	0	0	1
5	1	0	1	1	1

- En fait

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

16

Définition des ensembles fréquents (cont.)

- Une transaction T **supporte** un ensemble $X \subseteq I$ si elle contient tous les items de X ($X \subseteq T$).
 - T1 supporte {Café, Moutarde, Saucisse}
- Support de X (**Supp(X)**) : fraction de toutes les transactions dans D qui supportent X.
- Si $\text{supp}(X) \geq s_{\min}$ l'ensemble X est dit **fréquent**.
- Un ensemble d'items (*itemset*) X de cardinalité $k = |X|$ est appelé un *k-itemset*.
3-itemset : {Café, Moutarde, Saucisse}

17

Propriétés des ensembles fréquents

- **Propriété 1 : support pour les sous-ensembles**
 - Si $A \subseteq B$ pour les itemsets A, B alors $\text{supp}(A) \geq \text{supp}(B)$ car toutes les transactions dans D qui supportent B supportent aussi nécessairement A.
 $A=\{\text{Café, Moutarde}\}$, $B=\{\text{Café, Moutarde, Saucisse}\}$
- **Propriété 2 : les sous-ensembles d'ensembles fréquents sont fréquents**
- **Propriété 3 : les sur-ensembles d'ensembles non fréquents sont non fréquents (anti-monotonie)**

18

Définition des Règles d'association

- Une règle d'association est une implication de la forme

$$R : X \rightarrow Y$$

où X et Y sont des itemsets disjoints :
 $X, Y \subseteq I$ et $X \cap Y = \emptyset$.

Bière, Gâteaux \rightarrow Couches



19

Définition des Règles d'association (cont.)

- Confiance (*confidence*) dans une règle R
- Si une transaction supporte X , elle supporte aussi Y avec une certaine probabilité appelée **confiance** de la règle ($conf(R)$).

$$\begin{aligned} conf(R) &= p(Y \subseteq T \mid X \subseteq T) \\ &= p(Y \subseteq T \wedge X \subseteq T) / p(X \subseteq T) \\ &= support(X \cup Y) / support(X) \end{aligned}$$

$$conf(R) = \frac{Supp(\text{Bière, Gâteaux, Couches})}{Supp(\text{Bière, Gâteaux})} \geq \text{confiance ?}$$



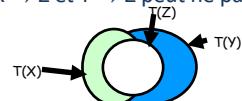
20

Propriétés des règles d'association

- **Propriété 4 : pas de composition des règles**
 - Si $X \rightarrow Z$ et $Y \rightarrow Z$ sont vrais dans D , $X \cup Y \rightarrow Z$ n'est pas nécessairement vrai.
 - Considérons le cas où $X \cap Y = \emptyset$ et les transactions dans D supportent Z si et seulement si elles supportent X ou Y , alors l'ensemble $X \cup Y$ a un support de 0 et donc $X \cup Y \rightarrow Z$ a une confiance de 0%.
- **Propriété 5 : décomposition des règles**
 - Si $X \cup Y \rightarrow Z$ convient, $X \rightarrow Z$ et $Y \rightarrow Z$ peut ne pas être vrai.



21



Propriétés des règles d'association

- Propriété 6 : pas de transitivité**

- Si $X \rightarrow Y$ et $Y \rightarrow Z$, nous ne pouvons pas en déduire que $X \rightarrow Z$.

- Propriété 7 : déduire si une règle convient**

- Si $A \rightarrow (L-A)$ ne vérifie pas la confiance alors nous n'avons pas $B \rightarrow (L-B)$ pour les itemsets L , A , B et $B \subseteq A$.



22

En résumé

- Itemsets : A , B ou B , E , F
- Support pour un itemset
 $\text{Supp}(A,D)=1$
 $\text{Supp}(A,C)=2$
- Itemsets fréquents ($\text{minSupp}=50\%$)
 $\{A,C\}$ est un itemset fréquent
- Pour $\text{minSupp} = 50\%$ et $\text{minConf} = 50\%$, nous avons les règles suivantes :
 $A \rightarrow C [50\%, 50\%]$
 $C \rightarrow A [50\%, 100\%]$

Trans. ID	Items
1	A, D
2	A, C
3	A, B, C
4	A, B, E, F



23

Schéma algorithmique de base

- La plupart des approches utilisent le même schéma algorithmique
- Pour construire les règles d'association, le support de tous les itemsets fréquents dans la base doit être calculé
- L'algorithme procède en deux phases :
 - 1) Génération de tous les ensembles fréquents
 - 2) Génération des règles d'association



24

Comptage des itemsets

- Une première approche
 - $I = \{A, B, C\}$
 - Génération de tous les cas possibles :
 - $\{\emptyset\}, \{A\}, \{B\}, \{C\},$
 - $\{A, B\}, \{A, C\}, \{B, C\}$
 - $\{A, B, C\}$
 - Comptage du support



25



Génération des ensembles fréquents

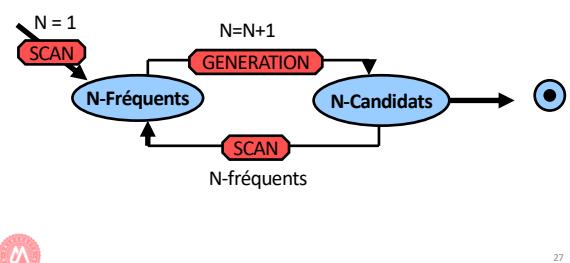
- Le nombre d ’ensemble fréquent potentiel est égal à la taille du produit cartésien de tous les items qui croit exponentiellement en fonction du nombre d ’items considérés.
 - Approche naïve : recherche exhaustive et test de tous les ensemble du produit cartésien pour savoir s ’ils sont fréquents
 - 1000 items => 2^{1000} ensembles à considérer



26



Vers un algorithme générique



27



Construction des règles

- Pour chaque ensemble fréquent X, chaque sous-ensemble est choisi comme antécédent de la règle, le reste devenant la partie conséquent.
- Comme X est fréquent, tous les sous-ensembles sont fréquents (Propriété 3) donc leur support est connu. La confiance d'une règle est calculée et une règle est conservée ou pas selon la confiance minimale.
- Amélioration : (Propriété 7) quand une règle échoue, aucun sous ensemble de l'antécédent n'est à considérer.



28

Bref historique

- Problématique initiée en 1993
- CPU vs. I/O
- De nombreux algorithmes ...
 - AIS - R. Agrawal, T. Imielinski and A. Swami - ACM SIGMOD 1993
 - SETM - Houtsma and Swami - IBM Technical Record
 - APRIORI - R. Agrawal and R. Srikant - VLDB 1994
 - PARTITION - A. Sarasere, E. Omiecinsky and S. Navathe - VLDB 1995
 - SAMPLING - H. Toivonen - VLDB 1996
 - DIC - S. Brin, R. Motwani, J. Ulman and S. Tsur - ACM SIGMOD 1997
 - PrefixSpan - J. Pei, J. Han, - ICDE'01
 - SPADE - M. Zaki - Machine Learning'01
 - ...2006, ...2010, 2014, 2016



29

L'algorithme APRIORI

- But : minimiser les candidats
- Principe : générer seulement les candidats pour lesquels tous les sous-ensembles ont été déterminés fréquents
- Génération des candidats réalisée avant et de manière séparée de l'étape de comptage



30

L'algorithme APRIORI

*Input : C_k : itemsets candidats de taille k
Output : L_k : itemsets fréquents de taille k*

```

 $L_1 = \{\text{items fréquents}\};$ 
for ( $k = 1; L_k \neq \emptyset; k++$ ) do
     $C_{k+1} = \text{candidats générés à partir de } L_k;$ 
    Pour chaque transaction  $t$  de la base de données, incrémenter le
    compteur de tous les candidats dans  $C_{k+1}$  qui sont contenus
    dans  $t$ 
     $L_{k+1} = \text{candidats dans } C_{k+1} \text{ avec minSupp}$ 
return  $\cup_k L_k$ ;

```



31

Détails d'APRIORI

- Comment générer les candidats ?
 - Etape 1: auto-jointure sur L_k
 - Etape 2: élagage
- Comment compter le support des candidats ?



32

Génération des candidats

- Les items de L_{k-1} sont ordonnés par ordre lexicographique
- Etape 1: auto-jointure sur L_{k-1}

```

        INSERT INTO  $C_k$ 
        SELECT p.item1, p.item2, ..., p.itemk-1, q.itemk-1
        FROM  $L_{k-1}$  p,  $L_{k-1}$  q
        WHERE p.item1=q.item1, ..., p.itemk-2=q.itemk-2, p.itemk-1 < q.itemk-1
    
```
- Etape 2: élagage


```

                For each itemset c in  $C_k$  do
                    For each (k-1)-subsets s of c do if (s is not in  $L_{k-1}$ ) then delete c from  $C_k$ 
            
```



33

Génération des candidats : exemple

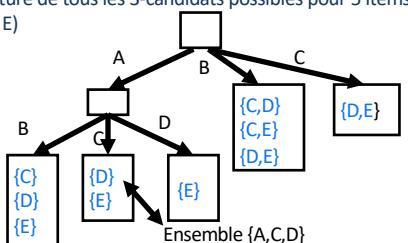
- $L_3 = \{abc, abd, acd, ace, bcd\}$
- Auto-jointure : $L_3 * L_3$
 - $abcd$ à partir de abc et abd
 - $acde$ à partir de acd et ace
- Élagage :
 - $acde$ est supprimé car ade n'est pas dans L_3
- $C_4 = \{abcd\}$



34

Stockage des candidats

- un arbre (structure de hash-tree)
- structure de tous les 3-candidats possibles pour 5 items (A, B, C, D, E)



35

Comptage du support des candidats

- Parcourir la base. Pour chaque tuple extrait t , compter tous les candidats inclus dedans
 - Rechercher toutes les feuilles qui peuvent contenir les candidats
 - Hachage sur chaque item du tuple et descente dans l'arbre des candidats
- Dans les feuilles de l'arbre vérifier ceux effectivement supportés par t
- Incrémenter leur support



36

Illustration

CID	Items
1	A B
2	A B C D E F
3	B D G
4	B E G
5	D F G
6	D E G
7	B E
8	B D E F

Support minimal = 1

37



Illustration

C1	Support
A	2
B	6
C	1
D	5
E	5
F	3
G	4

L1 = {{A},{B},{C},{D},{E},{F},{G}} 1-itemsets fréquents

38



Illustration

C2	Support	C2	Support
AB	2	CD	1
AC	1	CE	1
AD	1	CF	1
AE	1	CG	0
AF	1	DE	3
AG	0	DF	3
BC	1	DG	3
BD	3	EF	2
BE	4	EG	2
BF	2	FG	1
BG	2		

2-itemsets fréquents{{A,B},{A,C},{A,D},{A,E},{A,F},{B,C},{B,D},{B,E},{B,F},{B,G},
{C,D},{C,E},{C,F},{D,E},{D,F},{D,G},{E,F},{E,G},{F,G}}

39



Illustration

C3	Support	C3	Support
ABC	1	BDE	2
ABD	1	BDF	2
ABE	1	BDG	1
ABF	1	BEF	2
ACD	1	BEG	1
ACE	1	BFG	0
...
BCF	1	EFG	0

$L_3 = \{\{A, B, C\}, \{A, B, D\}, \{A, B, E\}, \{A, B, F\}, \{A, C, D\}, \dots \{D, F, G\}\}$

{B,C,G} élagué par Apriori-Gen car {C, G} n'appartient pas à L_2

40

Illustration

C4	Support	C4	Support
ABCD	1	ACEF	1
ABCE	1	ADEF	1
ABCF	1	BCDE	1
ABDE	1	BCDF	1
ABDF	1	BCEF	1
ABEF	1	BDEF	2
ACDE	1	BDEG	0
ACDF	1	CDEF	0

