SJSU SAN JOSÉ STATE UNIVERSITY

Sequence Data -

Sequential Pattern Mining

1

SJSU SAN JOSÉ STATE UNIVERSITY

Agenda

- What is Sequence Data?
- Sequential Pattern Mining
  - GSP
  - Prefixspan

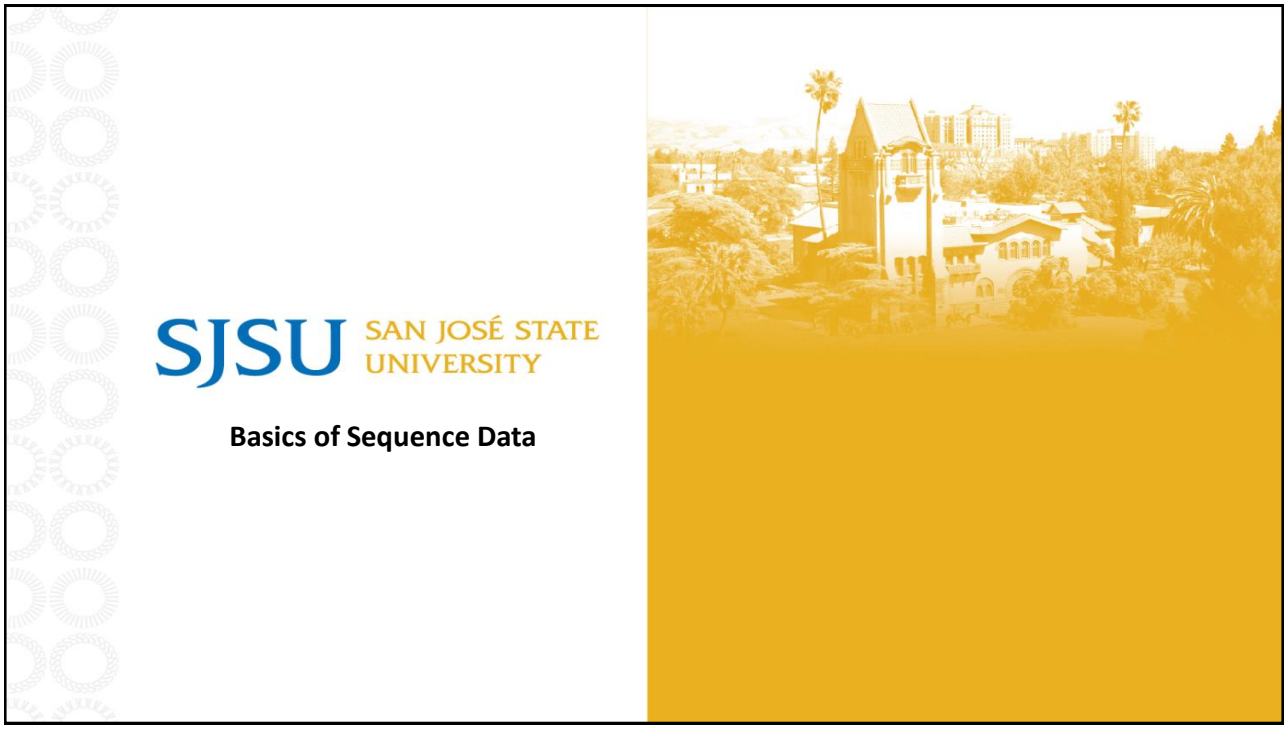
2

SJSU SAN JOSÉ STATE UNIVERSITY

Data Mining / Machine Learning Methods

	Vector Data	Set Data	Sequence Data / Time Series Data	Graph Data
Classification	Logistic Regression			
Clustering	K-means Hierarchical Clustering Gaussian Mixture Models			
Prediction / Regression	Linear Regression Generalized Linear Models			
Frequent Pattern Mining		Apriori	GSP; PrefixSpan	
Similarity Search				

3



4

SJSU SAN JOSÉ STATE UNIVERSITY

What is Sequence Data?

