



DECSAI

Departamento de Ciencias de la Computación e I.A.

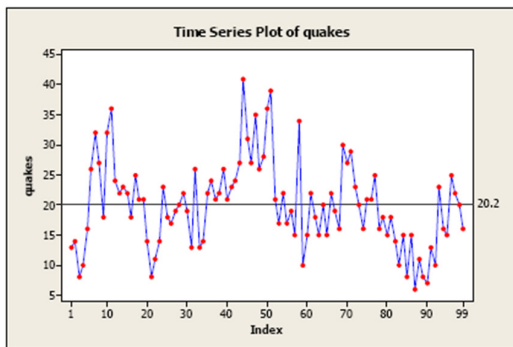
Universidad de Granada



Secuencias

© Fernando Berzal, berzal@acm.org

Datos secuenciales



Start

Human: GTTTTGAGG --- ATGTTCAACAAATGCTCC TTTTCATTCC TCTATTACAGACC TGCCGCA
 Chimpanzee: GTTTTGAGG --- ATGTTCAATAAATGCTCC TTTTCAC TCC TCTATTACAGACC TGCCGCA
 Macaque: GTTTTGAGG --- ATGCTCAATAAATGCTCC TTTTCATTCC TCCATTACAAACT TGCCGCA

Human: GACAAATTCGCTAGCAGCC TTTGTGCTATTATCTGTTTTCTAAACTTAGTAATTTGAGTGT
 Chimpanzee: GACAAATTCGCTAGCAGCC TTTGTGCTATTATCTGTTTTCTAAACTTAGTAATTTGAGTGT
 Macaque: GACAAATTCGCTAGCAGCC TTTGTGCTATTATCTGTTTTCTAAACTTAGTAATTTGAGTGT

Human: GATCTGGAGACTAACTGCTGAAATAAATAAGCTGATTATTTATTTATTTTCTCAAAAACAA
 Chimpanzee: GATCTGGAGACTAAACTGCTGAAATAAATAAGCTGATTATTTATTTATTTTCTCAAAAACAA
 Macaque: TATCTGGAGACTAAACTGCTGAAATAAATAAGCTGATTATTTATTTATTTTCTCAAAAACAA

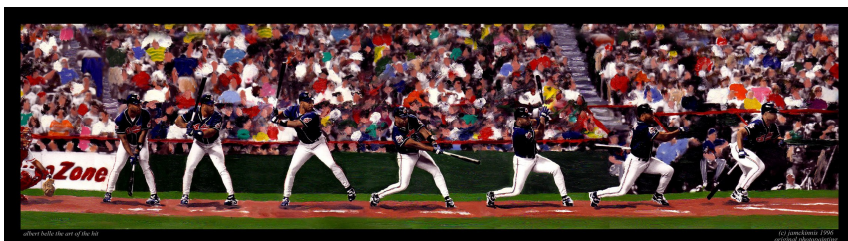
Human: CAGAATACGATTTAGCAAATTAAGTCTTTAAGATATTATTTTACATTTCTATATTCTCCTA
 Chimpanzee: CAGAATACGATTTAGCAAATTAAGTCTTTAAGATATTATTTTACATTTCTATATTCTCCTA
 Macaque: CAGAATATGATTTAGCAAATTAAGTCTTTAAGATATTATTTTGCAC TCTATATTCTCCTA

Human: CCC TGAGTTGATGTGTGAGCAATATGTCAC TTTTCATAAAGCCAGGTATACA --- TTATG
 Chimpanzee: CCC TGAGTTGATGTGTGAGCCG TATGTCAC TTTTCATAAAGCCAGGTATACA --- TTATG
 Macaque: CCC TGAGTTGATGTGTGAGCAATATGTCAC TTTTCATAAAGCCAGGTATATACATTACG

H I Y S F L S K

Human: GACAGTAAAGTAAAAACATATTATTTATCTACBTTTITGTCAAAATTTTAAATTTTC
 Chimpanzee: GACAGTAAAGTAAAAACATATTATTTATCTACBTTTITGTCAAAATTTTAAATTTTC
 Macaque: GACAGTAAAGTAAAAACATATTATTTATCTACBTTTITGTCAAAATTTTAAATTTTC

Human: AAC TGT TGC GGT GTT TGG TAA --- TGT AAAACA AACT CAGT ACA
 Chimpanzee: AAC TGT TGC GGT GTT TGG TAA --- TGT AAAACA AACT CAGT ACA
 Macaque: AAC TGT TGC ATGT TGT TGG TAA --- CBT AAAACA AAT TCA G T A G



Datos secuenciales



Datos genómicos

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```



Patrones secuenciales



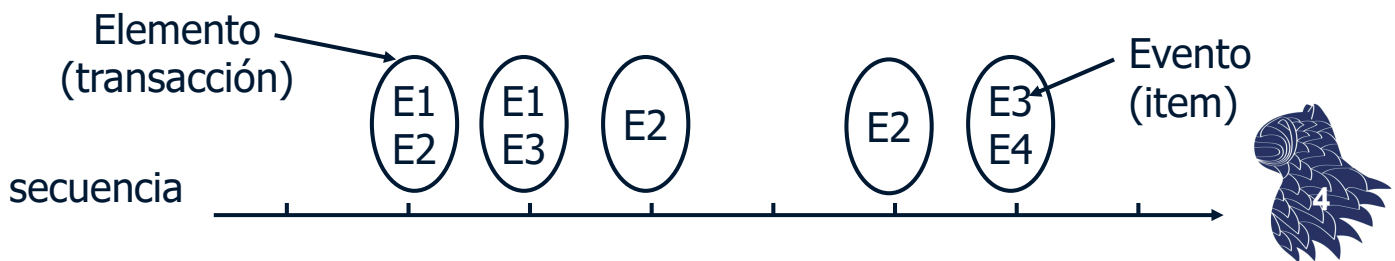
- Análisis de secuencias & patrones secuenciales.
- GSP [Generalized Sequential Patterns].
- SPADE [Sequential PAttern Discovery using Equivalent Class]
- PrefixSpan
- CloSpan
- Patrones secuenciales con restricciones.
- SSMiner [Similar Sequence Miner]
- Apéndice: Episodios



Análisis de secuencias



Base de datos	Secuencia	Elemento (Transacción)	Evento (Item)
Cientes	Historial de compras de un cliente determinado	Conjunto de artículos comprados por un cliente en un instante concreto	Libros, productos...
Web	Navegación de un visitante del sitio web	Colección de ficheros vistos por el visitante tras un único click de ratón	Página inicial, información de contacto, fotografía...
Eventos	Eventos generados por un sensor	Eventos generados por un sensor en un instante t	Tipos de alarmas generadas
Genoma	Secuencia de ADN	Elemento de la secuencia de ADN	Bases A,T,G,C



Análisis de secuencias



Una secuencia es una lista ordenada de elementos:

$$s = \langle e_1 e_2 e_3 \dots \rangle$$

- Cada elemento contiene una colección de eventos:

$$e_i = \{i_1, i_2, \dots, i_k\}$$

- Cada elemento tiene una localización temporal (o espacial) asociada.
- La longitud de la secuencia, $|s|$, es el número de elementos de la secuencia
- Una k-secuencia es una secuencia de k eventos.



