SS-ZG548: ADVANCED DATA MINING

11 Sequence Mining

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Oct 16, 2021 ONLINE (WILP @ BITS-Pilani July-Dec 2021)

http://ktiwari.in/adm

Sequence Data

- Sequence Data: $S =< e_1, e_2, e_3, ... >$ attributed with specific time
- Each element e_i is a list of events $\{i_1, i_2, ..., i_k\}$
- Subsequence: $< a_1, a_2, ..., a_n >$ is contained in $< b_1, b_2, ...b_m >$ if $\exists i_1 < i_2 < i_3 < ... < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, ... , $a_n \subseteq b_{i_n}$
- Support for a sequence in database is the fraction that contains it

Sequence Data

- Sequence Data: $S =< e_1, e_2, e_3, ... >$ attributed with specific time
- Each element e_i is a list of events $\{i_1, i_2, ..., i_k\}$
- Subsequence: $< a_1, a_2, ..., a_n >$ is contained in $< b_1, b_2, ...b_m >$ if $\exists i_1 < i_2 < i_3 < ... < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, ... , $a_n \subseteq b_{i_n}$
- Support for a sequence in database is the fraction that contains it

Sequential Pattern Mining

Given a set of sequences and minsup; report all subsequence with support grater then minsup

Consider sequences $A =< \{1,2,4\}, \{2,3\}, \{5\} >$
$B =< \{1,2\}, \{2,3,4\} >$ $C =< \{1,2\}, \{2,3,4\}, \{2,4,5\} >$
$D =< \{2\}, \{3,4\}, \{4,5\} >$ $E =< \{1,3\}, \{2,4,5\} >$

Sequence	Support
$< \{1,2\} >$	60%
$< \{2,4\} >$	80%
$< \{1\}, \{2\} >$	80%
$< \{1,2\}, \{2,3\} >$	60%

For n events $i_1, i_2, ..., i_n$.
1-sequence: $< \{i_1\} >, < \{i_2\} >, < \{i_n\} >$
2-sequence: $< \{i_k, i_j\} >, < \{i_k\}, \{i_j\} >$
3-seq $< \{i_k, i_j, i_l\} >, < \{i_k\}, \{i_j, i_l\} >, < \{i_k, i_j\}, \{i_l\} >, < \{i_k\}, \{i_j\}, \{i_l\} >$

Generalized Sequential Pattern (GSP)[1] Mining

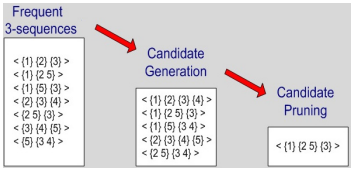
S-1 Make first pass over the sequence database to yield all 1-element frequent sequences

S-2 Repeat until new frequent sequences are found
- **Candidate Generation:** merge pairs found in $k-1^{th}$ pass
  - $< \{i_1\} >$ and $< \{i_2\} >$ yields $< \{i_1, i_2\} >$ and $< \{i_1\}, \{i_2\} >$
  - w_1 and w_2 can be merged if subsequences obtained by removal of first element of w_1 and last element of w_2 are same
- **Candidate Pruning:** Prune candidates containing infrequent $k-1$ subsequeces
- **Support Counting:** Make new pass to find support for new candidates
- **Candidate Elimination:** Eliminate candidate k-sequences whose support is less than minsup

$< \{1\}, \{2,3\}\{4\} >$ and $< \{2,3\}, \{4,5\} >$	$< \{1\}, \{2,3\}, \{4,5\} >$
$< \{1\}, \{2,3\}\{4\} >$ and $< \{2,3\}, \{4\}, \{5\} >$	$< \{1\}, \{2,3\}, \{4\}, \{5\} >$
$< \{1\}, \{2,6\}\{4\} >$ and $< \{1\}, \{2,6\}, \{4\} >$	Can not be merged

[1] Generalized Sequential Pattern (GSP), Srikant and Agrawal, In EDBT 1996

GSP: Candidate Generation



Candidate Generation

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

< {1} {2} {3} > , < {1} {2 5} > , < {1} {5} {3} > , < {2} {3} {4} > , < {2 5} {3} > , < {3} {4} {5} > , < {5} {3 4} > — each compared with:
< {1} {2} {3} >
< {1} {2 5} >
< {1} {5} {3} >
< {2} {3} {4} >
< {2 5} {3} >
< {3} {4} {5} >
< {5} {3 4} >

Candidate Generation

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

< {1} {2} {3} > , < {1} {2 5} > , < {1} {5} {3} > , < {2} {3} {4} > , < {2 5} {3} > , < {3} {4} {5} > , < {5} {3 4} > — each compared with:
< {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

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

< {1} {2} {3} {4} > : < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} >

< {1} {2 5} {3} > : < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} >

< {1} {5} {3 4} > : < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} >

< {2} {3} {4} {5} > : < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} >

< {2 5} {3 4} > : < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} >

Candidate Pruning

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

< {1} {2} {3} {4} > : < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} > → < {2} {3} {4} >, < {1} {3} {4} >, < {1} {2} {4} >, < {1} {2} {3} >

