Fouille de Séquences

Marc Plantevit

UCB Lyon 1 - LIRIS - CNRS

marc.plantevit@liris.cnrs.fr

Outline

Règles d'association : rappel

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

Règles d'association : rappel

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

- Itemsets : A,B ou B,E,F

Règles d'association : rappel

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

- Itemsets : A,B ou B,E,F
- Support d'un itemset :
  - $support(AD) = 1$
  - $support(AC) = 2$

Règles d'association : rappel

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

- Itemsets : A,B ou B,E,F
- Support d'un itemset :
  - $support(AD) = 1$
  - $support(AC) = 2$
- Itemsets fréquents ($minsupp = 50\%$) :
  - $\{A, C\}$ et $\{A, B\}$ sont fréquents

Règles d'association : rappel

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

- Itemsets : A,B ou B,E,F
- Support d'un itemset :
  - $support(AD) = 1$
  - $support(AC) = 2$
- Itemsets fréquents ($minsupp = 50\%$) :
  - $\{A, C\}$ et $\{A, B\}$ sont fréquents
- Pour $minsupp = 50\%$ et $minconf = 50\%$, on a les règles suivantes :
  - $A \rightarrow C$ $[50\%, 50\%]$
  - $C \rightarrow A$ $[50\%, 100\%]$
  - $A \rightarrow B$ et $B \rightarrow A$?

Association vs. motif séquentiel

Règles d'association

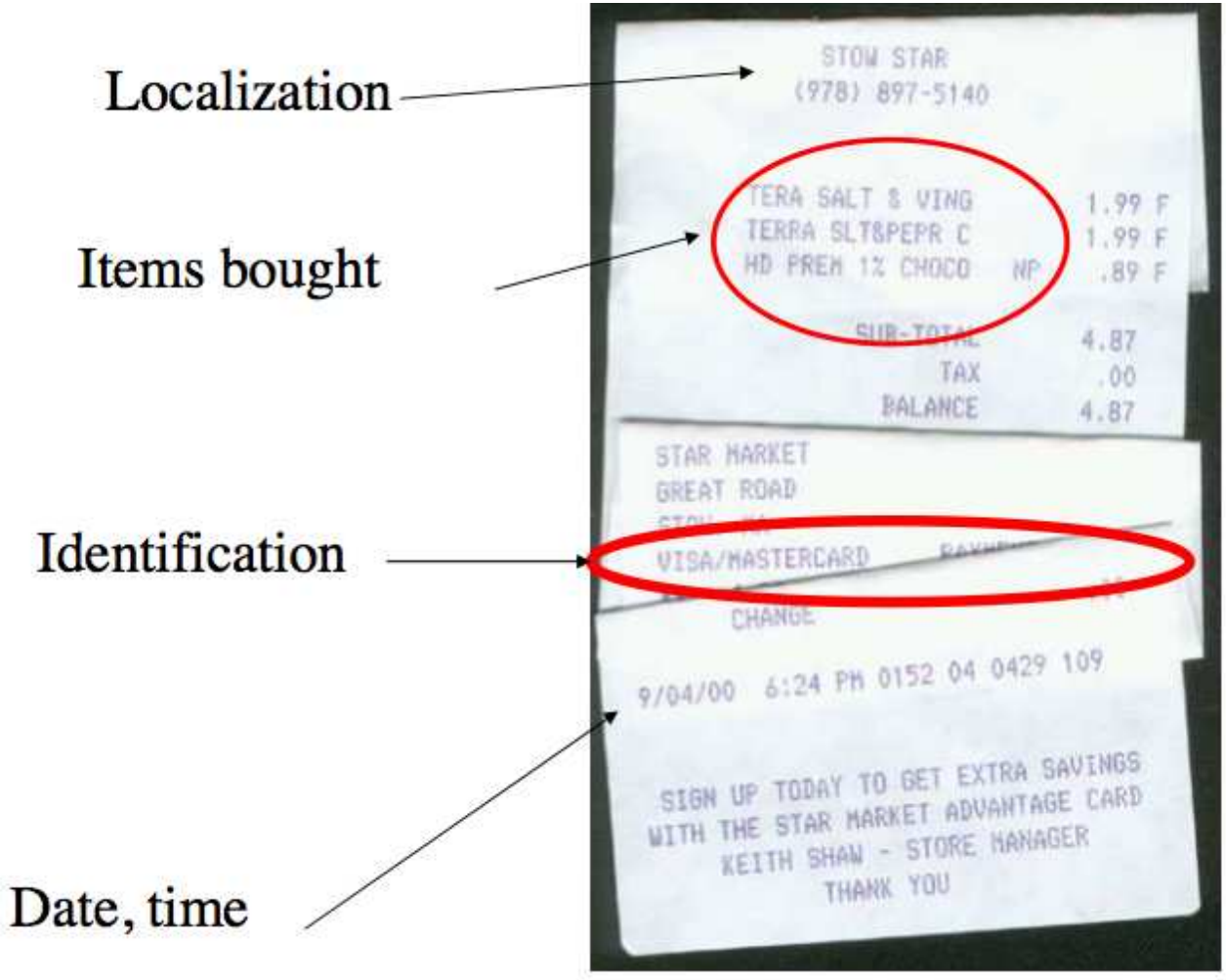
- ensemble d'items qui apparaît fréquemment dans une base de données
- Corrélation entre ces items

Motifs séquentiels

- Liste d'ensemble d'items qui apparaît fréquemment dans une base de données
- L'idée principale est de comprendre les habitudes d'un consommateurs, etc.

Analyse du panier de la ménagère

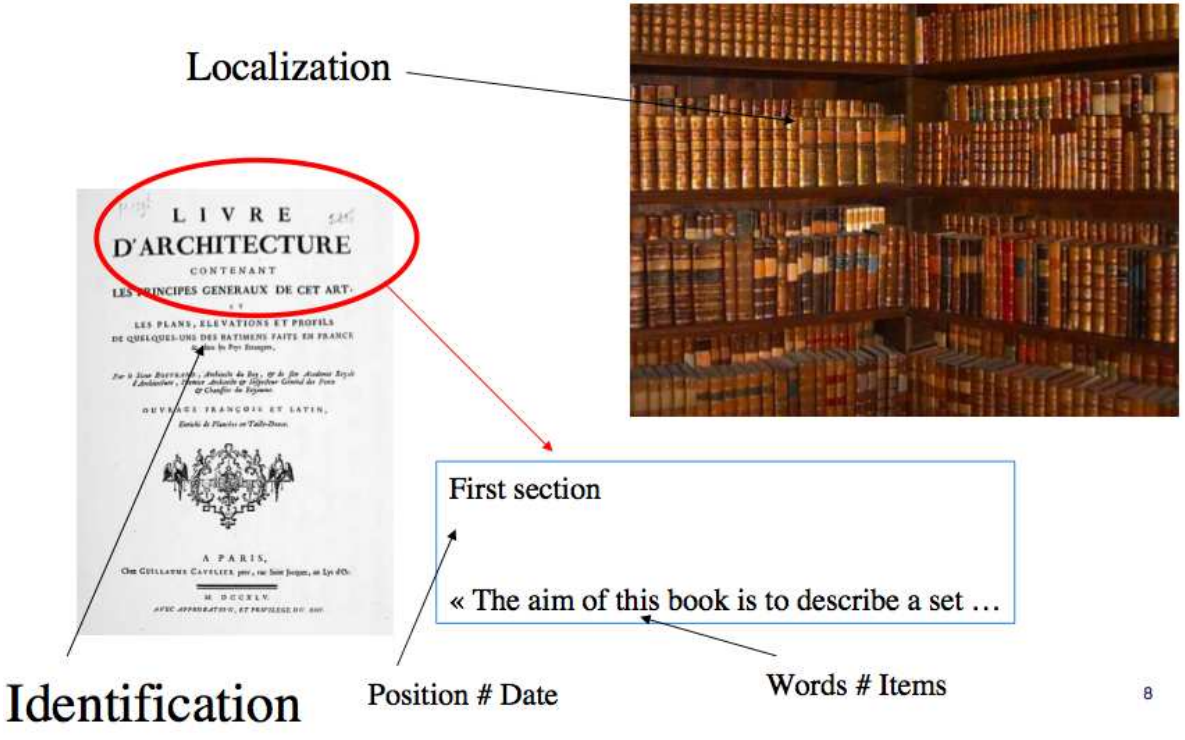
Analyse du panier de la ménagère



Encore

Textes

S_1 : ⟨"s-Tomosyn", "as", "well", "as", "m-tomosyn", "was", "mainly", ...⟩
S_2 :
⟨"This", "region", "was", "necessary", "but", "not", "sufficient", "for", ...⟩
...



Autres exemples de séquences de données

Biological Sequences : DNA, RNA and Proteins

GAATTCTCTGTAACACTAAGCTCTCTTCCTCAAAACCAGAGGTAGA
ATGTGTAATAATTTACAGAATTTCTAGACTTCAACGATCTGATTTTT
ATTTATTTTTTATTTTTTTCAGGTTGAGACTGAGCTAAAGTTAATCTG

Autres exemples de séquences de données

Biological Sequences : DNA, RNA and Proteins

GAATTCTCTGTAACACTAAGCTCTCTTCCTCAAAACCAGAGGTAGA
ATGTGTAATAATTTACAGAATTTCTAGACTTCAACGATCTGATTTTT
ATTTATTTTTTATTTTTTTCAGGTTGAGACTGAGCTAAAGTTAATCTG

Event Sequences : Weblogs, System Traces, Purchase Histories and Sales Histories

- weblog :
  $\langle 100, a\rangle, \langle 100, b\rangle, \langle 200, a\rangle, \langle 300, b\rangle, \langle 200, b\rangle, \langle 400, a\rangle, \langle 100, a\rangle, \langle 400, b\rangle$
- customer purchase history :
  $\langle 223100, 05/26/06, 10am, CentralStation, \{WholeMealBread, AppleJuice\}\rangle,$
  $\langle 225101, 05/26/06, 11am, CentralStation, \{Burger, Pepsi, Banana\}\rangle$
- Storewide sales history :
  $\langle 97100, 05/06, \{\langle Apple : \$85K\rangle, \langle Bread : \$100K\rangle, \langle Cereal : \$150K\rangle, \ldots\}\rangle,$
  $\langle 90089, 05/06, \{\langle Apple : \$65K\rangle, \langle Bread : \$105K\rangle, \langle Diaper : \$20K\rangle, \ldots\}\rangle$

Pourquoi l'extraction de motifs séquentiels ?