$L_4 = \{\{A, B, C, D\}, \{A, B, C, E\}, \{A, B, C, F\}, \dots \{C, D, E, F\}\}$

{B,D,F,G}, {B,E,F,G} élagués car {B,F,G} n'appartient pas à L_3

{D,E,F,G} élagué car {E,F,G} n'appartient pas à L_3

41

Illustration

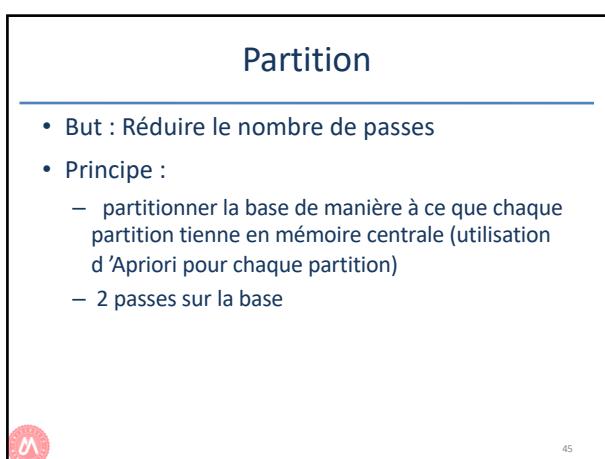
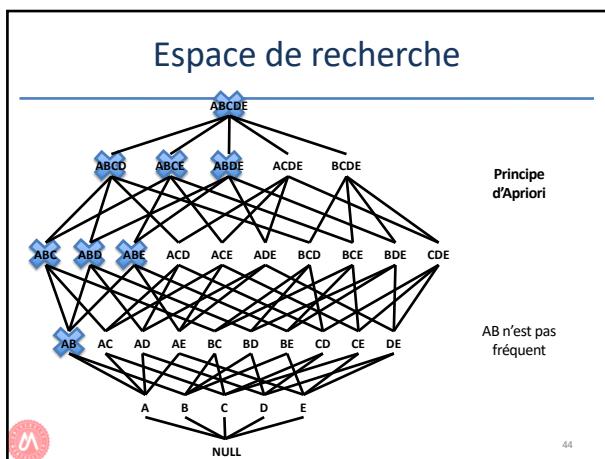
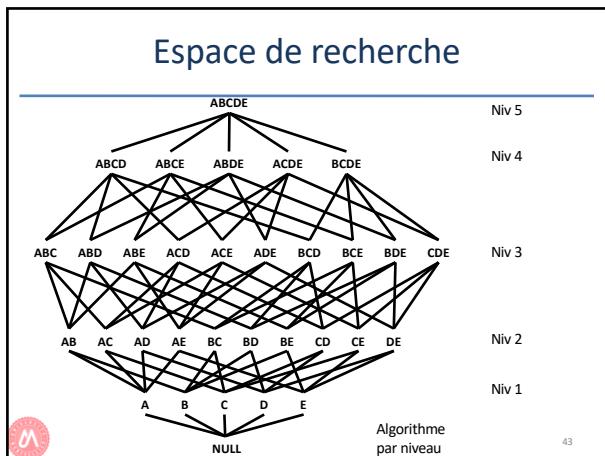
C6	Support
ABCDEF	1

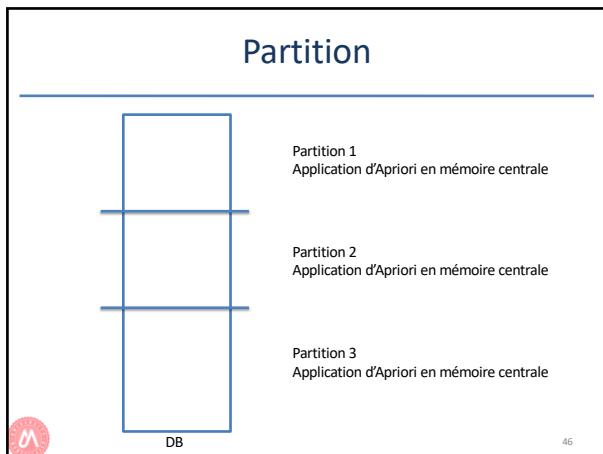
6-itemsets fréquents $L_6 = \{\{A, B, C, D, E, F\}\}$

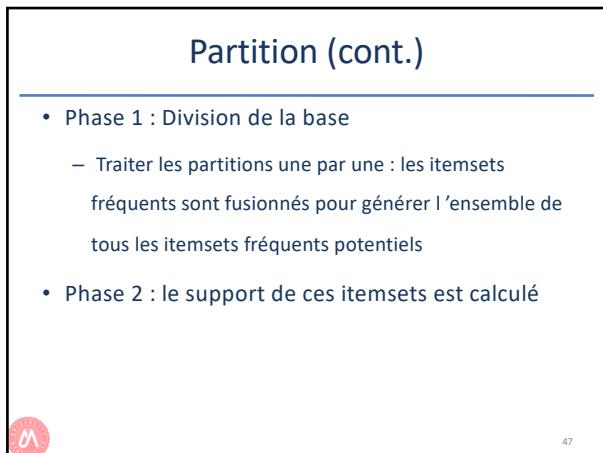
$C_7 = \{\emptyset\} \Rightarrow$ l'algorithme se termine.

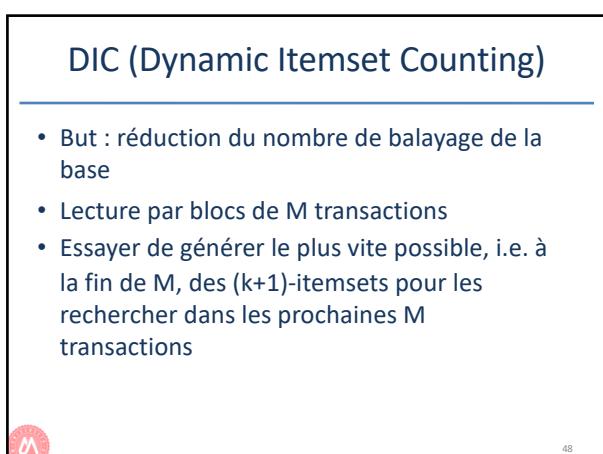
7 balayages pour déterminer tous les itemsets fréquents

42

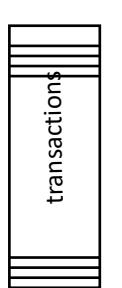








DIC (Cont.)



2-itemsets

3-itemsets

4-itemsets

2-itemsets

3-itemsets

4-itemsets

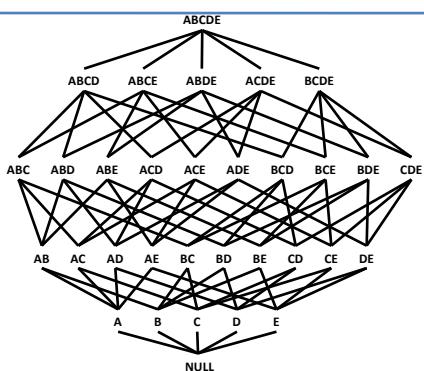
49

Sampling

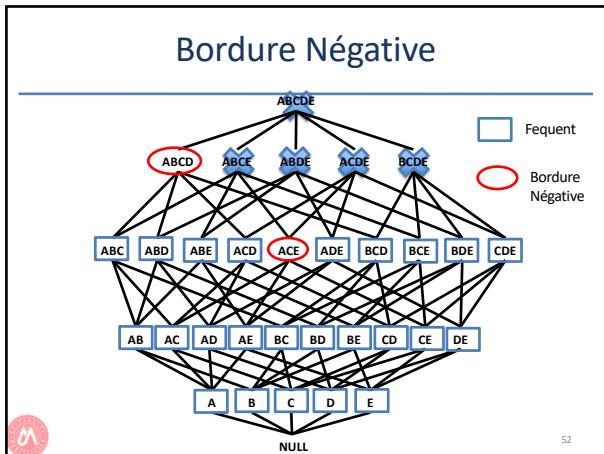
- Idée : prendre un ensemble aléatoire qui réside en mémoire centrale et rechercher tous les itemsets fréquents
- Très efficace : 1 passe, 2 passes au pire
- Basée sur la bordure négative

50

Bordure Négative



51



Sampling (cont.)

- **Algorithme**

support minimum, petit support minimum, une base et un échantillon de la base

 - 1 - prendre un échantillon de la base
 - 2 - Calculer les fréquents avec petit support minimum en mémoire centrale : Fréquents et Bordure
 - 3 - Evaluer la fréquence des itemsets fréquents et de la bordure négative sur le reste de la base
 - 4 - Retourner le résultat et les éventuels manques

53

Sampling (cont.)

- $D = 10$ millions de tuples - $A \dots F$ - support minimum = 2% - Echantillon s de 20 000 tuples petit support minimum = 1,5%

Pour l'échantillon avec 1,5% : $F = \{\{A, B, C\}, \{A, C, F\}, \{A, D\}, \{B, D\}\}$
 Bordure négative = $BN = \{\{B, F\}, \{C, D\}, \{D, F\}, \{E\}\}$

- Evaluer F et BD sur le reste de la base avec 2%
 - 1 - on trouve $\{A, B\}, \{A, C, F\}$ en une passe
 - 2 - si $\{B, F\}$ devient fréquent sur D => manque peut être $\{A, B, F\}$
 \Rightarrow reporter l'erreur et effectuer une seconde passe

54

MaxMiner : Mining Max-patterns

- But : rechercher les longs itemsets fréquents
- Max-patterns : bordures de motifs fréquents
 - Un sous-ensemble d'un max-pattern est fréquent
 - Un sur-ensemble d'un max-pattern est non fréquent
- Parcours en largeur et en profondeur



55

MaxMiner : Mining Max-patterns (cont.)

- 1er passage: rechercher les items fréquents
 - A, B, C, D, E
- 2nd passage: rechercher les support pour
 - AB, AC, AD, AE, **ABCDE**
 - BC, BD, BE, **BCDE**
 - CD, CE, **CDE**, DE,
- Comme BCDE est un max-pattern, il n'est pas nécessaire de vérifier BCD, BDE, CDE dans les parcours suivants

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

minSupp=2

56

Génération des candidats

- Depuis 2000 « La base peut tenir en mémoire »
- Constat : génération d'un trop grand nombre de candidats
 - s'il y a 10^4 1-itemset => génération de 10^7 candidats 2-itemsets
 - Pour un fréquent de 100, il faut générer plus de 10^{30} candidats au total
- Est-il possible de proposer une méthode qui évite de générer des candidats ?



57

FP-Tree

1 - Parcours de la base pour rechercher les 1-fréquents
 2 - Tri des fréquents dans l'ordre décroissant

TID	Items	Items triés
1	I1, I2, I5	I2, I1, I5
2	I2, I4	I2, I4
3	I2, I3	I2, I3
4	I1, I2, I4	I2, I1, I4
5	I1, I3	I1, I3
6	I2, I3	I2, I3
7	I1, I3	I1, I3
8	I1, I2, I3, I5	I2, I1, I3, I5
9	I1, I2, I3	I2, I1, I3

58

$L = [\quad I2:7,$
 $\quad I1:6,$
 $\quad I3:6,$
 $\quad I4:2,$
 $\quad I5:2]$

FP-Tree (cont.)

