Cours sur l'Analyse de Données Séquentielles

Chapitre 2 : Fouille des motifs séquentiels dans les bases de données transactionnelles

Master 2
Systèmes d'Information, Optimisation et Décision

Dr D. AKROUR

2024/2025

Plan du cours

1. Introduction
2. Concepts de base
   - Séquence, sous-séquence, support, ...
3. Visualisation de séquences
   - i-plot, f-plot, d-plot
4. Caractéristiques des séquences
   - Longitudinale et transversale
5. Mesure de similarité entre séquences
   - Sac de caractère, P-spectrum, LCP, LCS
6. Fouille des motifs séquentiels
   - AprioriAll, GSP, SPADE
7. Motifs séquentiels avec contraintes
8. Analyse de périodicité dans les motifs séquentiels

Introduction

Introduction

- Les **bases de données transactionnelles**
  - Les données sont organisées sous forme de **transactions**,
  - Chaque transaction regroupe plusieurs items, souvent collectés **simultanément**.
- **Exemple**:
  - un panier d'achat constitue une transaction contenant les produits achetés,
  - dans les banques, chaque opération (retrait, dépôt, transfert) est une transaction distincte.
  - en logistique, chaque livraison ou commande est également une transaction, incluant des détails sur les produits expédiés.

Introduction

- Le problème de la recherche de motifs séquentiels peut être considéré comme l'analogue temporel de la recherche de motifs fréquents.
- En effet, les algorithmes de fouille de motifs fréquents peuvent souvent être adaptés à la recherche de motifs séquentiels, bien que celle-ci soit plus complexe.
- Bien que l'analyse des paniers d'achat ait initialement motivé ces recherches, les motifs séquentiels sont désormais appliqués dans une variété de domaines temporels.

Concepts de base

Concepts de base

- Séquence
- Longueur d’une séquence
- Taille d’une séquence
- Sous-séquence
- Base de données séquentielles
- Support d’une séquence
- Séquences fréquentes
- Séquences fréquentes fermées
- Séquences fréquentes maximales

Séquence

- Une **séquence** S est une liste ordonnée **d'éléments** disposés dans un ordre temporel spécifique.

$$S = < s_1, s_2, \dots s_n >$$

Séquence

- Une **séquence** S est une liste ordonnée **d'éléments** disposés dans un ordre temporel spécifique.

$$S = < s_1, s_2, \dots s_n >$$

- Chaque élément, appelé **itemset**, est un ensemble non ordonné d'**items**.

$$s_i = \{i_1, i_2, \dots i_m\}$$

Séquence

- Une **séquence** S est une liste ordonnée **d'éléments** disposés dans un ordre temporel spécifique.

$$S = < s_1, s_2, \dots s_n >$$

- Chaque élément, appelé **itemset**, est un ensemble non ordonné d'**items**.

$$s_i = \{i_1, i_2, \dots i_m\}$$

- Un **item** peut représenter un état, un événement, une transaction, un produit acheté, etc.

Séquence

- Une **séq** isposés dans ur

- Chaque n ordonné d'**items**

- Un **item** ne transaction, un produit acheté, etc.

Par exemple:

$S = <\{a, b\}, \{a, b, c\}, \{d\}>$

Ici, {a, b} est un itemset, et a est un item à l'intérieur de cet itemset.

Longueur d’une séquence

- La longueur correspond au **nombre total d'items** qu'elle contient.

Longueur d’une séquence

- La longueur correspond au **nombre total d'items** qu'elle contient.
- Une séquence S est appelée ***k-sequence*** si elle contient k items.

Longueur d’une séquence

- La longueur correspond au **nombre total d'items** qu'elle contient.
- Une séquence S est appelée ***k-sequence*** si elle contient k items.
- Par exemple:

La séquence S = <{a, b, c, e}, {b, e}, {a} >

est une ***7-sequence.***

Taille d’une séquence

- La taille d'une séquence fait référence au **nombre total d’itemsets** qu'elle contient.

Taille d’une séquence

- La taille d'une séquence fait référence au **nombre total d’itemsets** qu'elle contient.
- Par exemple:

$$\text{La séquence } S = <\{a, b, c, e\}, \{b, e\}, \{a\}>$$

$$\text{est de taille } 3$$

Sous-séquence

- Soit les deux séquences:
  - $\alpha = < a_1, a_2, \ldots a_n >$
  - $\beta = < b_1, b_2, \ldots b_m >$
- α est une sous-séquenc de β si et seulement si il existe des entiers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ tels que $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}$, ..., $a_n \subseteq b_{jn}$
- Exemple:
  - $< \{a, c\} >$ est une sous-séquence de $< \{a, b, c\} >$
  - $< \{a, c\} >$ n'est pas une sous-séquence de $< \{a\},\{c\} >$
  - $< \{a\},\{c\} >$ est une sous-séquence de $< \{a, b\}, \{d\}, \{b, c\} >$
  - $< \{a\},\{c\} >$ n'est pas une sous-séquence de $< \{a, c\}, \{d\}>$

Sous-séquence

- **Exercice** :
  - La séquence <a(bc)> est une sous-séquence de <ae(abcd)> ?
  - La séquence <a(bc)> est une sous-séquence de <abc> ?
  - La séquence <b(fg)> est une sous-séquence de <(ab)c(fg)ge> ?
  - La séquence <bgf> est une sous-séquence de <(ab)c(fg)ge> ?

Base de données séquentielle

- Une base de données séquentielles **BDS** est une liste de séquences ayant chacune un identifiant unique.

$$BDS = \{S1, S2. . ., Sk\}$$

SID	Séquence
1	<{a, b}, {c}, {f, g}, {g}, {e}>
2	<{a, d}, {c}, {b}, {a, b, e, f }>
3	<{a}, {b}, {f }, {e}>
4	<{b}, {f, g}>

Support d'une séquence

- Le support (fréquence) d'une séquence α représente le nombre (ou la proportion) de séquences dans la base de données SDB qui contiennent α.

SID	Séquence
1	<{a, b}, {c}, {f, g}, {g}, {e}>
2	<{a, d}, {c}, {b}, {a, b, e, f }>
3	<{a}, {b}, {f }, {e}>
4	<{b}, {f, g}>

- $\alpha = < \{b\}, \{g\} >$ apparaît trois fois dans deux séquences. Par conséquent, elle a un support de 2.

Support d'une séquence

- Le support (fréquence) d'une séquence α représente le nombre (ou la proportion) de séquences dans la base de données SDB qui contiennent α.

SID	Séquence
1	<{a, b}, {c}, {f, g}, {g}, {e}>
2	<{a, d}, {c}, {b}, {a, b, e, f }>
3	<{a}, {b}, {f }, {e}>
4	<{b}, {f, g}>

- $\alpha = < \{b\}, \{g\} >$ apparaît trois fois dans deux séquences. Par conséquent, elle a un support de 2.

Séquences fréquentes

- Une séquence fréquente est aussi appelée un motif séquentiel
- Avec un **support minimum (*minsup)***, la séquence S_i est fréquente dans une base de données séquentielles si et seulement si:

$$\boldsymbol{supp\ (S_i, SDB) \geqslant minsup}$$

- En d'autres termes, S_i n'est fréquente que si elle apparaît au moins ***minsup*** fois dans la base de données.

Séquences fréquentes

SID	Séquence
1	<{a, b}, {c}, {f, g}, {g}, {e}>
2	<{a, d}, {c}, {b}, {a, b, e, f }>
3	<{a}, {b}, {f }, {e}>
4	<{b}, {f, g}>

Avec ***minsup=3***

Les motifs séquentiels avec un support de 3 sont:

{<{a}>, <{a}, {f}>, <{a}, {e}>, <{b}, {g}>, <{b}, {f, g}>, <{b}, {e}>; <{e}>, <{f}, {g}>, <{g}> }

Les motifs séquentiels avec un support de 4 sont:

{<{b}>, <{b}, {f}>, <{f}>}

Séquence fréquente fermée

- C’est une séquence fréquente qui n'est incluse dans aucune autre séquence fréquente ayant le même support.
- Autrement dit, une séquence est fermée si elle est fréquente et si aucune autre séquence plus longue contenant cette séquence n'a le même support.