Un important problème en DM avec de nombreuses applications :

- Analyse des achats des consommateurs
- Analyse ADN,
- Conséquences des catastrophes naturelles,
- Web log mining,
- Détection de tendances

Outline

Définitions

Soit $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ un ensemble de littéraux appelés **items**

- $\mathcal{I} = \{$PC, TV, bière, couches, etc.$\}$

Définitions

Soit $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ un ensemble de littéraux appelés **items**

- $\mathcal{I} = \{$PC, TV, bière, couches, etc.$\}$

Transaction

Une transaction est composée de :

- Un identifiant (id_client),
- Une date de transaction,
- Un ensemble d'items (**itemsets**)
- Exemple : $T = [Cid1, (Beer, Cake, Diaper)_5]$

Séquence et séquence de données

Séquence :

Une **liste ordonnée** d'itemsets

- Exemple : $\langle$(TV),(DVD-Reader, Star War box)$\rangle$

Séquence et séquence de données

Séquence :

Une **liste ordonnée** d'itemsets

- Exemple : $\langle$(TV),(DVD-Reader, Star War box)$\rangle$

Séquence de données

Soient $T_1, T_2, \ldots, T_m$, les transactions d'un client C, la séquence de données de C est la paire :

$$(C, \langle itemset(T_1), itemset(T_2), \ldots, itemset(T_m)\rangle)$$

Ex : Les achats d'un consommateur au cours du temps.

Séquence et séquence de données

Séquence :

Une **liste ordonnée** d'itemsets

- Exemple : $\langle$(TV),(DVD-Reader, Star War box)$\rangle$

Séquence de données

Soient $T_1, T_2, \ldots, T_m$, les transactions d'un client C, la séquence de données de C est la paire :

$$(C, \langle itemset(T_1), itemset(T_2), \ldots, itemset(T_m)\rangle)$$

Ex : Les achats d'un consommateur au cours du temps.

Base de séquences (de données)

$SDB =$

- $(C_1, \langle(Coffe, Mustard), (Coke), (Diaper)\rangle)$
- $(C_2, \langle$(Beer)(Coke)$\rangle)$

Inclusion de séquences

Inclusion

Soient deux séquenses $S_1 = \langle a_1, a_2, \ldots, a_n \rangle$ et $S_2 = \langle b_1, b_2, \ldots, b_{n'} \rangle$, S_1 est une sous-séquence de S_2 (S_2 superséquence de S_1, noté $S_1 \preceq S_2$)) si il existe des entiers $1 \leq i_1 < i_2 < \ldots < i_n \leq n'$ tels que :

$$a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \ldots, a_n \subseteq b_{i_n}$$

Exemple

- $S_1 = \langle (10)(20, 30), (40), (20) \rangle$
- $S_2 = \langle (20), (40) \rangle \preceq S_1$
- $S_3 = \langle (20), (30) \rangle \not\preceq S_1$

Support d'une séquence

L'occurence d'une séquence n'est considérée qu'une seule fois dans une séquence de données

Ex : $\langle(20)\rangle$ et $\langle(10)(20)(20,30),(40),(20)\rangle$
On dit aussi que la séquence de données **contient** la séquence $\langle(20)\rangle$.

Support d'une séquence

Le support d'une séquence s dans une base de données SDB correspond au nombre de séquences de données qui contiennent s.

Exemple

Customers	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

- $Absolute\_Support(\langle(10,30)(20)(20,60)\rangle) = 3$
- $Relative\_Support(\langle(10,30)(20)(20,60)\rangle) = \frac{3}{4} = 75\%$

Problématique

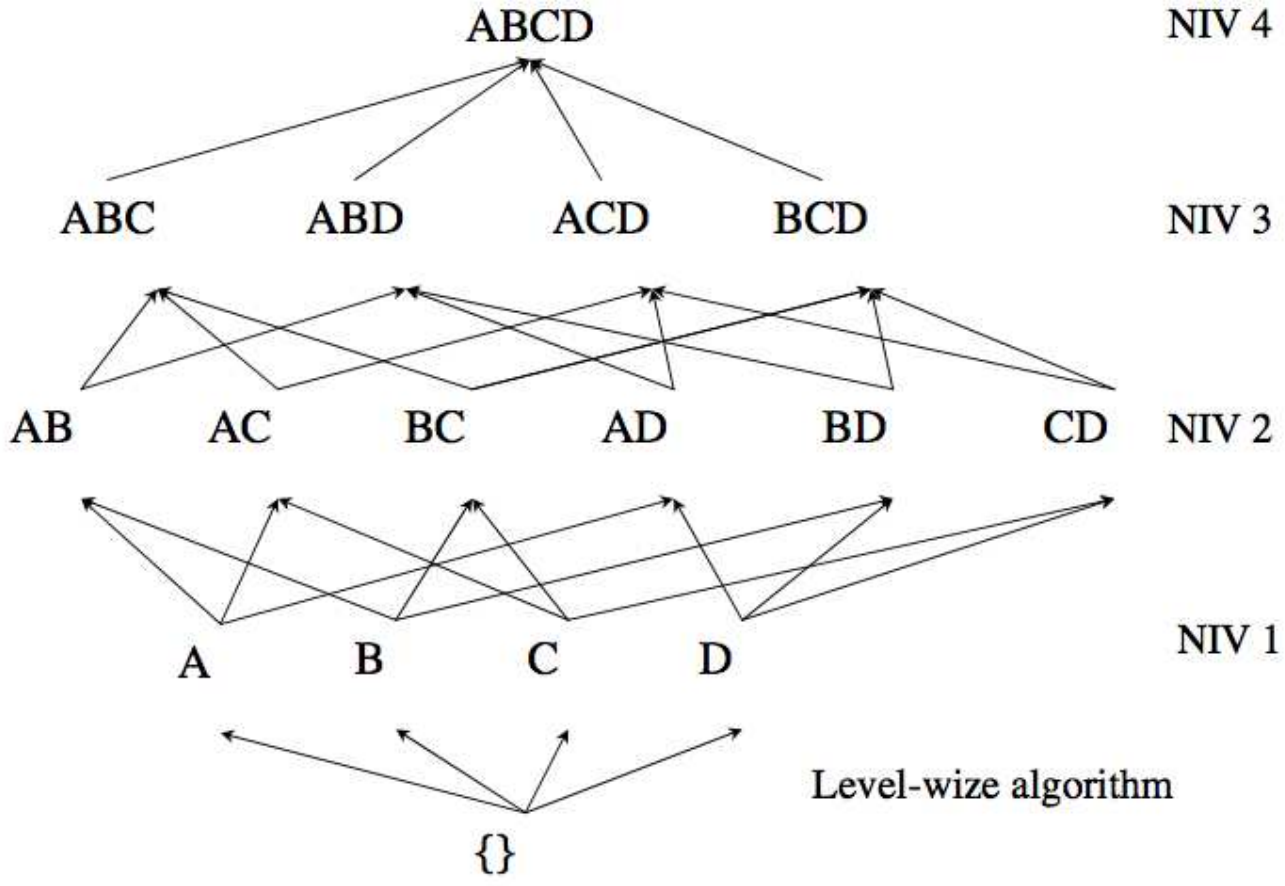
Etant donnés un ensemble de séquences de données *SDB* et un seuil de support minimum *minsupp*, le but de l'extraction des motifs séquentiels est de trouver l'ensemble complet des séquences qui ont un support supérieur ou égal à *minsupp*.

Les challenges de l'extraction de motifs séquentiels

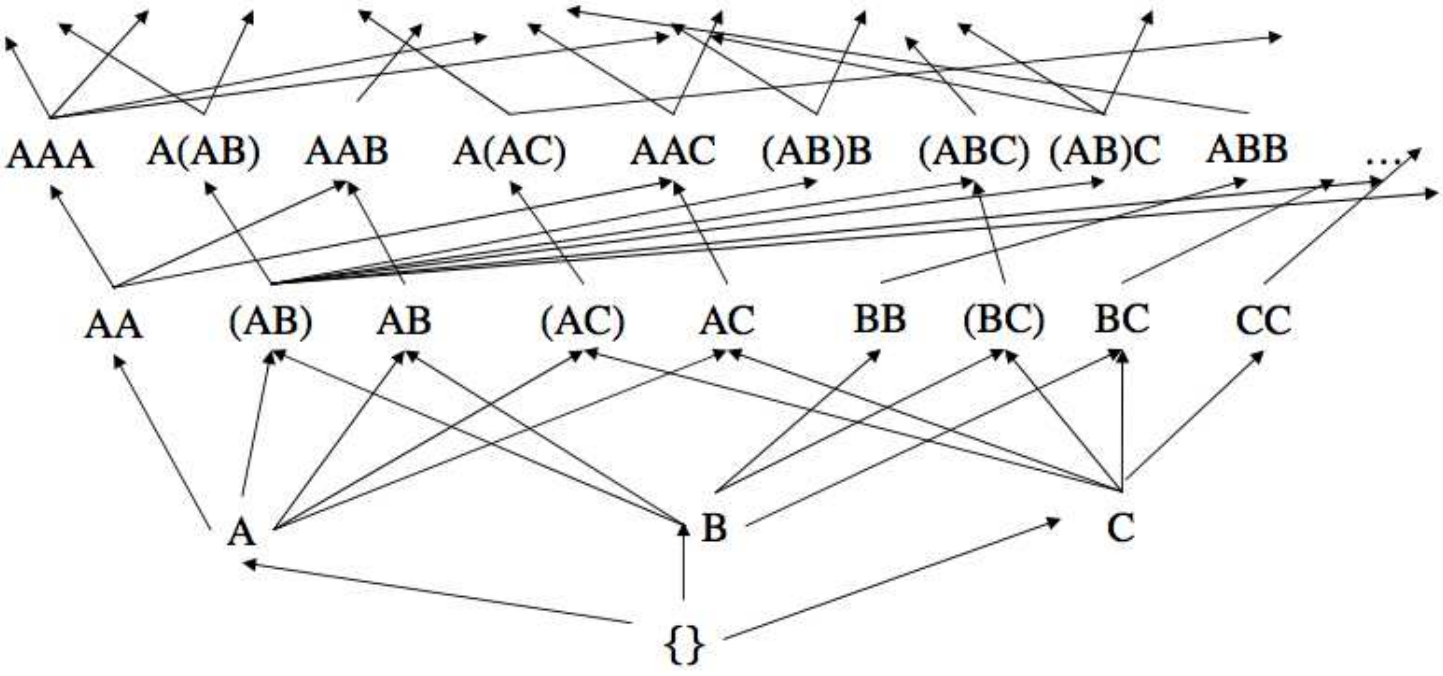
- Le nombre de motifs séquentiels « cachés » dans les séquences de données peut être très importants
- Un algorithme d'extraction doit donc :
  - trouver **l'ensemble complet** des motifs séquentiels
  - être efficace (passage à l'échelle) en limitant le nombre de passes sur les données
  - permettre d'incorporer un ensemble varié de contraintes définies par l'utilisateur (syntaxiques, temporelles, etc.)