Análisis de secuencias



Ejemplos

- Secuencia de visitas en una página web
< {Homepage} {Electronics} {Tablets} {Kindle Fire HD} {Shopping Cart}
{Order Confirmation} {Return to Shopping} >
- Secuencia de eventos que inició el accidente nuclear de 3-Mile Island http://en.wikipedia.org/wiki/Three_Mile_Island_accident
< {clogged resin} {outlet valve closure} {loss of feedwater}
{condenser polisher outlet valve shut} {booster pumps trip}
{main waterpump trips} {main turbine trips} {reactor pressure increases}>
- Secuencia de préstamos de libros en una biblioteca
<{Fellowship of the Ring} {The Two Towers} {Return of the King}>

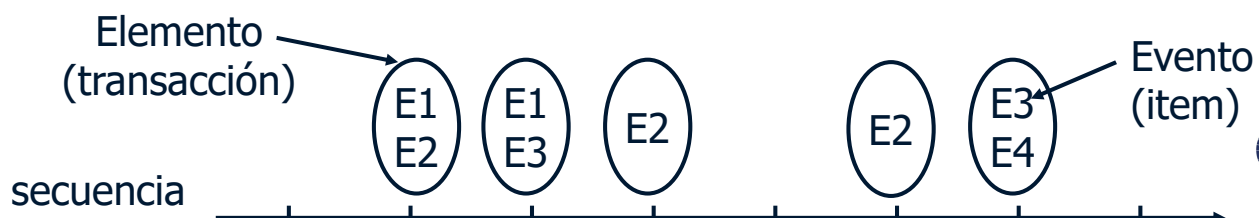


Análisis de secuencias



Una secuencia $\langle a_1 a_2 \dots a_n \rangle$
está contenida en otra secuencia $\langle b_1 b_2 \dots b_m \rangle$ ($m \geq n$)
si existe un conjunto de enteros $i_1 < i_2 < \dots < i_n$
tales que $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, ..., $a_n \subseteq b_{i_n}$

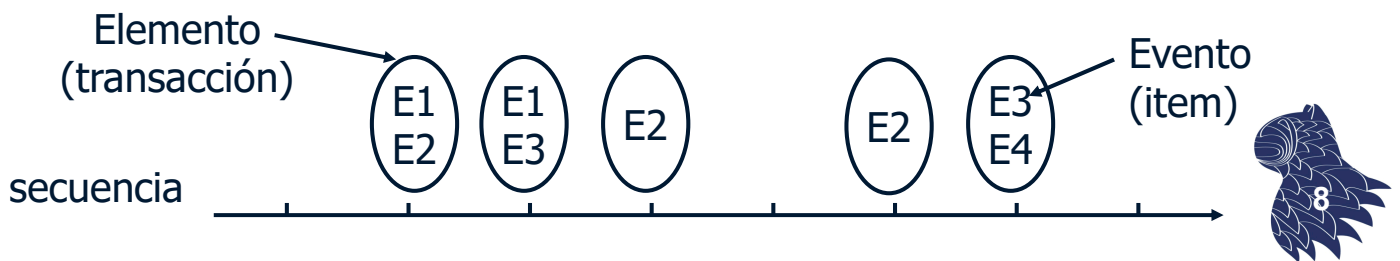
Secuencia	Subsecuencia	¿incluida?
$\langle \{2,4\} \{3,5,6\} \{8\} \rangle$	$\langle \{2\} \{3,5\} \rangle$	Sí
$\langle \{1,2\} \{3,4\} \rangle$	$\langle \{1\} \{2\} \rangle$	No
$\langle \{2,4\} \{2,4\} \{2,5\} \rangle$	$\langle \{2\} \{4\} \rangle$	Sí



Análisis de secuencias



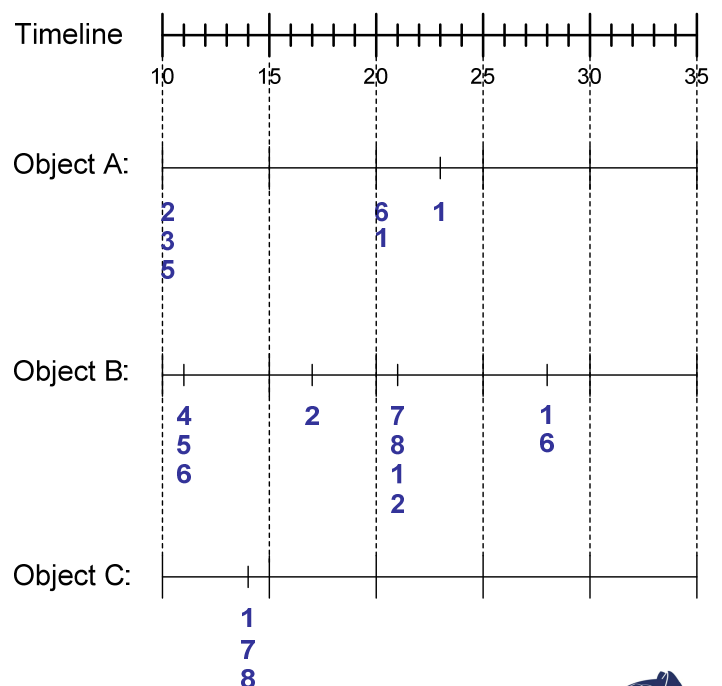
- El soporte de una subsecuencia S se define como la fracción de secuencias de la base de datos que incluyen la subsecuencia S .
- Un patrón secuencial es una subsecuencia frecuente (esto es, una subsecuencia con soporte $\geq \text{MinSupp}$)



Análisis de secuencias



Object	Timestamp	Events
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 7, 8



Base de datos de secuencias





Problema

Extracción de patrones secuenciales

Dados

- una base de datos de secuencias y
- un umbral de soporte mínimo $MinSupp$,
encontrar todas las subsecuencias frecuentes
(esto es, las subsecuencias con soporte $\geq MinSupp$).



Object	Timestamp	Events
A	1	1,2,4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1, 2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

$MinSupp = 50\%$

Ejemplos de subsecuencias frecuentes:

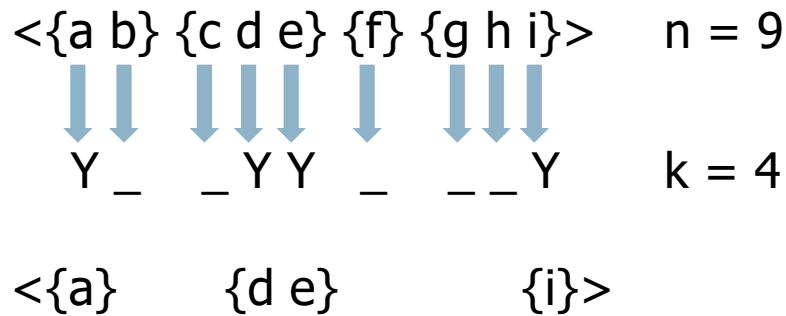
$\langle \{1,2\} \rangle$ $s=60\%$
 $\langle \{2,3\} \rangle$ $s=60\%$
 $\langle \{2,4\} \rangle$ $s=80\%$
 $\langle \{3\} \{5\} \rangle$ $s=80\%$
 $\langle \{1\} \{2\} \rangle$ $s=80\%$
 $\langle \{2\} \{2\} \rangle$ $s=60\%$
 $\langle \{1\} \{2,3\} \rangle$ $s=60\%$
 $\langle \{2\} \{2,3\} \rangle$ $s=60\%$
 $\langle \{1,2\} \{2,3\} \rangle$ $s=60\%$



Patrones secuenciales



¿Cuántas k -subsecuencias existen en una n -secuencia?



$$\binom{n}{k} = \binom{9}{4} = 126$$



Patrones secuenciales



Dados n eventos (items): $i_1, i_2, i_3, \dots, i_n$

- n 1-subsecuencias candidatas

$\langle \{i_1\} \rangle, \langle \{i_2\} \rangle, \langle \{i_3\} \rangle, \dots, \langle \{i_n\} \rangle$

- $n(n-1)/2 + n^2$ 2-subsecuencias candidatas:

$\langle \{i_1, i_2\} \rangle, \langle \{i_1, i_3\} \rangle, \dots, \langle \{i_1\} \{i_1\} \rangle, \langle \{i_1\} \{i_3\} \rangle, \dots, \langle \{i_{n-1}\} \{i_n\} \rangle$

- ¿? 3-subsecuencias candidatas:

$\langle \{i_1, i_2, i_3\} \rangle, \langle \{i_1, i_2, i_4\} \rangle, \dots, \langle \{i_1, i_2\} \{i_1\} \rangle, \langle \{i_1, i_2\} \{i_2\} \rangle, \dots,$
 $\langle \{i_1\} \{i_1, i_2\} \rangle, \langle \{i_1\} \{i_1, i_3\} \rangle, \dots, \langle \{i_1\} \{i_1\} \{i_1\} \rangle, \langle \{i_1\} \{i_1\} \{i_2\} \rangle,$
 \dots





Fase 1:

Recorrer la base de datos para obtener todas las secuencias frecuentes de 1 elemento.

