

Extraction de motifs :

Règles d'association et motifs séquentiels

Maguelonne Teisseire

TETIS – Irstea

teisseire@teledetection.fr

<http://www.lirmm.fr/~teisseire>



Plan

- Contexte général
- Règles d'association
- Motifs séquentiels
- Conclusions



Pourquoi fouiller les données ?

- De nombreuses données sont collectées et entreposées
 - Données du Web, e-commerce
 - Achats dans les supermarchés
 - Transactions de cartes bancaires
- Les ordinateurs deviennent de moins en moins chers et de plus en plus puissants
- La pression de la compétition est de plus en plus forte
 - Fournir de meilleurs services, s'adapter aux clients (e.g. dans les CRM)

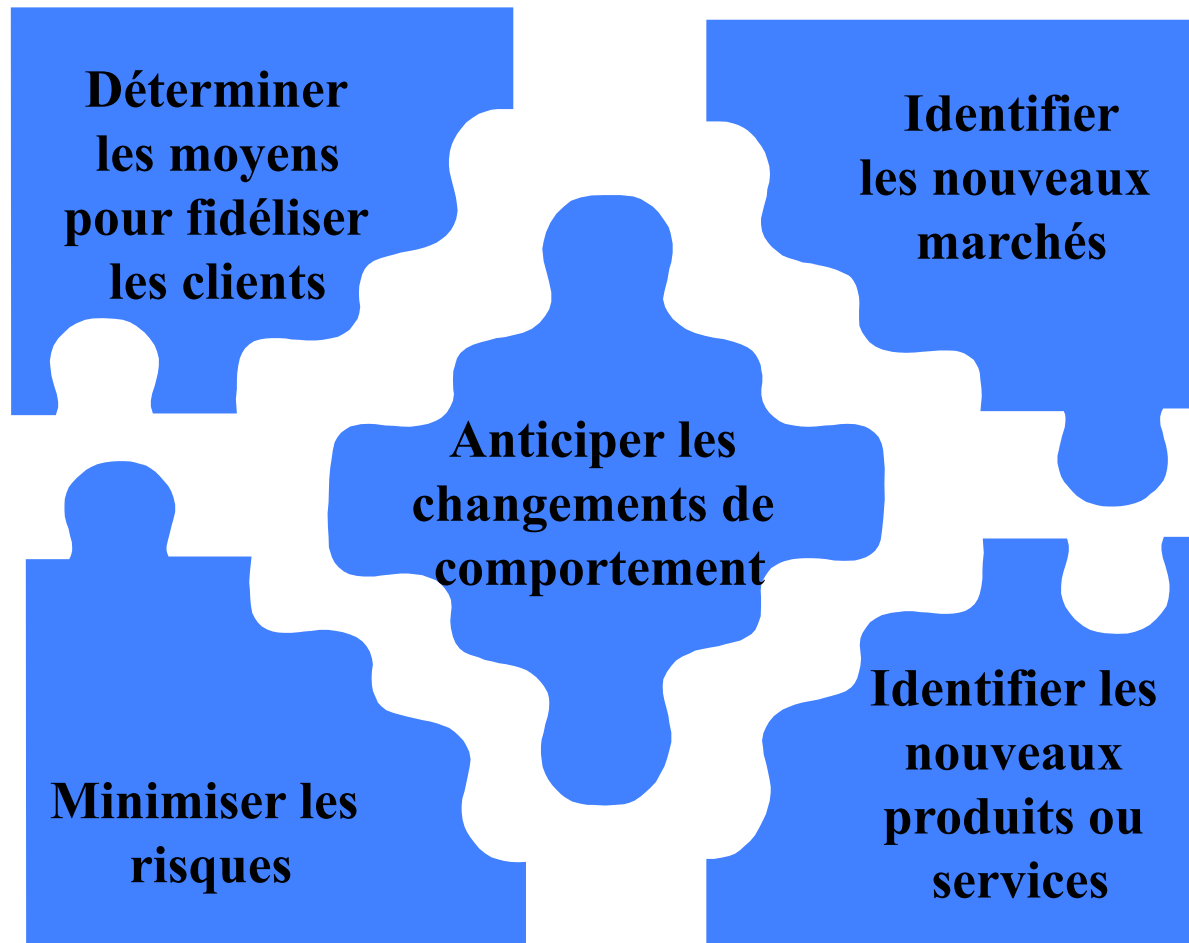


Pourquoi fouiller les données ?

- ❑ Les données sont collectées et stockées rapidement (GB/heures)
 - Capteurs : RFID, supervision de procédé
 - Télescopes
 - Puces à ADN générant des expressions de gènes
 - Simulations générant de téraoctets de données

- ❑ Les techniques traditionnelles ne sont pas adaptées

Un enjeu stratégique



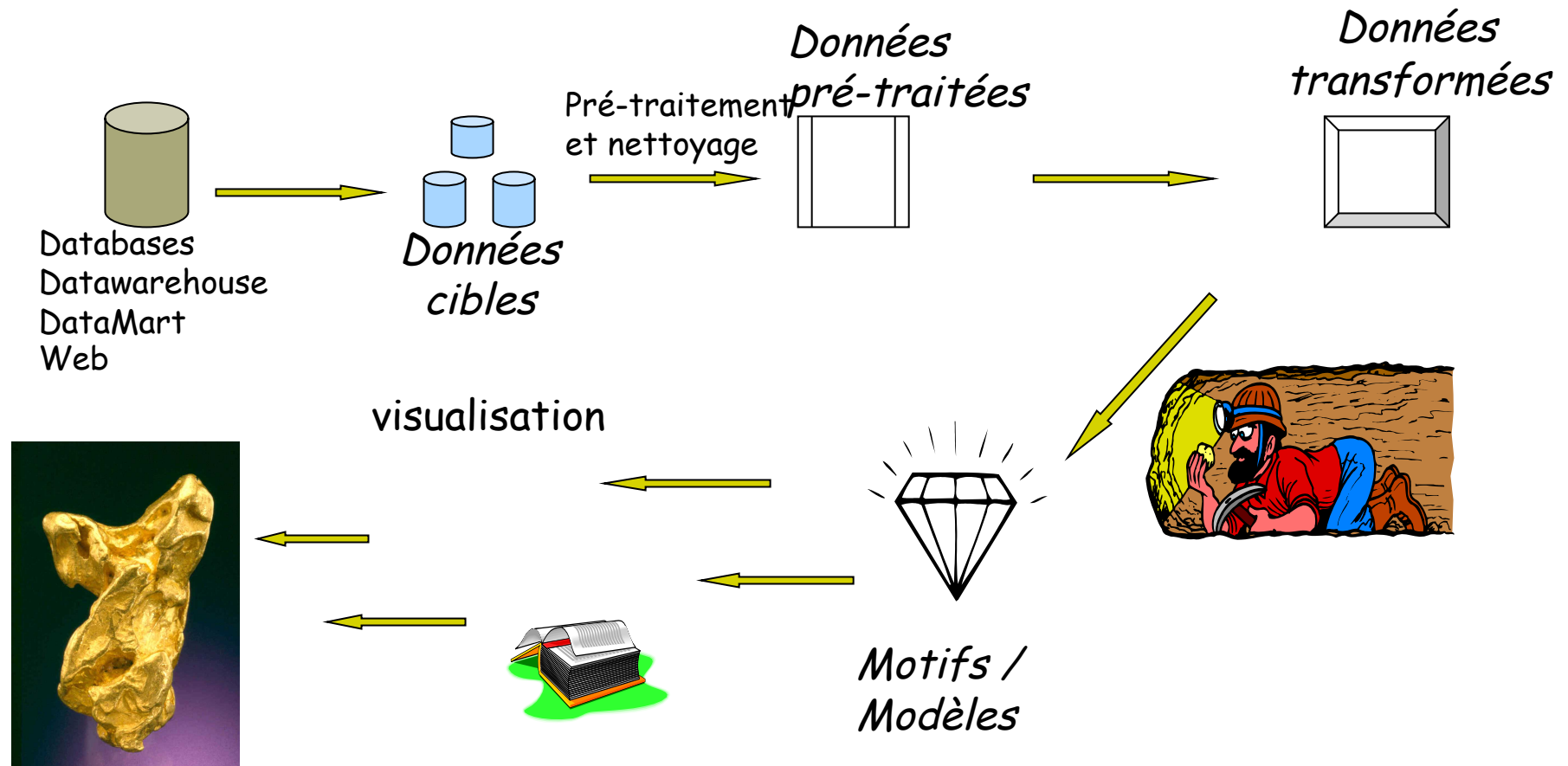


Qu'est ce que le Data Mining ?