Parcourir les transactions de la base
 Création du FP-Tree :

« faire glisser les transactions dans l'arbre »
 - Une branche existe : incrémenter le support
 - Créer la branche autrement

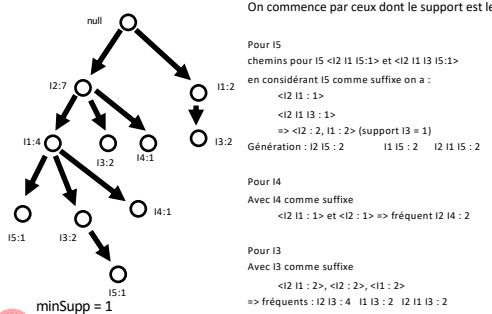
59

FP-Tree (cont.)

Association d'un tableau de pointeurs trié

60

FP-Tree (cont.)



61

Bénéfices de FP-tree

- Préserve l'information complète pour l'extraction d'itemsets
 - Pas de passage supplémentaire sur la base
- Approche Compacte
 - Les items sont triés dans un ordre décroissant de fréquence : plus ils apparaissent fréquemment plus ils seront partagés
 - Ne peut jamais être plus grand que la base d'origine (sans compter les liens, les nœuds et les compteurs)

62

Trop de fréquents

TID	Items
1	A,B,C,D
2	A,B,C
3	A,B,C
4	B,C,D

Avec support minimal = 2

A = 3/4
B = 4/4
C = 4/4
D = 2/4
AB = 3/4
AC = 3/4
BC = 4/4
BD = 2/4
ABC = 3/4
BCD = 2/4

4 tuples dans la base de données
10 itemsets extraits

63

Cas des données corrélées

$$M = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- D'autres types d'algorithmes
 - Utilisation du treillis et de ses propriétés
 - Recherche des itemsets fermés fréquents (les itemsets maximaux pour lesquels il n'existe pas de super ensemble avec la même valeur de support)
 - Recherche des générateurs
 - Recherche de représentation condensée (clos, libres, dérivables)
- Close, Close+, Charm ...



64

Quelques conclusions

- De nombreux travaux
 - De nouvelles approches condensées
 - De nouvelles contraintes (réduire l'espace de recherche)
 - Préservation de la vie privée
 - Approches Incrémentales
 - Règles plus générales
 - Définir de nouvelles mesures (lift, implication, ...)



65

Règles d'association incrémentales

- Générer les règles dans une base dynamique
- Problème : les algorithmes considèrent des bases statiques
- Objectifs :
 - Chercher les itemsets fréquents dans D
 - Chercher les itemsets fréquents dans $D \cup \{\Delta D\}$
- Doit être fréquent dans D ou ΔD
- Sauvegarder tous les fréquents, la bordure
- ... Data Streams (Flots de Données)



66

Des règles plus générales

- Les règles négatives
 $\text{Expr}(C_i) \rightarrow \text{Expr}(C_j)$ avec AND, OR, NOT
- Les règles sur plusieurs dimensions
- Les règles à attributs variables
 $\text{Age} \in [x,y] \Rightarrow \text{Salaire} > 45 \text{ K€} (5%; 30%)$
- Les règles approximatives
- Les règles avec généralisation
 Associée à une taxonomie



67

Utilité des règles

- La règle utile contenant des informations de qualité qui peuvent être mises en pratique
 ex : *le samedi, les clients des épiceries achètent en même temps de la bière et des couches*
- Résultats connus par quiconque
 ex : *les clients des épiceries achètent en même temps du pain et du beurre*
- Résultats inexplicables difficiles à situer et donc à expliquer
 ex : *lors de l'ouverture d'une quincaillerie, parmi les articles les plus vendus on trouve les abattants de toilette*



68

D'autres mesures

Articles	A	B	C	A, B	A, C	B, C	A, B, C
Fréquences (%)	45	42,5	40	25	20	15	5

- Si on considère les règles à trois articles, elles ont le même support 5%. Le niveau de confiance est alors :

Règle	Confiance
A, B → C	0,20
A, C → B	0,25
B, C → A	0,33

- La règle « B, C → A » possède la plus grande confiance. si B et C apparaissent simultanément dans un achat alors A y apparaît aussi avec une probabilité estimée de 33%.



69

D'autres mesures (cont.)

Articles	A	B	C	A, B	A, C	B, C	A, B, C
Fréquences (%)	45	42,5	40	25	20	15	5

- A apparaît dans 45% des achats. Il vaut donc mieux prédire A sans autre information que de prédire A lorsque B et C apparaissent.
 - l'*amélioration* permet de comparer le résultat de la prédiction en utilisant la fréquence du résultat

Amélioration = confiance / fréquence(résultat)



70



D'autres mesures (cont.)

- Une règle est intéressante lorsque l'amélioration est supérieure à 1. Pour les règles choisies, on trouve :

Règle	Confiance	Freq(résultat)	Amélioration
A, B → C	0.20	40%	0.50
A,C → B	0.25	42.5%	0.59
B,C → A	0.33	45%	0.74

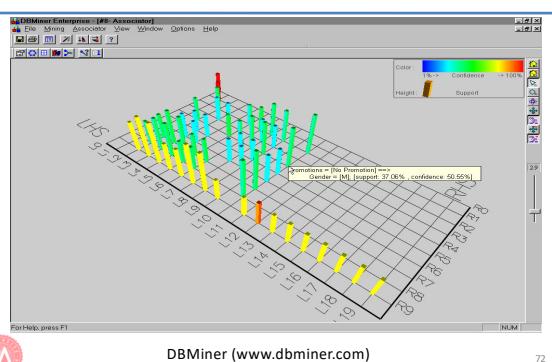
- Par contre, la règle si « A → B » possède un support de 25%, une confiance de 0.55 et une amélioration de 1.31, cette règle est donc la meilleure.
 - En règle générale, la meilleure règle est celle qui contient le moins d'articles.



71



Visualisation

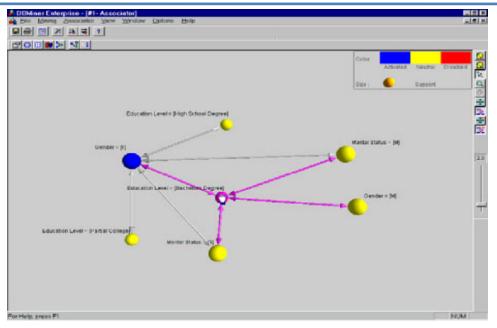


DBMiner (www.dbminer.com)

72

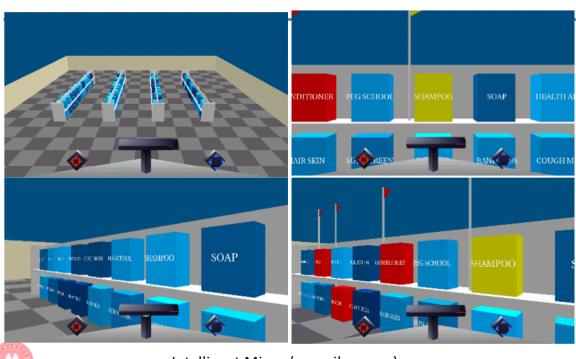


Visualisation

DBMiner (www.dbminer.com)

73

Visualisation

Intelligent Miner (www.ibm.com)

74

Pourquoi la recherche de séquence ?

- Un important domaine de recherche pour le data mining avec de très nombreuses applications
 - Analyse des achats des clients
 - Analyse de puces ADN
 - Processus
 - Conséquences de catastrophes naturelles
 - Web mining
 - Détection de tendances dans des données textuelles



75

Recherche de Motifs Séquentiels

- Même problématique mais avec le temps
- Item : « un article »
- Transaction : un client + un itemset + une estampille temporelle $T = [C, (a,b,c)_5]$
- Séquence : liste ordonnée d'itemsets
- Séquence de données : « activité du client »
Soit T_1, T_2, \dots, T_n , les transactions du client C, la séquence de données de C est :
 $[C, \langle \text{itemset}(T_1) \text{ itemset}(T_2) \dots \text{ itemset}(T_n) \rangle]$



76

Recherche de Motifs Séquentiels

- Support minimal : nombre minimum d'occurrences d'un motif séquentiel pour être considéré comme fréquent
- Attention l'occurrence n'est prise en compte qu'une fois dans la séquence

Support (20) dans $\langle(10)(20\ 30)(40)(20)\rangle = 1$



77

Inclusion

- Inclusion : Soient $S_1 = \langle a_1 a_2 \dots a_n \rangle$ et $S_2 = \langle b_1 b_2 \dots b_n \rangle$ $S_1 \subseteq S_2$ ssi
 $i_1 < i_2 < \dots < i_n / a_1 \subseteq b_{i1}, \dots, a_n \subseteq b_{in}$
- $S_1 = \langle(10)(20\ 30)(40)(20)\rangle$

$S_2 = \langle(20)(40)\rangle \subseteq S_1$

$S_3 = \langle(20)(30)\rangle$ n'est pas incluse dans S_1



78

Problématique

- Soit D une base de données de transactions de clients. Soit σ une valeur de support minimal
Rechercher toutes les séquences S telles que :
 $\text{support}(S) \geq \sigma$ dans D
- 50% des personnes qui achètent du vin et du fromage **le lundi** achètent aussi **du pain le vendredi**
<(French wine, cheese) (bread)>



79

Illustration

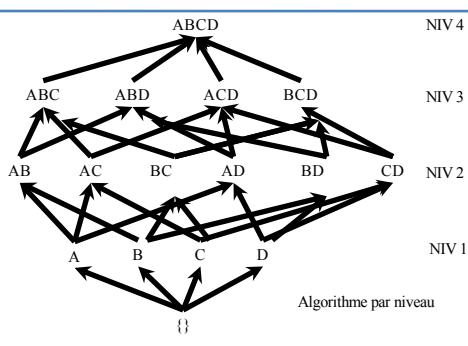
Clients	Date1	Date2	Date3	Date4
C1	10 20 30	20 40 50	10 20 60	10 40
C2	10 20 50	10 20 30		20 30 60
C3	20 30 50		10 40 60	10 20 30
C4	10 30 60	20 40	10 20 60	50

Support = 60% (3 clients) => <(10 30) (20) (20 60)>



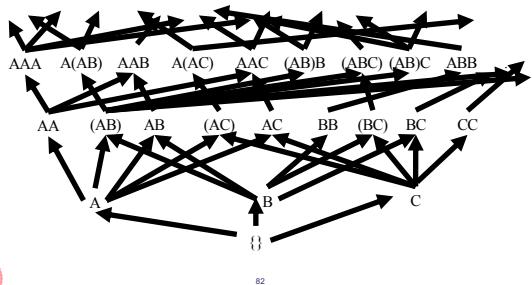
80

Itemsets : Espace de recherche



81

Motifs Séquentiels : l'espace de recherche



82

La propriété d'antimonotonie

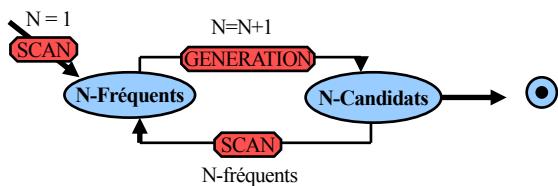
- Une propriété essentielle (c.f. Apriori [AIS93])

— Si une séquence n'est pas fréquente,
aucune des super-séquences de S n'est
fréquente!