Outline

Itemset : l'espace de recherche (un treillis)



Motifs séquentiels : l'espace de recherche



Propriété Apriori des motifs séquentiels

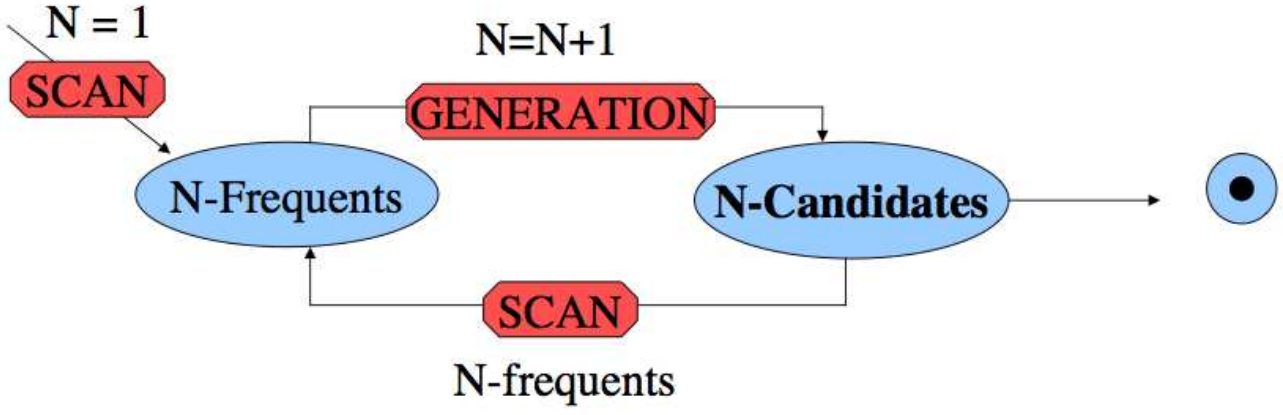
Une propriété fondamentale [Agrawal et Srikant, 94]

- Si une séquence S n'est pas fréquente
- Alors aucune de ses super-séquences est fréquente
- Ex : $\langle hb \rangle$ non fréquent alors $\langle hab \rangle$ $\langle (ah)b \rangle$ non plus

Seq. ID	Séquence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

minsupp =2

Approche générale : générer/élaguer



Génération de candidats : deux types d'extension

S-Extension

On ajoute un nouvel itemset (une séquence) à la séquence considérée

- $\langle (a, b)(c) \rangle \rightarrow \langle (a, b)(c)(d) \rangle$

I-Extension

On ajoute un nouvel item à la séquence considérée dans un itemset déjà créé.

- $\langle (a, b)(c) \rangle \rightarrow \langle (a, b)(c, d) \rangle$

GSP (Agrawal et Srikant) : basé sur le principe Apriori

L=1

While ($Result_L$!= NULL)

    Candidate Generate

    Prune

    Test

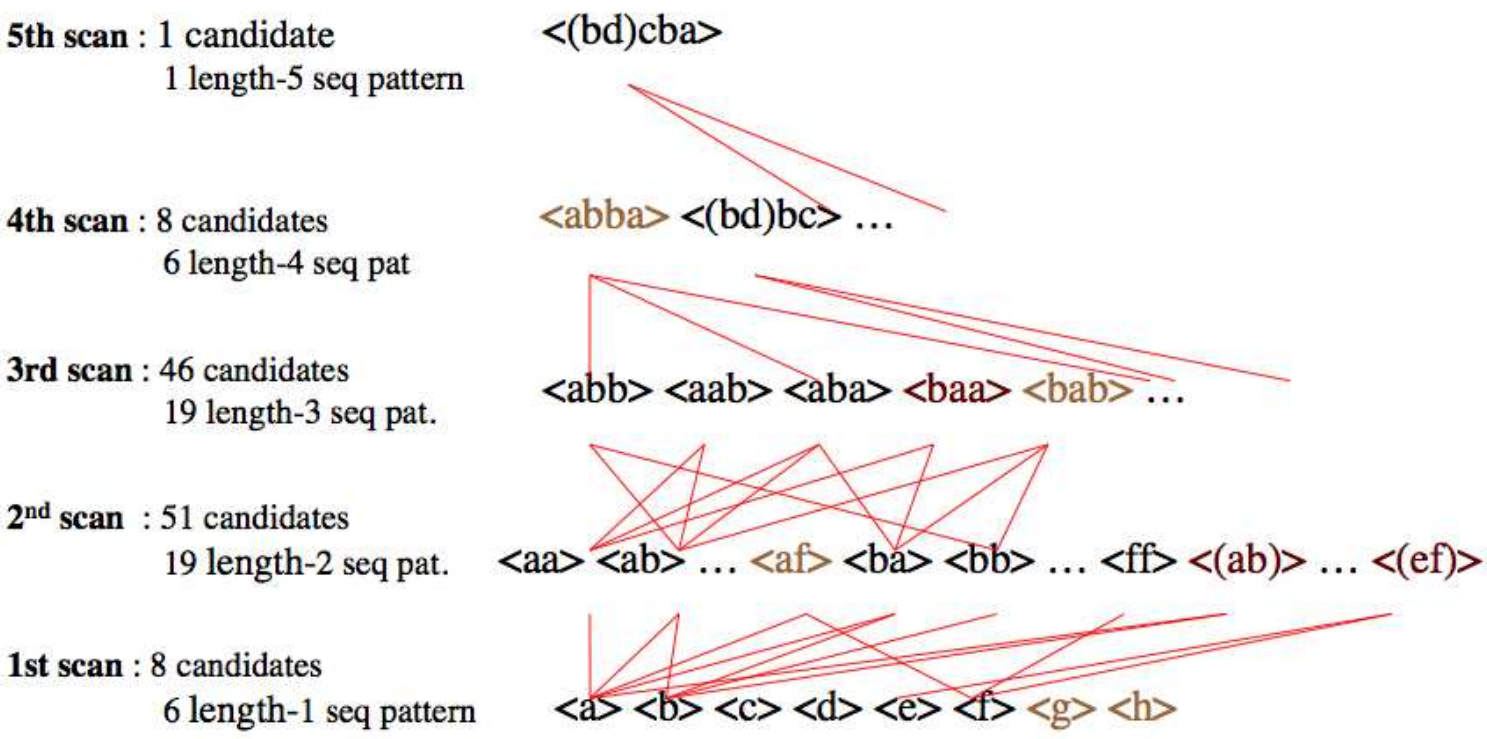
    L=L+1

Une passe sur les données pour extraire les séquences composées d'un unique item (dans un unique itemset).

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Cand	Sup
<a>	**3**
<b>	**5**
<c>	**4**
<d>	**3**
<e>	**3**
<f>	**2**
~~<g>~~	1
~~<h>~~	1

Les processus global



Génération des candidats de longueur 2

S-Extension

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<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>

51 2-Candidates

I-Extension

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

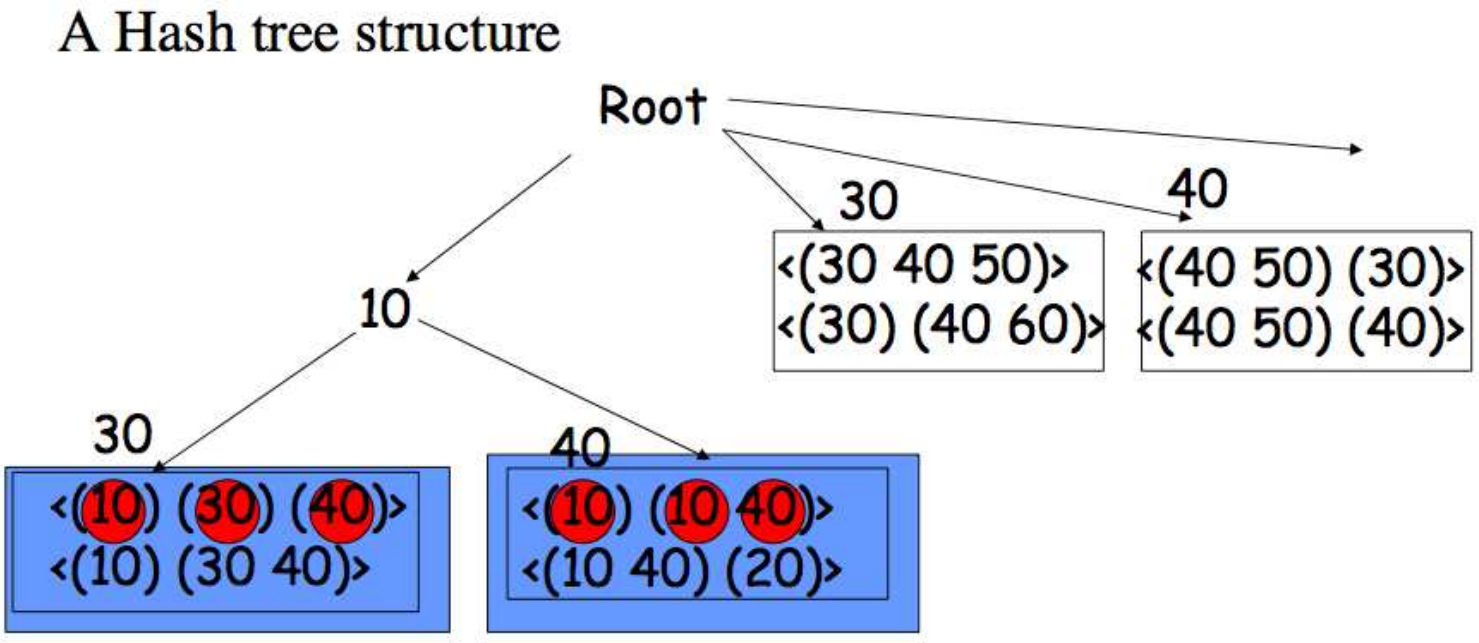
Without the antimonotonic property
8*8+8*7/2=92 candidates

