Sequential Pattern Mining

Ritesh Yadav

May 2020

Content

- Sequential Pattern and its Applications
- Sequential Pattern Mining
- Terminology in Sequential Pattern Mining
- Understanding a Sequence Database
- Sequential Pattern Mining Algorithms
- GSP : Generalized Sequential Patterns
- SPADE : Sequential Pattern Mining in Vertical Data Format
- CloSpan: Mining Closed Sequential Patterns
- Association Rule Mining

Sequential Pattern and its Applications

- A sequence pattern consists of sequences of ordered elements or events, recorded with or without a concrete notion of time.
- **Customer shopping sequences** : Purchase a smartphone first, then a sim card, and then a bluetooth earphone,within 6 months
- Medical treatments, science processes, stocks and markets
- Weblog click streams, calling patterns
- Biological sequences
- Common databases used in study of sequential pattern
  - Transaction DB
  - Sequence DB
  - Time-Series DB

Sequential Pattern Mining

- Data mining consists of extracting information from data stored in databases to understand the data and/or take decisions.
- Sequential Pattern mining is one of the most fundamental task of data mining.
- It is the mining of frequently occurring ordered events or sub sequences as pattern in sequence database.
- A sequence database stores a number of records, where all records are sequences of ordered events, with or without concrete notions of time.
- Sequential Patterns are used for targeted marketing and customer retention.

Terminology in Sequential Pattern Mining

- Given a pattern A, **support** of the sequence pattern A is the number of sequences in the database containing the pattern A.
- A pattern with support greater than the support threshold i.e. min-sup is called a **frequent pattern**.
- A sequential pattern of length l is called an **l-pattern**.
- A **subsequence** is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.
- An **element** may contain set of items. Items within an element are unordered and we list them alphabetically.
- Hence given a set of sequences, Sequential Pattern Mining find **the complete set of frequent subsequences which satisfies the min-sup thresold**.

Understanding a Sequence Database

A **Sequence Database** :

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

A **Sequence** : < (ef) (ab) (df) c b >

- The red box in a sequence denotes a single element.
- Alphabets in these elements are called items of that element.
- $< a(ab)d(cf) >$ is a **subsequence** of $< \mathbf{a}(\mathbf{ab}c)(ac)\mathbf{d}(\mathbf{cf}) >$
- For **support thresold** min-sup $= 2$, $< (ab)c >$ is a sequential pattern.

Sequential Pattern Mining Algorithms

- A huge number of possible sequential patterns are hidden in databases.
- Sequential Pattern Mining Algorithms comes into play here to select the important sequential patterns.
- A sequential pattern mining algorithm should
  - find the complete set of patterns, when possible, satisfying the minimum support (frequency) threshold
  - be highly efficient, scalable
  - be able to incorporate various kinds of user-specific constraints.
- **Apriori Property :** If a subsequence s is infrequent, none of s's super-sequences can be frequent.
- **Representative Algorithms :** GSP, SPADE, PrefixSpan
- **Mining Closed sequential Patterns** : CloSpan

GSP : Generalized Sequential Patterns

Apriori-Based Sequential Pattern Mining

- The GSP algorithm finds all the length-1 candidates (using one database scan) and orders them with respect to their support ignoring ones for which support $<$ min-sup i.e support thresold.
- Then for each level (i.e., sequences of length-k), the algorithm scans database to collect support count for each candidate sequence and generates candidate length-(k+1) sequences from length-k frequent sequences using Apriori.
- This is repeated until no frequent sequence or no candidate can be found.