- A sequence dataset consists of ordered elements or events
- Recorded with or without notion of time
- Transaction dataset vs. sequence dataset

Transaction Dataset

TID	itemsets
10	a, b, d
20	a, c, d
30	a, d, e
40	b, e, f

Sequence Dataset

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

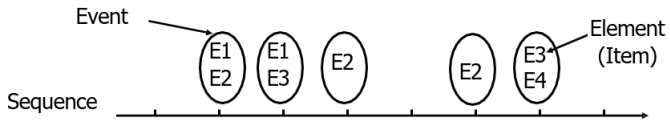
5

5

SJSU SAN JOSÉ STATE UNIVERSITY

Sequence Data

Sequence Database	Sequence	Event	Element
Customer	Purchase history of a given customer	A set of items bought by a customer at time t	Books, diary products, CDs, etc
Web Data	Browsing activity of a particular Web visitor	A collection of files viewed by a Web visitor after a single mouse click	Home page, index page, contact info, etc
Event data	History of events generated by a given sensor	Events triggered by a sensor at time t	Types of alarms generated by sensors
Genome sequences	DNA sequence of a particular species	An element of the DNA sequence	Bases A,T,G,C



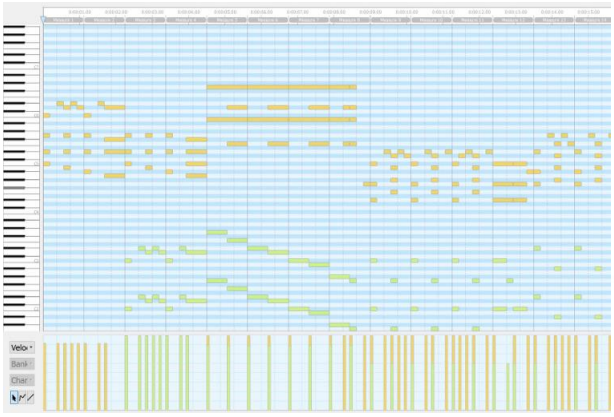
6

6

SJSU SAN JOSÉ STATE UNIVERSITY

Examples of Sequence Data

Music Sequence



DNA Sequence

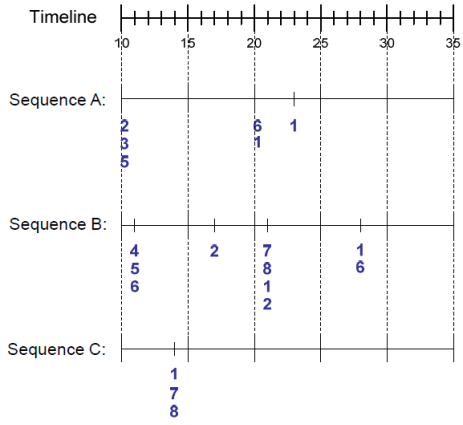
```
MD106  ATGCTTAGTAATCCCTACTTTAAGTCCGTTTTGTGGCTGATTGGCTTCGGAGGAATGGG
NEWC   ATGCTTAGTAATCCTTACTTTAAATCCGTTTTGTGGCTGATTGGCTTCGGAGGAATGGG
W501   ATGCTTAGTAATCCCTACTTTAAGTCCGTTTTGTGGCTGATTGGCTTCGGAGGAATGGG
MD199  ATGCTTAGTAATCCCTACTTTAAGTCCGTTTTGTGGCTGATTGGCTTCGGAGGAATGGG
C1674  ATGCTTAGTAATCCCTACTTTAAGTCCGTTTTGTGGCTGATTGGCTTCGGAGGAATGGG
SIM4   ATGCTTAGTAATCCCTACTTTAAGTCCGTTTTGTGGCTGATTGGCTTCGGAGGAATGGG

MD106  CTACGGCCTAATGGTGCTAACAGAGCCGAACGTCGACAAAATAGAGCGCATCAAAGCCT
NEWC   CTACGGCCTAATGGTGCTAACCGAGCCGAACGTCGACAAAATAGAGCGCATCAAAGCCT
W501   CTACGGCCTAATGGTGCTAACCGAGCCGAACGTCGACAAAATAGAGCGCATCAAAGCCT
MD199  CTACGGCCTAATGGTGCTAACCGAGCCGAACGTCGACAAAATAGAGCGCATCAAAGCCT
C1674  CTACGGCCTAATGGTGCTAACCGAGCCGAACGTCGACAAAATAGAGCGCATCAAAGCCT
SIM4   CTACGGCCTAATGGTGCTAACCGAGCCGAACGTCGACAAAATAGAGCGCATCAAAGCCT

MD106  CCGTTTCAAGTACCAAACTGAGTGCGGATGAGCAGCGAAAGGCTCTGTTTATGAAGAAG
NEWC   CCGTTTCAAGTACCAAACTGAGTGCGGATGAGCAGCGAAAGGCTCTGTTTATGAAGAAG
W501   CCGTTTCAAGTACCAAACTGAGTGCGGATGAGCAGCGAAAGGCTCTGTTTATGAAGAAG
MD199  CCGTTTCAAGTACCAAACTGAGTGCGGATGAGCAGCGAAAGGCTCTGTTTATGAAGAAG
C1674  CCGTTTCAAGTACCAAACTGAGTGCGGATGAGCAGCGAAAGGCTCTGTTTATGAAGAAG
SIM4   CCGTTTCAAGTACCAAACTGAGTGCGGATGAGCAGCGAAAGGCTCTGTTTATGAAGAAG

MD106  CTGCAGGAGGCGTCCACCACCAGTGCCCCAATCTACAGGTCAGCGGCCGAGAAATAG
NEWC   CTGCAGGAGGCGTCCACCACCAGTGCCCCAATCTACAGGTCATCGGCCGAGAAATAG
W501   CTGCAGGAGGCGTCCACCACCACTGCCCCAATCTACAGGTCATCGGCCGAGAAATAG
MD199  CTGCAGGAGGCGTCCACCACCAGTGCCCCAATCTACAGGTCAGCGGCCGAGAAATAG
C1674  CTGCAGGAGGCGTCCACCACCAGTGCCCCAATCTACAGGTCAGCGGCCGAGAAATAG
SIM4   CTGCAGGAGGCGTCCACCACCAGTGCCCCAATCTACAGGTCAGCGGCCGAGAAATAG
```