< {1} {2 5} {3} > : < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} > → < {2 5} {3} >, < {1} {5} {3} >, < {1} {2} {3} >, < {1} {2 5} >

< {1} {5} {3 4} > : < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} > → < {5} {3 4} >, < {1} {3 4} >, < {1} {5} {4} >, < {1} {5} {3} >

< {2} {3} {4} {5} > : < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} > → < {3} {4} {5} >, < {2} {4} {5} >, < {2} {3} {5} >, < {2} {3} {4} >

< {2 5} {3 4} > : < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} > → < {5} {3 4} >, < {2} {3 4} >, < {2 5} {4} >, < {2 5} {3} >

Candidate Pruning

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

< {1} {2} {3} {4} > : < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} > → < {2} {3} {4} >, < {1} {3} {4} >, < {1} {2} {4} >, < {1} {2} {3} >

< {1} {2 5} {3} > : < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} > → < {2 5} {3} >, < {1} {5} {3} >, < {1} {2} {3} >, < {1} {2 5} >

< {1} {5} {3 4} > : < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} > → < {5} {3 4} >, < {1} {3 4} >, < {1} {5} {4} >, < {1} {5} {3} >

< {2} {3} {4} {5} > : < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} > → < {3} {4} {5} >, < {2} {4} {5} >, < {2} {3} {5} >, < {2} {3} {4} >

< {2 5} {3 4} > : < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} > → < {5} {3 4} >, < {2} {3 4} >, < {2 5} {4} >, < {2 5} {3} >

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 Pruning

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

< {1} {2} {3} {4} > : < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} >, < {1} {2} {3} {4} > → < {2} {3} {4} >, < {1} {3} {4} >, < {1} {2} {4} >, < {1} {2} {3} >

< {1} {2 5} {3} > : < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} >, < {1} {2 5} {3} > → < {2 5} {3} >, < {1} {5} {3} >, < {1} {2} {3} >, < {1} {2 5} >

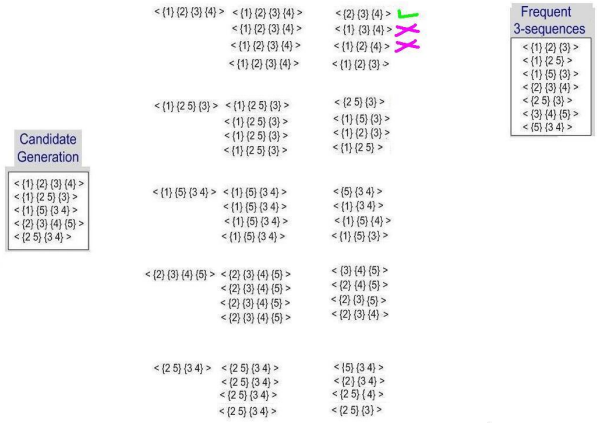
< {1} {5} {3 4} > : < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} >, < {1} {5} {3 4} > → < {5} {3 4} >, < {1} {3 4} >, < {1} {5} {4} >, < {1} {5} {3} >

< {2} {3} {4} {5} > : < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} >, < {2} {3} {4} {5} > → < {3} {4} {5} >, < {2} {4} {5} >, < {2} {3} {5} >, < {2} {3} {4} >

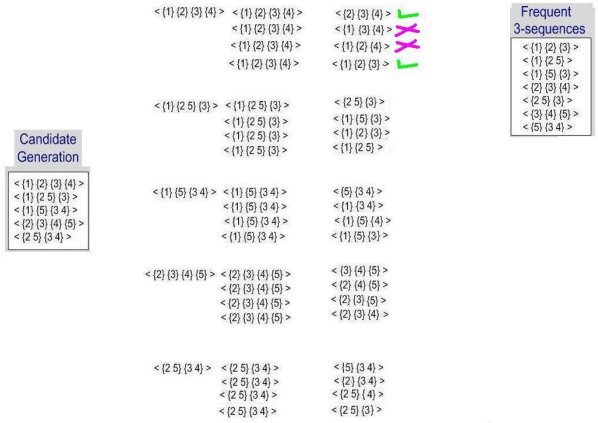
< {2 5} {3 4} > : < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} >, < {2 5} {3 4} > → < {5} {3 4} >, < {2} {3 4} >, < {2 5} {4} >, < {2 5} {3} >