□ De nombreuses définitions

- Processus **non trivial** d'extraction de connaissances d'une base de données pour obtenir de nouvelles données, valides, potentiellement utiles, compréhensibles,
- Exploration et analyse, par des moyens automatiques ou semi-automatiques, de large quantité de données en vue d'extraire des motifs intéressants

Le processus de KDD



Données, Informations, Connaissances

Décision

- Promouvoir le produit P dans la région R durant la période N
- Réaliser un mailing sur le produit P aux familles de profil F

Connaissance (data mining)

- Une quantité Q du produit P est vendue en région R
- Les familles de profil F utilisent M% de P durant la période N

Information (requêtes)

- X habite la région R
- Y a A ans
- Z dépense son argent dans la ville V de la région R

Données

- Consommateurs
- Magasins
- Ventes
- Démographie
- Géographie

Contexte général



Data Mining ou non ?

● NON

Rechercher le
salaire d'un
employé

Interroger un
moteur de
recherche Web
pour avoir des
informations sur le
Data Mining

● OUI

Les supporters achètent de la
bière le samedi et de l'aspirine
le dimanche

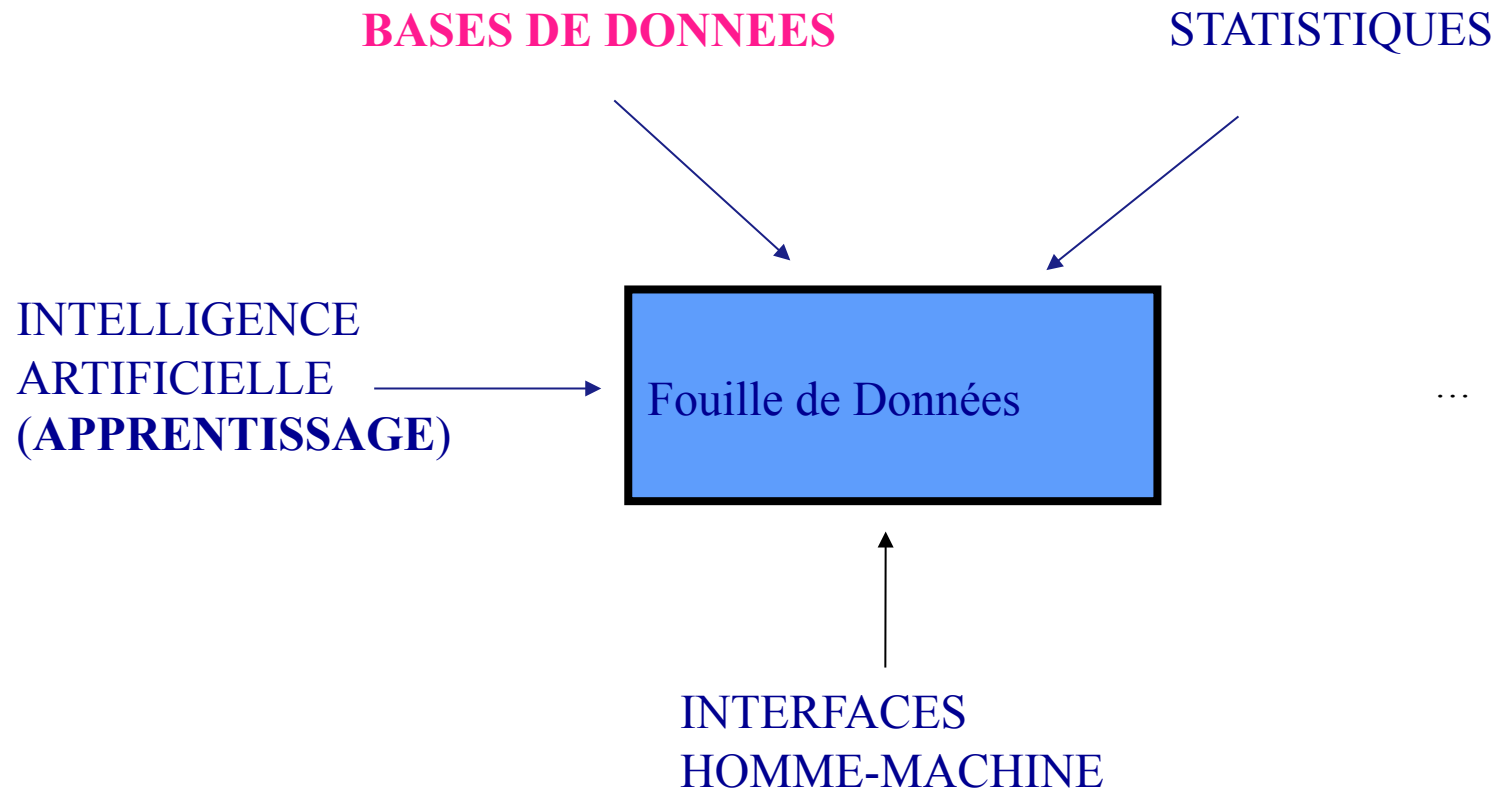
Regrouper ensemble des
documents retournés par un
moteur de recherche en
fonction de leur contenu



Applications

- ❑ Médecine : bio-médecine, drogue, Sida, séquence génétique, gestion hôpitaux, ...
- ❑ Finance, assurance : crédit, prédiction du marché, détection de fraudes, ...
- ❑ Social : données démographiques, votes, résultats des élections,
- ❑ Marketing et ventes : comportement des utilisateurs, prédiction des ventes, espionnage industriel, ...
- ❑ Militaire : fusion de données .. (secret défense)
- ❑ Astrophysique : astronomie, « contact » (;-))
- ❑ Informatique : agents, règles actives, IHM, réseau, Data-Warehouse, Data Mart, Internet (moteurs intelligent, profiling, text mining, ...)

Fouille de données



Recherche de motifs fréquents

- Qu'est ce qu'un motif fréquent ?
 - Un motif (ensemble d'items, séquences, arbres, ...) qui interviennent 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?



Recherche de motifs fréquents

□ Analyse des associations

- Panier de la ménagère, cross marketing, conception de catalogue, analyse de textes
- Corrélacion 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



« 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

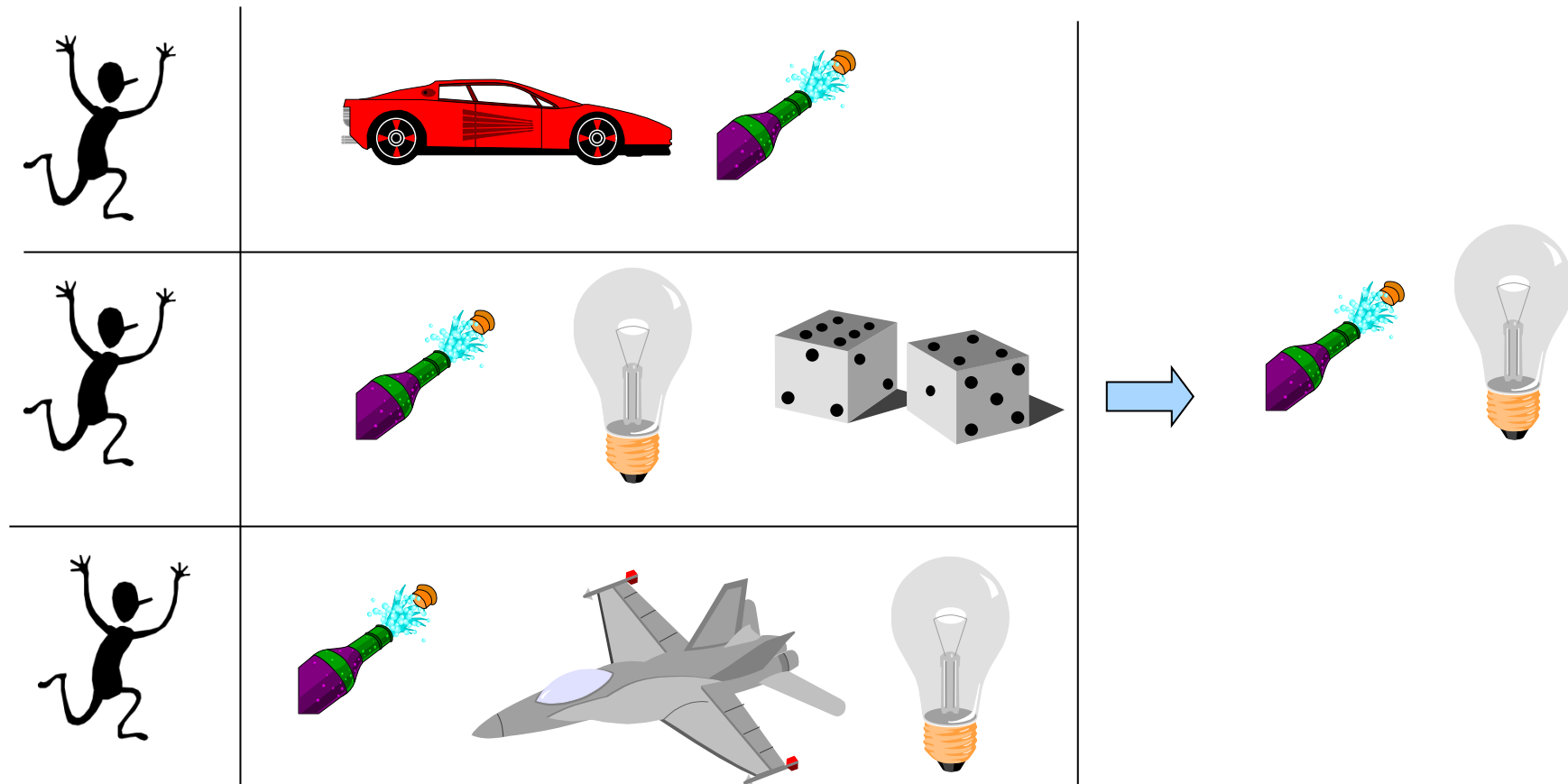
- recherche de corrélations entre attributs (items) mais en prenant en compte le temps entre items => comportement