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

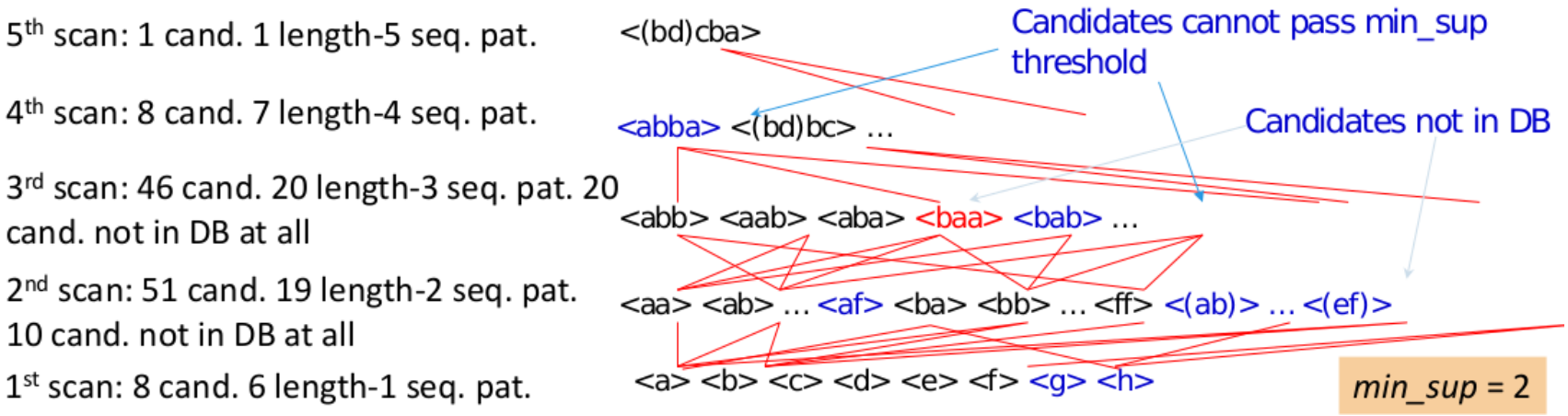
min_sup = 2

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

	<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>

	<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>						

- Intial Candidates of length-1 :
  $< a >, < b >, < c >, < d >, < e >, < f >, < g >, < h >$
- Scanning DB and counting support of these singelton candidates
- Generating lenth-2 candidates after removing those $<$ min-sup
- Length-2 Candidates with Apriori Pruning : $36 + 15 = 51$
- Without Apriori pruning: $8*8 + 8*7/2 = 92$



Drawbacks of GSP :

- a huge set of candidate sequences are generated
- multiple scans of database are needed
- it is inefficient for mining long sequential patterns.

Sequential Pattern Mining in Vertical Data Format: The SPADE Algorithm

- This is a vertical format sequential pattern mining method.
- In this algorithm a sequence database is mapped to : $< \mathbf{SID}, \mathbf{EID} >$
- Sequential pattern mining is performed by growing the subsequences (patterns) one item at a time by Apriori candidate generation.
- A sequence database:

SID	Sequence
1	<a(abc)(ac)d(cf)>
2	<(ad)c(bc)(ae)>
3	<(ef)(ab)(df)cb>
4	<eg(af)cbc>