Séquence fréquente fermée

- C’est une séquence fréquente qui n'est incluse dans aucune autre séquence fréquente ayant le même support.
- Autrement dit, une séquence est fermée si elle est fréquente et si aucune autre séquence plus longue contenant cette séquence n'a le même support.
- Par exemple:
  - $\alpha = <$ {a}, {b} $>$ avec un support de 3,
  - $\beta = <$ {a}, {b}, {c}$>$ avec un support de 2,

α est une séquence fréquente fermée car β n'a pas le même support.

Séquences fréquentes maximales

- C’est une séquence fréquente qui n'est incluse dans aucune autre séquence fréquente.
- Autrement dit, c'est la séquence fréquente la plus longue possible, et il n'existe pas de séquence fréquente plus longue qui contienne cette séquence.

Séquences fréquentes maximales

- C’est une séquence fréquente qui n'est incluse dans aucune autre séquence fréquente.
- Autrement dit, c'est la séquence fréquente la plus longue possible, et il n'existe pas de séquence fréquente plus longue qui contienne cette séquence.
- Par exemple:

  Si $\alpha = < \{a\}, \{b\} >$ est fréquente et qu'il n'existe aucune séquence fréquente contenant α alors α est une séquence fréquente maximale.

Visualisation de séquences

—

Visualisation de séquences

- Pour visualiser les séquences catégorielles, trois graphiques de base sont utilisés:
  - le **i-plot**, le **f-plot** et le **d-plot.**
- Nous prenons pour exemple le jeux de données **mvad** utilisé pour étudier la transition école-travail.
- Ces données proviennent d'une enquête auprès de 712 jeunes irlandais et indiquent les états mensuels successifs de chaque individu entre septembre 1993 et juin 1999.
- Les états sont :
  - EM : en emploi
  - FE : formation secondaire
  - HE : formation supérieure
  - JL : au chômage
  - SC : école
  - TR : en stage ou apprentissage.

Visualisation de séquences

- Pour visualiser les séquences catégorielles, trois graphiques de base sont utilisés:
  - le **i-plot**, le **f-plot** et le **d-plot.**
- Nous prenons pour exemple le jeux de données **mvad** utilisé pour étudier la transition école-travail.
- Ces données proviennent d'une enquête auprès de 712 jeunes irlandais et indiquent les états mensuels successifs de chaque individu entre septembre 1993 et juin 1999.
- Les états sont :
  - EM : en emploi
  - FE : formation secondaire
  - HE : formation supérieure
  - JL : au chômage
  - SC : école
  - TR : en stage ou apprentissage.

Dans ce cas, les séquences sont des séquences d'états

Visualisation de séquences

- Pour visualiser les séquences catégorielles, trois graphiques de base sont utilisés:
  - le **i-plot**, le **f-plot** et le **d-plot.**
- Nous prenons pour exemple le jeux de données **mvad** utilisé pour étudier la transition école-travail.
- Ces données proviennent d'une enquête auprès de 712 jeunes irlandais et indiquent les états mensuels successifs de chaque individu entre septembre 1993 et juin 1999.
- Les états sont :
  - EM : en emploi
  - FE : formation secondaire
  - HE : formation supérieure
  - JL : au chômage
  - SC : école
  - TR : en stage ou apprentissage.

Dans ce cas, les séquences sont des séquences d'états

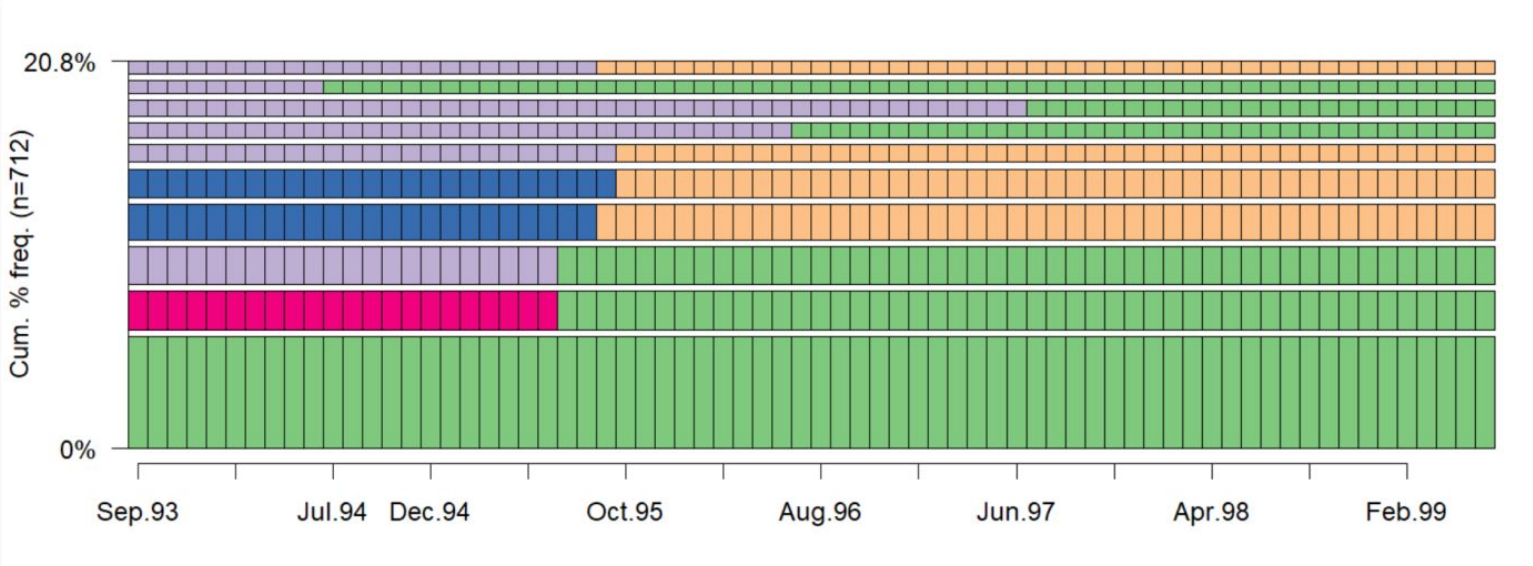
Chaque item représente donc un état qui peut durer dans le temps

Plot de séquences individuelles (i-plot)

- Le i-plot permet de visualiser chaque séquence par une barre horizontale en attribuant des couleurs spécifiques aux états.

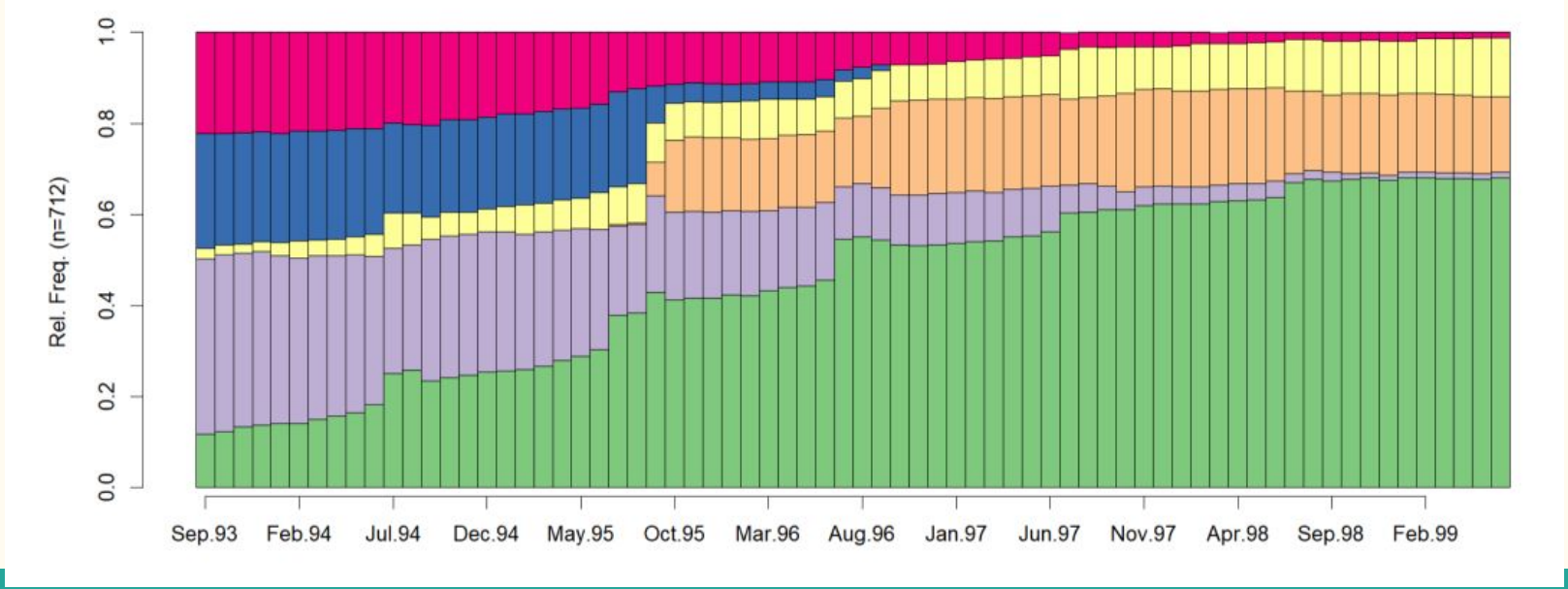
Plot de séquences selon la fréquence (f-plot)

- Le plot de séquences selon la fréquence (f-plot) est similaire au précédent mais permet de visualiser les séquences selon leurs fréquences déterminées par la hauteur de la barre.