Plan

- Contexte général
- Règles d'association
- Motifs séquentiels
- Conclusions

Recherche de règles d'association



Panier de la ménagère

Localisation →

Produits achetés →

Identification →

Date, heure →

STOW STAR
(978) 897-5140

TERA SALT & VING	1.99 F
TERRA SLT&PEPR C	1.99 F
HD PREM 1% CHOCO NP	.89 F
SUB-TOTAL 4.87	
TAX	.00
BALANCE	4.87

STAR MARKET
GREAT ROAD
STOW, MA

VISA/MASTERCARD PAYMENT

CHANGE

9/04/00 6:24 PM 0152 04 0429 109

SIGN UP TODAY TO GET EXTRA SAVINGS
WITH THE STAR MARKET ADVANTAGE CARD
KEITH SHAW - STORE MANAGER
THANK YOU

Aidons Mme Guénolé



Epicéris de
Mme. Guénolé.



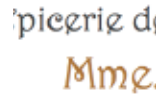
Epicéris de
Mme. Guénolé.



Epicéris de
Mme. Guénolé.



Epicéris de
Mme. Guénolé.



Epicéris de
Mme. Guénolé.



Epicéris de
Mme. Guénolé.

Camemb
Bière
Perrier
Oeufs,
Salade
Confitur
Couches





Epicéris de
Mme. Guénolé.

Biscuit
Crème
Couches
Pain,
Confiture



Couches
Confiture

mbert

de cacao



de cacao



mbert

de cacao

ouare de cacao
biscuit



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

Recherche de règles d'association

□ Règles de la forme

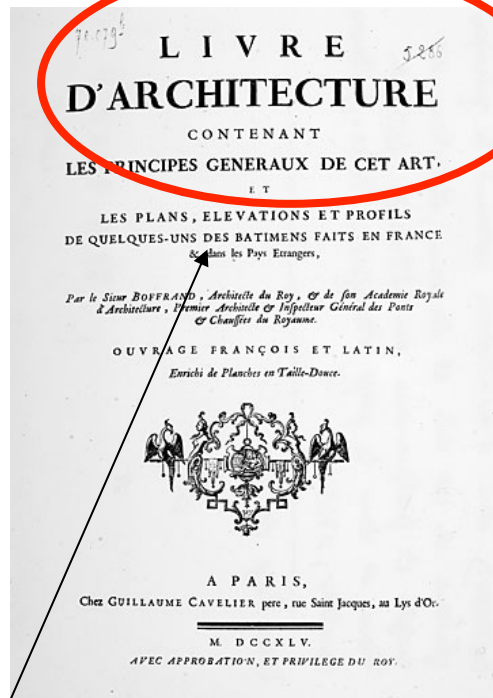
ANTECEDENT → CONSEQUENT [Support, Confiance]

(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%]

Panier de la ménagère

Localisation



Premier paragraphe

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

Identification

Position # Date

Mots # Produits

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 $\rightarrow$ Lait [70%, 80%]

■ Bière, Gâteaux $\rightarrow$ Couches [30%, 80%]

■ Caviar $\rightarrow$ Champagne [10%, 90%]



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

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 duplicats
- Soit une base de données D un ensemble de n transactions et chaque transaction est nommée par un identifiant (TID).
 - $D = \{\{T_1, \{\text{Café, Moutarde, Saucisse}\}\}, \{T_2, \{\text{Bière, Café, Gâteaux}\}\}, \dots\}$

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

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

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}

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)*

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

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)).

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

$$\text{conf}(R) = \frac{\text{Supp}(\text{Bi\`ere}, \text{G\^ateaux}, \text{Couches})}{\text{Supp}(\text{Bi\`ere}, \text{G\^ateaux})} \geq \text{confiance} ?$$

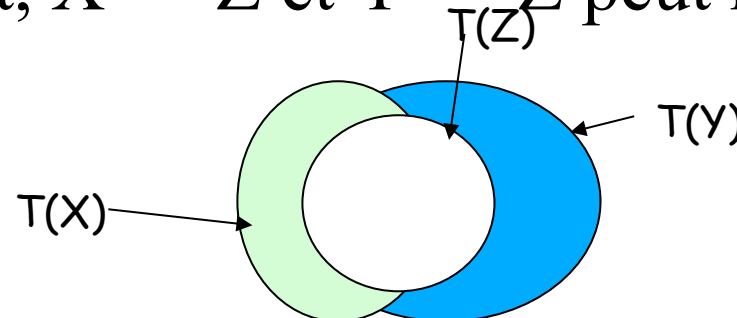
Propriétés des règles d'association

□ 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.



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 $B \subseteq A$.

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



Schéma algorithmique de base

- ❑ La plupart des approches utilise 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

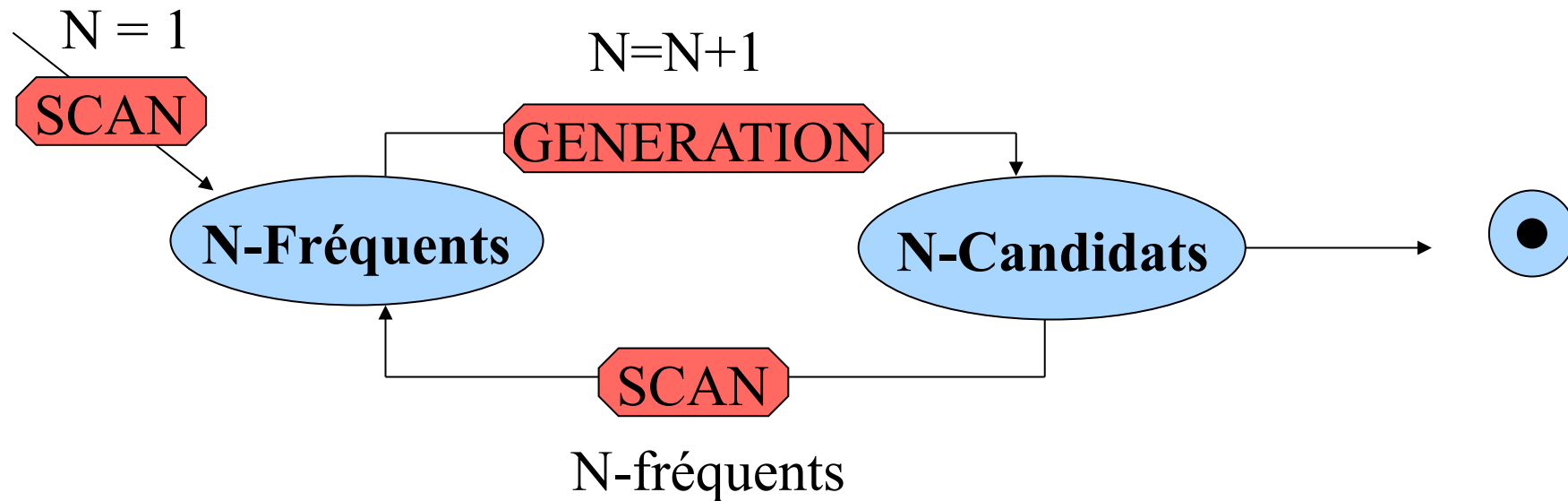
Comptage des itemsets

- 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

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 $\Rightarrow 2^{1000}$ ensembles à considérer

Vers un algorithme générique





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.

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, 2007



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

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} = 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} = candidats dans C_{k+1} avec minSupp

return $\bigcup_k L_k$;



Détails d'APRIORI

- Comment générer les candidats ?
 - Etape 1: auto-jointure sur L_k
 - Etape 2: élagage

- Comment compter le support des candidats ?