SJSU SAN JOSÉ STATE UNIVERSITY

Sequence Data with Timestamp

Sequence Database

Sequence ID	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, 8, 7



SJSU SAN JOSÉ STATE UNIVERSITY

Definitions of Sequences, Events…

- A sequence is an ordered list of events, denoted $< e_1\ e_2\ \dots\ e_l >$
- An event is a *non-empty* collection of items/elements: $e_1 =$ (abc)
- Given two sequences $\alpha = < a_1\ a_2\ \dots\ a_n >$ and $\beta = < b_1\ b_2\ \dots\ b_m >$
- α is called a subsequence of β, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \le j_1 < j_2 < \dots < j_n \le m$ such that $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}$,…, $a_n \subseteq b_{jn}$
- β is a super sequence of α
- Examples
  - α = <(ab)d> and β = < (abc)(de)>
  - α = <a(bc)dc> and β = <a(abc)(ac)d(cf)>

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

9

9

SJSU SAN JOSÉ STATE UNIVERSITY

Subsequence Exercises

Data Sequence	Subsequence	Contain?
< (2 4)(3 5 6)(8) >	< (2)(8) >	
< (1 2)(3 4) >	< (1)(2) >	
< (2 4)(2 4)(2 5) >	< (2)(4) >	
<(2 4)(2 5)(4 5)>	< (2)(4)(5) >	
<(2 4)(2 5)(4 5)>	< (2)(5)(5) >	
<(2 4)(2 5)(4 5)>	< (2 4 5) >	

10

10

SJSU SAN JOSÉ STATE UNIVERSITY

Definitions of Sequences, Events…

- Length of a sequence, |s|, is given by # of elements in the sequence

  e.g. What is the length of < (ef)(ab)(df)cb > ?

- A k-sequence is a sequence that contains k events:

  e.g. <(ab)(a)> is a 2-sequence and has a length of 3

SJSU SAN JOSÉ STATE UNIVERSITY

Support of a Sequence

- Support of a Sequence α: # of sequences in a sequence dataset D that contains a sequence α on and is denoted as sup(α):

$$\sup(\alpha) = |\{S \mid S \in D \text{ and } \alpha \sqsubseteq S\}|$$

- α is frequent if sup(α) ≥ σ (*minsup* threshold)
- A frequent sequence is called sequential pattern

SJSU SAN JOSÉ STATE UNIVERSITY

Example: Support of a Sequence

Consider the sequence dataset as shown:

- Example 1: sup(<(a)>) =?
- Example 2: sup(<(b)>) =?
- Example 3: sup(<(a)(b)>) =?
- Example 4: sup(<(a b)>) =?