Support (<(10) (20 30)>) < minsupp
Support (<(10) (20 30) (40)>) << minsupp

83

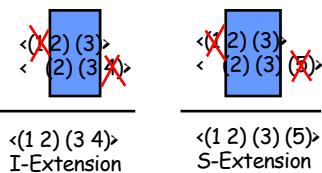
Vers un algorithme générique



84

Génération des candidats

- S-Extension : ajout d'une séquence
- I-Extension : ajout d'un itemset



85



GSP

- A la APRIORI [Srikant, Agrawal, EDBT'96]

```
L=1
While (ResultL != NULL)
    Candidate Generate
    Prune
    Test
    L=L+1
```

86



Recherche des séquences de taille 1

- Candidats initiaux : toutes les séquences réduites à un item
 - $\langle a \rangle, \langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle, \langle f \rangle, \langle g \rangle, \langle h \rangle$
- Un passage sur la base pour compter le support des candidats

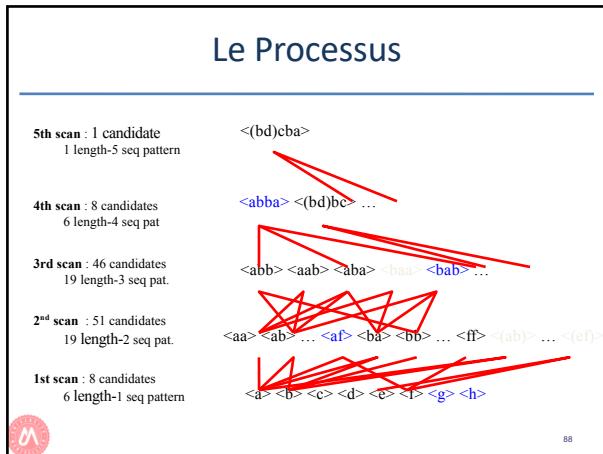
Seq. ID	Séquence
10	$\langle(bd)cb(ac)\rangle$
20	$\langle(bf)(ce)b(fg)\rangle$
30	$\langle(ab)(bf)abf\rangle$
40	$\langle(be)(ce)d\rangle$
50	$\langle(a bd)bc b(ade)\rangle$

 $minSupp = 2$

87

Cand	Sup
$\langle a \rangle$	3
$\langle b \rangle$	5
$\langle c \rangle$	4
$\langle d \rangle$	3
$\langle e \rangle$	3
$\langle f \rangle$	2
$\langle g \rangle$	1
$\langle h \rangle$	1





Génération des candidats de taille 2

S-Extension

	$\langle a \rangle$	$\langle b \rangle$	$\langle c \rangle$	$\langle d \rangle$	$\langle e \rangle$	$\langle f \rangle$
$\langle a \rangle$		$\langle aa \rangle$	$\langle ab \rangle$	$\langle ac \rangle$	$\langle ad \rangle$	$\langle ae \rangle$
$\langle b \rangle$		$\langle ba \rangle$	$\langle bb \rangle$	$\langle bc \rangle$	$\langle bd \rangle$	$\langle be \rangle$
$\langle c \rangle$			$\langle ca \rangle$	$\langle cb \rangle$	$\langle cc \rangle$	$\langle cd \rangle$
$\langle d \rangle$				$\langle da \rangle$	$\langle db \rangle$	$\langle dc \rangle$
$\langle e \rangle$					$\langle ea \rangle$	$\langle eb \rangle$
$\langle f \rangle$						$\langle fa \rangle$

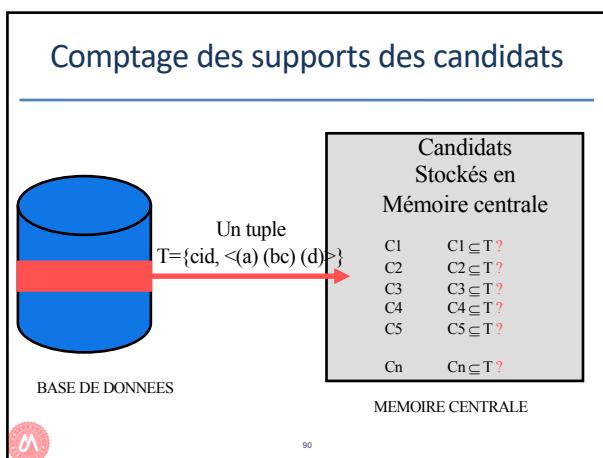
51 2-Candidats

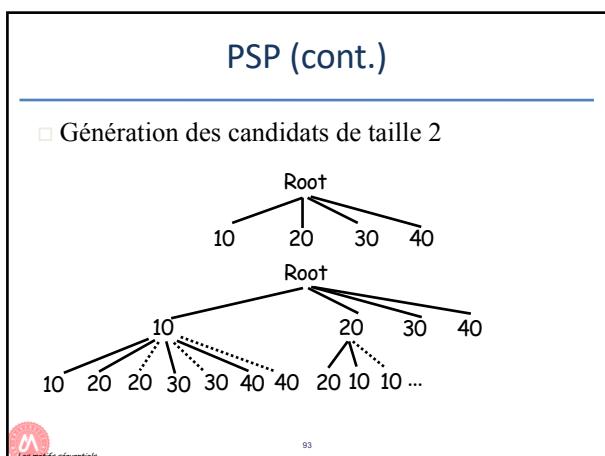
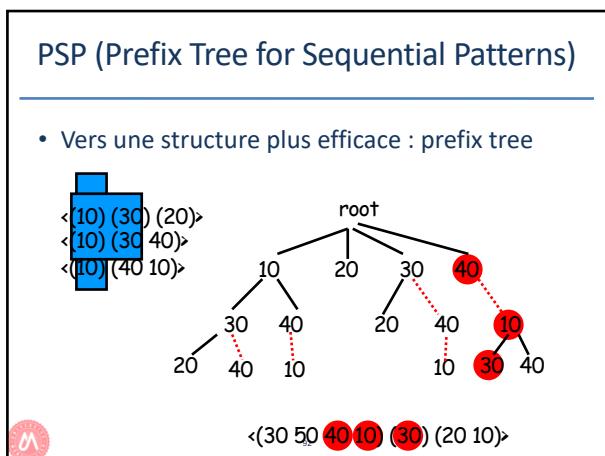
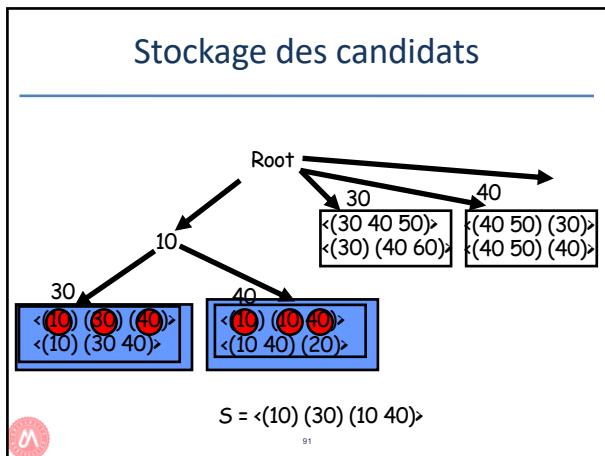
I-Extension

	$\langle a \rangle$	$\langle b \rangle$	$\langle c \rangle$	$\langle d \rangle$	$\langle e \rangle$	$\langle f \rangle$
$\langle a \rangle$		$\langle(ab) \rangle$	$\langle(ac) \rangle$	$\langle(ad) \rangle$	$\langle(af) \rangle$	
$\langle b \rangle$			$\langle(bc) \rangle$	$\langle(bd) \rangle$	$\langle(be) \rangle$	$\langle(bf) \rangle$
$\langle c \rangle$				$\langle(cd) \rangle$	$\langle(ce) \rangle$	$\langle(cf) \rangle$
$\langle d \rangle$					$\langle(de) \rangle$	$\langle(df) \rangle$
$\langle e \rangle$						$\langle(ef) \rangle$
$\langle f \rangle$						

Sans la propriété d'anti-monotonie
 $8*8+8*7/2=92$ candidats

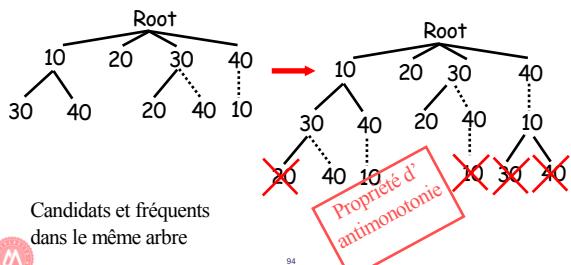
89





PSP (cont.)

- #### □ Génération des candidats de taille > 2



SPAM

- Utilisation de bitmaps pour rechercher les motifs fréquents
 - Hypothèse : la base tient toujours en mémoire
 - On construit d'un arbre lexicographique contenant toutes les branches possibles – élimination des branches en fonction du support
 - Nouvelle représentation des données

05

SPAM (cont.)

- Représentation verticale des données

$$C1 = \langle (1)_3 (1)_5 \rangle$$

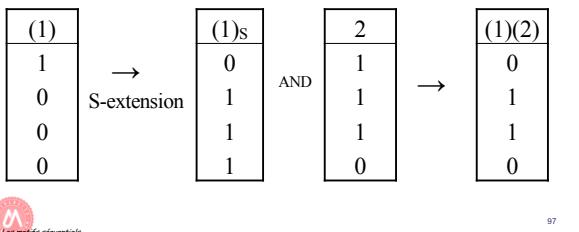
- S-Extension
 - I-Extension

		(1)
C1	T1	0
	T2	0
	T3	1
	T4	0
	T5	1

96

SPAM (cont.)

- S-Extension : un bitmap transformé + AND
- I-Extension : AND
- Exemple : recherche du candidat (1) (2)



Les motifs séquentiels

L'algorithme SPADE

- SPADE (*Sequential PAttern Discovery using Equivalent Class*) - M. Zaki (Machine Learning 01)
- Représentation verticale des données
- Une base de séquence est transformée en :
 - Item : <SID, Eid>
- La recherche des motifs est réalisée en étendant les sous séquences un item à la fois via la génération des candidats d'Apriori

Les motifs séquentiels

98

L'algorithme SPADE

The diagram shows the transformation of a sequence database into a transactional format for SPADE processing.

Sid	Sequence
1	(A) (A B C) (A C) (D) (C F)
2	(A D) (C) (B C) (A E)
3	(E F) (A B) (D F) (C) (B)
4	(E) (G) (A F) (C) (B) (C)

→

SID	Eid	Items
1	1	A
1	2	A B C
1	3	A C
1	4	D
1	5	C F
2	1	A D
2	2	C
2	3	B C
2	4	A E
3	1	E F
3	2	A B
3	3	D F
3	4	C
3	5	B
....	...	99

99

Les motifs séquentiels

L'algorithme SPADE

SID	Eid	Items
1	1	A
1	2	A B C
1	3	A C
1	4	D
1	5	C F
2	1	A D
2	2	C
2	3	B C
2	4	A E
3	1	E F
3	2	A B
3	3	D F
3	4	C
3	5	B
...

A	
Sid	eid
1	1
1	2
1	3
2	1
2	4
3	2
4	3

B	
Sid	eid
1	2
1	3
2	1
2	4
3	2
3	5
4	5

(A)(B)	
Sid	eid
1	2
2	3
3	5
4	5

Listes d'occurrences

100

L'algorithme SPADE

A	
Sid	eid
1	1
1	2
1	3
2	1
2	4
3	2
4	3

B	
Sid	eid
1	2
2	3
3	2
3	5
4	5

(A)(B)	
Sid	eid
1	2
2	3
3	5
4	5

Jointure temporelle

Seuls les couples <(1,1), (1,2)>, <(2,1) (2,3)>, <(3,2), (3,5)> et <(4,3), (4,5)> donnent lieu à de nouvelles occurrences du motifs (A)(B)

101

L'algorithme SPADE

A	
Sid	eid
1	1
1	2
1	3
2	1
2	4
3	2
4	3

B	
Sid	eid
1	2
2	3
3	2
3	5
4	5

(AB)	
Sid	eid
1	2
3	2

Jointure équivalente

Seuls les couples <(1,2), (1,2)> et <(3,2), (3,2)> donnent lieu à de nouvelles occurrences du motifs (A B)

102

Motifs généralisés

- Pour certains domaines d'applications il est nécessaire de limiter les résultats

corrélations entre achat du caviar le 1er janvier et de champagne le 31 décembre ?