Fase 2:

Mientras se encuentren nuevas secuencias frecuentes:

- **Generación:** Generar k-secuencias candidatas a partir de las (k-1)-secuencias frecuentes.
- **Podar:** Podar k-secuencias candidatas que contengan alguna (k-1)-secuencia no frecuente.
- **Conteo:** Recorrer nuevamente el conjunto de datos para obtener el soporte de las candidatas.
- **Eliminación:** Eliminar las k-secuencias candidatas cuyo soporte real esté por debajo de MinSupp.



Generación de candidatos

Caso base (k=2)

La combinación de dos 1-secuencias $\langle \{i_1\} \rangle$ e $\langle \{i_2\} \rangle$ produce dos 2-secuencias candidatas

$$\langle \{i_1\} \{i_2\} \rangle \quad \& \quad \langle \{i_1 i_2\} \rangle$$





Generación de candidatos

Caso general ($k > 2$)

Dos $(k-1)$ -secuencias frecuentes, w_1 y w_2 , se combinan para producir una k -secuencia candidata si la subsecuencia obtenida al eliminar el primer evento de w_1 es la misma que la subsecuencia obtenida al eliminar el último evento de w_2

La k -secuencia candidata vendrá dada por la extensión de w_1 con el último evento de w_2 .

- Si los dos últimos eventos de w_2 corresponden al mismo elemento, el último evento de w_2 pasa a formar parte del último elemento de w_1



Generación de candidatos

Ejemplos

- $w_1 = \langle \{1\} \{2\ 3\} \{4\} \rangle$ y $w_2 = \langle \{2\ 3\} \{4\ 5\} \rangle$
producen la secuencia candidata $\langle \{1\} \{2\ 3\} \{4\ 5\} \rangle$
- $w_1 = \langle \{1\} \{2\ 3\} \{4\} \rangle$ y $w_2 = \langle \{2\ 3\} \{4\} \{5\} \rangle$
producen la secuencia candidata $\langle \{1\} \{2\ 3\} \{4\} \{5\} \rangle$
- $w_1 = \langle \{1\} \{2\ 6\} \{4\} \rangle$ y $w_2 = \langle \{1\} \{2\} \{4\ 5\} \rangle$
no es necesario combinarlas para producir la secuencia candidata $\langle \{1\} \{2\ 6\} \{4\ 5\} \rangle$, ya que, si dicha secuencia es una posible candidata, se obtendrá combinando w_1 con $\langle \{1\} \{2\ 6\} \{5\} \rangle$



GSP [Generalized Sequential Patterns]



Frequent 3-sequences

- < {1} {2} {3} >
- < {1} {2 5} >
- < {1} {5} {3} >
- < {2} {3} {4} >
- < {2 5} {3} >
- < {3} {4} {5} >
- < {5} {3 4} >



Candidate Generation

- < {1} {2} {3} {4} >
- < {1} {2 5} {3} >
- < {1} {5} {3 4} >
- < {2} {3} {4} {5} >
- < {2 5} {3 4} >



Candidate Pruning

- < {1} {2 5} {3} >



GSP [Generalized Sequential Patterns]



- Candidatos iniciales: "singleton sequences":
<a>, , <c>, <d>, <e>, <f>, <g>, <h>
- Recorrido inicial de la base de datos:
Conteo del soporte de cada candidato

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)>

MinSupp = 2

Cand	Supp
<a>	3
	5
<c>	4
<d>	3
<e>	3
<f>	2
<g>	1
<h>	1



GSP [Generalized Sequential Patterns]



51 candidatos de tamaño 2 cumplen la propiedad Apriori

	<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>

	<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>						

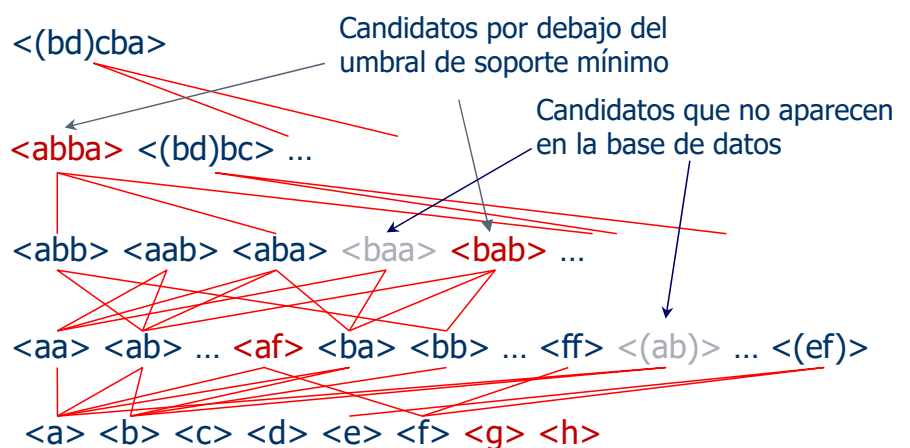
Se poda el 44.57% de las 92 secuencias posibles $(8*8+8*7/2)$



GSP [Generalized Sequential Patterns]



- 5th scan: 1 cand. 1 length-5 seq. pat.
- 4th scan: 8 cand. 6 length-4 seq. pat.
- 3rd scan: 46 cand. 19 length-3 seq. pat.
- 2nd scan: 51 cand. 19 length-2 seq. pat.
- 1st scan: 8 cand. 6 length-1 seq. pat.



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)>

MinSupp = 2





Sequential PAttern Discovery using Equivalent Class [Zaki]

- Método de identificación de secuencias frecuentes basado en una representación vertical (tipo Eclat).
- La base de datos de secuencias se transforma en un conjunto enorme de items $\langle \text{SID}, \text{EID} \rangle$.
- Las subsecuencias van creciendo usando el algoritmo de generación de candidatos de Apriori.



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	





Limitaciones

- Se recorre la base de datos varias veces.
- Búsqueda en anchura [Breadth-first search].
 - Los patrones largos se construyen a partir de patrones más cortos, lo que obliga a considerar un número exponencial de candidatos.
 - Se genera un conjunto enorme de candidatos: 1000 patrones de tamaño 1 dan lugar a 1.5M patrones de tamaño 2.

$$1000 \times 1000 + \frac{1000 \times 999}{2} = 1,499,500$$

- Descubrir una secuencia de tamaño 100 necesita considerar 10^{30} subsecuencias

$$\sum_{i=1}^{100} \binom{100}{i} = 2^{100} - 1 \approx 10^{30}$$



PrefixSpan



Mining Sequential Patterns by Prefix Projections

Prefijos de una secuencia

Secuencia <a(abc)(ac)d(cf)>
 Prefijos <a>, <aa>, <a(ab)>, <a(abc)>...

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





Estrategia

Paso 1: Encontrar patrones secuenciales de longitud 1.

`<a>`, ``, `<c>`, `<d>`, `<e>`, `<f>`

Paso 2: Dividir el espacio de búsqueda.