Plot des séquences des distributions transversales (d-plot)

- Le plot des séquences des distributions transversales (d-plot) représente la distribution transversale des états à chaque position des séquences considérées

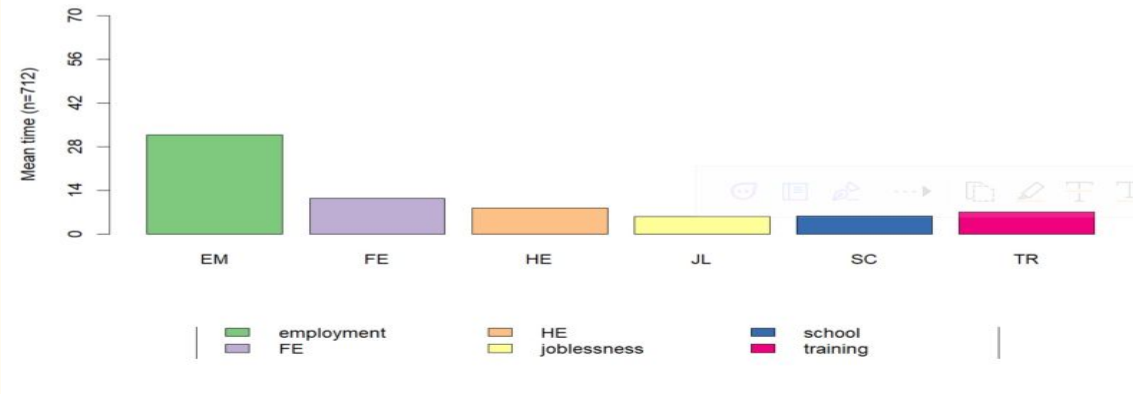


Caractéristiques des séquences

- Caractéristiques des séquences **individuelles** (**longitudinale**): observer et de comprendre les transitions, les états et les tendances d'une séquence au fil du temps.
  - Durée moyenne
  - Taux de transition
  - Entropie longitudinale
  - Turbulence
- Caractéristiques d'un **ensemble de séquences** (**transversale**): analyse d'un ensemble de séquences à un instant donné. Ici, les séquences sont étudiées de manière collective pour dégager des tendances globales, des modèles et des similarités entre elles.
  - Séquence des états modaux
  - Entropie transversale

Durée moyenne

- La durée moyenne (pas nécessairement consécutive) passée dans les différents états représente le nombre moyen de fois où chaque état est observé dans une séquence.
- Cette mesure donne un aperçu de la répartition générale des états au niveau des séquences.



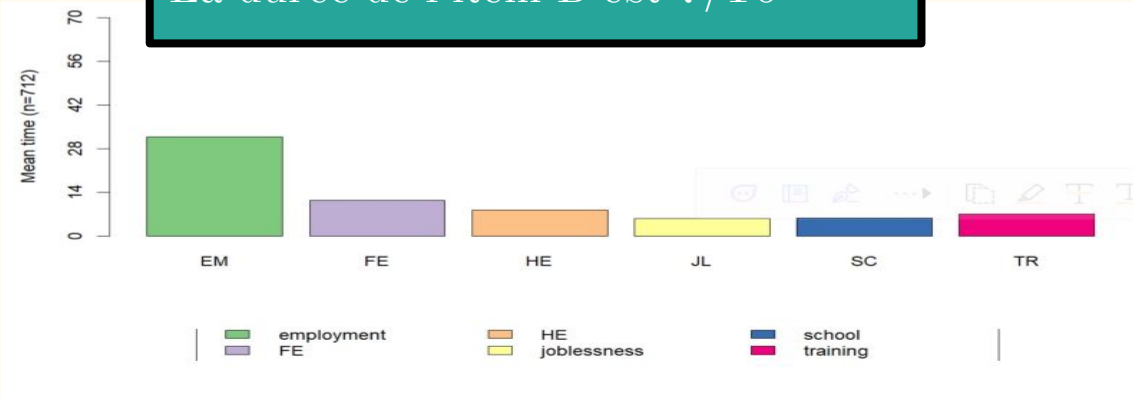
Durée moyenne

- La durée moyenne (pas nécessairement consécutive) passée dans les différents états r… où chaque état est observé dans une…
- Cette mesure do… rale des états au niveau des séquences.

Exemple

<AABBABBBBB>

La durée de l'item A est 3/10
La durée de l'item B est 7/10



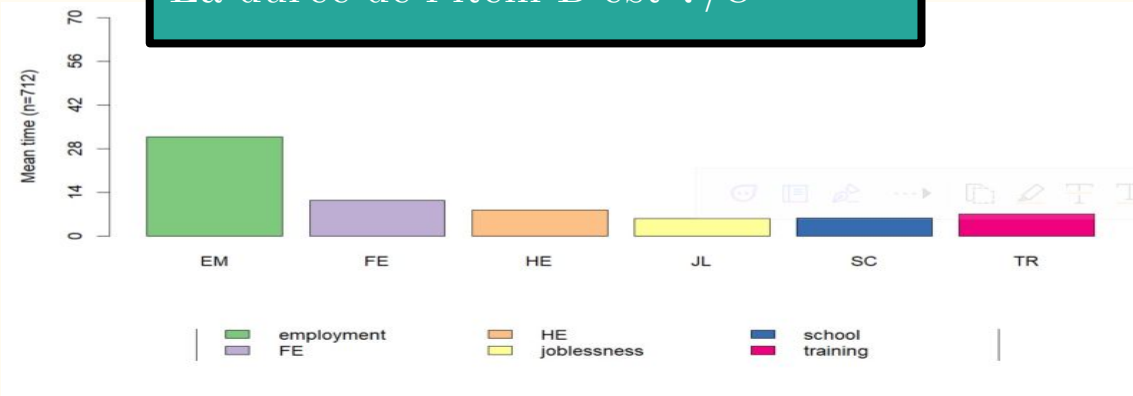
Durée moyenne

- La durée moyenne (pas nécessairement consécutive) passée dans les différents états r… où chaque état est observé dans une…
- Cette mesure do… rale des états au niveau des séquences.

Exemple

<A(AB)B(AB)BBBB>

La durée de l'item A est 3/8
La durée de l'item B est 7/8



Taux de transitions

- Elle représente l'estimation des probabilités de transition à une position donnée d'un état à l'autre pour une séquence donnée.
- Par exemple, le taux de transition pour les items de la séquence

S=<{A}, {B}, {B}, {A}, {B}, {B}, {B}>

est

	A	B
A	0	2/2
B	1/4	3/4

Entropie longitudinale

- Mesure la diversité.
- Elle est :
  - nulle lorsque la séquence garde le même item tout au long de la période d'observation (A-A-A-A-A-A-A par exemple),
  - maximale lorsque la séquence comprend un même nombre de chacun de ces items (par exemple A-A-B-B-C-C-D-D).

$$H(S) = \sum_{i=0}^{n} P(E_i) \log_2 P(E_i)$$

Turbulence

- La turbulence, contrairement à l'entropie, elle tient compte de l'ordre des états en plus de leurs occurrences.

$$T(x) = \log_2\left(\phi(x)\frac{s^2_{t,max}(x)+1}{s^2_t(x)+1}\right)$$

Turbulence

- La turbulence se base sur

  - $\phi(x)$: le **nombre de sous-séquences** distinctes contenues dans la suite des états distincts composant la séquence.