- Contraintes de temps

windowSize : regrouper des événements

minGap : considérer des événements comme trop proches

minGap : considérer des événements comme trop proches
maxGap : considérer des événements comme trop éloignés



Illustration

Client	Date	Items
C1	1	Ringworld
C1	2	Foundation
C1	15	Ringworld Engineers, Second Foundation
C2	1	Foundation, Ringworld
C2	20	Foundation and Empire
C2	50	Ringworld Engineers

Support = 50% : <(Ringworld) (Ringworld Engineers)> et
<(Foundation) (Ringworld Engineers)>

windowSize=7 jours : <(Foundation, Ringworld) (Ringworld Engineers)>

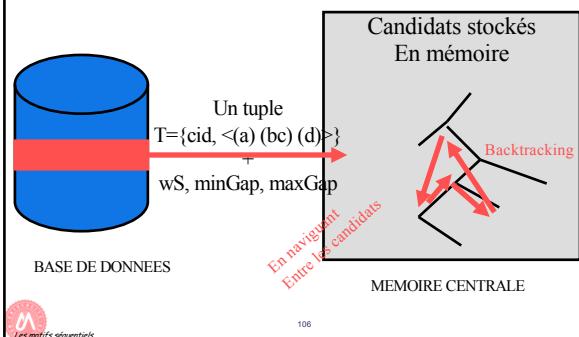


Contraintes temporelles

- $d = \langle 1^2 2^3 3^4 5^5 7^1 \rangle$
 - Candidat : $C = \langle 1 2 3 4 5 6 7 \rangle$
 - $\text{windowSize} = 3$, $\text{minGap}=0$, $\text{maxGap} = 5$,
 - $d = \langle 1 2 3 4 5 6 7 \rangle$ - Donc C est inclus dans d
 - Candidat : $C = \langle 1 2 3 6 7 \rangle$
 - $\text{windowSize} = 1$, $\text{minGap}=3$, $\text{maxGap} = 4$,
 - $d = \langle 1 2 3 4 5 6 7 \rangle$
 - minGap pas respecté entre 3 et 5 ! C pas inclus dans d



Comment gérer les contraintes ?



Inclusion des contraintes

Client	Date	Items
C1	1	10
C1	7	20
C1	13	30
C1	17	40
C1	18	50
C1	24	60

minGap=1 windowSize=5

<(10) (20) (30) (50) (60)>
 <(10) (20) (30) (40) (60)>
 <(10) (20) (30) (40 50) (60)>
 <(10) (20) (30 40) (60)>
 <(10) (20) (30 40 50) (60)>

107

Recherche des inclusions

Date	1	7	13	17	18	24
C	1	2	3	4	5	6

windowSize = 5, minGap = 1

Via minGap

- <(1) (2) (3) (4) (6)>
- <(1) (2) (3) (5) (6)>

Puis avec windowSize

- <(1) (2) (3) (4 5) (6)>
- <(1) (2) (3 4) (6)>
- <(1) (2) (3 4 5) (6)>

En fait:

- <(1) (2) (3) (4 5) (6)>
- et <(1) (2) (3 4 5) (6)>

108

Recherche des inclusions (cont.)

Date	1	7	13	17	18	24
C	1	2	3	4	5	6

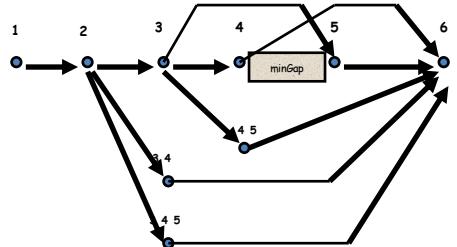


3 4
3 4 5

109



Recherche des inclusions (cont.)

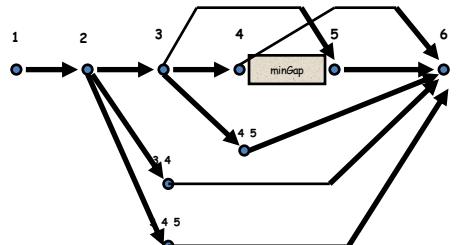


Un chemin = une séquence
Tous les chemins mais quid des inclusions

110

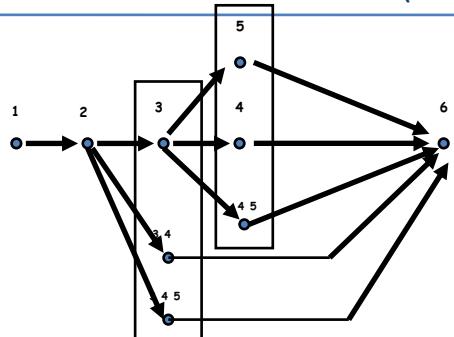


Recherche des inclusions (cont.)



111

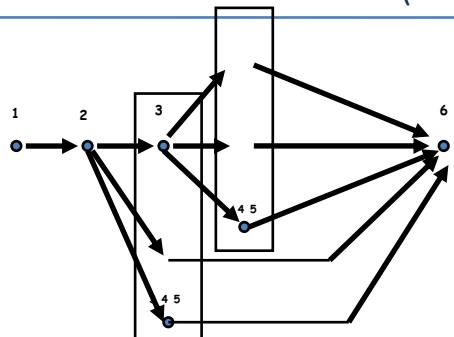
Recherche des inclusions (cont.)



Les motifs séquentiels

112

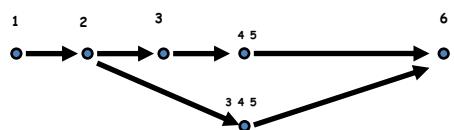
Recherche des inclusions (cont.)



Les motifs séquentiels

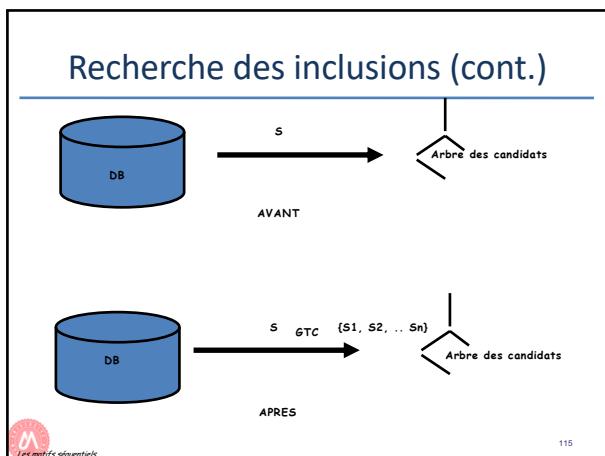
113

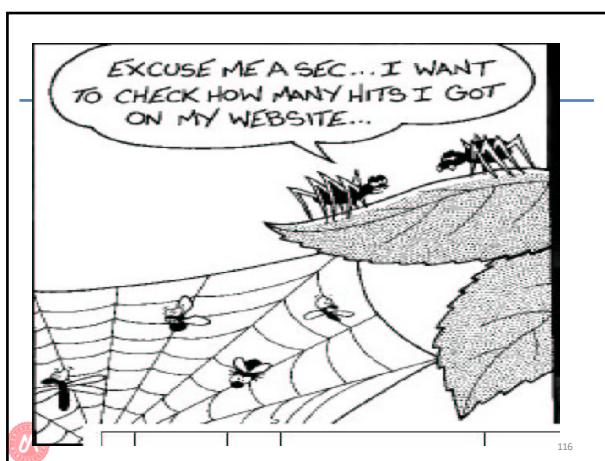
Recherche des inclusions (cont.)

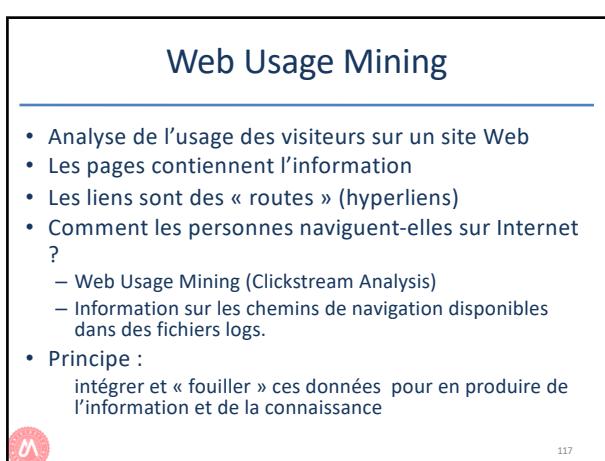


$\leftarrow(1) (2) (3) (4 \ 5) (6) \rightleftharpoons$ et $\leftarrow(1) (2) (3 \ 4 \ 5) (6) \rightleftharpoons$

114







Web Usage Mining

- Pourquoi analyse l'usage des sites Web ?
- La connaissance sur la manière dont les visiteurs utilisent un site Web permet de :
 - Fournir une aide pour réorganiser site
 - Aider le concepteur à positionner l'information importante que les visiteurs recherchent.
 - Précharger et cacher les pages
 - Fournir des sites adaptatifs (personnalisation)
 - Eviter le « zapping »
- Utile dans le cas du e-commerce



118

Exemple d'utilisation

Statistiques générales	Performance du site	Retenir les clients
Analyse du contenu	Groupement des clients	Campagne adaptée
Point d'entrée	Ciblages des clients	Campagne ciblée
Parcours	Comportement des clients	Modification dynamique



119

Web Usage Mining

- De nombreux outils disponibles
- Statistiques générales :
 - Nombre de hits
 - Quelle est la page la plus populaire du site ?
 - Qui a visité le site ?
 - Qu'est ce qui a été téléchargé ?
 - Quels sont les mots clés utilisés pour venir sur le site ?



120

Web Usage Mining

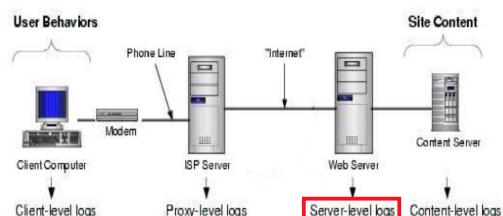


« 75% des parisiens qui achètent une raquette de tennis achètent trois mois après des chaussures »
Modification dynamique

121

Log or Logs?