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  $p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$`
`FROM  $L_{k-1} p, L_{k-1} q$`
`WHERE  $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} <$`
 `$q.item_{k-1}$`
- 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  $C_k$`

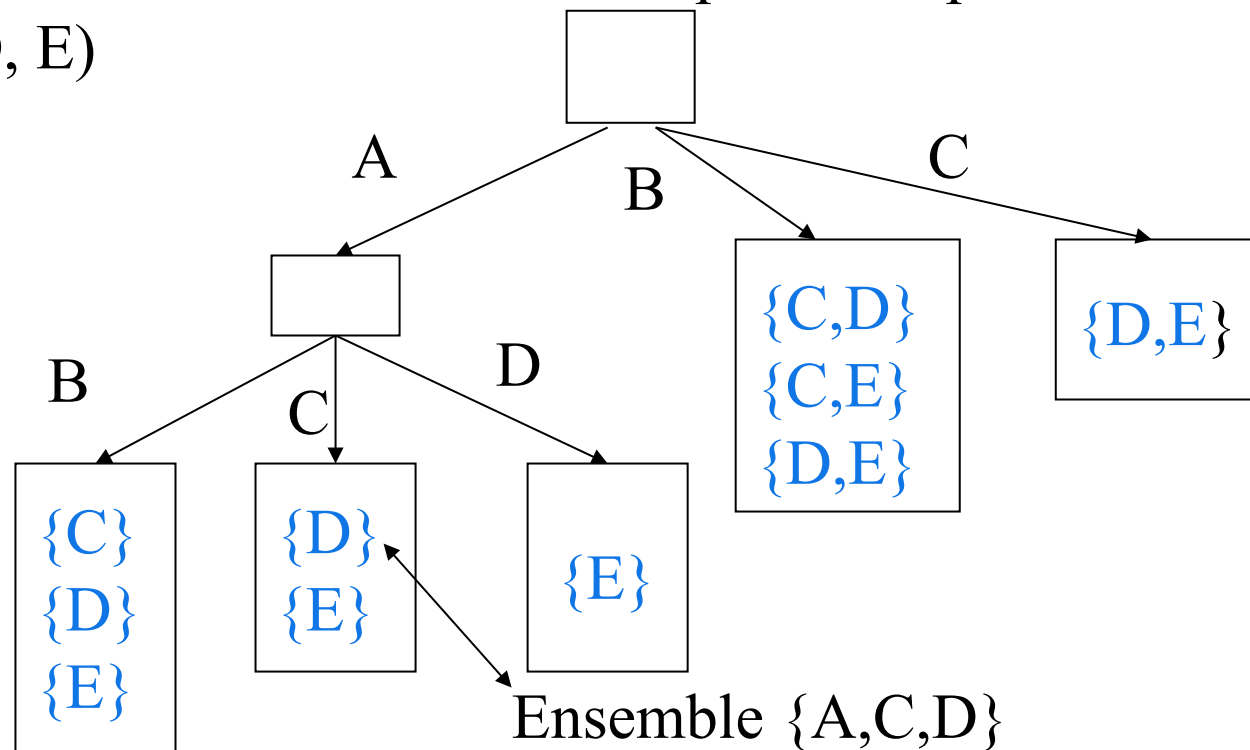
Génération des candidats : exemple

- $L_3 = \{abc, abd, acd, ace, bcd\}$
- 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
- $C_4 = \{abcd\}$

Stockage des candidats

- un arbre (structure de hash-tree)