El conjunto completo de patrones secuenciales se puede dividir en 6 subconjuntos:

- Los patrones que tienen como prefijo `<a>`
- Los patrones que tienen como prefijo ``
- ...
- Los patrones que tienen como prefijo `<f>`



- Para encontrar los patrones con prefijo `<a>` sólo hay que considerar las proyecciones con respecto a `<a>`

Base de datos proyectada sobre `<a>`:

`<(abc)(ac)d(cf)>`
`<(_d)c(bc)(ae)>`
`<(_b)(df)cb>`
`<(_f)cbc>`

SID	Secuencia
10	<code><a(abc)(ac)d(cf)></code>
20	<code><(ad)c(bc)(ae)></code>
30	<code><(ef)(ab)(df)cb></code>
40	<code><eg(af)cbc></code>

- A continuación, se identifican los patrones de longitud 2 que tienen `<a>` como prefijo:

`<aa>`, `<ab>`, `<(ab)>`, `<ac>`, `<ad>`, `<af>`

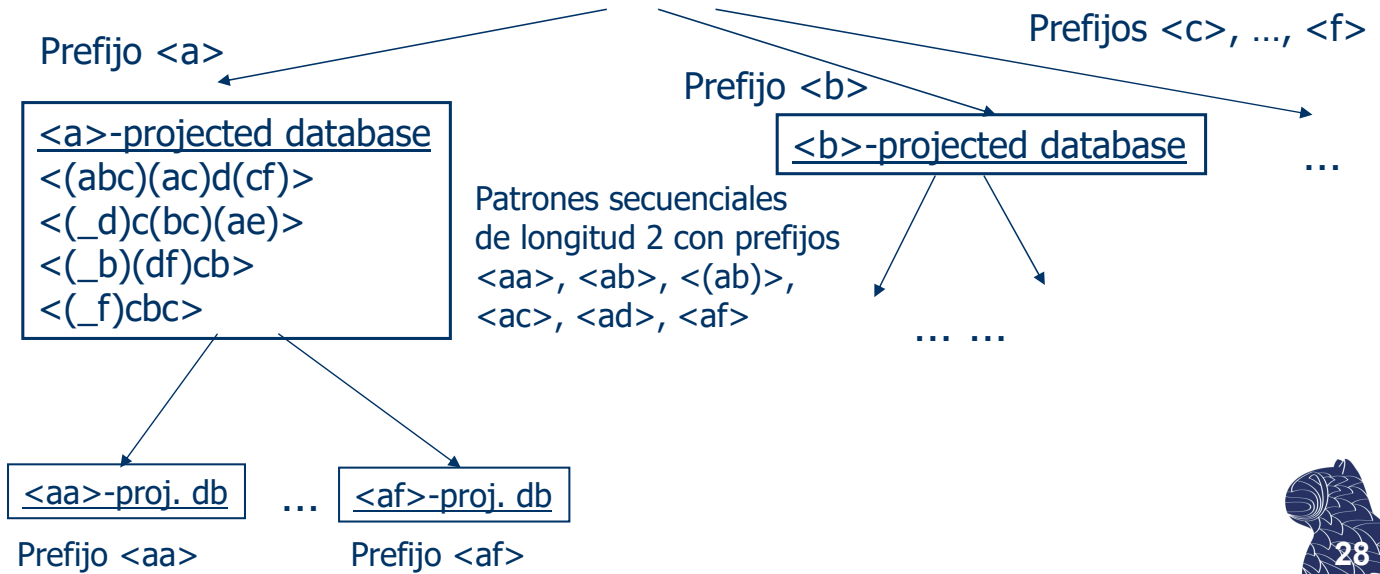


PrefixSpan



SID	Secuencia
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Patrones de secuenciales de longitud 1
<a>, , <c>, <d>, <e>, <f>

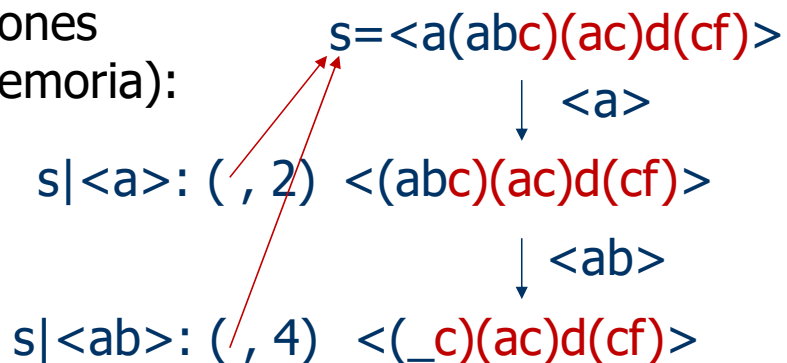


PrefixSpan



Ventajas de PrefixSpan

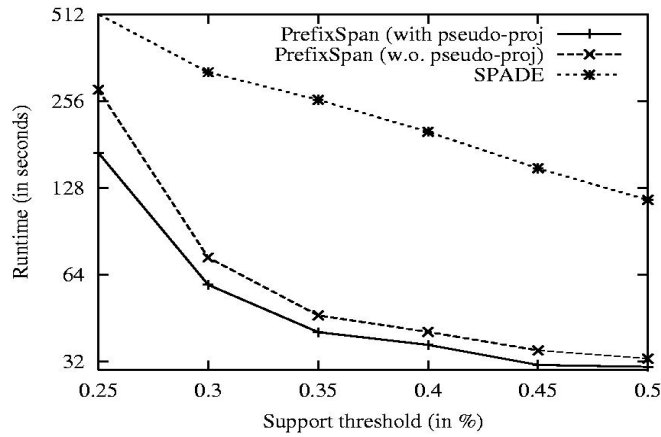
- No se generan secuencias candidatas.
- Las proyecciones son cada vez más pequeñas.
- Pseudoproyecciones (punteros en memoria):



PrefixSpan



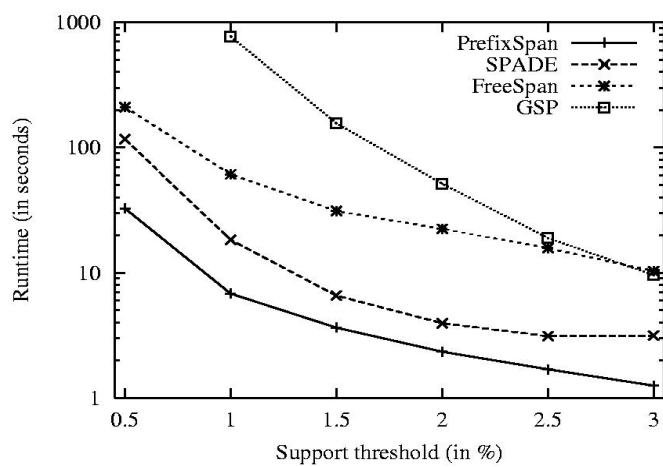
Pseudoproyecciones



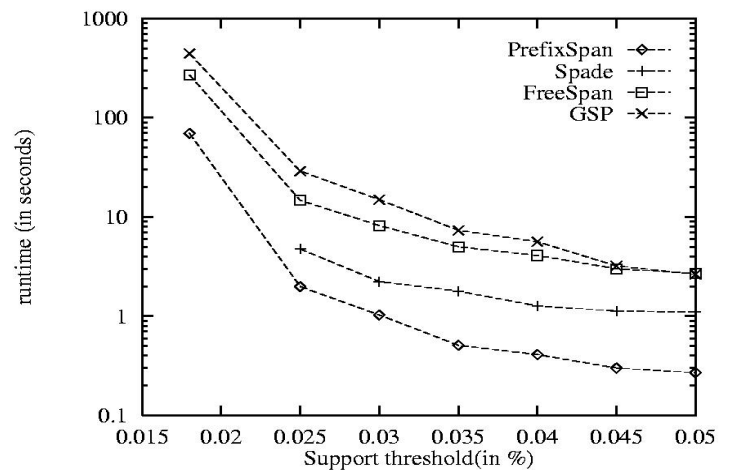
Performance comparison:
with pseudo-projection vs.
without pseudo-projection