Calcul du support des candidats

L'étape la plus coûteuse

- Les candidats sont stockés en mémoire centrale.
- Il faut limiter les accès disques sur la base de données
- Possibilité de charger la base en mémoire quand c'est possible

Comment stocker efficacement les candidats ?

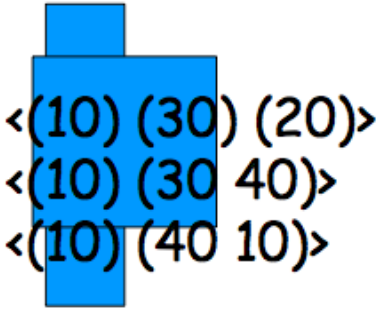


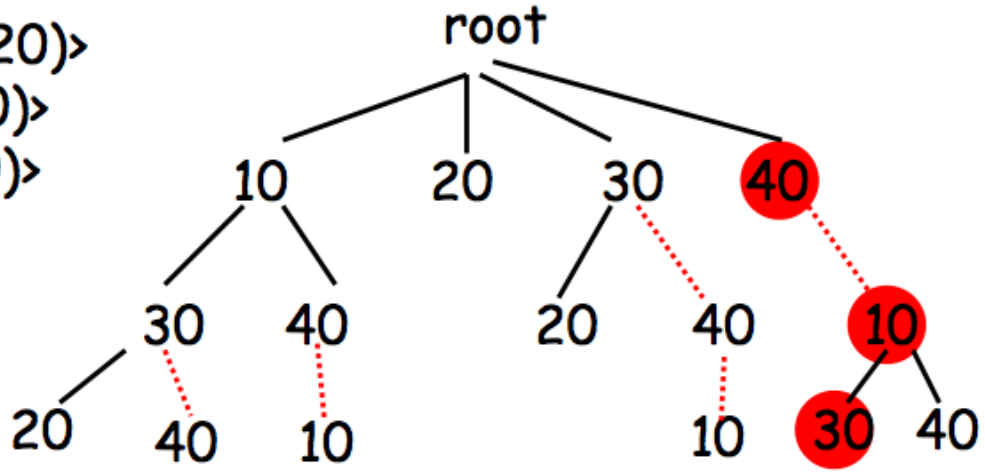
$S = \langle (10)(30)(10,40) \rangle$

PSP (Prefix Tree for SP) [Masseglia et al. 98]

Une structure plus efficace basée sur un arbre des préfixes.

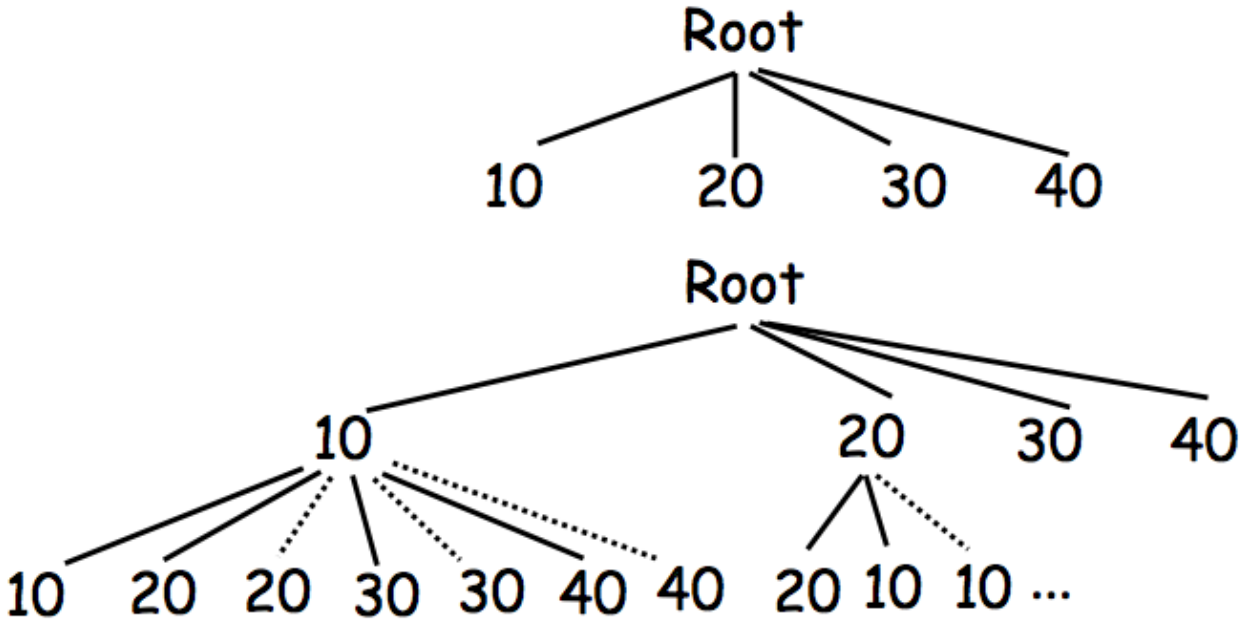
- 2 types d'arc



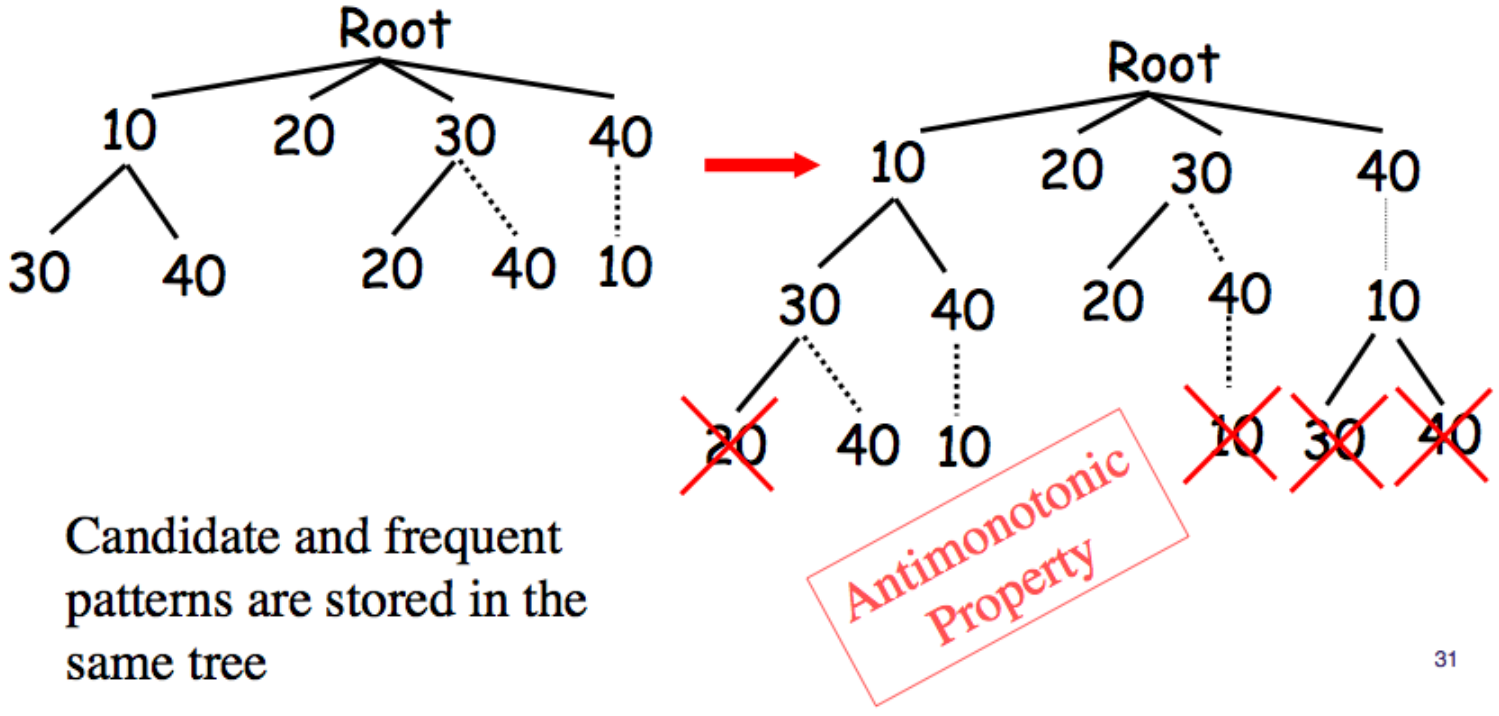


‹(30 50 40 10) (30) (20 10)›

PSP : génération des 2-candidats



PSP : génération des k-candidats ($k > 2$)