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

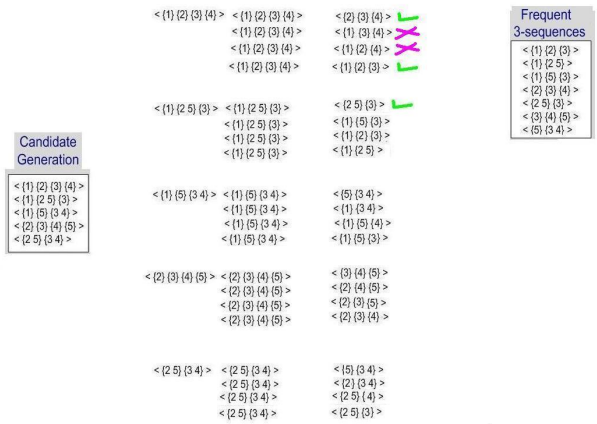
Candidate Pruning



Candidate Pruning

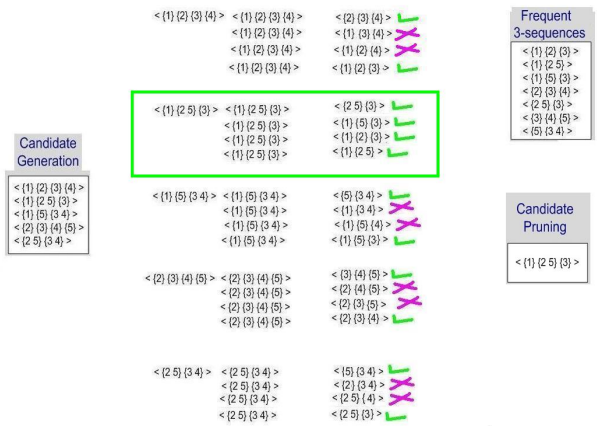


Candidate Pruning

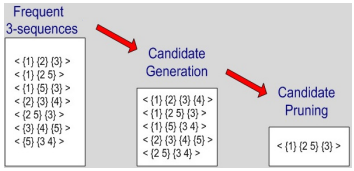


Candidate Pruning

Candidate Pruning



GSP: Candidate Generation



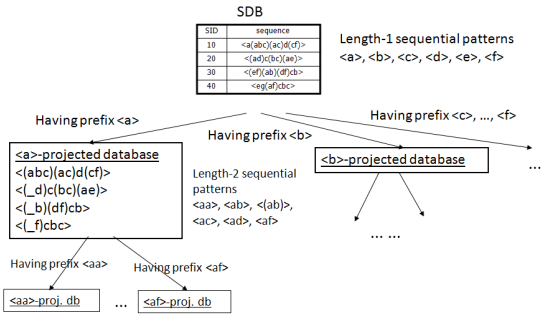
Issues:

- Huge number of candidate sets. n frequent 1-length candidate could generate $n^2 + \frac{n*(n-1)}{2}$ 2-length candidate
- Multiple scans of the database
- Mining n-length sequential patterns need $\sum_{i=1}^{n} {}^nC_i = 2^n - 1$ number of short candidates. It is exponential

One can use prefix projections approach similar to FP-Growth

Pseudo-Projections[2]

When things can fit in main memory

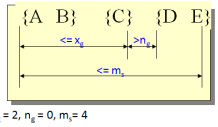


[2] Han, Jiawei and Pei, Jian and Mortazavi-Asl, Behzad and Pinto, Helen and Chen, Qiming and Dayal, Umeshwar and Hsu, MC, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth" In proceedings of international conference on data engineering, pages 215–224, 2001

Application: Hotlink Assignment

- Hotlink are the mostly visited pages
- Website is modeled as graph, where pages are nodes and links are edges
- Web logs stores sequences of user clicks
- One sequence one session
- Intermediate pages could be navigational or target
- Use sequence mining to Mark start and end page of frequent sessions

Time Constraints



x_g: max-gap
n_g: min-gap
m_s: maximum span

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

Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {4,7} {4,5} {8} >	< {6} {5} >	Yes
< {1} {2} {3} {4} {5}>	< {1} {4} >	No
< {1} {2,3} {3,4} {4,5}>	< {2} {3} {5} >	Yes
< {1,2} {3} {2,3} {3,4} {2,4} {4,5}>	< {1,2} {5} >	No

Approaches

1. Mine without timing constraint and post-process discovered patterns
2. Modify GSP to directly prune candidates violating timing

Anti Monotone Property

If a set is frequent all its subset must be frequent. If a set fails the test all its super set also must fail.

Consider sequences $A =< \{1,2,4\},\{2,3\},\{5\} >$
$B =< \{1,2\},\{2,3,4\} >$ $C =< \{1,2\},\{2,3,4\},\{2,4,5\} >$
$D =< \{2\},\{3,4\},\{4,5\} >$ $E =< \{1,3\},\{2,4,5\} >$

- Let $x_g = 1(max - gap)$, $n_g = 0(min - gap)$, $m_s = 5(maxspan)$, $minSup = 60\%$
- What is support for $< \{2\},\{5\} >$? 40%
- What is support for $< \{2\},\{3\},\{5\} >$? 60%

Anti Monotone Property does **not** holds, so these properties can not be pushed in GSP

Thank You!

Thank you very much for your attention!

Queries ?