PrefixSpan



Performance comparison
on data set C10T8S8I8



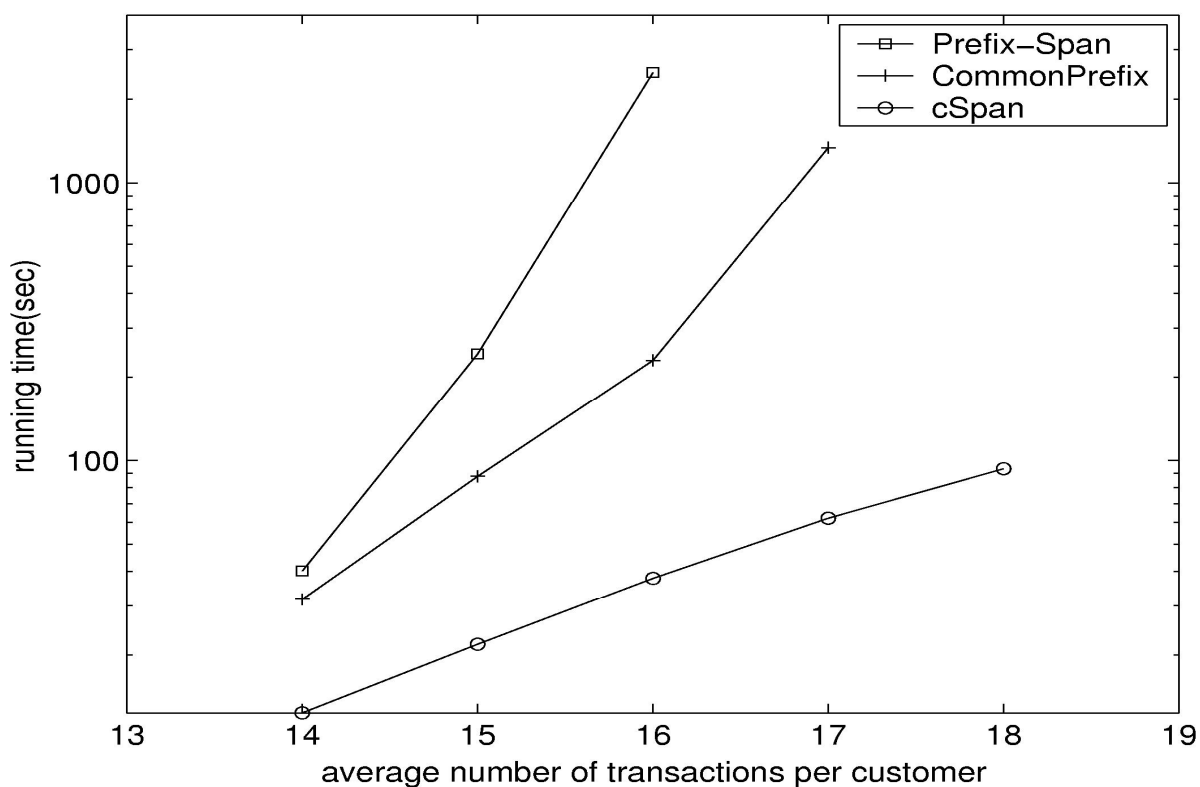
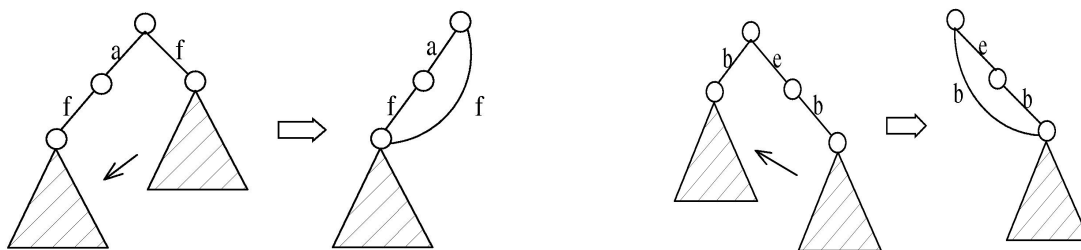
Performance comparison
on Gazelle data set





Mining Closed Sequential Patterns

- Un patrón secuencial s es cerrado si no existe otro patrón s' con el mismo soporte tal que $s \subset s'$.
- Identificar directamente patrones cerrados permite reducir el número de candidatos considerados.



Restricciones



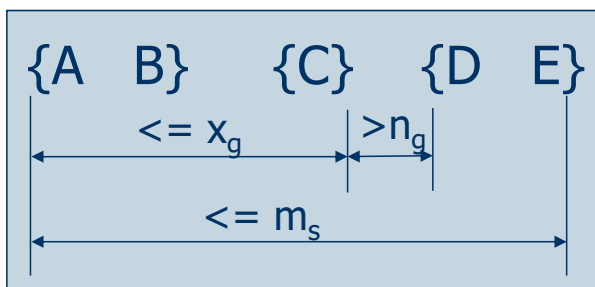
El uso de restricciones permite optimizar la exploración del espacio de patrones candidatos.

Clasificación de las restricciones

- **Anti-monótonas:** $\text{value_sum}(S) < 150, \text{min}(S) > 10$
- **Monótonas:** $\text{count}(S) > 5, S \supseteq \{\text{PC}, \text{digital_camera}\}$
- **Sucintas:** $\text{length}(S) \geq 10, S \in \{\text{Pentium}, \text{MS/Office}\}$
- **Convertibles:** $\text{value_avg}(S) < 25, \text{profit_sum}(S) > 160, \text{max}(S)/\text{avg}(S) < 2, \text{median}(S) - \text{min}(S) > 5$
- **Inconvertibles:** $\text{avg}(S) - \text{median}(S) = 0$



Restricciones



x_g : max-gap

n_g : min-gap

m_s : maximum span

$$x_g = 2, n_g = 0, m_s = 4$$

Secuencia de datos	Subsecuencia	¿Incluida?
$\langle \{2,4\} \{3,5,6\} \{4,7\} \{4,5\} \{8\} \rangle$	$\langle \{6\} \{5\} \rangle$	Sí
$\langle \{1\} \{2\} \{3\} \{4\} \{5\} \rangle$	$\langle \{1\} \{4\} \rangle$	No
$\langle \{1\} \{2,3\} \{3,4\} \{4,5\} \rangle$	$\langle \{2\} \{3\} \{5\} \rangle$	Sí
$\langle \{1,2\} \{3\} \{2,3\} \{3,4\} \{2,4\} \{4,5\} \rangle$	$\langle \{1,2\} \{5\} \rangle$	No





Identificación de patrones secuenciales con restricciones temporales

Estrategia 1

- Identificar patrones ignorando las restricciones.
- Postprocesar los patrones identificados.

Estrategia 2

- Modificar el algoritmo (p.ej. GSP) para podar los candidatos que no satisfacen las restricciones.

¡OJO! Hay que comprobar la propiedad Apriori...



Object	Timestamp	Events
A	1	1,2,4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1, 2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

$x_g = 1$ (max-gap)

$n_g = 0$ (min-gap)

$m_s = 5$ (maximum span)

$MinSupp = 60\%$

$\langle \{2\} \{5\} \rangle$ support = 40%

$\langle \{2\} \{3\} \{5\} \rangle$ support = 60% !!!

La restricción max-gap es la que causa el problema (no posee la propiedad Apriori).





Modificación de la etapa de generación de candidatos

- **Sin** la restricción max-gap: Una k-secuencia candidata se elimina si al menos una de sus (k-1)-subsecuencias no es frecuente.
- **Con** la restricción max-gap: Una k-secuencia candidata se elimina si al menos una de sus (k-1)-subsecuencias contiguas no es frecuente.