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

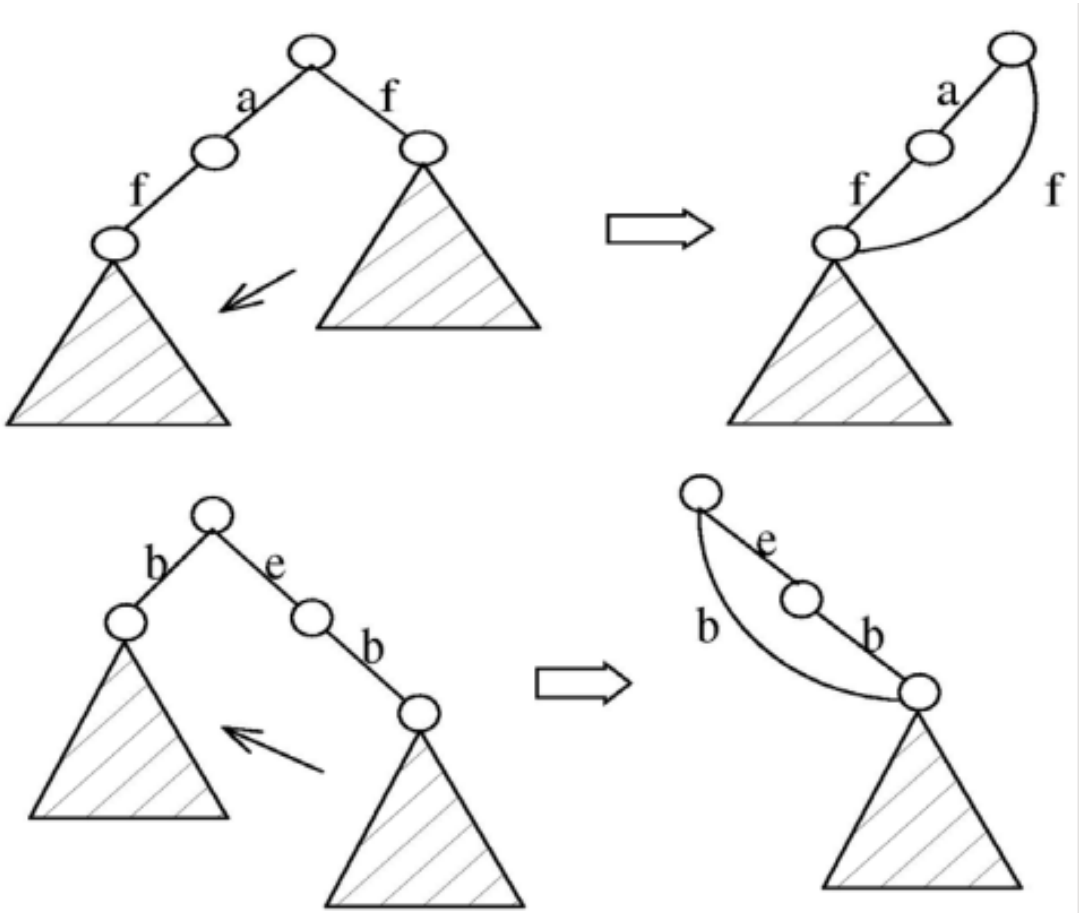
a		b		...
SID	EID	SID	EID	...
1	1	1	2	
1	2	2	3	
1	3	3	2	
2	1	3	5	
2	4	4	5	
3	2			
4	3			

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	

CloSpan: Mining Closed Sequential Patterns

- CloSpan is an algorithm for the mining of closed repetitive gapped subsequences.
- **Closed Sequential Patterns** : There exists no superpattern $s^{'}$ such that $s^{'} \subsetneq s$, and $s^{'}$ and s have the same support.
- **Open Sequential Patterns** : There exists superpattern $s^{'}$ such that $s^{'} \subsetneq s$, and $s^{'}$ and s have the same support.
- Given $< abc >: 20$, $< abcd >: 20$, $< abcde >: 15$, we know that $< abcd >$ is closed.
- Directly mining closed sequences reduces number of redundant patterns and attain same expressive power.
- **Note :** If s $\subsetneq$s$^{'}$, s is closed iff two project DBs have the same size.
- CloSpan uses backward subpattern and backward superpattern pruning to prune redundant search space thereby preventing unnecessary computations.



First one is backward super pattern pruning and second one is backward sub pattern pruning.

Association Rule Mining

- Association rule mining finds interesting associations and relationships among large sets of data items.
- This rule shows how frequently a itemset occurs in a transaction.
- Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.
- **Support Count :** Frequency of occurrence of a itemset.
- **Frequent Itemset :** An itemset whose support is greater than or equal to minsup threshold.
- **Association Rule :** An implication expression of the form A -> B, where A and B are any 2 itemsets.

Rule Evaluation Metrics

- **Support :** The number of transactions that include items in the A and B parts of the rule as a percentage of the total number of transaction. It measures how frequently collection of items occur together as a percentage of all transactions.
- **Confidence(A=>B) = Supp(A $\cup$ B)/ Sup(A) :** It is the ratio of the no of transactions that includes all items in B as well as the no of transactions that includes all items in A to the no of transactions that includes all items in A. It measures how often each item in B appears in transactions that contains items in A also.
- **Lift(A=>B) = Conf(A=>B)/ Supp(B) :** Lift value near 1 indicates X and Y almost often appear together as expected, greater than 1 means they appear together more than expected and less than 1 means they appear less than expected.Greater lift values indicate stronger association.

The End