Candidate and frequent patterns are stored in the same tree

31

Algorithme SPADE

SPADE (Sequential PAttern Discovery using Equivalent Class) [Zaki 2001]

- représentation verticale de la base de données
- base de séquences transformée en un ensemble de triplet (item, SID,EID)
- Extraction de motifs est effectuée avec une génération de candidats basée sur Apriori

SPADE : pour un calcul du support plus efficaces

SID	EID	Items
1	1	a
1	2	abc
1	3	ac
1	4	d
1	5	cf
2	1	ad
2	2	c
2	3	bc
2	4	ae
3	1	ef
3	2	ab
3	3	df
3	4	c
3	5	b
4	1	e
4	2	g
4	3	af
4	4	c
4	5	b
4	6	c

a		b		...
SID	EID	SID	EID	...
1	1	1	2	
1	2	2	3	
1	3	3	2	
2	1	3	5	
2	4	4	5	
3	2			
4	3			

	ab			ba		...
SID	EID (a)	EID(b)	SID	EID (b)	EID(a)	...
1	1	2	1	2	3	
2	1	3	2	3	4	
3	2	5				
4	3	5				

	aba			...
SID	EID (a)	EID(b)	EID(a)	...
1	1	2	3	
2	1	3	4	

Problème des approches type générer/élaguer

- Une surgénération de candidats :
  - Pour 1000 1-séquences fréquentes, on génère $1000 \times 1000 \times \frac{1000 \times 999}{2} = 1,499,500$ 2 candidats
  - De multiples passages sur la bases de données
  - Approche en largeur sont coûteuse en gestion mémoire
- Pour extraire de longues séquences, ce type d'approche n'est pas adapté :
  - un nombre exponentiel de sous séquences candidates générées
  - pour une séquence de longueur 100 : $2^{100} - 1 \approx 10^{30}$

Approche « pattern growth »

- pas de génération de candidats
- extraction d'items fréquents sur des bases projetées
- approche gloutonne en profondeur

Préfixe et suffixe

- $\langle a \rangle$, $\langle aa \rangle$ et $\langle a(abc) \rangle$ sont des **préfixes** de la séquence $\langle a(abc)(ac)d(cf) \rangle$

Préfixe et suffixe

- $\langle a \rangle$, $\langle aa \rangle$ et $\langle a(abc) \rangle$ sont des **préfixes** de la séquence $\langle a(abc)(ac)d(cf) \rangle$
- Etant donnée la séquence $\langle a(abc)(ac)d(cf) \rangle$

Préfixe et suffixe

- $\langle a \rangle$, $\langle aa \rangle$ et $\langle a(abc) \rangle$ sont des **préfixes** de la séquence $\langle a(abc)(ac)d(cf) \rangle$
- Etant donnée la séquence $\langle a(abc)(ac)d(cf) \rangle$

Préfixe	Suffixe (Prefix-Based Projection)
<a>	<(abc)(ac)d(cf)>
<aa>	<(_bc)(ac)d(cf)>
<ab>	<(_c)(ac)d(cf)>

- Des items particuliers : (_b)

Extraction de motifs séquentiels par projection des préfixes

- Etape 1 : extraction des 1-séquences fréquentes :

$$\langle a \rangle, \langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle, \langle f \rangle$$

- Etape 2 : l'ensemble complet des motifs séquentiels peut être partitionné en 6 sous-ensembles :
  - ceux de préfixe $\langle a \rangle$,
  - ceux de préfixe $\langle b \rangle$,
  - ceux de préfixe $\langle c \rangle$,
  - ceux de préfixe $\langle d \rangle$,
  - ceux de préfixe $\langle e \rangle$,
  - ceux de préfixe $\langle f \rangle$.

Trouver les motifs séquentiels de préfixe $\langle a \rangle$

Simplement considérer les projections par rapport à $\langle a \rangle$:

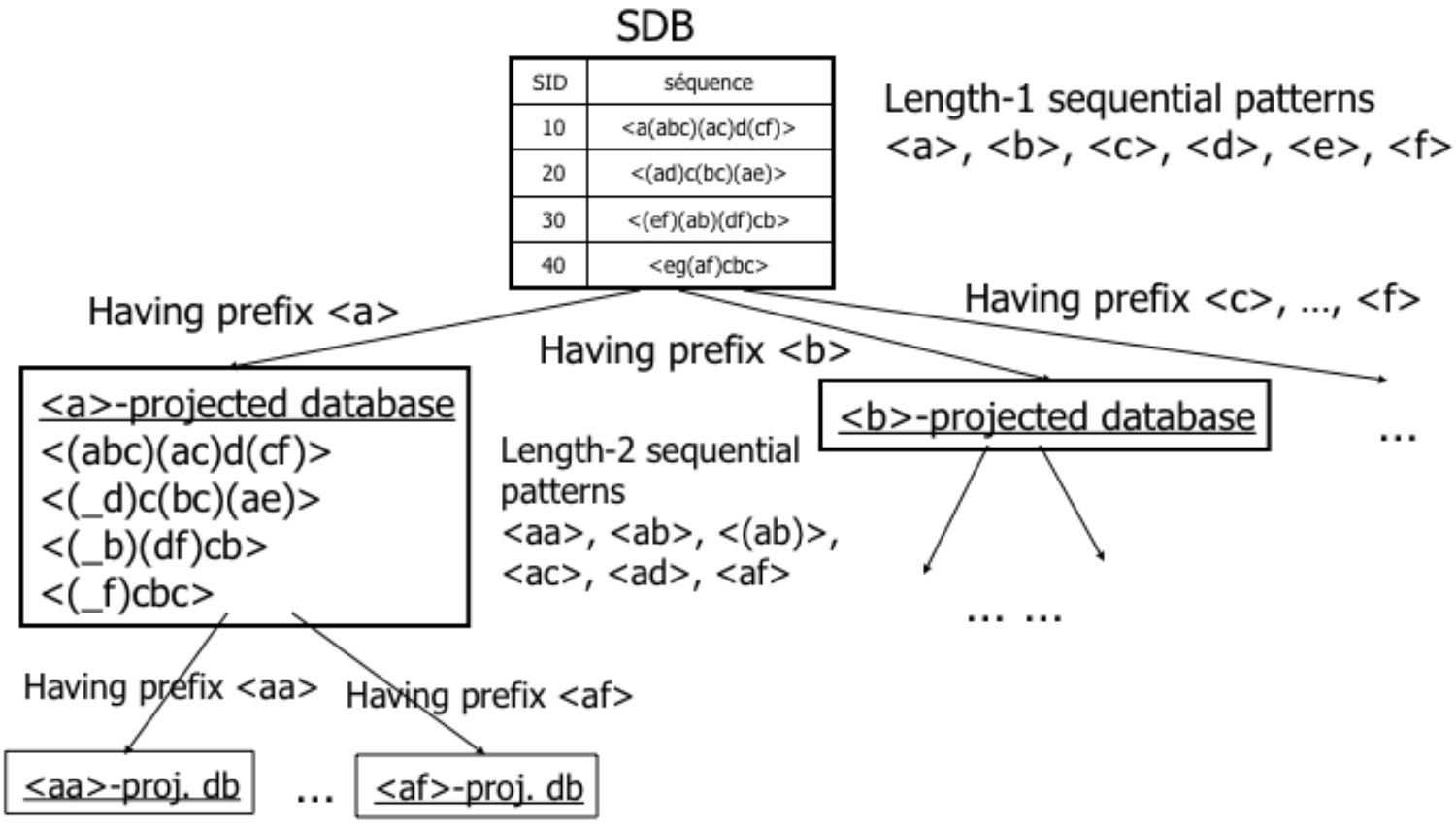
- $< (abc)(ac)d(cf) >$,
- $< (\_d)c(bc)(ae) >$,
- $< (\_b)(df)cb >$,
- $< (\_f)cbc >$

Motifs séquentiels de longueur 2 de préfixe $\langle a \rangle$:

- $< aa >$,
- $< ab >$,
- $< (ab) >$,
- $< ac >$,
- $< ad >$,
- $< af >$

Nouvelle partition en 6 ss-ensembles

Complétude de PrefixSpan

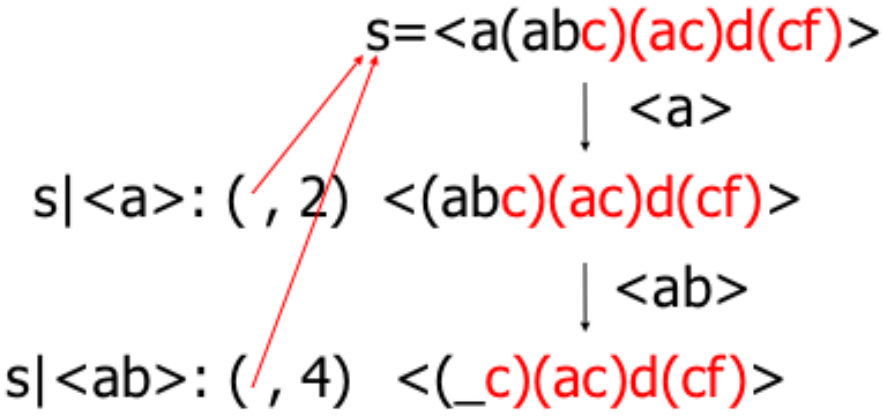


Efficacité de PrefixSpan

- Pas de génération de candidats
- La taille des bases projetées diminue sans cesse
- Le principal coût de PrefixSpan : Construire les bases projetées
  - Est amélioré à l'aide de **pseudo projections**

Pseudo projection

- Quand la base de données considérée peut être gérée en mémoire, utilisation de pointeurs pour les projections
- Pointeur sur les séquences
- Offset sur le suffixe



Pseudo projection vs. projection physique

- La pseudo projection évite de stocker physiquement les suffixes
  - Efficacité (temps d'exécution et espace mémoire) quand la base peut être stocker en mémoire
- Quid des bases qui ne peuvent pas être stockées en mémoire ?
  - Intégration de projections physiques et de pseudo projections
  - Utilisation de pseudo projections quand la base tient en mémoire (la taille des bases projetées diminuant)

Motifs séquentiels fermés

Définition ?