Information sur les chemins de navigation dans les fichiers logs



122

Web logs

```

IP or domain name    User Id      Date and Time      Request
123.456.78.9 - [24/Oct/1999:19:13:44 -0400] "GET /Images/tagline.gif HTTP/1.0"
200 1449 http://www.teced.com/ "Mozilla/4.51 [en] (Win98; i)"

Status               File Size   Referrer URL     Browser
/                   1449       /                Mozilla/4.51 [en] (Win98; i)

Cookies
  
```

123

Web logs						
IP Address	Time	Method/URL/Protocol	Status	Size	Referer	Agent
123.456.78.9	[25/Apr/1998:03:04:41-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:05:34-0500	GET B.html HTTP/1.0	200	2050	A.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:05:39-0500	GET L.html HTTP/1.0	200	4130	-	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:06:02-0500	GET F.html HTTP/1.0	200	5096	B.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:06:58-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:07:42-0500	GET B.html HTTP/1.0	200	2050	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:07:55-0500	GET R.html HTTP/1.0	200	8140	L.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:09:50-0500	GET C.html HTTP/1.0	200	1820	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:10:02-0500	GET O.html HTTP/1.0	200	2270	F.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:10:45-0500	GET J.html HTTP/1.0	200	9430	C.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:12:23-0500	GET G.html HTTP/1.0	200	7220	B.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:05:05:22-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (Win95, I)



124

Web logs						
IP Address	Time	Method/URL/Protocol	Sta	Size	Referer	Agent
123.456.78.9	[25/Apr/1998:03:04:41-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:05:34-0500	GET B.html HTTP/1.0	200	2050	A.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:05:39-0500	GET L.html HTTP/1.0	200	4130	-	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:06:02-0500	GET F.html HTTP/1.0	200	5096	B.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:06:58-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (X11_1, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:07:42-0500	GET B.html HTTP/1.0	200	2050	A.html	Mozilla/3.01 (X11_1, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:07:55-0500	GET R.html HTTP/1.0	200	8140	L.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:09:50-0500	GET C.html HTTP/1.0	200	1820	A.html	Mozilla/3.01 (X11_1, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:10:02-0500	GET O.html HTTP/1.0	200	2270	F.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:10:45-0500	GET J.html HTTP/1.0	200	9430	C.html	Mozilla/3.01 (X11_1, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:12:23-0500	GET G.html HTTP/1.0	200	7220	B.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:05:05:22-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (Win95, I)



125

↑ Items

125

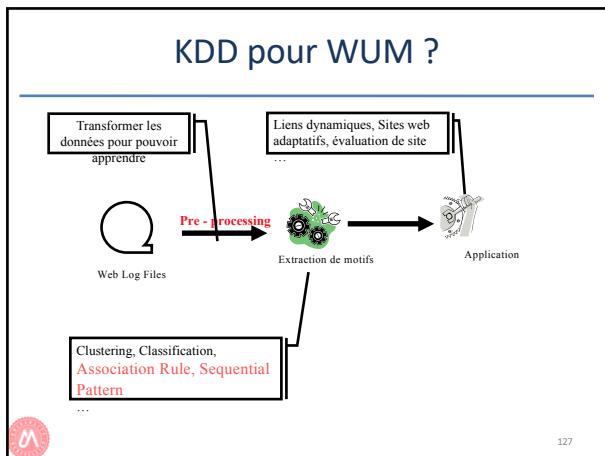
Web logs						
IP Address	Time	Method/URL/Protocol	Sta	Size	Referer	Agent
123.456.78.9	[25/Apr/1998:03:04:41-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:05:44-0500	GET B.html HTTP/1.0	200	2050	A.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:05:39-0500	GET L.html HTTP/1.0	200	4130	-	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:06:02-0500	GET F.html HTTP/1.0	200	5096	B.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:06:58-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:07:42-0500	GET B.html HTTP/1.0	200	2050	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:07:55-0500	GET R.html HTTP/1.0	200	8140	L.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:09:50-0500	GET C.html HTTP/1.0	200	1820	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:10:02-0500	GET O.html HTTP/1.0	200	2270	F.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:03:10:45-0500	GET J.html HTTP/1.0	200	9430	C.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
123.456.78.9	[25/Apr/1998:03:12:23-0500	GET G.html HTTP/1.0	200	7220	B.html	Mozilla/3.01 (Win95, I)
123.456.78.9	[25/Apr/1998:05:02:25-0500	GET A.html HTTP/1.0	200	3290	-	Mozilla/3.01 (Win95, I)



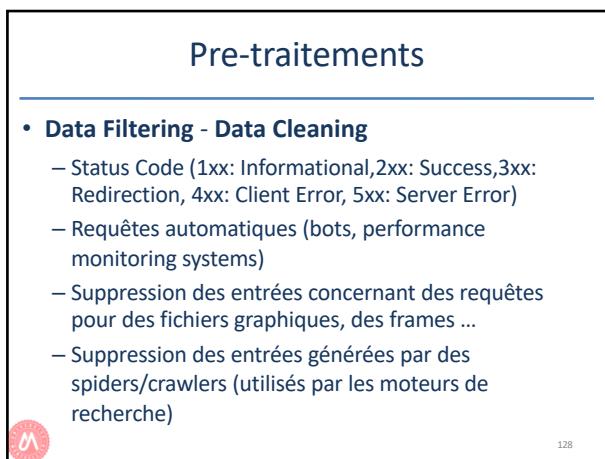
126

Items

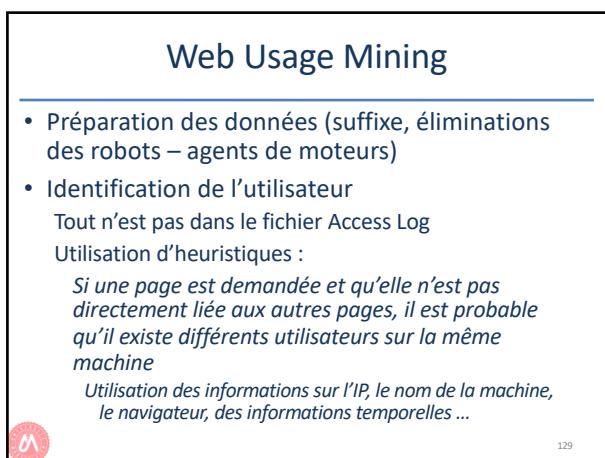
126



127



128



129

Web Usage Mining

- Problèmes :
 - ID utilisateurs supprimées pour des raisons de sécurité
 - IP individuelles cachées par les proxy
 - Les caches des proxy et du côté clients
- Solutions actuelles :
 - Enregistrement de l'utilisateur – pratique ??
 - Cookies – difficile ??
 - « Cache busting » - augmente le trafic sur le réseau (inutile avec certains proxy)

130

Web Usage Mining

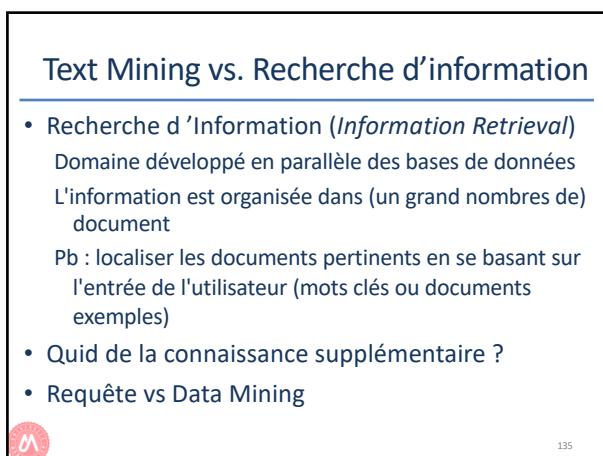
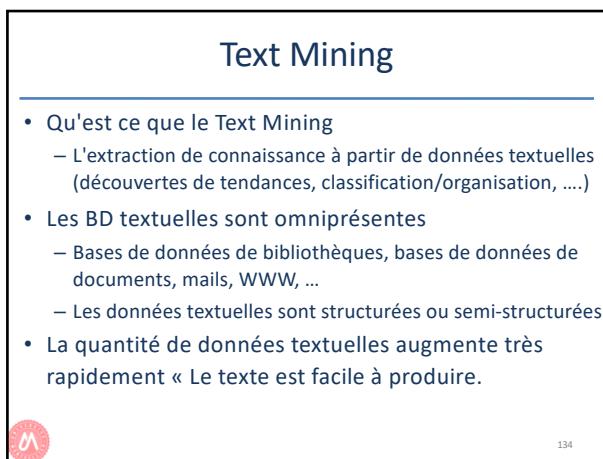
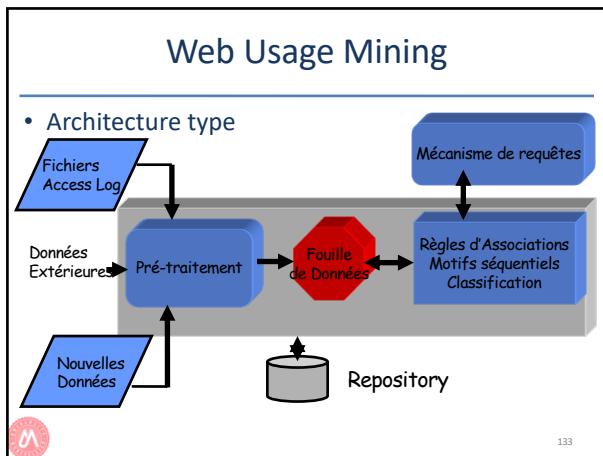
- Sessions : Comment identifier/définir une transaction d'un visiteur ?
- « Time Oriented »
 - Durée totale d'une session : ≤ 30 minutes
 - Par temps passé sur une page : ≤ 10 minutes/page
- « Navigation Oriented »
 - Le « referrer » est la page précédente, ou le « referrer » n'est pas défini mais demandé dans les 10 secondes, ou le lien de la page précédente à la page courante dans le site web

131

Web Usage Mining

- Sources de données
 - Utilisation de fichiers logs
 - Mais aussi cookies, bases de données des clients,

132



Text Mining - Classification

- Classification automatique
 - Classification automatique d'un grand nombre de documents (pages Web, mails, fichiers textuels) basée sur un échantillon de documents pré-classifié
 - Mise en oeuvre
 - *Echantillon* : des experts génèrent l'échantillon
 - *Classification* : l'ordinateur découvre les règles de classification
 - *Application* : les règles découvertes peuvent être utilisées pour classer des nouveaux documents et les affecter à la bonne classe



136



Text Mining - Classification

- Quelques problèmes
 - Synonymie : un mot T peut ne pas apparaître dans un document mais si le document est très lié à T (data mining / software product)
 - Polysémie : le même mot peut avoir plusieurs sens (mining)
 - Représentation des documents (vecteurs de termes, choix des termes représentatifs, calcul de la distance entre un vecteur représentant le groupe de documents et celui du nouveau document, ...)
 - Evolution des classes dans le temps



137



Text Mining - Corrélations

- Analyse d'associations basée sur des mots clés
 - Rechercher des associations/correlations parmi des mots clés ou des phrases
 - Mise en œuvre
 - *Pré-traitement des données* : parser, supprimer les mots inutiles (le, la, ...) => prise en compte d'une analyse morpho-syntaxique (e.g. lemmatiseur)
 - Un document est représenté par : (document_id, {ensemble de mots clés})
 - Appliquer des algorithmes de recherche de règles d'association



138



Text Mining - Corrélations

- Quelques problèmes
 - Ceux du traitement de la langue naturelle
 - Les mots inutiles (ordinateur ? Utile ?) – Réduction de l'espace de recherche
 - Les associations de mots, phrase, paragraphe, ...



139

- ---

Text Mining – Analyse de tendances

- Rechercher des tendances dans les documents
 - Mise en œuvre
 - Pré-traitement : attention l'ordre est important
 - Document représenté par : (document_id, <phrases simplifiées : ensemble de mots ordonnés>)
 - Appliquer des algorithmes de motifs séquentiels
 - Générer l'historique des phrases
 - Recherche les phrases qui correspondent à des tendances



140

- ---

Text Mining – Analyse de tendances

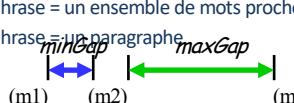
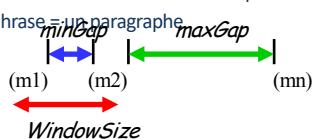
- Principes
 - Un mot : (m)
 - Une phrase : <(m1) (m2) (m3) ... (mn)>
 - Paramètres : WindowSize, MaxGap, MinGap)
Une phrase = une phrase
 - Une phrase = un ensemble de mots proches
 - Une phrase = un ensemble de mots proches

Diagram illustrating the parameters for a phrase. It shows three tokens: (m1), (m2), and (mn). A blue double-headed arrow between (m1) and (m2) is labeled 'minGap'. A green double-headed arrow between (m1) and (mn) is labeled 'maxGap'. A red double-headed arrow below (m1) and (m2) is labeled 'WindowSize'.



