

finding frequent subsequences from set of sequences.

Sequential Pattern Mining (SPM)

Sequence = Set of ordered events
represented by $\langle \rangle$
Challenge \rightarrow finding all subsequences

Example:-

CID	TID	Transactions	Sequential data
1	100	a, b, c, d	CID Sequences
3	111	a, f, d, e	1 $\langle (abcd), (def) \rangle$
1	122	d, e, f	
3	133	b, f, s, a	3 $\langle (afde), (bfse) \rangle$

Sid	Sequence	min supp = 2
10	$\langle (abc) (cd) d (cf) \rangle$ (1)	$\langle (ab) c \rangle \checkmark$
20	$\langle (cd) c (bc) (ac) \rangle$	
30	$\langle (ef) (ab) (df) (cb) \rangle$ (2)	$\langle eg \rangle \times$
40	$\langle eg (af) (ebc) \rangle$	

SPM vocabulary

- Itemset (non-empty set of items)
- Sequence (ordered list of events) order matters
- Event (itemset (unordered list of items))

\downarrow
($I_1 I_2 I_3$) where I_1, I_2, I_3 belongs to set of item I.

$e_1 = abc$

$e_2 = adc$

$S = \langle (abc) (adc) \rangle$

length = 6

\downarrow
no. of instances of items
in a sequence.

1-sequence \rightarrow length 1

Sequence with length-1 is called 1-pattern.

- Subsequence (α) is a sequence which is part of another sequence (β)

Agar kisi integers exist krty hain jisme α ke elements β main order main dikhte hain, toh α β ka subsequence and β is α supersequence

$\alpha = \langle (ab) d \rangle$
 $\beta = \langle (abc) (d.e) \rangle$
 α subsequence of β

~~$\beta = \langle (abd) \rangle$~~

- Sequence database

collection of sequences and often stores its tuples

~~Ident~~ $\langle S_{10}, S \rangle$
↓
Identifier → sequence

- Support of a sequence in a sequence database.

- Support of sequence (α) in a sequence database (S) is (no. of tuples that contains sequence (α))

calculated as no. of times sequence appears in a database.

- Frequent Sequence.

- Sequence (α) is considered frequent if its support is greater or equal to specified minimum support threshold.

Algo:- Apriori based (GSP)

Pattern-growth method (FreeSpan & PrefixSpan)

Vertical format based mining (SPADE)

Mining closed sequential pattern (CLOSPAN)

Sequence pattern.

Example:- min. supp = 2

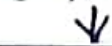
length of 1 is 1 &
there are 3 items in the first
3 events from 1, it contributes
to supp (a) to be 1.

SID	Sequence
1	< a (abc) (ac) d (ef) >
2	"
3	"
4	"

Subsequence & support

Example:-

< a (bc) d > is sub sequence of < a (abc) (ac) d (ef) >



It means all key events

~~sequence~~ sequence main order key

Set as main key in

Challenges of SPMs-

- Many number of possible sequential pattern are hidden within large datasets. which makes mining complex

Aditya

Notes / Uni: example.

Example:-

min-sup = 3

SID	Sequence
1	$\langle (bd) cb (ac) \rangle$
2	$\langle (bf) (ce) b (fg) \rangle$
3	$\langle (ah) (bdf) a bf \rangle$
4	$\langle (be) (ce) d \rangle$
5	$\langle a (bd) bcb (ade) \rangle$

candidate	Support		candidate	Support
a	3		a	3
b	5	\rightarrow	b	5
c	4		c	4
d	4		d	4
e	3		e	3
f	2 X			
g	1 X			
h	1 X			

~~| 2-length seq (A) | Support |
|------------------|---------|
| ab | 2 |
| ac | 2 |
| ad | 2 |
| ae | 2 |
| (ab) | 2 |
| (ac) | 2 |
| (ad) | 2 |
| (ae) | 2 |~~

2) 2 events Candidate.