Motifs séquentiels fermés

Définition ?

Des motivations doubles

- Réduire le nombre de motifs (redondants) tout en maintenant la qualité de la connaissance extraite (pas de perte d'information)
- Améliorer l'extraction des motifs séquentiels en introduisant des propriétés d'élagages supplémentaires.

Motifs séquentiels fermés

Définition ?

Des motivations doubles

- Réduire le nombre de motifs (redondants) tout en maintenant la qualité de la connaissance extraite (pas de perte d'information)
- Améliorer l'extraction des motifs séquentiels en introduisant des propriétés d'élagages supplémentaires.

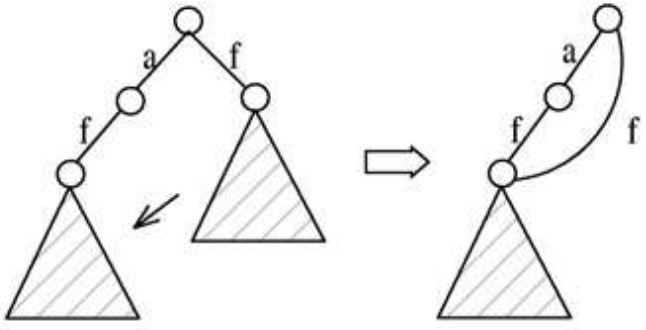
Deux approches :

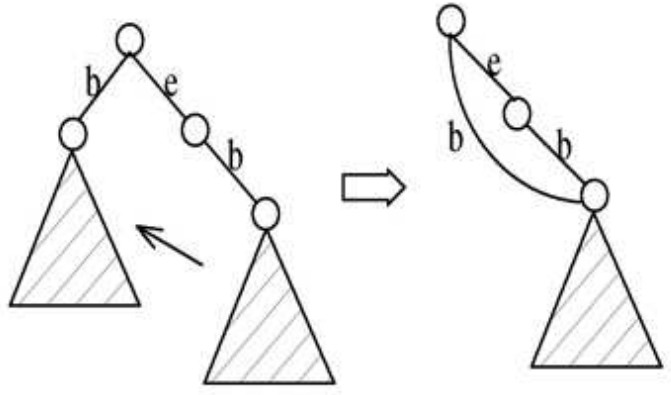
- CloSpan (Yan et al. 2003)
- Bide (Wang et al. 2007)

Clospan

Eviter de parcourir plusieurs fois les mêmes bases projetées

≪ *Using Backward Subpattern and Backward Superpattern pruning to prune redundant search space* ≫





Algorithme

Les limites de la gestion d'ensemble de candidats

- post-traitement : $O(n^2)$

Il faut éviter de gérer des candidats

BIDE : Idée Principale

Extension d'une g-k-séquence préfixe $\langle s_1, s_2, \ldots, s_g \rangle$:

BIDE : Idée Principale

Extension d'une g-k-séquence préfixe $\langle s_1, s_2, \ldots, s_g \rangle$:

1. vers l'avant inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$

BIDE : Idée Principale

Extension d'une g-k-séquence préfixe $\langle s_1, s_2, \ldots, s_g \rangle$:

1. vers l'avant inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$
2. vers l'avant intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\} \rangle$

BIDE : Idée Principale

Extension d'une g-k-séquence préfixe $\langle s_1, s_2, \ldots, s_g \rangle$:

1. vers l'avant inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$
2. vers l'avant intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\} \rangle$
3. vers l'arrière inter itemset $S' = \langle s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g \rangle$

BIDE : Idée Principale

Extension d'une g-k-séquence préfixe $\langle s_1, s_2, \ldots, s_g \rangle$:

1. vers l'avant inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$
2. vers l'avant intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\} \rangle$
3. vers l'arrière inter itemset $S' = \langle s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g \rangle$
4. vers l'arrière intra itemset $S' = \langle s_1, s_2, \ldots s_i \cup \{e'\}, s_{i+1}, \ldots, s_g \rangle$

BIDE : Idée Principale

Extension d'une g-k-séquence préfixe $\langle s_1, s_2, \ldots, s_g \rangle$:

1. vers l'avant inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\} \rangle$
2. vers l'avant intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\} \rangle$
3. vers l'arrière inter itemset $S' = \langle s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g \rangle$
4. vers l'arrière intra itemset $S' = \langle s_1, s_2, \ldots s_i \cup \{e'\}, s_{i+1}, \ldots, s_g \rangle$

Séquence Close :

- Si pas d'extension qui conserve le support de la séquence

Intervalles de $\langle s_1, s_2, \ldots, s_g \rangle$

Exhiber des items qui apparaissent tous les $i^{èmes}$ intervalles

$\underbrace{l_1}\, s_1, \underbrace{l_2}\, , s_2, \underbrace{l_3}\, , s_3, \ldots, s_{g-1} \underbrace{l_g}\, , s_g$

Intervalles de $\langle s_1, s_2, \ldots, s_g \rangle$

Exhiber des items qui apparaissent tous les $i^{èmes}$ intervalles

$\underbrace{l_1}\, s_1, \underbrace{l_2}, s_2, \underbrace{l_3}, s_3, \ldots, s_{g-1} \underbrace{l_g}, s_g$

Une séquence peut apparaître plusieurs fois dans une séquence de données

- Séquence $\langle (a,b)(a,c) \rangle$
- Séquence de données $\langle (a,b)(a,c)(a,b)(a,c)(a,b)(a,c)(a,b)(a,c) \rangle$

Intervalles de $\langle s_1, s_2, \ldots, s_g \rangle$

Exhiber des items qui apparaissent tous les $i^{èmes}$ intervalles

$\underbrace{l_1}\, s_1, \underbrace{l_2}, s_2, \underbrace{l_3}, s_3, \ldots, s_{g-1} \underbrace{l_g}, s_g$

Une séquence peut apparaître plusieurs fois dans une séquence de données

- Séquence $\langle (a,b)(a,c) \rangle$
- Séquence de données $\langle (a,b)(a,c)(a,b)(a,c)(a,b)(a,c)(a,b)(a,c) \rangle$

Maximiser ces intervalles

Intervalles de $\langle s_1, s_2, \ldots, s_g \rangle$

Exhiber des items qui apparaissent tous les $i^{èmes}$ intervalles

$\underbrace{I_1}\ s_1, \underbrace{I_2}, s_2, \underbrace{I_3}, s_3, \ldots, s_{g-1}\ \underbrace{I_g}, s_g$

Une séquence peut apparaître plusieurs fois dans une séquence de données

- Séquence $\langle (a,b)(a,c) \rangle$
- Séquence de données
  $\langle (a,b)(a,c)(a,b)(a,c)(a,b)(a,c)(a,b)(a,c) \rangle$

Maximiser ces intervalles

$I_1 : \langle (a,b)(a,c) \rangle$

$$\langle \underbrace{(a,b)(a,c)(a,b)(a,c)(a,b)(a,c)}\ (a,b)(a,c) \rangle$$

Intervalles de $\langle s_1, s_2, \ldots, s_g \rangle$

Exhiber des items qui apparaissent tous les $i^{èmes}$ intervalles

$\underbrace{l_1}\ s_1, \underbrace{l_2}, s_2, \underbrace{l_3}, s_3, \ldots, s_{g-1}\ \underbrace{l_g}, s_g$

Une séquence peut apparaître plusieurs fois dans une séquence de données

- Séquence $\langle (a,b)(a,c) \rangle$
- Séquence de données $\langle (a,b)(a,c)(a,b)(a,c)(a,b)(a,c)(a,b)(a,c) \rangle$

Maximiser ces intervalles

$l_2 : \langle (a,b)(a,c) \rangle$

$$\langle (a,b) \underbrace{(a,c)(a,b)(a,c)(a,b)(a,c)(a,b)} (a,c) \rangle$$

Les motifs séquentiels sous contraintes

Extraction de motifs sous contraintes :

Permet de surmonter deux difficultés majeures de l'extraction de motifs :

- améliorer les connaissances extraites
- améliorer le processus d'extraction

Les contraintes : plus proche de l'utilisateur

- Les contraintes peuvent permettre de modéliser l'intérêt de l'utilisateur.

Une large gamme de contraintes

- item constraint
- length constraint
- aggregate constraint
- etc.

Des contraintes spécifiques aux séquences

Des contraintes spécifiques aux séquences

- min-gap
- max-gap
- durée (window size)

Exemple

Au tableau

Algorithmes d'extraction de motifs séquentiels sous contraintes

- Spirit [Garofalakis et al, 02]
- [Pei et al, 02]