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



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

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	DEG
7	B E
8	B D E F

Support minimal = 1

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

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} }

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

$L3 = \{\{A,B,C\}, \{A,B,D\}, \{A,B,E\}, \{A,B,F\}, \{A,C,D\}, \dots \{D,F,G\}\}$

$\{B,C,G\}$ élagué par Apriori-Gen car $\{C, G\}$ n'appartient pas à $L2$

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

$L4 = \{\{A,B,C,D\}, \{A,B,C,E\}, \{A,B,C,F\}, \dots \{C,D,E,F\}\}$

$\{B,D,F,G\}, \{B,E,F,G\}$ élagués car $\{B,F,G\}$ n'appartient pas à L3

$\{D,E,F,G\}$ élagué car $\{E,F,G\}$ n'appartient pas à L3

Illustration

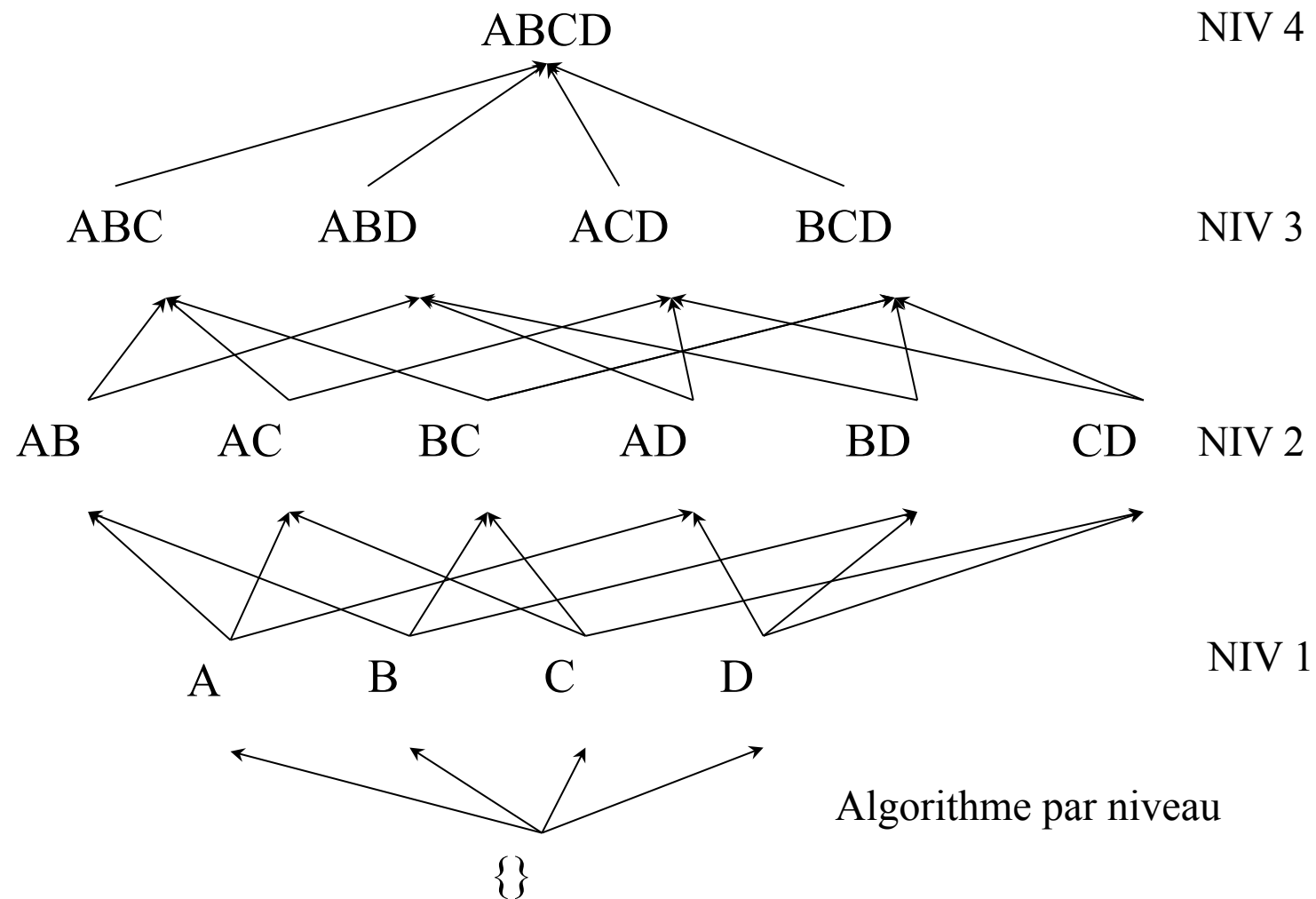
C6	Support
ABCDE	1

6-itemsets fréquents $L6 = \{\{A,B,C,D,E,F\}\}$

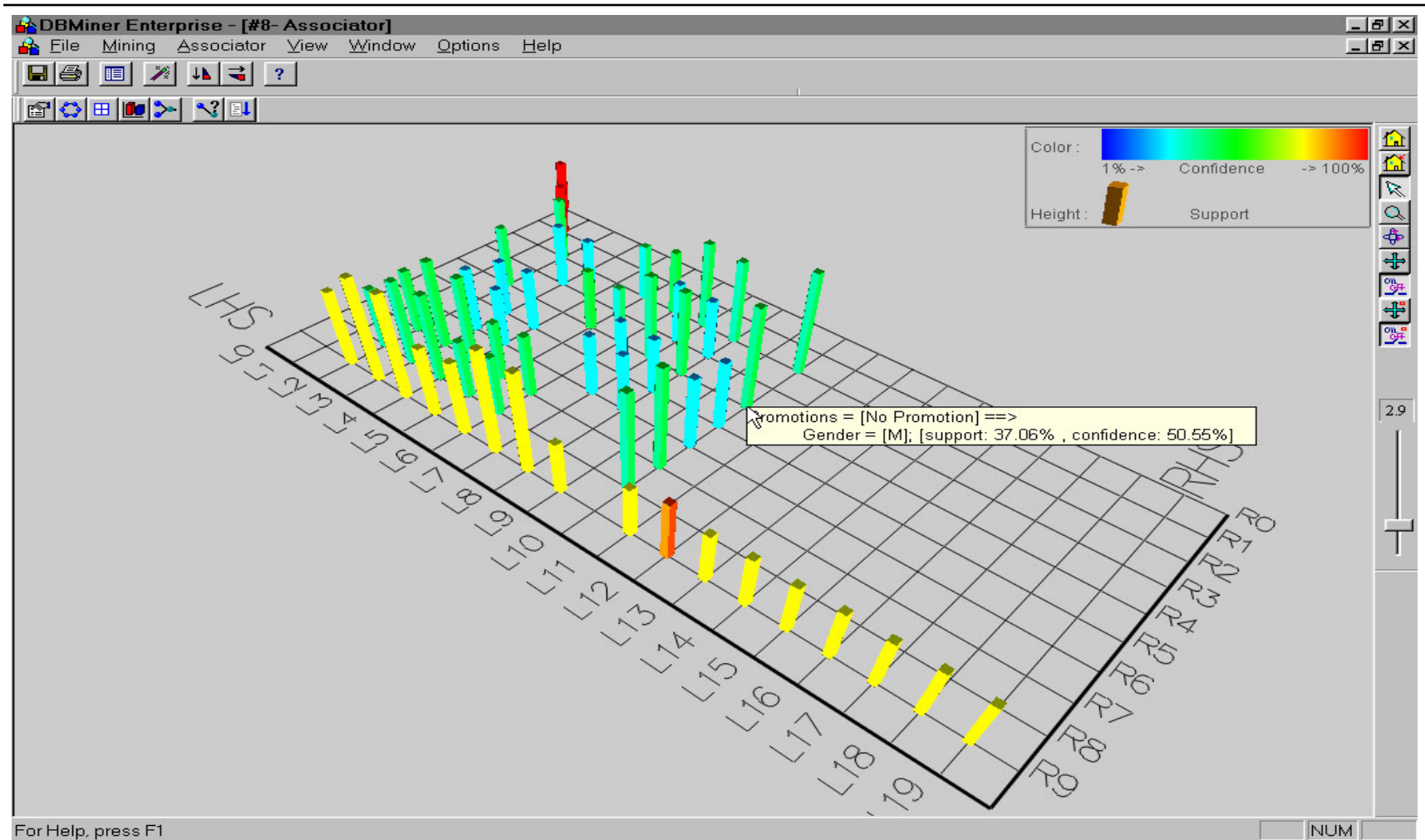
$C7 = \{\emptyset\} \Rightarrow 1$ 'algorithme se termine.