    Par exemple x = S − U − M − C (16 sous-squences : Φ; S; SU; SM; SC; SUM; SUC; SMC; SUMC; U; UM; UC; UMC; M; MC; C) plus turbulente que y = S − U − S − C (15 sous-squences)

  - $s_t^2(x)$: la **variance des durées passées** dans chacun des états distincts.

Turbulence

- La turbulence se base sur

- $s^2_{t,max}(x)$: la **valeur maximum** que peut atteindre la variance compte tenu de la dure totale de la séquence

$$s^2_{t,max}(x) = (n-1)(1-\bar{t})^2$$

Tel que **n** est la taille du vecteur des durées passés dans chaque état et **t barre** est la moyenne des durées consécutives passées dans les états distincts (durée de la séquence / nombre de ses états distincts)

Turbulence

- Exemple:

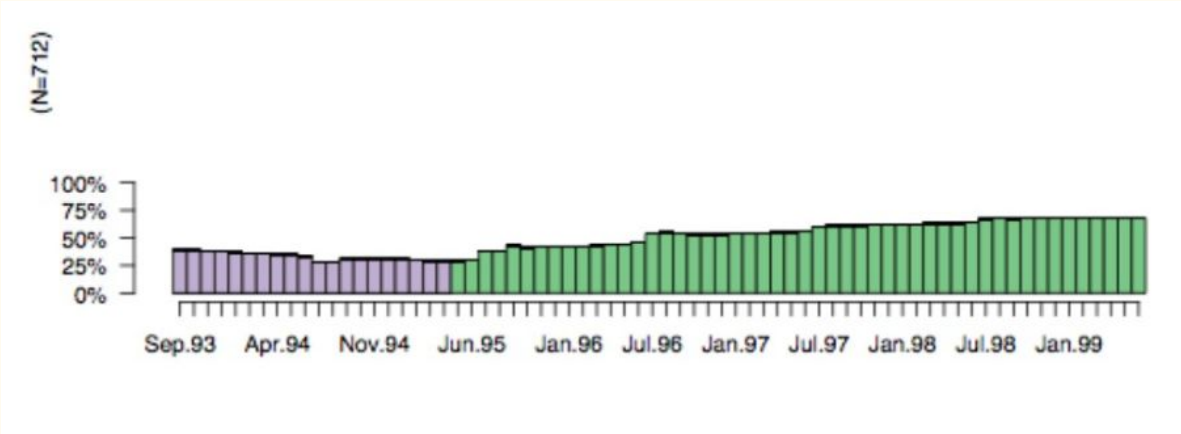
  S = <ABB>

  - Le nombre de sous séquences est 5 {ϕ, A, B, AB, BB}
  - Durée passée dans chaque état V={1, 2}

    - $s_t^2(x)=\frac{1}{2}((1-1.5)^2+(2-1.5)^2)=0.25$
    - $s_{t,max}^2(x)=(2-1)(1-\frac{3}{2})^2=0.25$
    - $T(x)=\log_2(5\frac{0.25+1}{0.25+1})=2$

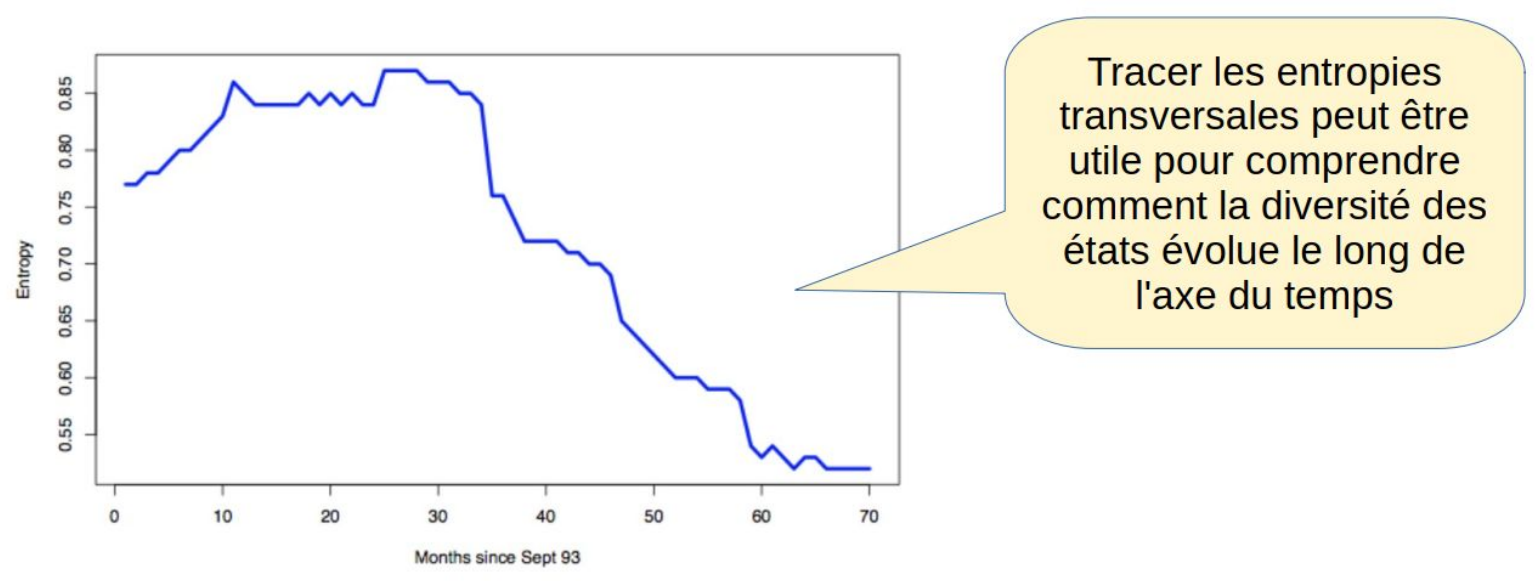
Séquence des états modaux

- représente l'item ou l'état qui se reproduit le plus fréquemment à chaque instant dans un ensemble de séquences.
- Elle sert de modèle de référence pour comparer les autres séquences.



Entropie transversale

- Représente l'entropie des séquence pour chaque état temporel



Mesure de similarité entre les séquences

- Les mesures de similarité ont une importance particulière pour les séquences parce qu'elles sont la base de plusieurs tâches effectuées dans ce domaine
  - Sac-de-caractère
  - p-Spectrum
  - LCP
  - LCS

Sac de caractère

- Ne prend pas en compte l'ordre des états,
- L'importance est plutôt donnée à la fréquence des états,
- On représente chaque séquence S par un vecteur V des fréquences d'apparition des états de l'alphabet dans cette séquence.

$$D(S_1, S_2) = \frac{\sum_{k=1}^{m} V_{1k} V_{2k}}{\sqrt{(\sum_{k=1}^{m} V_{1k}^2)(\sum_{k=1}^{m} V_{2k}^2)}}$$

Sac de caractère

Exemple :

- S1 = <NNRCRNRCDD>
- S2 = <NNBCNNNDDD>

- Représentation en sac de caractères pour l'alphabet

{N, R, C, D, B}

- V1 ={3, 3, 2, 2, 0}
- V2 ={5, 0, 1, 3, 1}

- Calcul de la distance $$D(S1,S2)=\frac{(5*3)+(2*1)+(2*3)}{\sqrt{(3^2+3^2+2^2+2^2)(5^2+1^2+3^2+1^2)}}$$

p-spectrum

- Elle permet de calculer le nombre de sous-séquences contiguës que peuvent avoir deux séquences en commun.
- Cette distance tire son principe du fait que plus le nombre de sous-séquences communes entre deux séquences est important, moins est la distance entre ces deux séquences
- La distance compte toutes les sous-séquences de longueur p dans les deux séquences puis calcule la somme des produits.

- **Exemple** : la distance 2-Spectrum entre ces deux séquences est **7**
  - VVVRCCCVRV
  - VRCRCVVCR

VV	VR	VC	RV	RR	RC	CV	CR	CC
2	2	0	1	0	1	1	0	2
1	1	1	0	0	2	1	2	0

Longest Common Prefix (LCP)

- Le LCP donne le plus long préfixe commun entre deux séquences.
- Le LLCP est sa longueur.
- La distance est donne par :

$$D_{LCP}(x,y) = 1 - \frac{LLCP(x,y)}{\sqrt{|x|.|y|}}$$

Où |x| et |y| sont les longueurs des séquences x et y

Longest Common Subsequence (LCP)

- Cette mesure calcule la taille des sous-séquences partagées par 2 séquences.
- Par exemple, dans les séquences
  - <agbfcgdhei>
  - <afbgchdiei>
- La plus longue sous-séquence commune est <afghei>.
- LLCS(S1, S2) = 6,
- et donc la distance égale à 1-610.10=0.4

Longest Common Subsequence (LCP)

- Comment trouvez la sous-séquence commune?