La prise en compte des hiérarchies

Motivations

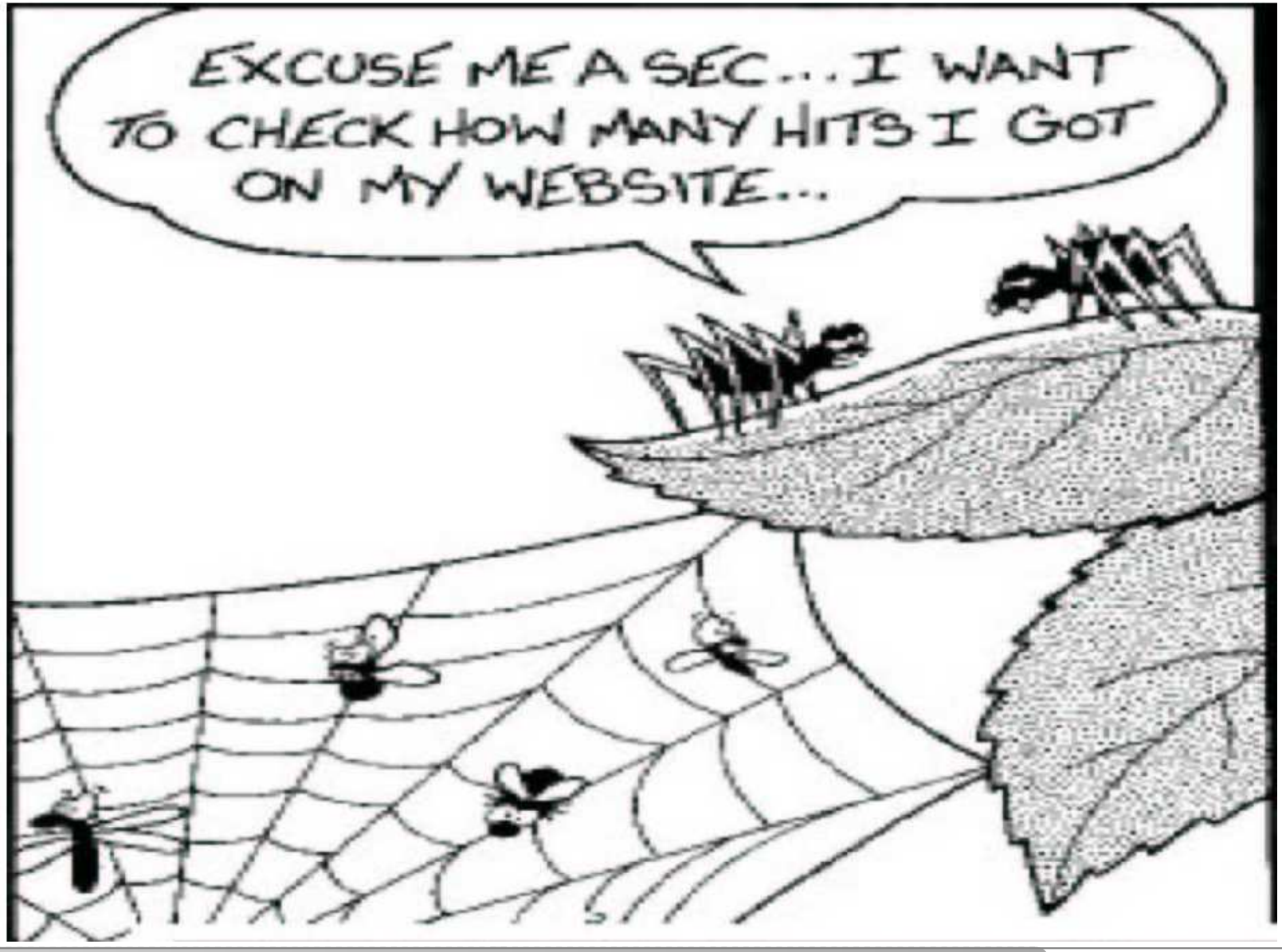
Difficulté de fixer le bon seuil de support :

- coca-cola non fréquent
- pepsi non fréquent
- MAIS : soda fréquent

Peu d'approches permettant la prise en compte des hiérarchies dans l'extraction de motifs séquentiels

Outline

Web Usage Mining



Web usage mining

La découverte de motifs intéressants sur la navigation des visiteurs à partir de logs d'un site web

- Les pages contiennent des informations
- Les liens sont des « routes »
- Comment les gens naviguent sur Internet ?
- WUM = clickstream analysis

Applications

Hyper liens dynamiques

- générer des hyper liens dynamiques en regardant la navigation du visiteur

Applications

Hyper liens dynamiques

- générer des hyper liens dynamiques en regardant la navigation du visiteur

Personnalisation

Fournir au visiteur ce qu'il veut sans le lui demander explicitement

- générer des recommandations pour chaque utilisateur pendant sa navigation

Applications

Hyper liens dynamiques

- générer des hyper liens dynamiques en regardant la navigation du visiteur

Personnalisation

Fournir au visiteur ce qu'il veut sans le lui demander explicitement

- générer des recommandations pour chaque utilisateur pendant sa navigation

Site web adaptatif

Changer le site web en fonction des motifs extraits

Applications (suite)

Amélioration de site web

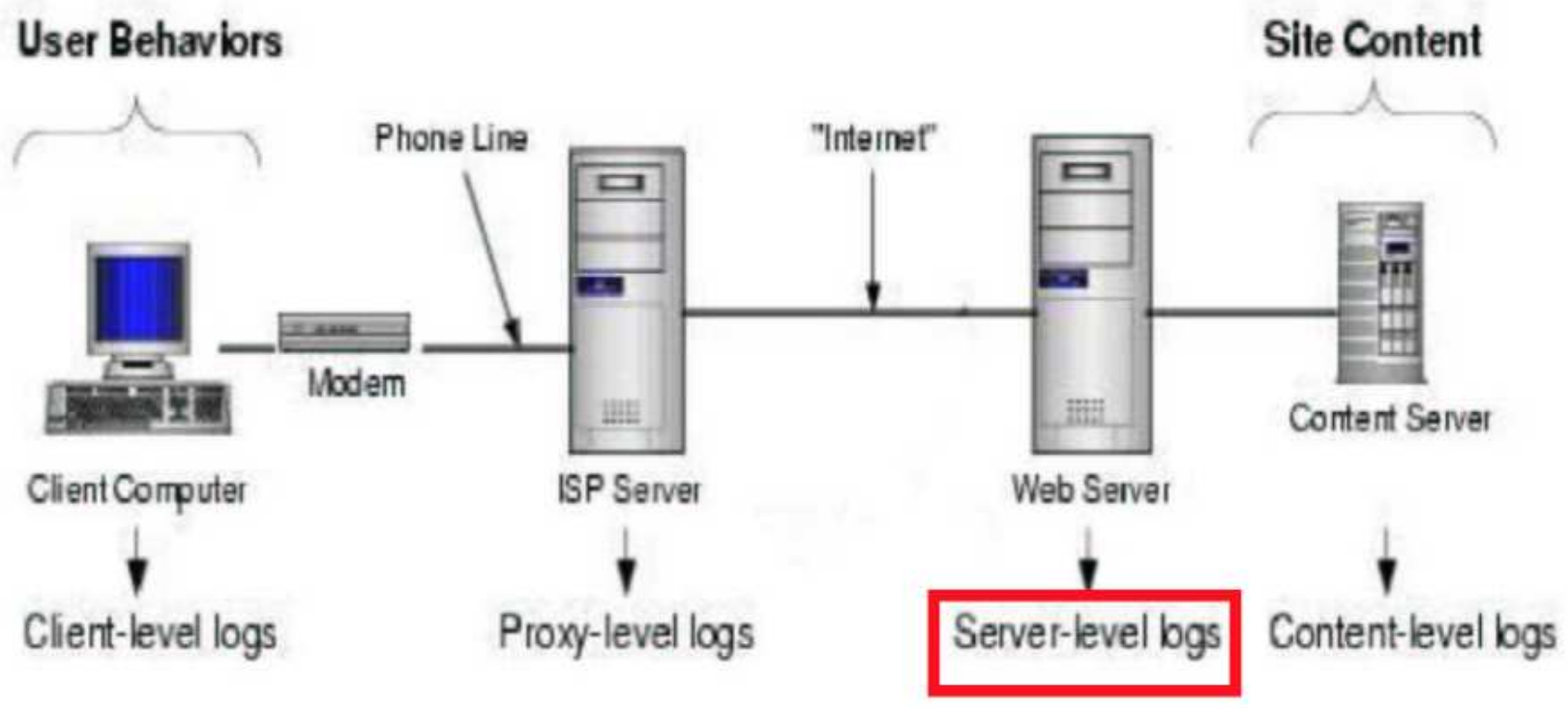
- comparer les caractéristiques des « clients »et des « non clients »
- changer le site afin que les « non clients » deviennent « clients »
- e.g. changer des hyper liens entre certaines pages

Web Usage Mining

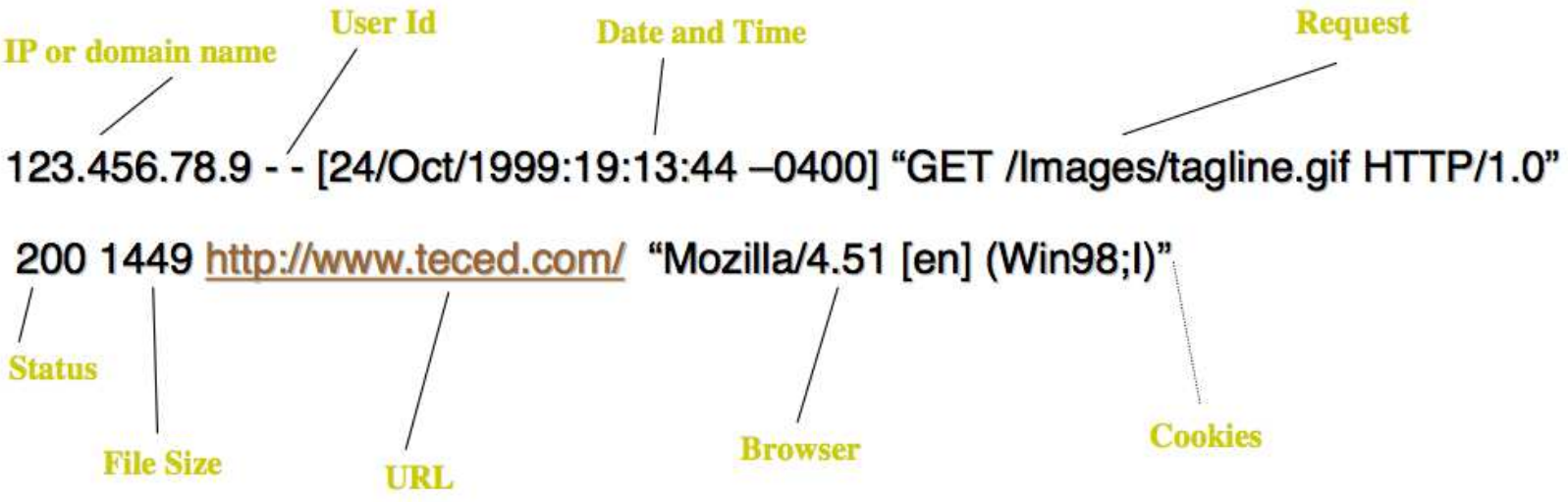
- 75% des Français achètent une raquette de tennis et moins de trois après, des chaussures de sports.
- hyper lien dynamique