7 balayages pour déterminer tous les itemsets fréquents

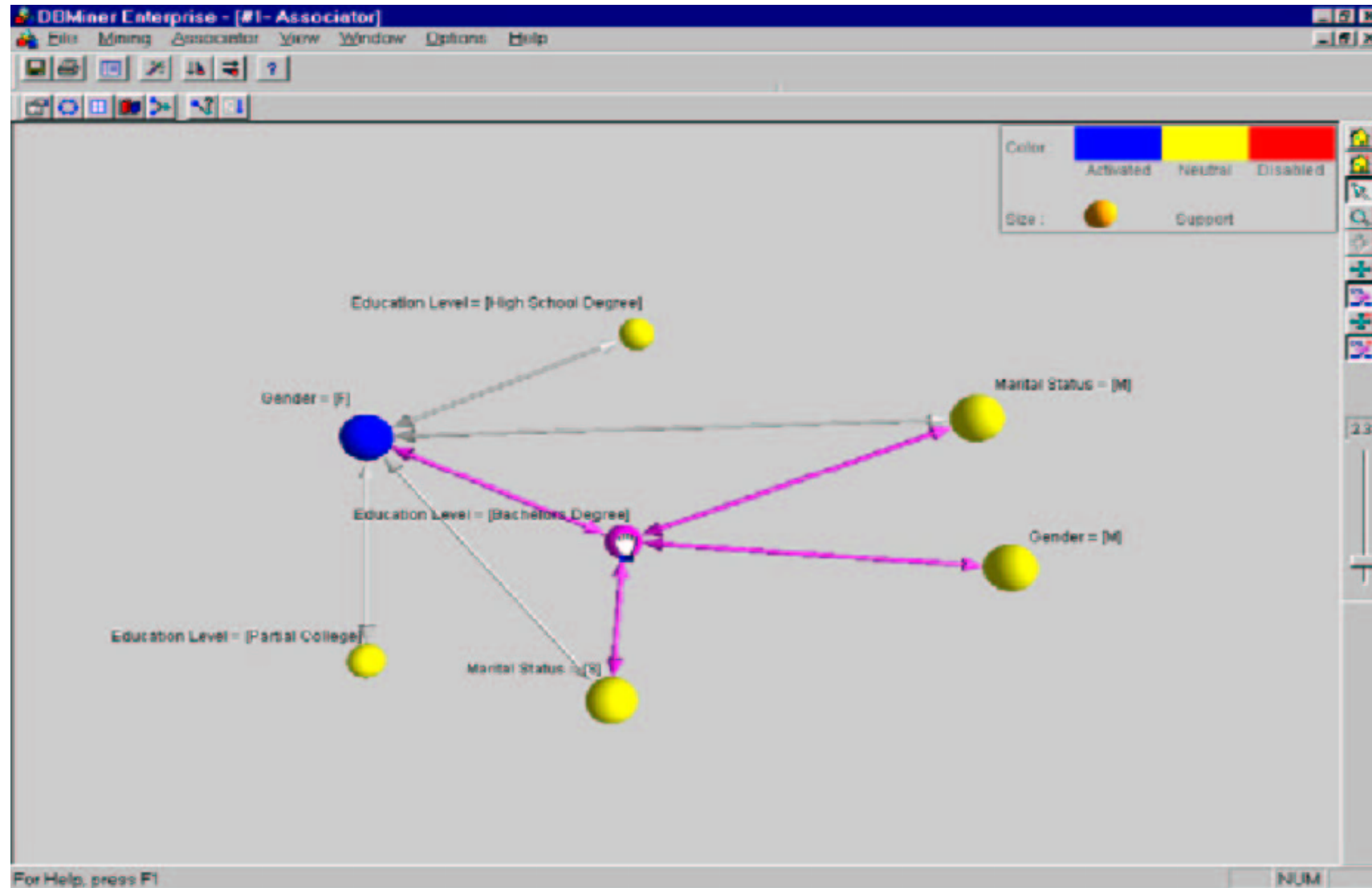
Espace de recherche



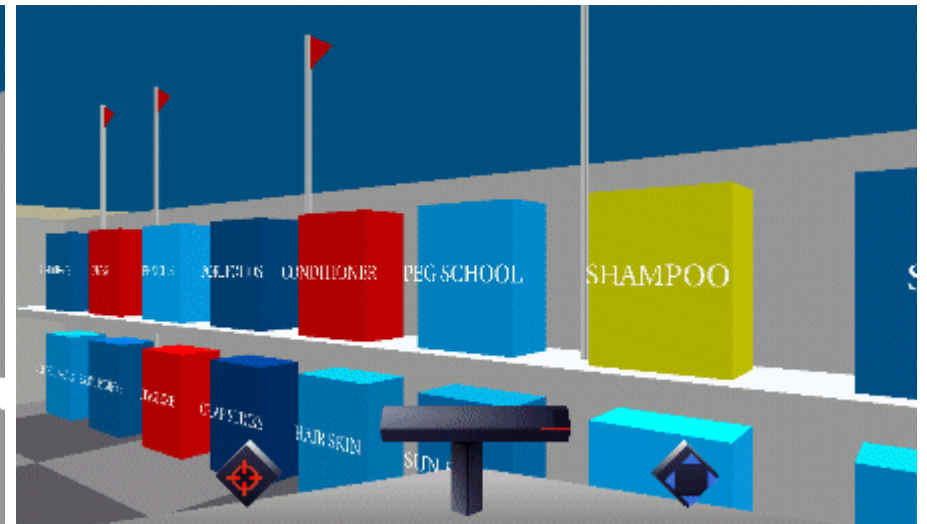
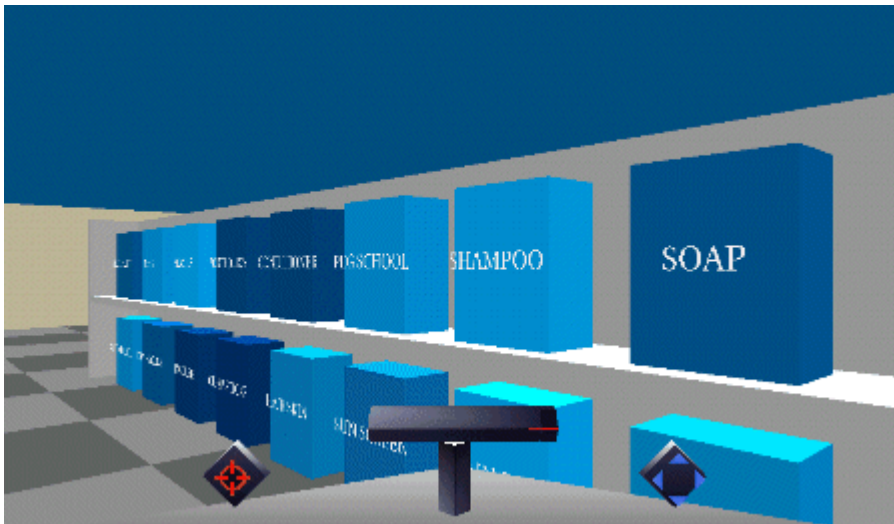
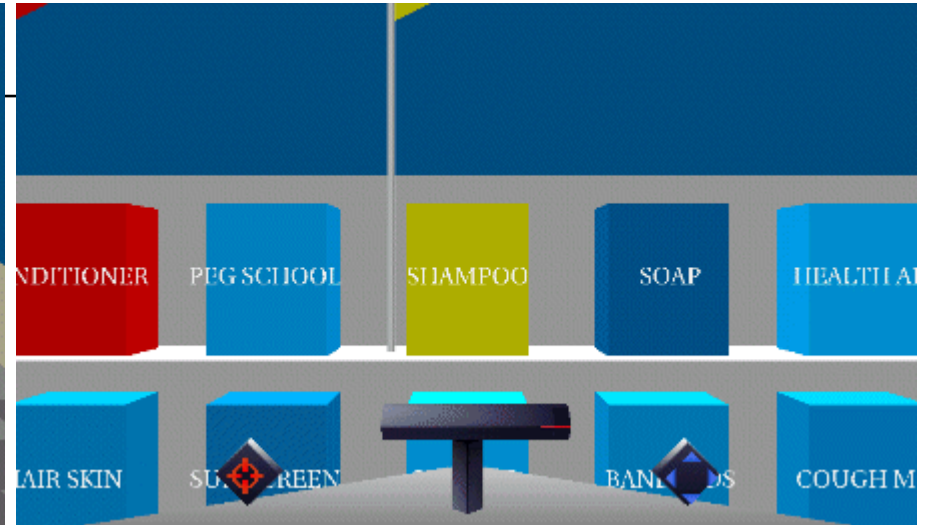
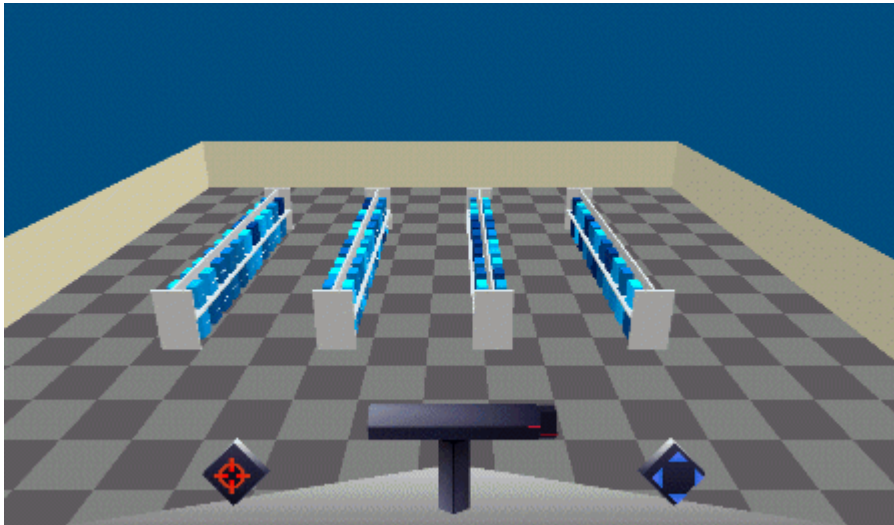
Visualisation



DBMiner (www.dbminer.com)



DBMiner (www.dbminer.com)



Intelligent Miner (www.ibm.com)



Plan

- Contexte général
- Règles d'association
- Motifs séquentiels
- Conclusions



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

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 :

$$[C, \langle \text{itemset}(T_1) \text{ itemset}(T_2) \dots \text{itemset}(T_n) \rangle]$$

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

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

$$i_1 < i_2 < \dots < i_n / a_1 \subseteq b_{i_1}, \dots, a_n \subseteq b_{i_n}$$

- $S1 = \langle (10) (20\ 30) (40) (20) \rangle$

$$S2 = \langle (20) (40) \rangle \subseteq S1$$

$$S3 = \langle (20) (30) \rangle \text{ n'est pas incluse dans } S1$$

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**

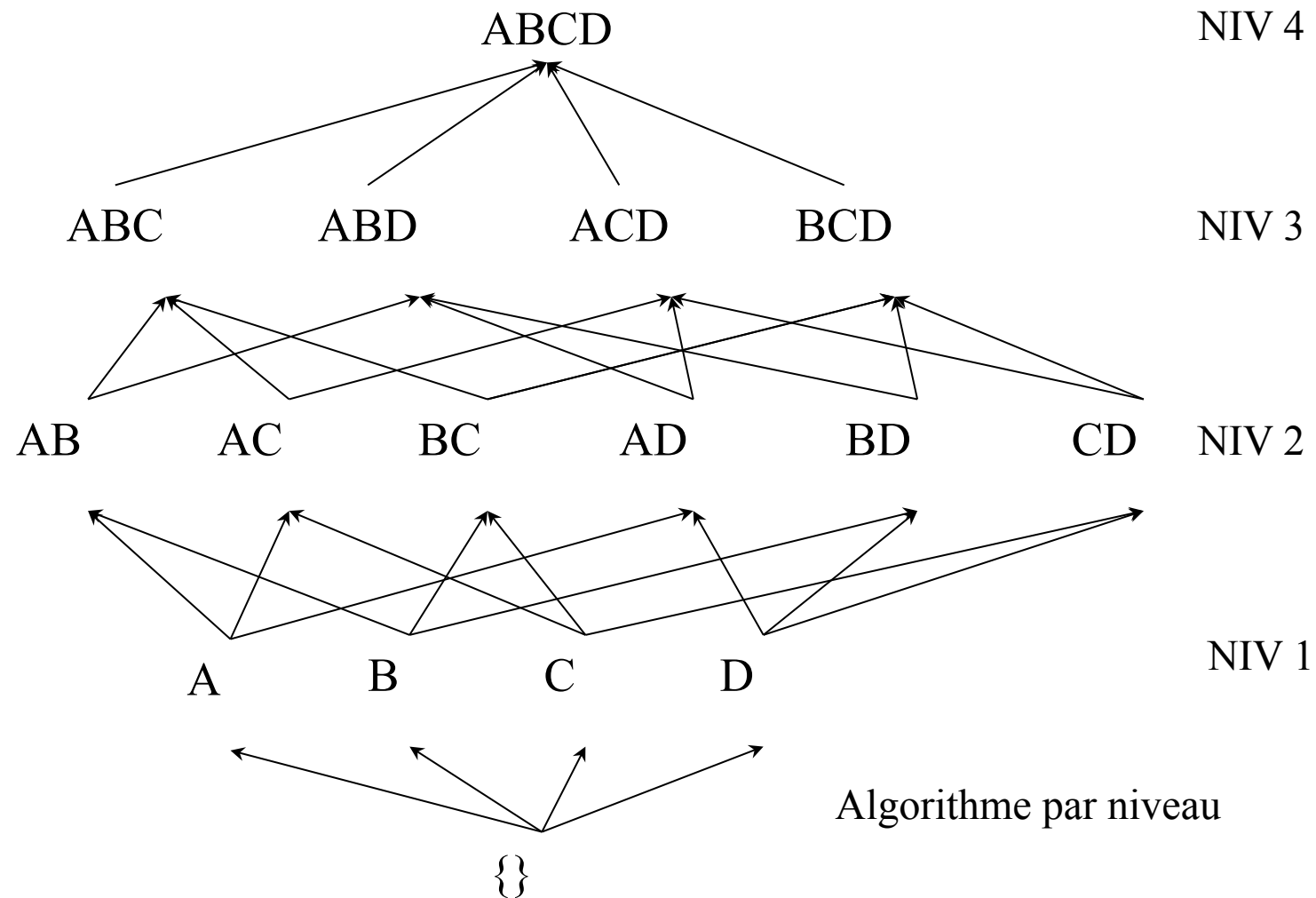
$\langle (\text{French wine, cheese}) (\text{bread}) \rangle$