Soit deux séquences $X = (x_1 \dots x_m)$ et $Y = (y_1 \dots y_n)$. On définit $LCS(i, j)$ comme la longueur de la plus longue sous-séquence commune entre les segments X_i (les i premiers symboles de X) et X_j (les j premiers symboles de Y).

La fonction $LCS(i, j)$ peut être exprimée récursivement comme suit :

$$LCS(i, j) = max \begin{cases} LCS(i-1, j-1) + 1 & si\ x_i = y_j \\ LCS(i-1, j) & sinon \\ LCS(i, j-1) & sinon \end{cases} \quad (8)$$

Les conditions aux limites sont que $LCS(i, 0) = 0$ et $LCS(0, j) = 0$, car une séquence vide ne peut avoir de sous-séquence commune avec une autre séquence.

Longest Common Subsequence (LCP)

- Comment trouvez la sous-séquence commune?

	a	g	b	f	c	g	d	h	e	i
a	1	1	1	1	1	1	1	1	1	1
f	1	1	1	2	2	2	2	2	2	2
b	1	1	2	2	2	2	2	2	2	2
g	1	2	2	2	2	3	3	3	3	3
c	1	2	2	2	3	3	3	3	3	3
h	1	2	2	2	3	3	3	4	4	4
d	1	2	2	2	3	3	4	4	4	4
i	1	2	2	2	3	3	4	4	4	5
e	1	2	2	2	3	3	4	4	5	5
i	1	2	2	2	3	3	4	4	5	6

Fouille de motifs séquentiels

- La fouille de motifs séquentiels consiste à extraire l'ensemble des sous-séquences qui sont fréquentes dans une séquence ou un ensemble de séquences.
- Plus précisément, le problème se résume à identifier toutes les sous-séquences αi dans une base de données séquentielle SDB telles que leurs supports supp(αi, SDB)>= minsup.

Fouille de motifs séquentiels

- **Complexité?**
- La fouille de motifs séquentiels est une tâche complexe pour plusieurs raisons.
  - Les contraintes temporelles entraînent un ensemble potentiellement immense de séquences candidates.
  - La fouille de motifs séquentiels longs présente d'autres défis, notamment la difficulté d'extraire le grand nombre de motifs potentiels cachés, comme dans l'analyse des séquences d'ADN ou la prédiction de séquences boursières.

Fouille de motifs séquentiels

- **Challenge?**
- L'approche naïve, consistant à calculer le support de toutes les sous-séquences possibles, est inefficace.
- Il est donc essentiel de concevoir des algorithmes capables de réduire l'espace de recherche et d'identifier l'ensemble complet des motifs séquentiels tout en étant performants, évolutifs et en limitant le nombre d'analyses de la base de données.
- Ces algorithmes doivent également intégrer des contraintes spécifiques à l'utilisateur et inclure des techniques de "pruning" (ou élagage) pour éliminer rapidement les candidats non pertinents, rendant ainsi le processus plus efficace.

Fouille de motifs séquentiels

- **Trois algorithme:**
  - **Apriori-All**
  - **GSP**
  - **SPADE**

AprioriAll

- L'algorithme AprioriAll est une adaptation de l'algorithme de base Apriori pour les séquences, où la génération des candidats et le calcul de support sont faits d'une manière différentes.
- **Avantages**:
  - Il est le premier algorithme de fouille séquentielle, constituant une base pour les développements ultérieurs.
  - Implémente un processus bien structuré pour gérer les motifs séquentiels.
- **Inconvénients**:
  - Nécessite plusieurs passages sur la base de données, ce qui le rend lent et inefficace pour des bases de données volumineuses.
  - Génère un grand nombre de candidats et augmentant la complexité de calcul.

AprioriAll

Méthode:

1. **Tri des données :** La base de données transactionnelle est triée par customer-id et transaction-time et mappée en une base de données séquentielle.
2. **Extraction des itemsets fréquents :** La base séquentielle est scannée et ensuite toutes les sous-séquences fréquentes de taille 1 (Large-itemsets) sont identifiées selon un seuil de support minimal prédéfini. Les Large-itemsets sont ensuite mappés à des entiers pour rendre la fouille plus efficace.
3. **Transformation** : Les séquences de la base séquentielle sont remplacées par les Large-itemsets qu'elles contiennent (avec les entiers mappés) tout en éliminant les itemsets non fréquents.
4. **Génération des séquences fréquentes** : Les motifs séquentiels fréquents sont générés en utilisant l'algorithme AprioriAll.
5. **Pruning** : Les motifs séquentiels maximaux sont retenus et ceux contenus dans d'autres motifs sont supprimés, car seuls les motifs maximaux sont d'intérêt. (Une recherche basée sur un arbre de hachage)

AprioriAll

Algorithme AprioriAll

1. $L_1 \leftarrow \{générer\ les\ séquences\ fréquentes\ de\ taille\ 1\ (Large\ -\ itemsets)\}$;
2. $K \leftarrow 2$;
   TantQue $L_k \neq \phi$ ***faire***
   $C_k \leftarrow générerCandidats\ (L_{k-1})$;
   Pour *chaque séquence S dans la base de séquences* ***faire***
3. *Incrémenter les compteurs pour tous les candidats dans* C_k *contenus dans S*
   Fin Pour
4. $L_k \leftarrow \{candidats\ dans\ C_k\ ayant\ le\ support\ minimum\}$;
   $K \leftarrow K + 1$;
5. ***Fin TantQue***

AprioriAll

Algorithme AprioriAll

1. $L_1 \leftarrow \{$*générer les séquences fréquentes de taille* 1 $(Large - itemsets)\}$;
2. $K \leftarrow 2$;
   TantQue $L_k \neq \phi$ ***faire***
   $C_k \leftarrow$ *générerCandidats* (L_{k-1});
3. ***Pour*** *chaque séquence S dans la base de séquences* ***faire***
   Incrémenter les compteurs pour tous les candidats dans C_k *contenus dans S*
   Fin Pour
4. $L_k \leftarrow \{$*candidats dans* C_k *ayant le support minimum*$\}$;
   $K \leftarrow K + 1$;
5. ***Fin TantQue***

AprioriAll

Algorithme AprioriAll

1. $L_1 \leftarrow \{génére$
2. $K \leftarrow 2;$
   TantQue $L_k \neq$
   $C_k \leftarrow g$
   Pour
3. $ontenus\ dans\ S$
   Fin Po
4. $L_k \leftarrow \{$
   $K \leftarrow K + 1;$
5. ***Fin TantQue***

Procédure $générerCandidats\ (L_{k-1})$

1. **Jointure:** Lier les séquences de L_{k-1} entre elles avec la condition d'avoir les mêmes sous-séquences préfixes de longueur $K - 2$;
2. **Élagage (Pruning):** Supprimer toutes les séquences candidates ayant des sous-séquences non fréquentes;

AprioriAll

Algorithme AprioriAll

1. $L_1 \leftarrow \{génér$
2. $K \leftarrow 2;$
   TantQue L_k
   $C_k \leftarrow$
   Pour
3. ntenus dans S
   Fin Po
4. $L_k \leftarrow \{$
   $K \leftarrow K + 1;$
5. ***Fin TantQue***

candidates ayant des sous-séquences non fréquentes;

Par exemple, si <{1}, {2}, {3}> et <{1}, {2}, {4}> sont deux séquences fréquentes de longueur 3, en appliquant la jointure, on remarque que les deux séquences possèdent la même sous-séquence préfixe de longueur 2. Ainsi, les séquences candidates de longueur 4 sont générées comme suit : <{1}, {2}, {3}, {4}> et <{1}, {2}, {3}, {4}>.