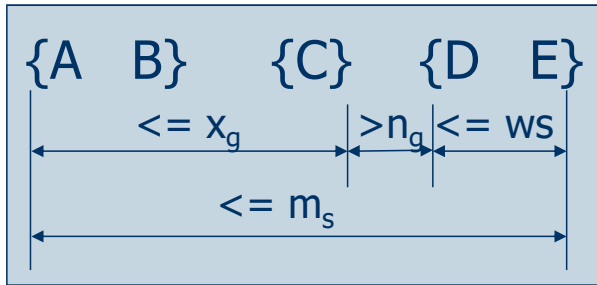
Ejemplo: Subsecuencias contiguas

$s = \langle \{1\} \{2\} \rangle$

- s es una subsecuencia contigua de
 $\langle \{1\} \{2\} \{3\} \rangle$
 $\langle \{1\} \{2\} \{3\} \rangle$
 $\langle \{3\} \{1\} \{2\} \{3\} \{4\} \rangle$
- s no es una subsecuencia contigua de
 $\langle \{1\} \{3\} \{2\} \rangle$
 $\langle \{2\} \{1\} \{3\} \{2\} \rangle$



Restricciones



x_g : max-gap

n_g : min-gap

ws: window size

m_s : maximum span

$x_g = 2, n_g = 0, ws = 1, m_s = 5$

Secuencia de datos	Subsecuencia	¿Incluida?
$\langle \{2,4\} \{3,5,6\} \{4,7\} \{4,6\} \{8\} \rangle$	$\langle \{3\} \{5\} \rangle$	No
$\langle \{1\} \{2\} \{3\} \{4\} \{5\} \rangle$	$\langle \{1,2\} \{3\} \rangle$	Sí
$\langle \{1,2\} \{2,3\} \{3,4\} \{4,5\} \rangle$	$\langle \{1,2\} \{3,4\} \rangle$	Sí



Restricciones



Modificación de la fase de conteo del soporte

Dado un patrón candidato $\langle \{a, c\} \rangle$, contribuirá al soporte del candidato cualquier secuencia que contenga

$\langle \dots \{a\} \{c\} \dots \rangle$,

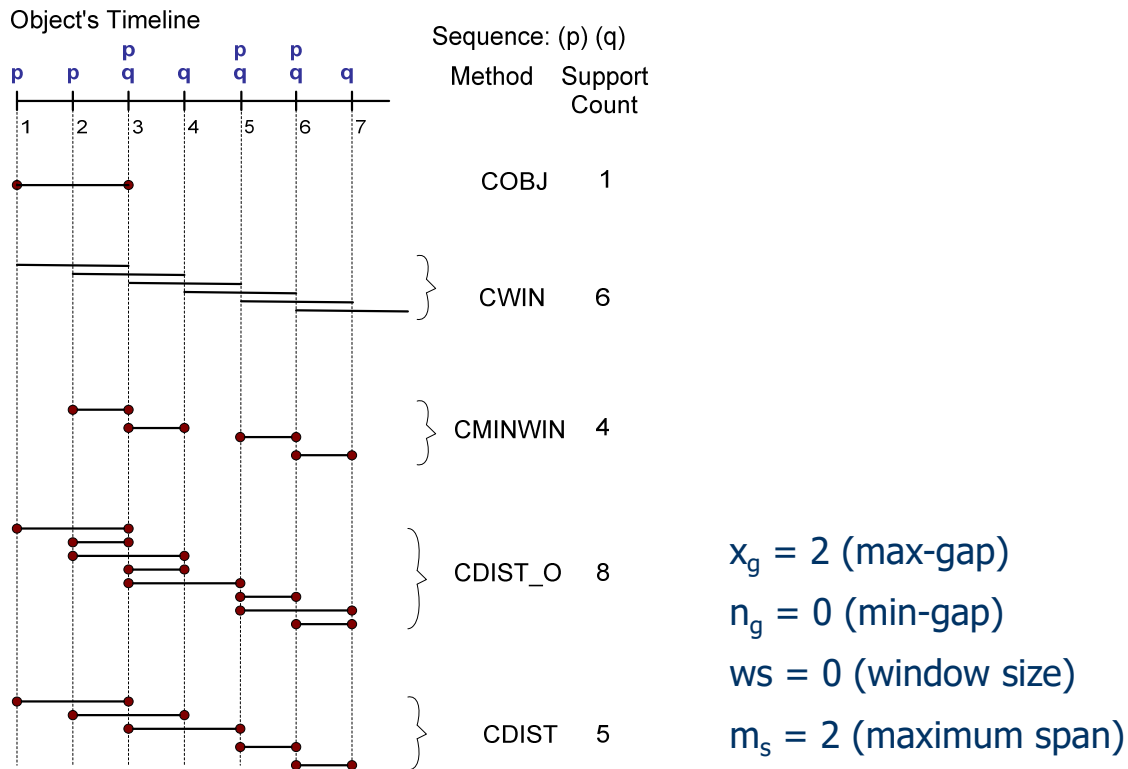
$\langle \dots \{a\} \dots \{c\} \dots \rangle$ con $\text{time}(\{c\}) - \text{time}(\{a\}) \leq ws$

$\langle \dots \{c\} \dots \{a\} \dots \rangle$ con $\text{time}(\{a\}) - \text{time}(\{c\}) \leq ws$





Distintas formas de calcular el soporte



SSMiner [Similar Sequence Miner]



Idea

- Generación de candidatos: Secuencias candidatas de $k+1$ elementos a partir de secuencias frecuentes de k elementos con $k-1$ elementos en común.
- Conteo del soporte: Listas de ocurrencias de cada secuencia frecuente en las que se preserva información acerca de la similitud entre patrones.

NOTA: Sólo se lee el conjunto de datos **2 veces**.





Algoritmo

Obtain frequent elements (frequent patterns of size 1)
Build candidate classes C_1 from the frequent elements
for $k=2$ **to** MaxSize
 for each class $P \in C_{k-1}$
 for each element $p \in P$.
 Compute the frequency of p
 if p is frequent
 then
 Create a new class P' from p .
 Add P' **to** C_k



Listas de ocurrencias

Tuplas (t, m, p, d, Θ) asociadas a cada patrón

- t : Identificador de la secuencia.
- m : elementos de la secuencia que corresponden a los $(k-1)$ elementos del prefijo del patrón.
- p : posición del último elemento en el patrón.
- d : parámetro utilizado para garantizar que los elementos del patrón son contiguos en la secuencia.
- Θ indica la similitud entre la ocurrencia y el patrón.





Listas de ocurrencias

Reunión de listas de ocurrencias

Let $(t_x, m_x, p_x, d_x, \Theta_x)$ and $(t_y, m_y, p_y, d_y, \Theta_y)$ be two tuples to be joined.

if

1. $t_x = t_y = t$ **and**
2. $m_x = m_y = m$ **and**
3. $d_x = 1$ (only if $k \neq 2$) **and**
4. $p_x < p_y$ **and**
5. $\Theta_x = \Theta_y$

then add $[t, m \cup \{p_x\}, p_y, d_y - d_x, \Theta_y]$ to the occurrence list



Eficiencia

$$O(L^K * n * (S - K + 1)^2)$$

L	Número de items frecuentes
K	Tamaño máximo de los patrones
n	Número de secuencias
S	Longitud de las secuencias

- Lineal con respecto al número de secuencias.
- Cuadrático con respecto a su longitud.
- Proporcional al número de patrones considerados.





Aplicación: Patrones musicales

- Una melodía puede verse como una secuencia de elementos musicales.
- “Motivo” [motif]: Elemento melódico más pequeño con significado.
- La identificación de “motivos” frecuentes puede servir...
 - ... para descubrir automáticamente el estribillo.
 - ... para crear miniaturas [audio-thumbnailing].
 - ... para indexar grandes bases de datos de canciones.