141

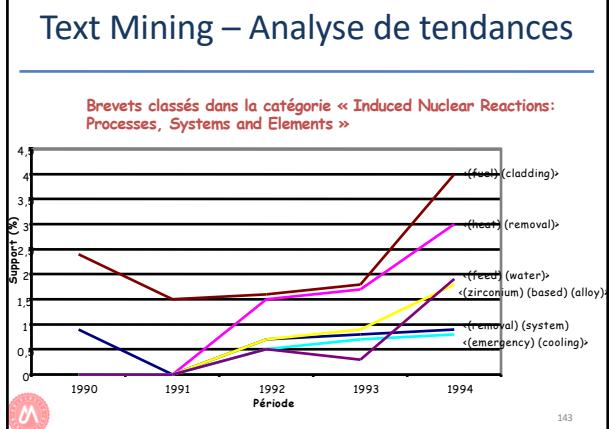
- ---

Text Mining – Analyse de tendances

- Gérer l'historique des phrases
 - Partitionner les documents en fonction de leur estampille (ex : année pour les brevets, mois pour des documents sur le Web)
 - Pour chaque partition, génération des ensembles fréquents de phrases
 - Maintenir l'historique des supports pour chaque phrase
 - Interroger l'historique des phrases pour connaître les tendances (tendance récente à monter, transition récente, résurgence d'usage,)



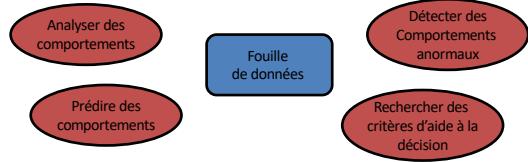
142



143

Fouille de données de santé

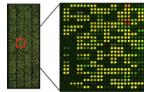
- **Données particulières:** hétérogènes, souvent imprécises, subjectives, non déterministes, bruitées, avec des valeurs manquantes et des erreurs



144

Puces à ADN

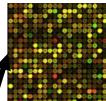
- Incontournables pour comprendre les maladies génétiques complexes :** perturbation des processus naturels de croissance, de division et de mort des cellules
- Utilisées par les biologistes** pour acquérir de grandes quantités de données sur l'expression des gènes et identifier les lois suivies par ces expressions en fonction des maladies et des traitements :
 - gènes impliqués dans la maladie ?
 - gènes dont les expressions sont corrélées ?
 - gènes qui inhibent ou activent une fonction ?
 - ...
- Difficultés pour extraire automatiquement** des connaissances liées aux gros volumes de données



145

Puces à ADN

- Le principe :** propriété de l'ADN dénaturé de reformer spontanément sa double hélice lorsqu'il est porté face à un brin complémentaire (réaction d'hybridation).
- A ≡ T
- T ≡ A
- G ≡ C
- C ≡ G
- Concrètement...** un ensemble de molécules d'ADN fixées en rangées ordonnées sur une petite surface



Expression (couleur) ≈ mesure de la quantité d'ADN dénaturé qui se reforme

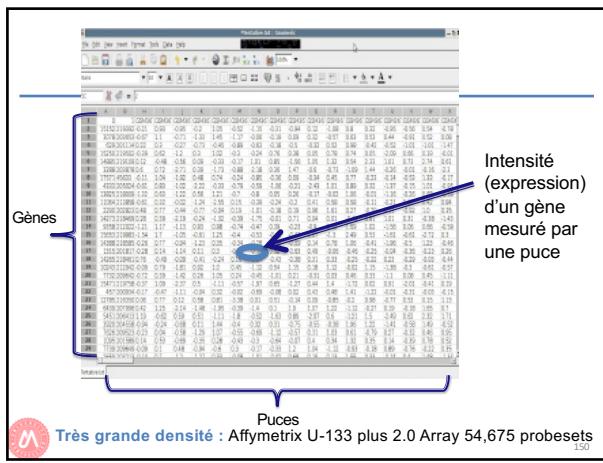
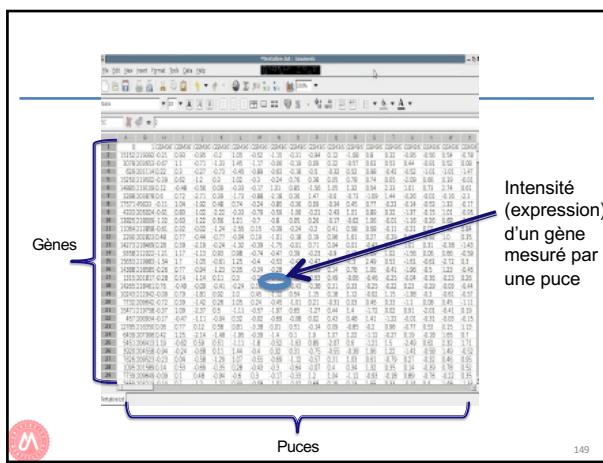
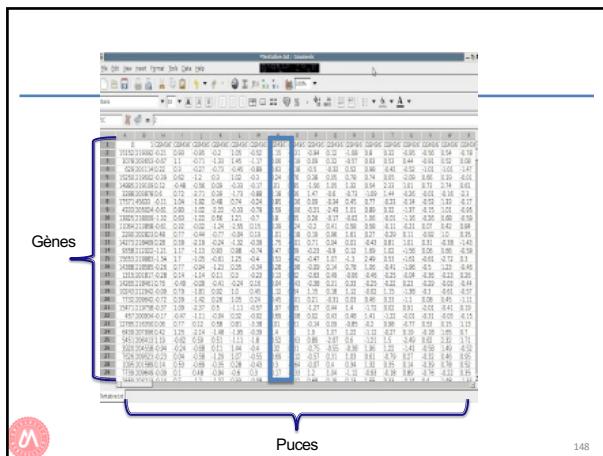


146

Gènes

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
Gènes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

147



Les motifs séquentiels dans ce contexte...

- **Motifs séquentiels** : séquences fréquentes d'itemsets ordonnés
< ( ) () >
 - Rechercher des motifs séquentiels pour mettre en évidence des gènes dont les expressions sont fréquemment ordonnées de la même manière
< (G5 G4) (G6) >
 - **2 exemples avec**
 - MMDN sur la maladie d'Alzheimer
 - IRCM sur le cancer du sein



151

Maladie d'Alzheimer : problème majeur de la société moderne

- **Maladie d'Alzheimer (AD)** : la forme la plus commune de démence
 - 26,6 millions de personnes atteintes (2006)
 - Augmentation du nombre de patients (*4 en 2050)
 - Intérêt de la communauté biomédicale pour la découverte génique impliquée dans le développement de la maladie
 - **MMDN** : travaillent sur l'AD et sur le vieillissement à partir d'un modèle animal, *Microcebus murinus*
 - Objectifs : comparer les tissus du cortex cérébral de lémuriens jeunes (sains) avec ceux de lémuriens âgés (malades) pour étudier le vieillissement (la maladie d'Alzheimer).



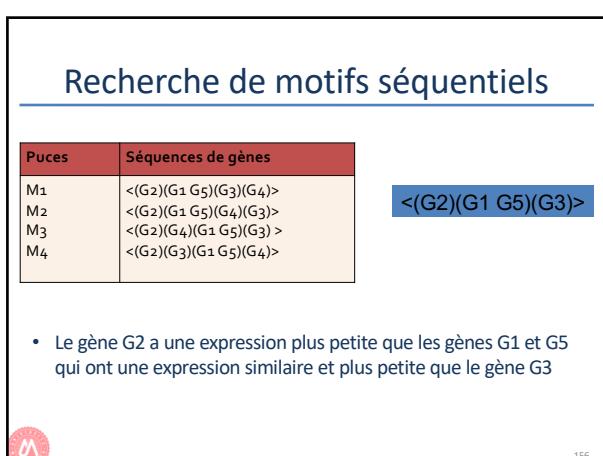
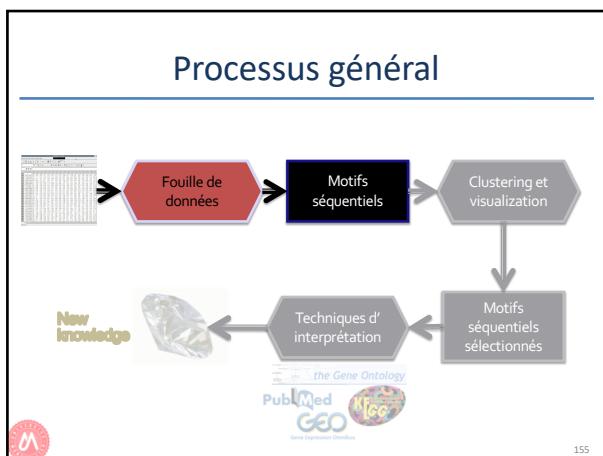
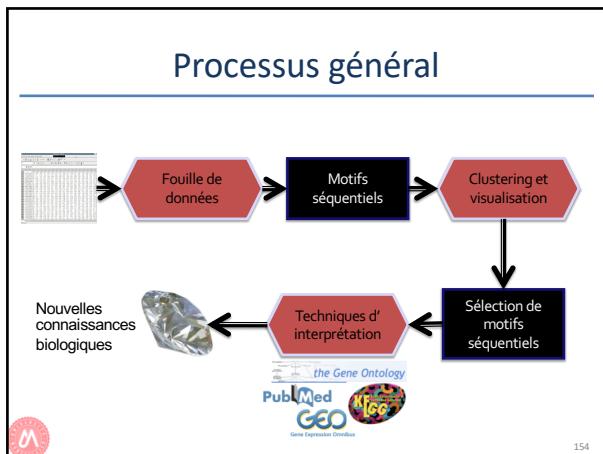
152

Cancer du sein : première cause de mortalité entre 45 et 64 ans (2004)

- **Perturbation de la communication cellulaire**, associée à une **absence de mort cellulaire**, engendrant le **développement d'amas de cellules cancéreuses** (appelées tumeurs) qui échappent aux règles de fonctionnement du corps.
 - **IRCM** : utilisent les puces ADN pour comparer les tissus issus de tumeurs du sein, répertoriés selon différents grades.
 - **Objectif** : déterminer un ensemble de biomarqueurs suffisants pour **typer ces tumeurs**.
 - **Enjeu considérable** : Les thérapies sont + ou - toxiques et fonctionnent sur un patient mais pas sur un autre. Typer une tumeur s'avère crucial pour le choix d'une thérapie.