	$\langle a \rangle$	$\langle b \rangle$	$\langle c \rangle$	$\langle d \rangle$	$\langle e \rangle$
$\langle a \rangle$	aa	ab	ac	ad	ae
$\langle b \rangle$	ba	bb	bc	bd	be
$\langle c \rangle$	ca	cb	cc	cd	ce
$\langle d \rangle$	da	db	dc	dd	de
$\langle e \rangle$	ea	eb	ec	ed	ee

Candidate	Supp	Validity	Candidate	Supp	Validity
aa	2		dd	2	
ab	2		de	1	
ac	1		ea	0	
ad	2		eb	1	
ae	1		ec	1	
ba	3	✓	ed	1	
bb	4	✓	ee	1	
bc	4	✓			
bd	2				
be	3	✓			
ca	2				
cb	3	✓			
cc	1				
cd	2				
ce	1				
da	3	✓			
db	3	✓			
dc	2				

3) Generating 1-event Candidate

	Candidate	Supp	Valid
a, b	$\langle a, b \rangle$	0	
a, c	$\langle a, c \rangle$	1	
a, d	$\langle a, d \rangle$	1	
a, e	$\langle a, e \rangle$	1	
a, f	$\langle a, f \rangle$	0	
a, g	$\langle a, g \rangle$	3	✓
a, h	$\langle a, h \rangle$	1	
b, c	$\langle b, c \rangle$	0	
b, d	$\langle b, d \rangle$	2	
b, e	$\langle b, e \rangle$	1	
b, f	$\langle b, f \rangle$	1	
c, d	$\langle c, d \rangle$	1	

Frequent 2-Sequences	
$\langle a, b \rangle$	✓
$\langle a, c \rangle$	✓
$\langle a, d \rangle$	✓
$\langle a, e \rangle$	✓
$\langle a, f \rangle$	✓
$\langle a, g \rangle$	✓
$\langle a, h \rangle$	✓
$\langle b, c \rangle$	✓
$\langle b, d \rangle$	✓
$\langle b, e \rangle$	✓
$\langle b, f \rangle$	✓
$\langle c, d \rangle$	✓

→ for 4th step

→ for 5th step

4) Generating 3-event candidate

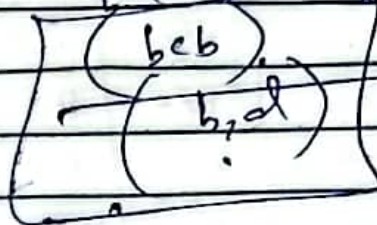
	$\langle a \rangle$	$\langle b \rangle$	$\langle c \rangle$	$\langle d \rangle$	$\langle e \rangle$
$\langle ab \rangle$	baa	bab	bac	bud	bne
$\langle bb \rangle$	bbq	bbb	bbc	bnd	bbe
$\langle bc \rangle$	bca	bcb	bcc	bcd	bce
$\langle be \rangle$	bea	beb	bec	bed	bee
$\langle cb \rangle$	cba	cbb	cac	cba	cbe
$\langle da \rangle$	daa	deb	dac	dad	dae
$\langle db \rangle$	dba	dbb	dbc	dbd	dbe

baa	0		beb	1	dba	2
bab	1		bec	0	dbd	1
bac	0		bed	1	dbe	1
bud	0		bee	0		
bne	0		cba	2		
bbb	2		cbb	0		
bbc	1		cbe	1		
bbd	1		cbf	0		
bbe	1		cbe	1		
bca	2		daa	0		
bcb	3	✓	deb	1		
bcc	1		dac	0		
bcd	2		dad	0		
bce	1		dne	0		
bea	0		dba	2		
			dbb	1		

5) 2-erent candidates.

	a	b	c	d	e
b, d	(b, d) a	(b, d) b	(b, d) c	(b, d) d	(b, d) e

(b, d) a	3	✓
(b, d) b	2	X
(b, d) c	2	X
(b, d) d	1	X
(b, d) e	1	X



5/1/1