Log ou logs

Informations sur les chemins de navigations sont disponibles dans des fichiers de logs



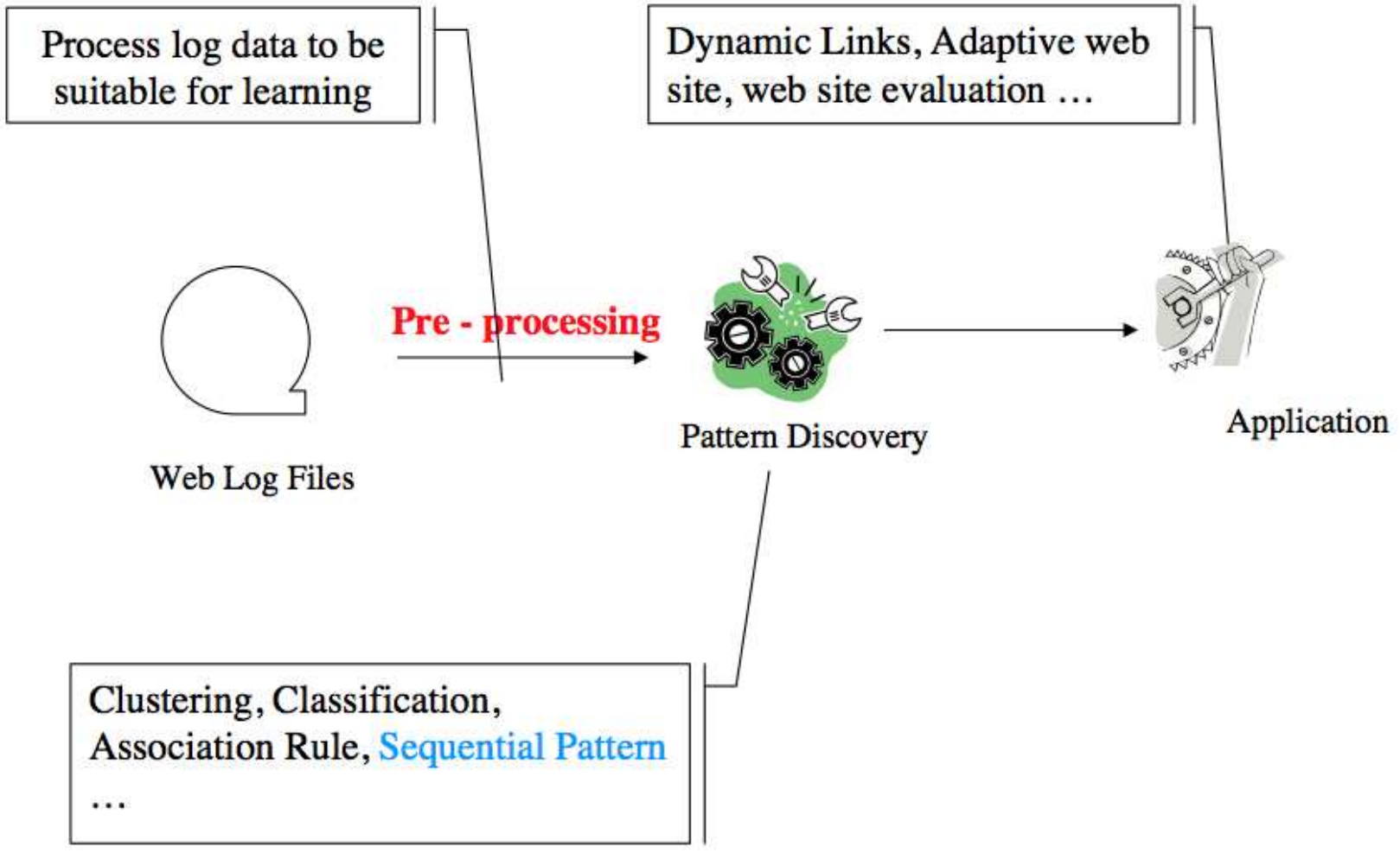
Web logs


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
URL
Browser
Cookies

Web logs

IP Address	Time	Method/URL/Protocol	Sta	Size	Referred	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)

KDD for WUM


Process log data to be suitable for learning
Dynamic Links, Adaptive web site, web site evaluation …
Pre - processing
Web Log Files
Pattern Discovery
Application
Clustering, Classification, Association Rule, Sequential Pattern …

Prétraitement des données

Filtrage, nettoyage

- Status code (1xx : Informational,2xx : Success,3xx : Redirection, 4xx : Client Error, 5xx : Server Error)
- Requêtes automatiques (bots, performance monitoring systems)
- Supprimer toutes les entrées représentant des requêtes pour des objets graphiques, etc.
- Supprimer les entrées générées par des spiders/crawlers (google, etc.)

Prétraitement des données

User Identification - Session Identification

- Comment identifier un utilisateur unique ? une transaction d'un utilisateur ?
- Pb :
  - Les identifiants des utilisateurs sont souvent supprimés par mesure de sécurité
  - Les adresses IP sont parfois cachées derrière des proxy
  - $\longrightarrow$ web logs pas toujours fiables

Prétraitement des données

User Identification - Session Identification

- Comment identifier un utilisateur unique ? une transaction d'un utilisateur ?
- Pb :
  - Les identifiants des utilisateurs sont souvent supprimés par mesure de sécurité
  - Les adresses IP sont parfois cachées derrière des proxy
  - $\longrightarrow$ web logs pas toujours fiables
- Solutions : identification du client, cookies côté client, etc.

Prétraitement des données

Identifier une session

Temps de navigation :

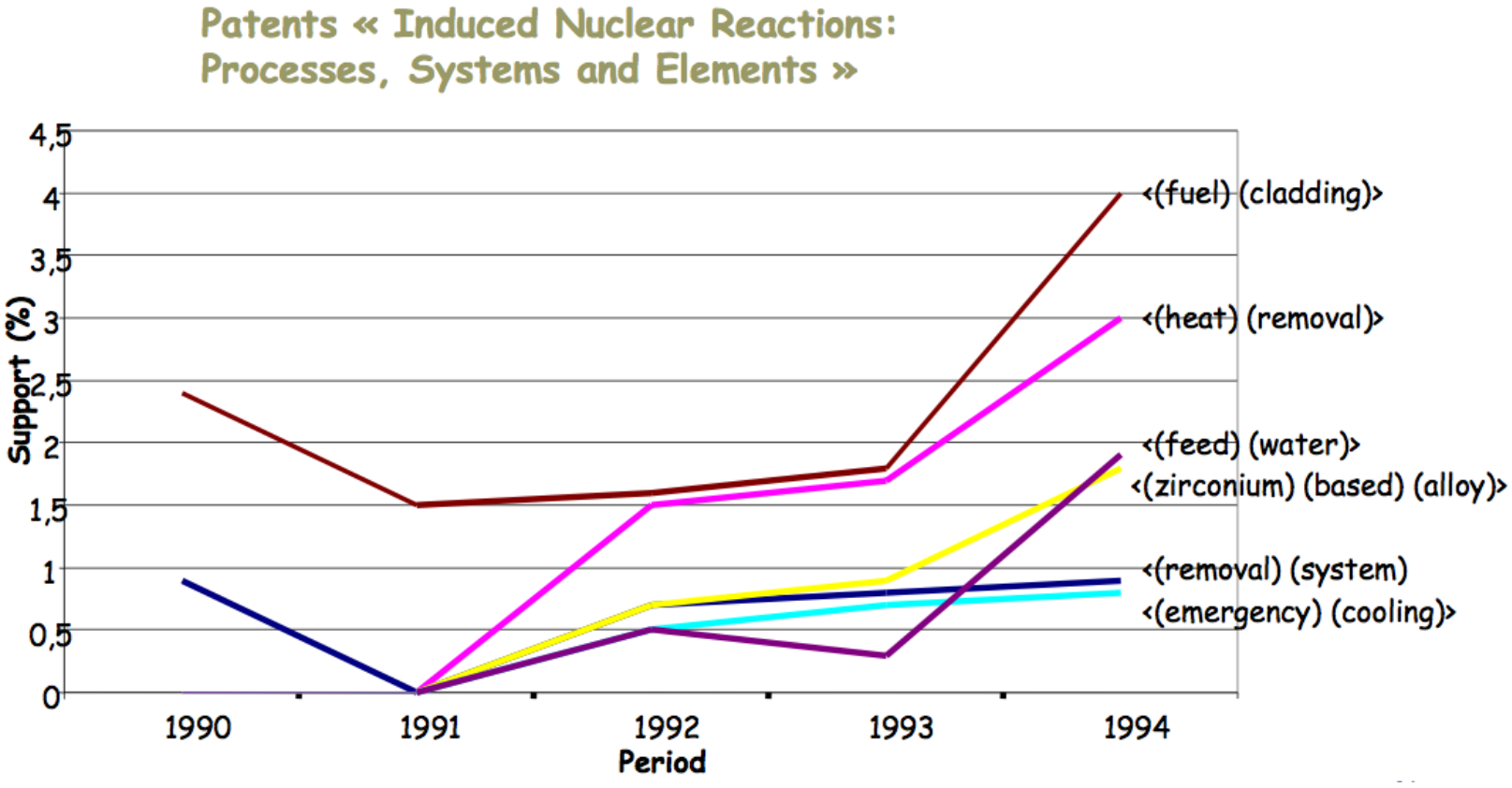
- Durée totale d'une session ($\leq$ 30 minutes)
- Durée par page ($\leq$ 10 minutes par page)

Web Usage Mining

Sources de données :

- fichiers logs
- mais aussi cookies, bd d'utilisateurs (clients)

Analyse de tendance avec des motifs séquentiels



Outline

Nouveaux challenges

- privacy preserving mining (k anonymous sequential patterns)
- d'autres types de motifs (arbres, graphes)
- motifs séquentiels inattendus
- motifs séquentiels dans les flux de données.