Ejemplo



G4	A4	E4	D5	E5	B4
{1,_,1,1,=}	{1,_,2,2,=}	{1,_,4,4,=}	{1,_,5,5,=}	{1,_,6,6,=}	{1,_,8,8,=}
{1,_,3,3,=}	{1,_,10,10,=}	{1,_,12,12,=}	{1,_,7,7,=}		{1,_,15,15,=}
{1,_,9,9,=}	{1,_,13,13,=}	{1,_,19,19,=}			
{1,_,11,11,=}	{1,_,17,17,=}	{1,_,8,8,+5}			
{1,_,14,14,=}	{1,_,6,6,+5}	{1,_,15,15,+5}			
{1,_,16,16,=}					
{1,_,18,18,=}					
{1,_,5,5,+5}					
{1,_,7,7,+5}					

Secuencia

G4 A4 G4 E4 D5 E5 D5 B4 G4 A4 G4 E4 A4 G4 B4 G4 A4 G4 E4





Ejemplo



Prefix: G4

G4 G4

- {1,1,3,2,=}
- {1,1,11,10,=}
- {1,1,16,15,=}
- {1,3,9,6,=}
- {1,3,14,11,=}
- {1,3,18,15,=}
- {1,9,14,5,=}
- {1,9,18,9,=}
- {1,11,16,5,=}
- {1,14,16,2,=}
- {1,16,18,2,=}
- {1,1,9,8,=}
- {1,1,14,13,=}
- {1,1,18,17,=}
- {1,3,11,8,=}
- {1,3,16,13,=}
- {1,9,11,2,=}
- {1,9,16,7,=}
- {1,11,14,3,=}
- {1,11,18,7,=}
- {1,14,18,4,=}
- {1,5,7,2,+5}

G4 A4

- {1,1,2,1,=}**
- {1,1,13,12,=}
- {1,3,10,7,=}
- {1,3,17,14,=}
- {1,9,13,4,=}
- {1,11,13,2,=}
- {1,14,17,3,=}
- {1,5,6,1,+5}**
- {1,1,10,9,=}
- {1,1,17,16,=}
- {1,3,13,10,=}
- {1,9,10,1,=}**
- {1,9,17,8,=}
- {1,11,17,6,=}
- {1,16,17,1,=}**

G4 E4

- {1,1,4,3,=}
- {1,1,19,18,=}
- {1,3,12,9,=}
- {1,9,12,3,=}
- {1,11,12,1,=}**
- {1,14,19,5,=}
- {1,18,19,1,=}**
- {1,5,15,10,+5}
- {1,7,15,8,+5}
- {1,1,12,11,=}
- {1,3,4,1,=}**
- {1,3,19,16,=}
- {1,9,19,10,=}
- {1,11,19,8,=}
- {1,16,19,3,=}
- {1,5,8,3,+5}
- {1,7,8,1,+5}**

Secuencia

G4 A4 G4 E4 D5 E5 D5 B4 G4 A4 G4 E4 A4 G4 B4 G4 A4 G4 E4

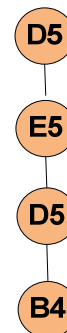


Aplicación: Patrones musicales

- G4
- A4
- G4
- E4
- D5
- E5
- D5
- B4
- A4
- G4
- A4
- G4
- E4
- B4



Exact
Pattern

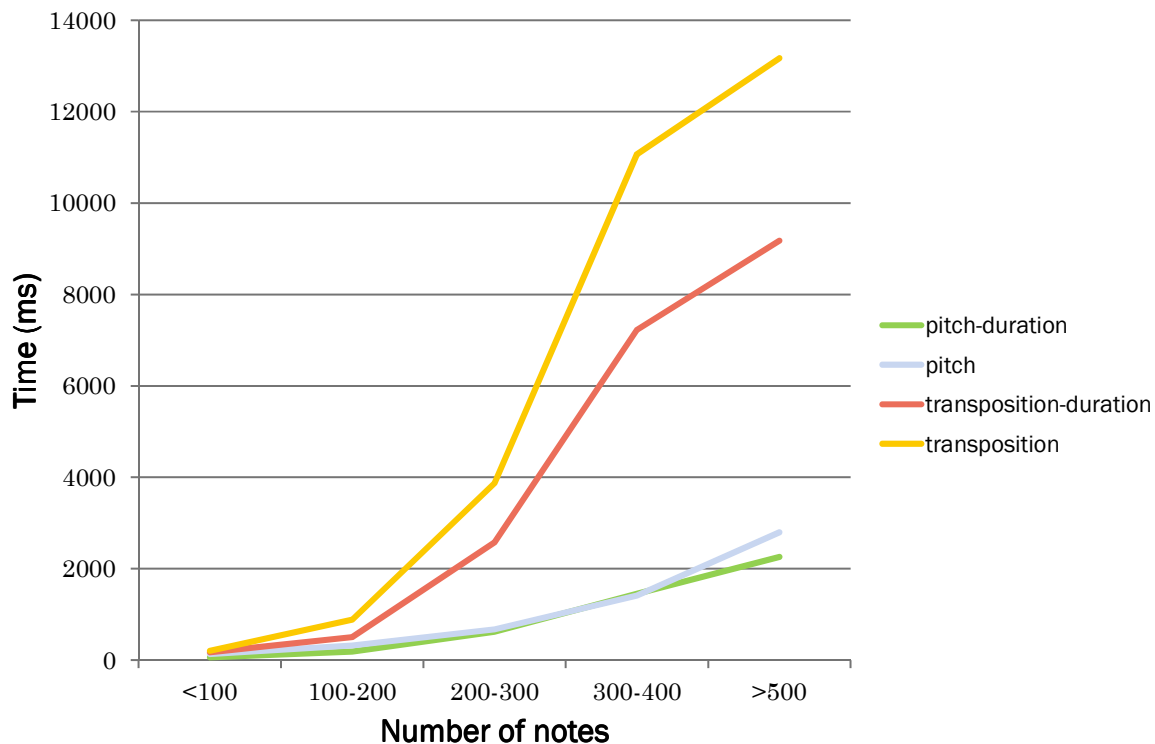


Similar
Pattern





Aplicación: Patrones musicales



Algoritmos



Derivados de Apriori

Candidate Generation: Horizontal Database Format		
Apriori (All, Some, Dynamic Some)	[Agrawal and Srikant]	1995
Generalised Sequential Patterns (GSP)	[Srikant and Agrawal]	1996
PSP	[Massegia et al.]	1998
Sequential Pattern Mining with Regular expression constraints (SPIRIT)	[Garofalakis et al.]	1999
Maximal Frequent Sequences (MFS)	[Zhang et al.]	2001
Regular Expression-Highly Adaptive Constrained Local Extractor (RE-Hackle)	[Albert-Lorincz and Boulicaut]	2003
Maximal Sequential Patterns using Sampling (MSPS)	[Luo and Chung]	2004
Candidate Generation: Vertical Database Format		
Sequential Pattern Discovery using Equivalence classes (SPADE)	[Zaki]	2001
Sequential Pattern Mining (SPAM)	[Ayres et al.]	2002
LAst Position INduction (LAPIN)	[Yang and Kitsuregawa]	2004
Cache-based Constrained Sequence Miner (CCSM)	[Orlando et al.]	2004
Indexed Bit Map (IBM)	[Savary and Zeitouni]	2005
LAst Position INduction Sequential PAttern Mining (LAPIN-SPAM)	[Yang and Kitsuregawa]	2005

Derivados de FP-Growth

Algorithm Name	Author	Year
Pattern Growth		
FREquent pattern-projected Sequential PAtterN mining (FreeSpan)	[Han et al.]	2000a
PREFIX-projected Sequential PAtterN mining (PrefixSpan)	[Pei et al.]	2001b
Sequential pattern mining with Length-decreasing support (SLPMiner)	[Seno and Karypis]	2002

Algoritmos paralelos

ACME	Advanced Parallel Motif Extractor
DGSP	Distributed GSP
DPF	Data Parallel Formulation
EVE	Event Distribution
EVECAN	Event and Candidate Distribution
HPSPM	Hash Partitioned Sequential Pattern Mining
MG-FSM	Large-Scale Frequent Sequence Mining
NPSPM	Non-Partitioned Sequential Pattern Mining
Par-ASP	Parallel PrefixSpan with Sampling
Par-CSP	Parallel CloSpan with Sampling
PLUTE	Parallel Sequential Patterns Mining
pSPADE	Parallel SPADE



Bibliografía



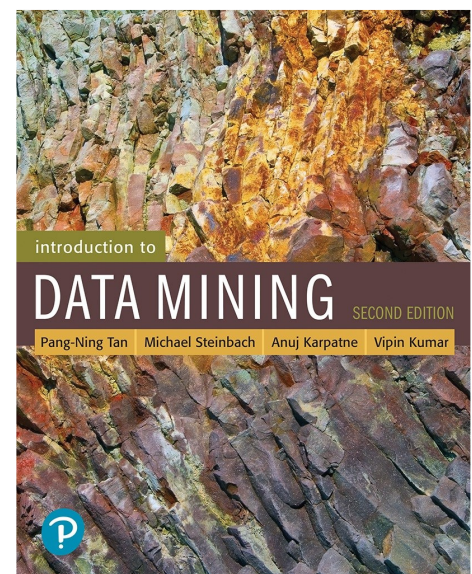
- Charu C. Aggarwal & Jiawei Han(editors):
Frequent Pattern Mining.
Springer, 2014.
ISBN 3319078208.