Illustration

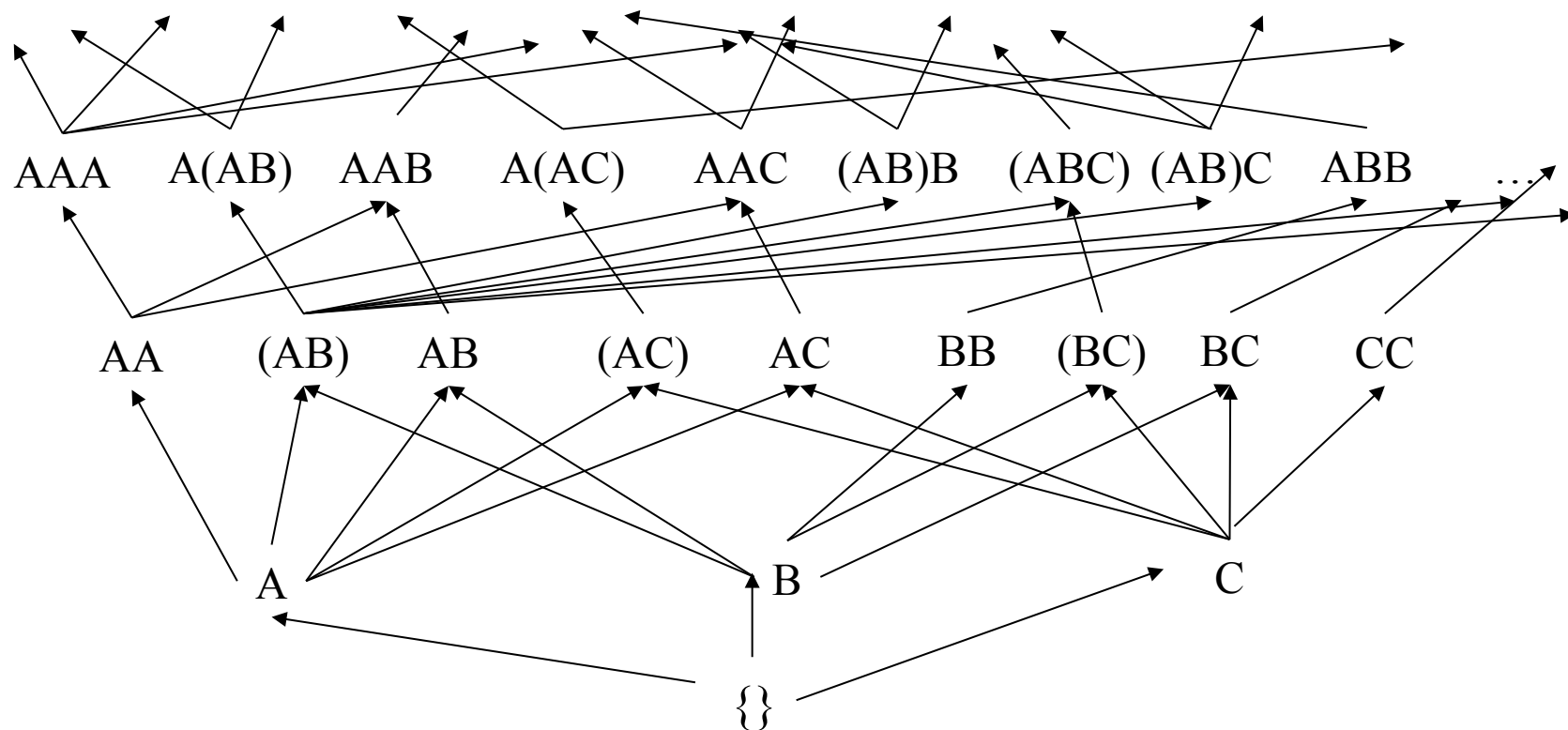
Clients	Date1	Date2	Date3	Date4
C1	10 20 30	20 40 50	10 20 60	10 40
C2	10 20 30	10 20 30		20 30 60
C3	20 30 50		10 40 60	10 20 30
C4	10 30 60	20 40	10 20 60	50

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

Itemsets :Espace de recherche



Motifs Séquentiels : l'espace de recherche



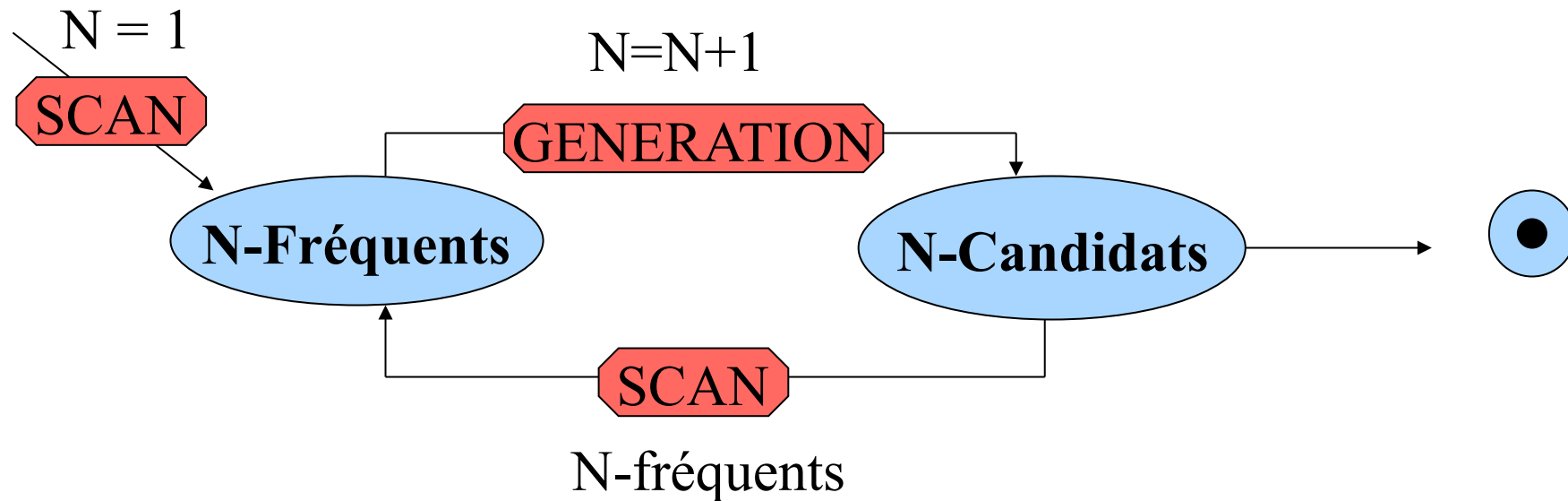
La propriété d'antimonotonie

- Une propriété essentielle (c.f. Apriori [AIS93])
 - Si une séquence n'est pas fréquente, aucune des super-séquences de S n'est fréquente!

Support ($\langle (10) (20\ 30) \rangle$) $<$ minsupp

Support ($\langle (10) (20\ 30) (40) \rangle$) $\ll$ minsupp

Vers un algorithme générique

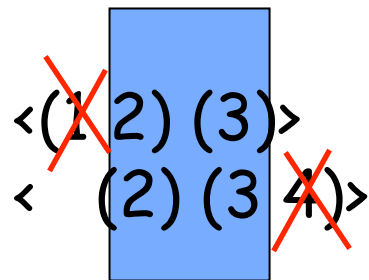




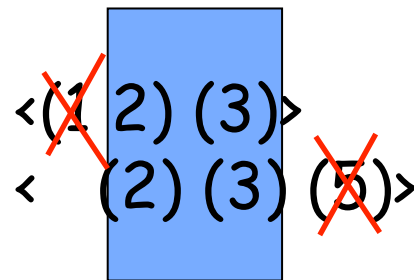
□ Comment générer les candidats ?

Génération des candidats

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



<(1 2) (3 4)>
I-Extension



<(1 2) (3) (5)>
S-Extension

GSP

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

L=1

While ($\text{Result}_L \neq \text{NULL}$)

 Candidate Generate

 Prune

 Test

 L=L+1

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	$\langle (ah)(bf)abf \rangle$
40	$\langle (be)(ce)d \rangle$
50	$\langle a(bd)bcb(ade) \rangle$

$minSupp = 2$

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

Le Processus

5th scan : 1 candidate
1 length-5 seq pattern

<(bd)cba>

4th scan : 8 candidates
6 length-4 seq pat

<abba> <(bd)bc> ...