AprioriAll

Exemple :

Considérons la base transactionnelle suivante et supposons un support minimum de 40% (0.4):

Transaction Time	Customer Id	Items Bought
June 10 '93	2	10, 20
June 12 '93	5	90
June 15 '93	2	30
June 20 '93	2	40, 60, 70
June 25 '93	4	30
June 25 '93	3	30, 50, 70
June 25 '93	1	30
June 30 '93	1	90
June 30 '93	4	40, 70
July 25 '93	4	90

AprioriAll

Etape 1:

Customer Id	TransactionTime	Items Bought
1	June 25 '93	30
1	June 30 '93	90
2	June 10 '93	10, 20
2	June 15 '93	30
2	June 20 '93	40, 60, 70
3	June 25 '93	30, 50, 70
4	June 25 '93	30
4	June 30 '93	40, 70
4	July 25 '93	90
5	June 12 '93	90

Customer Id	Customer Sequence
1	⟨(30) (90)⟩
2	⟨(10 20) (30) (40 60 70)⟩
3	⟨(30 50 70)⟩
4	⟨(30) (40 70) (90)⟩
5	⟨(90)⟩

AprioriAll

Etape 2:

1-Itemset	support	1-Itemset	support
(30)	4	(40, 60)	1
(90)	3	(40, 70)	2
(10, 20)	1	(60, 70)	1
(10)	1	(50)	1
(20)	1	(30, 50)	1
(40)	2	(30, 70)	1
(60)	1	(50, 70)	1
(70)	3	(30, 50,70)	1

AprioriAll

Etape 2:

Les séquences fréquentes

{ (30), (90), (40), (70), (40, 70) }

1-Itemset	support	1-Itemset	support
(30)	4	(40, 60)	1
(90)	3	(40, 70)	2
(10, 20)	1	(60, 70)	1
(10)	1	(50)	1
(20)	1	(30, 50)	1
(40)	2	(30, 70)	1
(60)	1	(50, 70)	1
(70)	3	(30, 50,70)	1

AprioriAll

Etape 2:

Les séquences fréquentes

{ (30), (90), (40), (70), (40, 70) }

1-Itemset	Mapped to
(30)	1
(40)	2
(70)	3
(40, 70)	4
(90)	5

1-Itemset	support	1-Itemset	support
(30)	4	(40, 60)	1
(90)	3	(40, 70)	2
(10, 20)	1	(60, 70)	1
(10)	1	(50)	1
(20)	1	(30, 50)	1
(40)	2	(30, 70)	1
(60)	1	(50, 70)	1
(70)	3	(30, 50,70)	1

AprioriAll

Etape 3:

Customer Id	Original Customer Sequence	Transformed Customer Sequence	After Mapping
1	⟨(30) (90)⟩	⟨{(30)} {(90)}⟩	⟨{1} {5}⟩
2	⟨(10 20) (30) (40 60 70)⟩	⟨{(30)} {(40), (70), (40 70)}⟩	⟨{1} {2, 3, 4}⟩
3	⟨(30 50 70)⟩	⟨{(30), (70)}⟩	⟨{1, 3}⟩
4	⟨(30) (40 70) (90)⟩	⟨{(30)} {(40), (70), (40 70)} {(90)}⟩	⟨{1} {2, 3, 4} {5}⟩
5	⟨(90)⟩	⟨{(90)}⟩	⟨{5}⟩

AprioriAll

Etape 4: générer les motifs candidats de taille 2 à partir des motifs séquentiels de taille 1

C2	supp	C2	supp	C2	supp
11	0	31	0	51	0
12	2	32	0	52	0
13	2	33	0	53	0
14	2	34	0	54	0
15	2	35	1	55	0
21	0	41	0		
22	0	42	0		
23	0	43	0		
24	0	44	0		
25	1	45	1		

AprioriAll

Etape 4: générer les motifs candidats de taille 2 à partir des motifs séquentiels de taille 1

c2	supp	c2	supp	c2	supp
11	0	31	0	51	0
12	2	32	0	52	0
13	2	33	0	53	0
14					0
15					0
21					
22					
23	0	43	0		
24	0	44	0		
25	1	45	1		

Les motifs séquentiels de taille 2 sont: L2={12, 13, 14, 15}

AprioriAll

Etape 4: générer les motifs candidats de taille 3 à partir des motifs séquentiels de taille 2

L2	c3 après jointure	c3 aprés élagage
12	123	/
	132	
	124	
	142	
	152	
	125	
13	.....	

AprioriAll

Etape 4: générer les motifs candidats de taille 3 à partir des motifs séquentiels de taille 2

L2	c3 après jointure	c3 aprés élagage
12	123	/
	132	
	124	
	142	
	152	
	125	
13	.....	

Aucun motif séquentiel de taille 3:
L3=ensemble vide

AprioriAll

Etape 4:

- Les séquences fréquentes sont :

(1,2,3,4,5,12,13,14,15)

- c’est à dire :

{ (30), (40), (70), (40 70), (90), { (30) (40) }, { (30) (70) }, { (30) (40 70) }, { (30) (90) } }

AprioriAll

Etape 5:

- Les motifs séquentiels maximums sont:

{ { (30) (40 70) }, { (30) (90) } }

GSP

- GSP (Generalized Sequential Patterns) a été l’une des premières approches proposées pour résoudre le problème de la fouille de motifs séquentiels. Il s'agit d'un algorithme de recherche en largeur qui, pour générer des séquences de longueur k, s'appuie sur les séquences fréquentes de longueur k-1.

GSP

- **Avantages:**
  - Réduit l'espace de recherche par l'introduction de contraintes temporelles.
  - Génère moins de séquences candidates que AprioriAll.
- **Inconvénients:**
  - Nécessite de multiples passages sur la base de données, ce qui peut être inefficace pour de grandes bases de données.
  - Génère un grand nombre de candidats qui ne sont pas présents dans la base de données, ce qui fait perdre beaucoup de temps de calcul.
  - Consomme beaucoup de mémoire car il garde toutes les séquences de longueur k en mémoire afin de pouvoir générer des modèles de longueur k+1.

GSP

Algorithme GSP

$L_1 \leftarrow \{générer\ les\ séquences\ 1\ -\ sequence\}$
$K \leftarrow 2;$
TantQue $L_k \neq \phi$ ***faire***
    $C_k \leftarrow générerCandidats\ (L_{k-1});$
    Pour *chaque séquence S dans la base de séquences* ***faire***
        Incrémenter les compteurs pour tous les candidats dans C_k *contenus dans S*
    Fin Pour
    $L_k \leftarrow \{candidats\ dans\ C_k\ ayant\ le\ support\ minimum\};$
    $K \leftarrow K + 1;$
Fin TantQue

GSP

Algorithme GSP

$L_1 \leftarrow \{générer\ les\ séquences\ 1\ -\ sequence\}$

$K \leftarrow 2;$

TantQue $L_k \neq \phi$ ***faire***

$C_k \leftarrow générerCandidats\ (L_{k-1});$

Pour *chaque séquence S dans la base de séquences* ***faire***

Incrémenter les compteurs pour tous les candidats dans C_k *contenus dans S*

Fin Pour

$L_k \leftarrow \{candidats\ dans\ C_k\ ayant\ le\ support\ minimum\};$

$K \leftarrow K + 1;$

Fin TantQue

GSP

La procédure générerCandidats (), utilisée pour générer des candidats de longueur k+1, diffère de celle de l'algorithme AprioriAll dans la phase de jointure. En effet, la jointure est effectuée en combinant des paires de motifs séquentiels de longueur k qui partagent tous leurs éléments à l'exception d'un seul.

Algori

$L_1 \leftarrow$ {

$K \leftarrow 2$

TantQ

Pour *chaque séquence S dans la base de séquences* ***faire***

Incrémenter les compteurs pour tous les candidats dans C_k *contenus dans* S

Fin Pour

$L_k \leftarrow \{candidats\ dans\ C_k\ ayant\ le\ support\ minimum\};$

$K \leftarrow K + 1;$

Fin TantQue

GSP

Exemple

SID	Sequence
1	$\langle\{a,b\},\{c\},\{f,g\},\{g\},\{e\}\rangle$
2	$\langle\{a,d\},\{c\},\{b\},\{a,b,e,f\}\rangle$
3	$\langle\{a\},\{b\},\{f,g\},\{e\}$
4	$\langle\{b\},\{f,g\}\rangle$

GSP

Exemple

- Candidat (item) de longueur 1

SID	Sequence
1	$\langle\{a,b\},\{c\},\{f,g\},\{g\},\{e\}\rangle$
2	$\langle\{a,d\},\{c\},\{b\},\{a,b,e,f\}\rangle$
3	$\langle\{a\},\{b\},\{f,g\},\{e\}$
4	$\langle\{b\},\{f,g\}\rangle$