153



Recherche de motifs séquentiels

Puces	Séquences de gènes
M1	$\langle(G_2)(G_1 G_5)(G_3)(G_4)\rangle$
M2	$\langle(G_2)(G_1 G_5)(G_4)(G_3)\rangle$
M3	$\langle(G_2)(G_4)(G_1 G_5)(G_3)\rangle$
M4	$\langle(G_2)(G_3)(G_1 G_5)(G_4)\rangle$

$\langle(G_2)(G_1 G_5)(G_3)\rangle$
Support = 3/4

 157

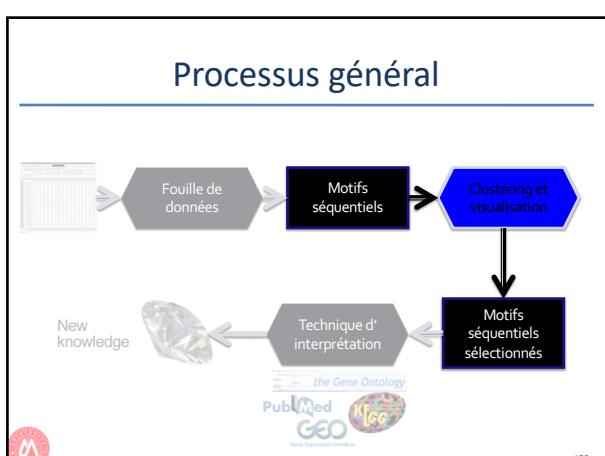
Recherche de motifs séquentiels

Puces	Séquences de gènes
M1	$\langle(G_2)(G_1 G_5)(G_3)(G_4)\rangle$
M2	$\langle(G_2)(G_1 G_5)(G_4)(G_3)\rangle$
M3	$\langle(G_2)(G_4)(G_1 G_5)(G_3)\rangle$
M4	$\langle(G_2)(G_3)(G_1 G_5)(G_4)\rangle$

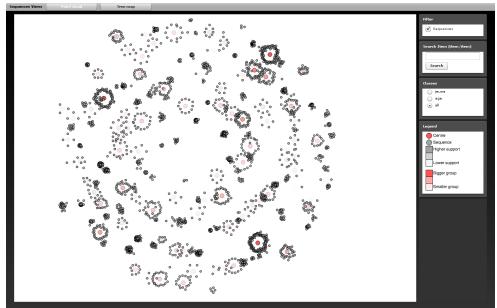
$\langle(G_2)(G_1 G_5)(G_3)\rangle$
Support = 3/4

- Motifs séquentiels discriminants
 - Fréquents dans une classe (malades)
 - Non fréquents dans la classe complémentaire (sains)

 158

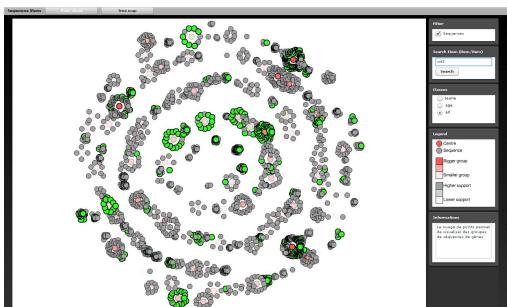


Clustering simple (k-means)



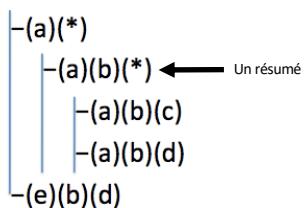
160

Clustering simple (k-means)



Clustering hiérarchique

- Exemple: (a)(b)(c), (a)(b)(d), (e)(b)(d)



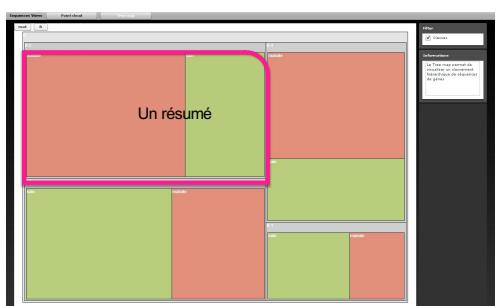
162

Clustering hiérarchique



163

Clustering hiérarchique

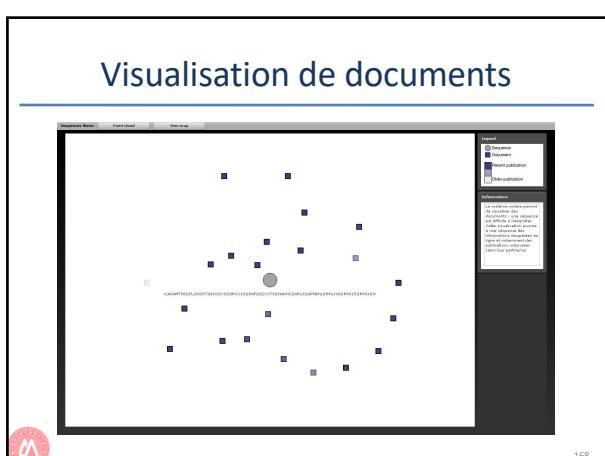
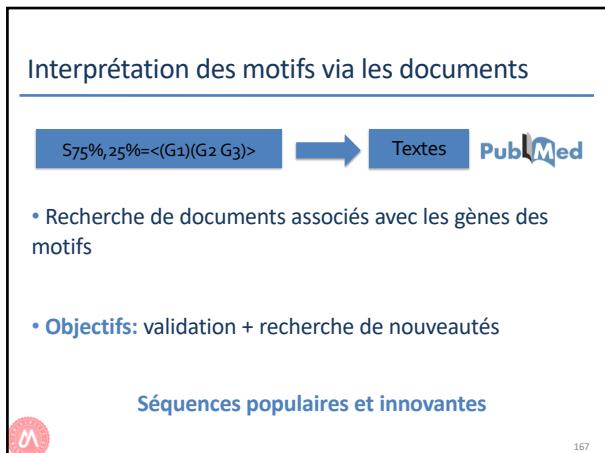
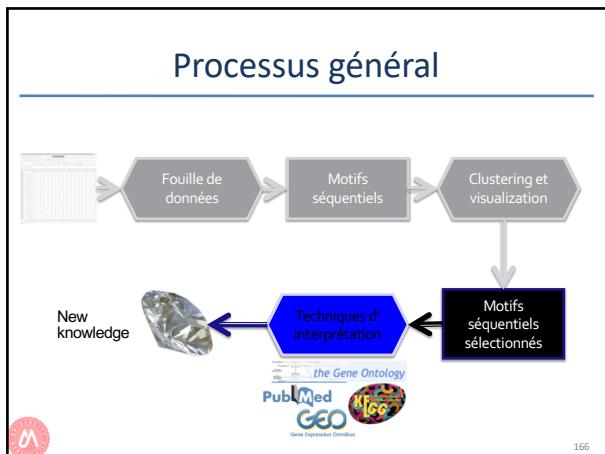


164

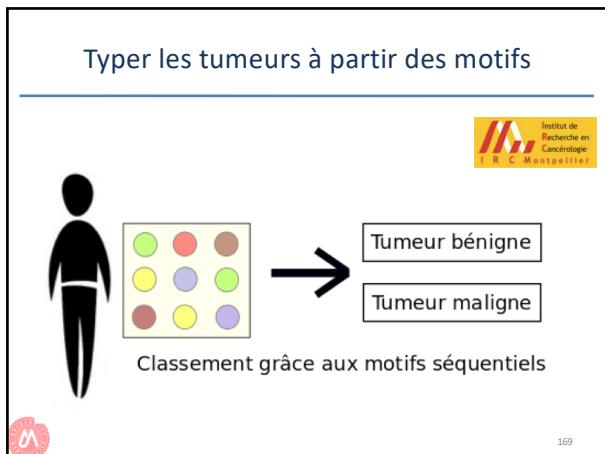
Clustering hiérarchique



165



Typer les tumeurs à partir des motifs



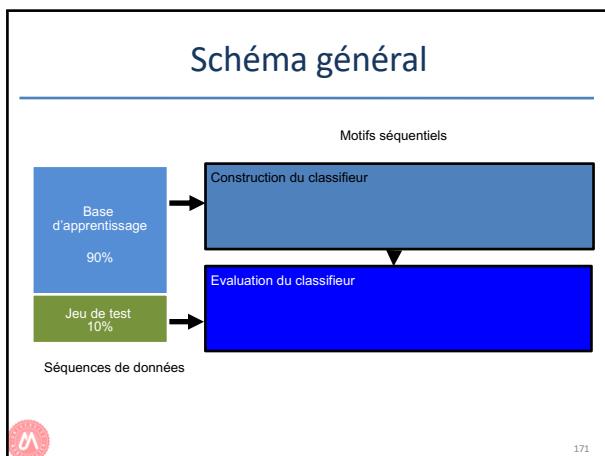
Classement selon 3 classes

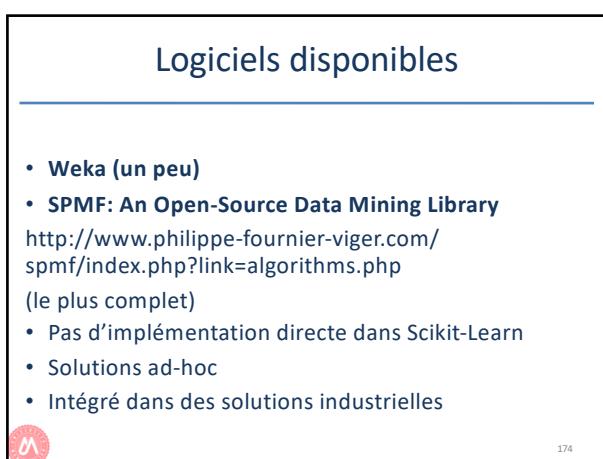
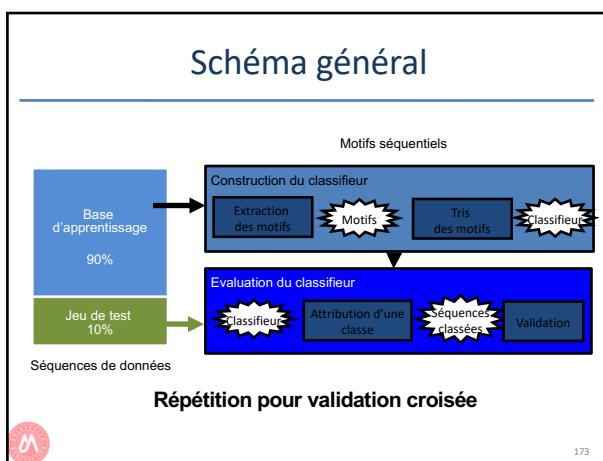
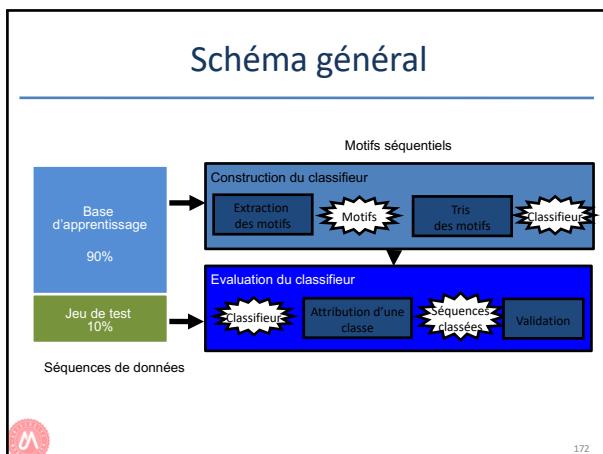
- Classe 1 : faible
 - Classe 2 : moyen
 - Classe 3 : forte

**0.96% de rappel et 0.97% de précision
selon le jeu de données**



Schéma général





Voir notebook

- ExtractSequentialPattern.ipynb
- Utilisation de AprioriAll, PrefixSpan (Python-Java)

175

Conclusions

- Les motifs séquentiels sont une petite partie des patterns à extraire ...
 - Arbres, graphes, multigraphes ...
 - De nombreuses approches existent
- Ce qu'il faut retenir : les patterns sont différents, les usages sont différents mais les contraintes existent aussi quelques soient les types de patterns

176

- Des questions ?

177