Bibliografía

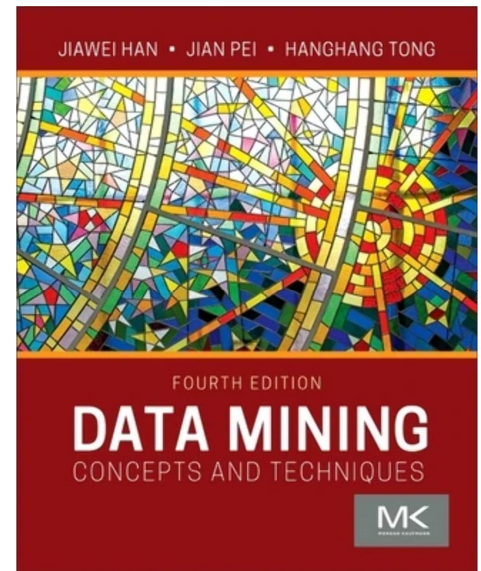


- Pang-Ning Tan,
Michael Steinbach,
Vipin Kumar &
Anuj Karpatne:
Introduction to Data Mining,
2nd edition, Addison Wesley, 2018.
ISBN 0133128903





Jiawei Han,
Jian Pei &
Hanghang Tong:
**Data Mining:
Concepts and Techniques**,
4th edition, Morgan Kaufmann, 2022.
ISBN 0128117605



5.4 Mining sequential patterns
7.6.2 Sequence classification



Bibliografía – Algoritmos



Algoritmo original (tipo Apriori)

- Rakesh Agrawal & Ramakrishnan Srikant: **Mining Sequential Patterns**. In Proceedings of the Eleventh International Conference on Data Engineering (ICDE '95), pp. 3-14. ISBN 0-8186-6910-1.

GSP [Generalized Sequential Patterns]

- Ramakrishnan Srikant & Rakesh Agrawal: **Mining Sequential Patterns: Generalizations and Performance Improvements**. In Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '96), pp. 3-17. ISBN 3-540-61057-X.

SPADE [Sequential Pattern Discovery using Equivalent Class]

- Mohammed J. Zaki: **SPADE: an efficient algorithm for mining frequent sequences**. Machine Learning 42(1/2):31-60, January 2001. DOI 10.1023/A:1007652502315

SSMiner [Similar Sequence Miner]

- Aída Jiménez, Miguel Molina-Solana, Fernando Berzal & Waldo Fajardo: **Mining transposed motifs in music**. Journal of Intelligent Information Systems 36(1):99-115, February 2011. DOI 10.1007/s10844-010-0122-7





FreeSpan

- Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal & Mei-Chun Hsu: **FreeSpan: Frequent pattern-projected sequential pattern mining**. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge discovery and Data Mining (KDD '00), pp. 355-359. DOI 10.1145/347090.347167.

PrefixSpan

- Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal & Meichun Hsu: **PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth**. Proceedings of the 17th International Conference on Data Engineering (ICDE'2001), pp. 215-224. DOI 10.1109/ICDE.2001.914830
- Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal & Mei-Chun Hsu: **Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach**. IEEE Transactions on Knowledge and Data Engineering 16(11): 1424-1440, November 2004. DOI 10.1109/TKDE.2004.77

IncSpan

- Hong Cheng, Xifeng Yan & Jiawei Han: **IncSpan: incremental mining of sequential patterns in large database**. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04), pp. 527-532. DOI 10.1145/1014052.1014114



CloSpan

- Xifeng Yan, Jiawei Han & Ramin Afshar: **CloSpan: Mining closed sequential patterns in large databases**. SIAM International Conference on Data Mining (SDM'2003)

BIDE [BI-Directional Extension]

- Jianyong Wang & Jiawei Han: **BIDE: Efficient Mining of Frequent Closed Sequences**. Proceedings of the 20th International Conference on Data Engineering (ICDE '04), pp. 79-90. DOI 10.1109/ICDE.2004.1319986

Par-CSP [Parallel CloSpan with samPLing]

- Shengnan Cong, Jiawei Han & David Padua: **Parallel mining of closed sequential patterns**. Proceedings of the 11th ACM SIGKDD international conference on Knowledge Discovery in Data Mining (KDD '05), pp. 562-567. DOI 10.1145/1081870.1081937

MG-FSM [Mind the Gap – Frequent Sequence Mining]

- Iris Miliaraki, Klaus Berberich, Rainer Gemulla & Spyros Zoupanos: **Mind the gap: large-scale frequent sequence mining**. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13), pp. 797-808. DOI 10.1145/2463676.2465285





pSPADE [Parallel SPADE]

- Mohammed J. Zaki: **Parallel sequence mining on shared-memory machines**. Journal of Parallel and Distributed Computing 61(3):401-426, March 2001. DOI 10.1006/jpdc.2000.1695

Patrones secuenciales con restricciones

- Minos N. Garofalakis, Rajeev Rastogi & Kyuseok Shim: **SPIRIT: Sequential Pattern Mining with Regular Expression Constraints**. Proceedings of the 25th International Conference on Very Large Data Bases (VLDB '99), pp. 223-234. ISBN 1-55860-615-7
- Jian Pei, Jiawei Han & Wei Wang: **Mining sequential patterns with constraints in large databases**. Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM '02), pp. 18-25. DOI 10.1145/584792.584799

Survey

- Carl H. Mooney and John F. Roddick. 2013. **Sequential pattern mining -- approaches and algorithms**. ACM Computing Surveys 45, 2, Article 19, March 2013, 39 pages. DOI 10.1145/2431211.2431218



Apéndice: Episodios



Patrones secuenciales alternativos

Episodios y expresiones regulares

- Episodios secuenciales: $A \rightarrow B$
- Episodios paralelos: $A \& B$
- Expresiones regulares: $(A|B)C^*(D \rightarrow E)$

Algoritmos para la identificación de patrones episódicos:

- Variaciones de Apriori/GSP
- Extensiones de FP-Growth:
"Projection-based pattern growth"



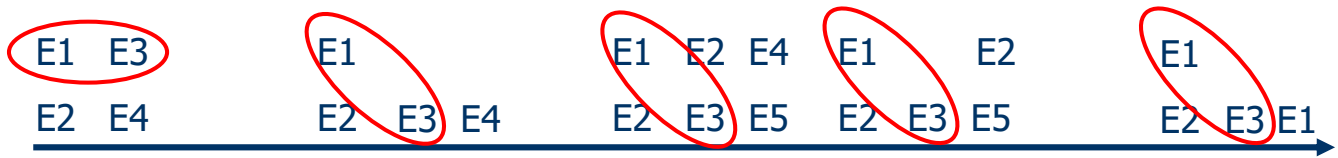
Apéndice: Episodios



En dominios en los que se tiene una única secuencia,
p.ej. Monitorización del tráfico de una red

Objetivo

Encontrar secuencias frecuentes de eventos, también conocidas como episodios [episodes].



Patrón < {E1} {E3} >