→

1 - seq	fréquence
a	3
b	4
c	2
d	1
e	3
f	4
g	3

GSP

Exemple

Générer les séquences candidates de longueur 2 en utilisant la méthode de jointure et élagage

GSP

Exe

Gér… …éthode de …

1 - seq
a
b
c
d x
e
f
g

2 - seq	supp
ab	2
ac	2
ae	3
af	3
ag	2
(ab)	2
(ac)	0
(ae)	1
(af)	1
(ag)	0

2 - seq	supp
ba	1
bc	1
be	3
bf	4
bg	3
(bc)	0
(be)	1
(bf)	1
(bg)	1

2 - seq	supp
ca	1
cb	1
ce	2
cf	2
cg	1
(ce)	0
(cf)	0
(cg)	0

2 - seq	supp
ea	0
eb	0
ec	0
ef	0
eg	0
(ef)	2
(eg)	1

2 - seq	supp
fa	0
fb	0
fc	0
fe	2
fg	1
(fg)	3

2 - seq	supp
ga	0
gb	0
gc	0
ge	2
gf	0

GSP

Exemple

Générer les séquences candidates de longueur 3 en utilisant la méthode de jointure et élagage

2 - seq	2 - seq
ab	bg
ac	ce
ae	cf
af	(ef)
ag	fe
(ab)	(fg)
be	ge
bf	

GSP

Exemple

G équences candidates de longueur 3 en utilisant la méthode de élagage

2 - seq	2 - seq
ab	bg
ac	ce
ae	cf
af	(ef)
ag	fe
(ab)	(fg)
be	ge
bf	

ab+ac= abc et acb
ab+(ab)=(ab)b
ab+be=abe
ab+ce=jointure impossible
ab+(ef)=jointure impossible
(ef)+(fg)=jointure impossible
.....

GSP

Exemple

G équences candidates de longueur 3 en utilisant la méthode de élagage

2 - seq	2 - seq
ab	bg
ac	ce
ae	cf
af	(ef)
ag	fe
(ab)	(fg)
be	ge
bf	

Elagage:
on peut supprimer abc et acb car ils contiennent une sous-séquence non fréquente

ab+ac= abc et acb
ab+(ab)=(ab)b
ab+be=abe
ab+ce=jointure impossible
ab+(ef)=jointure impossible
(ef)+(fg)=jointure impossible
.....

Elagage:
on peut garder et calculer le supp de abe car toutes ses sous-séquence sont fréquentes

GSP

Exemple

G équences candidates de longueur 3 en utilisant la méthode de élagage

2 - seq	2 - seq
ab	bg
ac	ce
ae	cf
af	(ef)
ag	fe
(ab)	(fg)
be	ge
bf	

Les motifs séquentiels de longueur 3 (après jointure et élagage) sont:

L3 {<abe}>, <abf>, <ace>, <acf>, <afe>, <age>, <a{f,g}>, <bfe>, <b{f,g}>, <{f,g}e>, <bge>}.

GSP

Exemple

Générer les séquences candidates de longueur 4 en utilisant la méthode de jointure et élagage

GSP

Exemple

Générer les séq tilisant la méthode de jointure et é

L_3	C_4 Après jointure	C_4 Après élagage	supp
<a,b,e>	<a, b, e, f> <a, b, f, e> <a, b, c, e> <a, c, b, e> <a, f, b, e> <a, b, g, e>	<a, b, f, e>	1
<a,b,f>	<a, b, c, f> <a, c, b, f> <a, b, (f,g)>	<a, c, b, f>	1
<a,c,e>	<a, c, e, f> <a, c, f, e> <a, f, c, e> <a, c, g, e> <a, g, c, e>	/	/
<a,c,f>	<a, c, (f,g)>	/	/
<a,f,e>	<b, a, f, e> <a, (f,g)e>	<a, (f,g)e>	2
<b,g,e>	<b, a, g, e>		
<a,(f,g)>	<b, (f,g),e> <b, a, (f,g)>	<b, (f,g),e>	2
<b,f,e>	<b, f, g, e> <b, g, f, e>	/	/
<b,(f,g)>	/	/	/
<(f,g),e>	/	/	/
<a,g,e>	/	/	/

GSP

Exemple

Générer les séq tilisant la méthode de jointure et é

L_3	C_4 Après jointure	C_4 Après élagage	supp
<a,b,e>	<a, b, e, f> <a, b, f, e> <a, b, c, e> <a, c, b, e> <a, f, b, e> <a, b, g, e>	<a, b, f, e>	1
<a,b,f>	<a, b, c, f> <a, c, b, f> <a, b, (f,g)>	<a, c, b, f>	1
<a,c,e>	<a, c, e, f> <a, c, f, e> <a, f, c, e> <a, c, g, e>	/	/
<a,c,f>			/
<a,f,e>			2
<b,g,e>	<b, a, g, e>		
<a,(f,g)>	<b, (f,g),e> <b, a, (f,g)>	<b, (f,g),e>	2
<b,f,e>	<b, f, g, e> <b, g, f, e>	/	/
<b,(f,g)>	/	/	/
<(f,g),e>	/	/	/
<a,g,e>	/	/	/

Les motifs séquentiels de longueur 4 sont : L_3={ <a, (f, g), e>, <b, (f, g), e> }.

SPADE

- SPADE (Sequential PAttern Discovery using Equivalence classes) est basé sur l'extraction de sous-séquences en utilisant un format de données vertical de la base de données. Il transforme la base de données transactionnelle en une structure de ID-lists, permettant une exploration efficace des motifs séquentiels par des jointures simples.
- **Avantages** :
    - SPADE est plus rapide que AprioriAll et GSP grâce à son utilisation des ID-lists pour calculer le support.
    - Il réduit le nombre de passages sur la base de données à un seul, nécessaire pour créer les ID-lists des 1-séquences.
- **Inconvénients** :
    - Nécessite de stocker les ID-lists en mémoire, ce qui peut être coûteux en termes de mémoire pour de très grandes bases de données.

SPADE

- **Étapes Algorithmiques :**
  - **Transformation en ID-lists :** La base de données est convertie en format vertical, associant chaque item à une liste de paires (sid, eid), où sid est l'identifiant de la séquence dans laquelle il apparaît et eid est l'horodatage de l'événement correspondant.
  - **Génération des séquences fréquentes :** Les K-séquences sont formées en combinant des paires de motifs séquentiels de longueur K-1 qui partagent le même identifiant de séquence et respectent l'ordre séquentiel des événements ou d'itemsets.
  - **Exploration de l'espace des motifs :** L'exploration s'effectue en largeur (BFS) ou en profondeur (DFS) :
    - Recherche en largeur (BFS) : On explore toutes les séquences d'une même longueur (par ex. les 2-séquences), puis on passe aux séquences plus longues, niveau par niveau.
    - Recherche en profondeur (DFS) : On prolonge chaque séquence au maximum pour trouver les plus longues, puis on revient en arrière pour explorer les autres chemins.

SPADE

- **Exemple**

SID	Séquence
1	<{a, b}, {c}, {f, g}, {g}, {e}>
2	<{a, d}, {c}, {b}, {a, b, e, f }>
3	<{a}, {b}, {f }, {e}>
4	<{b}, {f, g}>

a	
SID	Itemsets
1	1
2	1
2	4
3	1

b	
SID	Itemsets
1	1
2	3
2	4
3	2
4	1

c	
SID	Itemsets
1	2
2	2

d	
SID	Itemsets
2	1

e	
SID	Itemsets
1	5
2	4
3	4

f	
SID	Itemsets
1	3
2	4
3	3
4	2

g	
SID	Itemsets
1	3
1	4
3	3
4	2

transformer la base en une représentation verticale avec des ID-lists pour chaque item

SPADE

- **Exemple**

SID	Séquence
1	<{a, b}, {c}, {f,
2	<{a, d}, {c}, {b}
3	<{a}, {b}, {f }, {e
4	<{b}, {f, g}>