3rd scan : 46 candidates
19 length-3 seq pat.

<abb> <aab> <aba> <baa> <bab> ...

2nd scan : 51 candidates
19 length-2 seq pat.

<aa> <ab> ... <af> <ba> <bb> ... <ff> <(ab)> ... <(ef)>

1st scan : 8 candidates
6 length-1 seq pattern

<a> <c> <d> <e> <f> <g> <h>

Génération des candidats de taille 2

S-Extension

51 2-Candidates

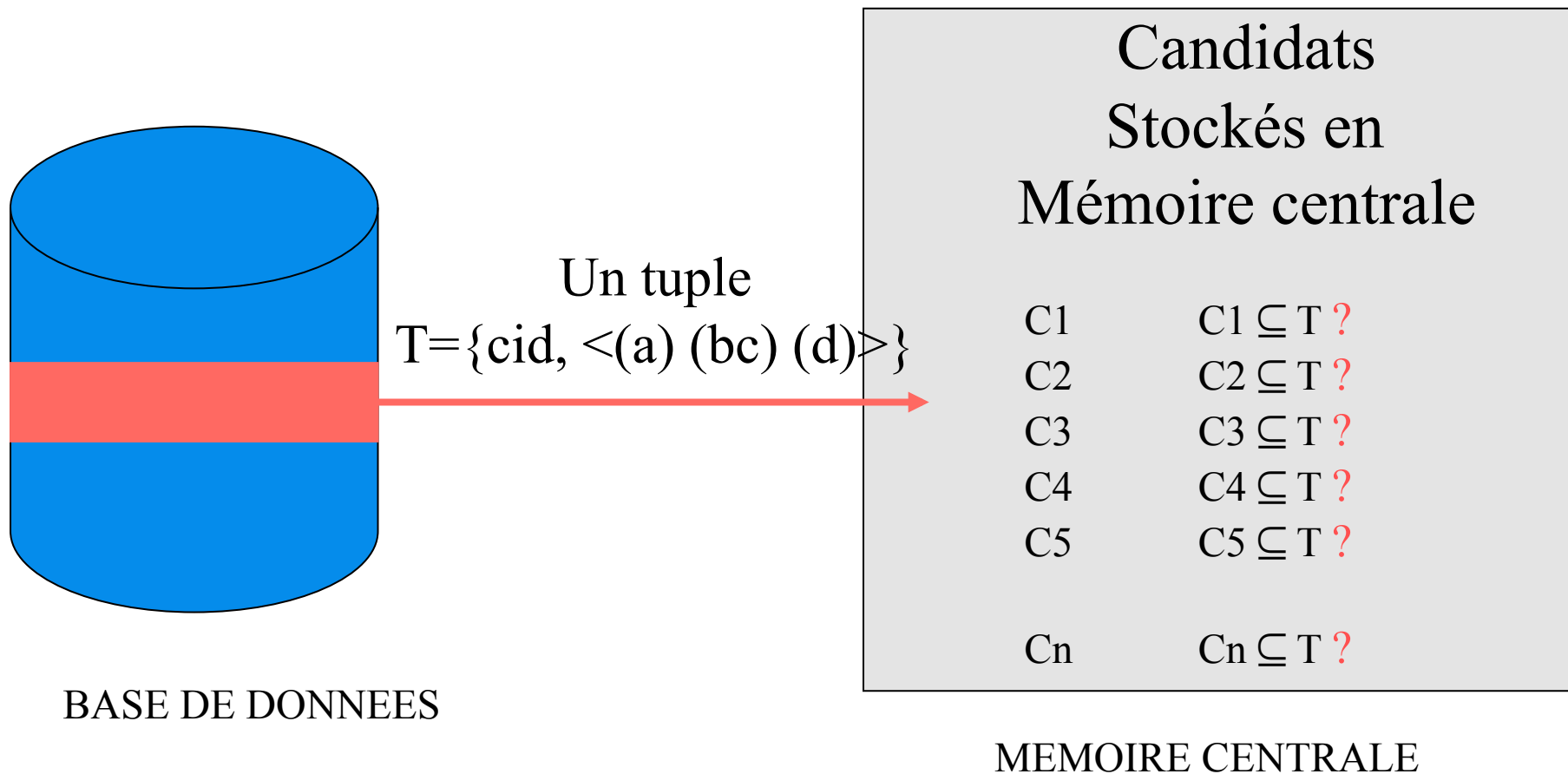
	<a>		<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

I-Extension

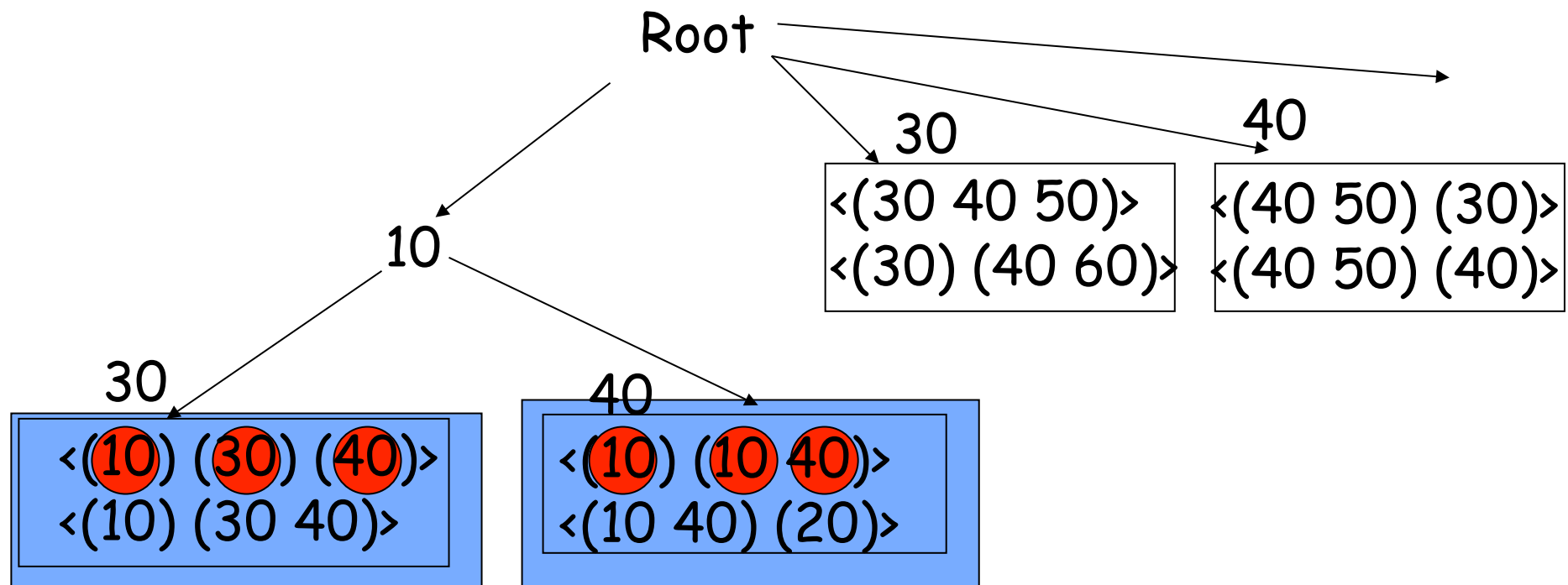
	<a>		<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

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

Comptage des supports des candidats



Stockage des candidats



$S = \langle (10) (30) (10) (40) \rangle$



Conclusions

- Les points forts :
 - Résultats clairs et explicites.
 - Adaptée à l'exploitation non dirigée des données
 - Travaille sur des données de taille variable
 - Calculs utilisés simples à comprendre
- Les points faibles :
 - Volume de calculs (fonction du nombre d'items)
 - Difficulté de sélectionner le bon nombre d'articles



Conclusions

- ❑ Depuis 1996 :
- ❑ Problème de recherche ouvert
- ❑ Données de plus en plus complexes (représentations, ...), obtenues de plus en plus rapidement (incrémental, flots de données), avec de nouvelles contraintes (préservation de la vie privée, contraintes de dimensions, temporelles), avec valeurs manquantes, ...
- ❑ Besoins de nouveaux indicateurs de qualité

Conclusions

- ❑ Une URL : KDD Mine [ttp://www.kdnuggets.com](http://www.kdnuggets.com)
- ❑ Google, citeseer, ...
- ❑ Quelques outils

Intelligent Miner (www.ibm.com)

Enterprise Miner (SAS Institute)

MineSet (Silicon Graphics Inc.)

Clementine (Integral Solutions Ltd, racheté par SPSS)

DBMiner (www.dbminer.com)

- ❑ Le projet Weka (bibliothèque de classes Java)

<http://www.cs.waikato.ac.nz/ml/weka>



□ La suite en Master 2