Les motifs séquentiels de longueur 1 sont ensuite définis. Le support correspond au nombre de séquences distinctes dans lesquelles les motifs apparaissent. Avec un support minimum égal à 2 les motifs sont L1={a, b, c, e, f}

d	
SID	Itemsets
2	1

2	4
3	4

2	4
3	3
4	2

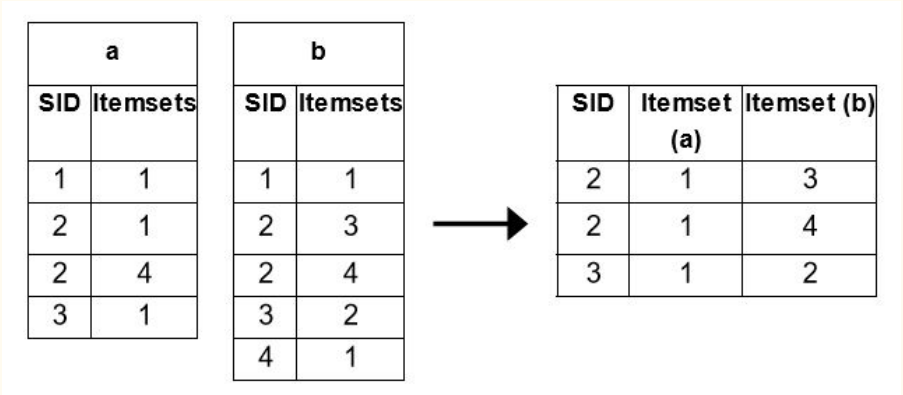
g	
	Itemsets
	3
1	4
3	3
4	2

transformer la base en une représentation verticale avec des ID-lists pour chaque item

SPADE

- **Exemple**

Pour trouver des séquences candidates 2-fréquentes, nous devons joindre les ID-lists des items qui partagent le même identifiant de séquence et identifiant d' itemset, tout en suivant le même ordre séquentiel:



a	
SID	Itemsets
1	1
2	1
2	4
3	1

b	
SID	Itemsets
1	1
2	3
2	4
3	2
4	1

SID	Itemset (a)	Itemset (b)
2	1	3
2	1	4
3	1	2

SPADE

- **Exemple**

Pour trouver des séquences candidates 2-fréquentes, nous devons joindre les ID-lists des items qui partagent le même [illegible] suivant le même ordre séquentiel:

Il faut continuer à joindre et à élaguer les séquences de manière récursive et ascendante (BFS) jusqu'à ce qu'aucun candidat ne respecte le seuil de support minimum, ou que toutes les longueurs de séquence possibles aient été explorées.

2	4
3	1

2	4
3	2
4	1

3	1	2

Fouille des motifs séquentiels sous contraintes

- La découverte de tous les motifs séquentiels possibles peut générer une grande quantité de résultats souvent non exploitables.
- Il est donc essentiel d'introduire des contraintes pour orienter la recherche vers des motifs plus significatifs et pertinents.
- Ainsi, l’utilisateur peut définir des contraintes pour focaliser l'analyse sur des motifs spécifiques d'intérêt tout en réduisant l’espace de recherche et en améliorant l'efficacité des algorithmes de fouille.

Fouille des motifs séquentiels sous contraintes

- **Types de contraintes:**
  - **Contraintes de longueur:** Limitation de la longueur minimale ou maximale des motifs extraits. Par exemple, un utilisateur peut être intéressé uniquement par des motifs contenant au moins 20 items.
  - **Contraintes de gap et de span**: Imposition d'une contrainte sur le temps écoulé entre deux événements consécutifs dans une séquence (gap). Par exemple, en analyse de séquences web, on peut fixer une durée maximale de 20 minutes entre deux pages visitées. De plus, une contrainte peut être imposée sur la durée totale d'un motif (span).
  - **Contraintes de super-motif :** Recherche de motifs incluant une séquence spécifique, comme l'achat de l'article X suivi de l'achat de l'article Y.

Fouille des motifs séquentiels sous contraintes

- **Types de contraintes:**
  - **Contraintes d'inclusion et exclusion d’items :** Filtrage des motifs contenant ou excluant des éléments spécifiques. Par exemple, en analyse de panier d'achat, un motif peut inclure l'achat d'un produit particulier.
  - **Contraintes d'expression régulière :** Utilisées pour détecter des motifs qui respectent un schéma particulier, comme des séquences de navigation sur des sites web. Par exemple, une séquence peut commencer par la page d'accueil Google et se terminer par une recherche d'hôtel à Biskra.
  - **Contrainte d’agrégat:** Cette contrainte repose sur des agrégats statistiques comme la somme ou la moyenne des items dans un motif, par exemple, trouvez des motifs où le prix moyen des articles est supérieur à 100 $.

Fouille des motifs séquentiels sous contraintes

- **Introduction des contraintes:**
  - **Lors de la génération des candidats** : Les contraintes (de longueur, de gap, de span, etc.) sont utilisées pour filtrer les motifs potentiels avant même de les explorer. Par exemple, lors de la génération des séquences candidates de longueur k, seules celles respectant les contraintes définies par l'utilisateur sont considérées.
  - **Lors du calcul du support :** Une fois les séquences candidates générées, leur support est calculé en tenant compte des contraintes. Par exemple, pour les contraintes de gap, seuls les motifs respectant les intervalles temporels spécifiés entre les événements seront retenus.
  - **Filtrage final des motifs :** Après avoir calculé le support des candidats, ceux qui ne respectent pas les contraintes finales (comme un support minimal ou des conditions d'expression régulière) sont éliminés.

Analyse de périodicité pour les données séquentielles temporelles

Les motifs périodiques sont des schémas récurrents dans les bases de données séquentielles temporelles, avec une périodicité quotidienne, hebdomadaire, mensuelle, saisonnière ou annuelle.

- Catégories de l'extraction des motifs périodiques
  - **Extraction de motifs périodiques complets** : Chaque point de la base de séquentielle temporelle contribue au comportement cyclique. Par exemple, chaque jour de la semaine participe au cycle hebdomadaire.
  - **Extraction de motifs périodiques partiels** : Seuls certains points de la base contribuent au comportement périodique. Par exemple, Mary regarde des films tous les vendredis de 19 h à 21 h.
  - **Extraction de règles d'association périodiques** : Identifie des règles associant des événements récurrents. Par exemple, si un restaurant reçoit de nombreux clients pour le thé entre 15h00 et 17h00, les ventes du dîner seront bonnes entre 20h00 et 21h00.

Analyse de périodicité pour les données séquentielles temporelles

Les motifs périodiques sont des schémas récurrents dans les bases de données séquentielles temporelles, avec une périodicité quotidienne, hebdomadaire, mensuelle, saisonnière ou annuelle.

- Approches d’extraction de motifs périodiques
  - Replier les événements dans des fenêtres de taille appropriée et en trouvant des sous-séquences fréquentes dans ces fenêtres.
  - Dériver des motifs d'ordre partiel en assouplissant l'exigence d'un ordre séquentiel strict lors de l'extraction de motifs de sous-séquences.

Exercices

Exercice 1

1. Quels sont les principaux avantages et inconvénients de l'algorithme GSP de fouille de motifs séquentiels ?
2. Quelle est la principale différence entre les deux algorithmes GSP et SPADE ?
3. Expliquer pourquoi la tâche de détection de motifs séquentiels dans la fouille de données séquentielle est un problème difficile.
4. Quelle est la solution pour faire face à cette difficulté ?

Exercices

Exercice 2

Soit la base de données suivante représentant des événements générés par divers capteurs triés par rapport au temps.

Capteur	pas de temps	évènement
S1	1 2 3 4	A, B C D, E C
S2	1 2 3	A, B C, D E
S3	1 2	B A
	3 4	B D, E
S4	1 2 3 4	C D, E C E
S5	1 2 3 4	B A B, C A, D

Exercices

Exercice 2

1. Transformer la base de données en une base de données séquentielle.
2. Calculer pour le capteur S3 la matrice des taux de transitions.
3. Calculer et comparer les entropies longitudinales des séquences de données des capteurs S1 et S2.
4. En utilisant l'algorithme AprioriAll et un support minimum de 45 %, trouver tous les motifs fréquents séquentiels.

Exercices

Exercice 3

Soit la base de données séquentielle suivante représentant l'historique des achats des clients:

ID séquence	Séquence
1	< (1,5) (2) (3) (4) >
2	< (1) (3) (4) (3,5) >
3	< (1) (2) (3) (4) >
4	< (1) (3) (5) >
5	< (4) (5) >

Exercices

Exercice 3

1. Calculer et comparer les turbulences des séquences d'achats des clients trois et quatre.
2. Générer une seule matrice des taux de transitions comportant les probabilités de transition à une position donnée d'un item à l'autre pour toutes les séquences du tableau.
3. Appliquer l'algorithme GSP à l'ensemble des données du tableau en utilisant un support minimum s = 33 % pour déterminer toutes les séquences fréquentes.
4. En déduire les motifs fréquents maximums.