SID	Sequence
10	<(ab)(c)(a)>
20	<(a)(b)(c)>
30	<(b)(c)(d)>
40	<(b)(ab)(c)>

sup(<(a)>) = 3 (75%)
sup(<(b)>) = 4 (100%)
sup(<(c)>) = 4 (100%)
sup(<(a)(c)>) = 3 (75%)
sup(<(ab)>) = 2 (50%)
sup(<(b)(c)>) = 4 (100%)
sup(<(ab)(c)>) = 2 (50%)

SID	Sequence
10	<(ab)(c)(a)>
20	<(a)(b)(c)>
30	<(b)(c)(d)>
40	<(b)(ab)(c)>

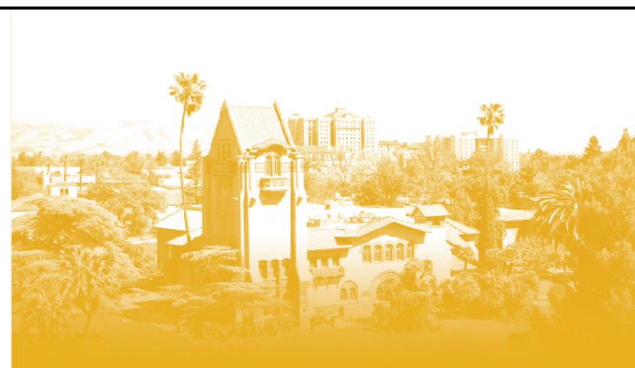
SID	Sequence
10	<(ab)(c)(a)>
20	<(a)(b)(c)>
30	<(b)(c)(d)>
40	<(b)(ab)(c)>

SID	Sequence
10	<(ab)(c)(a)>
20	<(a)(b)(c)>
30	<(b)(c)(d)>
40	<(b)(ab)(c)>

SID	Sequence
10	<(ab)(c)(a)>
20	<(a)(b)(c)>
30	<(b)(c)(d)>
40	<(b)(ab)(c)>

13

13



SJSU SAN JOSÉ STATE UNIVERSITY

Sequential Pattern Mining

14

SJSU SAN JOSÉ STATE UNIVERSITY

What Is Sequential Pattern Mining?

- Given a set of sequences and support threshold, find the complete set of **frequent subsequences**.

Sequence Data

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

Given *support threshold* *min_sup* =2, <(ab)c> is a *sequential pattern*

SJSU SAN JOSÉ STATE UNIVERSITY

Applications

Applications of Sequential Pattern Mining

- Customer shopping sequences:
  - First buy computer, then CD-ROM, and then digital camera, within 3 months.
- Medical treatments, natural disasters (e.g., earthquakes), science & eng. processes, stocks and markets, etc.
- Telephone calling patterns
- Weblog click streams
- DNA sequences and gene structures

SJSU SAN JOSÉ STATE UNIVERSITY

Example: Sequential Pattern Mining

Sequence ID	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

Frequent Subsequences:

sup<(1,2)> = 60%
sup<(2,3)> = 60%
sup<(2,4)> = 80%
sup<(3)(5)> = 80%
sup<(1)(2)> = 80%
sup<(2)(2)> = 60%
sup<(1)(2,3)> = 60%
sup<(2)(2,3)> = 60%
sup<(1,2)(2,3)> = 60%

17

SJSU SAN JOSÉ STATE UNIVERSITY

Extracting Sequential Patterns

- Given n items: $i_1, i_2, i_3, \ldots, i_n$
- Candidate 1-subsequences:

  $\langle(i_1)\rangle, \langle(i_2)\rangle, \langle(i_3)\rangle, \ldots, \langle(i_n)\rangle$
- Candidate 2-subsequences:

  $\langle(i_1, i_2)\rangle, \langle(i_1, i_3)\rangle, \ldots,$
  $\langle(i_1)\ (i_1)\rangle, \langle(i_1)\ (i_2)\rangle, \ldots, \langle(i_n)\ (i_n)\rangle$
- Candidate 3-subsequences:

  $\langle(i_1, i_2, i_3)\rangle, \langle(i_1, i_2, i_4)\rangle, \ldots,$
  $\langle(i_1, i_2)\ (i_1)\rangle, \langle(i_1, i_2)\ (i_2)\rangle, \ldots,$
  $\langle(i_1)\ (i_1, i_2)\rangle, \langle(i_1)\ (i_1, i_3)\rangle, \ldots,$
  $\langle(i_1)\ (i_1)\ (i_1)\rangle, \langle(i_1)\ (i_1)\ (i_2)\rangle, \ldots$

18

SJSU SAN JOSÉ STATE UNIVERSITY

Challenges on Sequential Pattern Mining

- A huge number of possible sequential patterns are hidden in dataset
- A mining algorithm should
  - find the complete set of patterns, when possible, satisfying the minimum support (frequency) threshold
  - be highly efficient, scalable, involving only a small number of database scans
  - be able to incorporate various kinds of user-specific constraints

SJSU SAN JOSÉ STATE UNIVERSITY

Generalized Sequential Pattern (GSP)

SJSU SAN JOSÉ STATE UNIVERSITY

The Apriori Property of Sequential Patterns

A basic property: Apriori (Agrawal & Sirkant'94)

- If a sequence S is not frequent, then none of the super-sequences of S is frequent.

Example: <abc> is infrequent ➔ <abcde> and <(abc)(de)> are infrequent too!

SJSU SAN JOSÉ STATE UNIVERSITY

Methods for Sequential Pattern Mining

- Apriori-based Approaches
  - GSP
  - SPADE
- Pattern-Growth-based Approaches
  - FreeSpan
  - PrefixSpan

SJSU SAN JOSÉ STATE UNIVERSITY

Recall The Apriori Algorithm

C_k: Candidate itemsets of size k
L_k: Frequent itemsets of size k

sup_{min} = 2

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan →

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

→ L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

1st Iteration

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan →

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

→ L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

2nd Iteration

Why no {A, D} or {B, D}?

C_3

Itemset
{B, C, E}

3rd scan →

Itemset	sup
{B, C, E}	2

→ L_3

Itemset	sup
{B, C, E}	2

3rd Iteration

Why no {A, C, E}?

23

23

SJSU SAN JOSÉ STATE UNIVERSITY

Generalized Sequential Pattern (GSP) Mining

- Step 1
  - Make the first pass over the sequence database D to yield all candidate of length-1
- Step 2 (Repeat until no new frequent sequences are found)
  - Candidate Generation:
    - Merge pairs of frequent subsequences found in the (k-1)th pass to generate candidate sequences that contain k items
  - Candidate Pruning:
    - Prune candidate length k sequences that contain infrequent length (k-1) subsequences
  - Support Counting:
    - Make a new pass over the sequence database D to find the support for these candidate sequences
  - Candidate Elimination:
    - Eliminate candidate length k sequences whose actual support is less than minsup

24

24

SJSU SAN JOSÉ STATE UNIVERSITY

Example: Finding Length-1 Sequential Patterns

- Initial candidates: all singleton sequences: <a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>
- Scan dataset once, count support for candidate.

min_sup =2

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

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

25

25

SJSU SAN JOSÉ STATE UNIVERSITY

Example: Finding Length-2 Sequential Patterns

- Now, generate length-2 candidates.

	<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>						

- 51 length-2 candidates!!!
- Without Apriori property, 8*8+8*7/2 = 92 candidates

➔ Apriori prunes 44.6% candidates!

26

26

SJSU SAN JOSÉ STATE UNIVERSITY

Generating Candidates in General

From L_{k-1} to C_k

- Step 1: Join
  - Sequences α and β can join, if dropping first item in α is the same as dropping the last item in β

  Examples:

  <(12)3> join <(2)34> ➔ <(12)34>

  <(12)3> join <(2)(34)> ➔ <(12)(34)>
- Step 2: Pruning
  - Check whether all length k-1 subsequences of a candidate is contained in L_{k-1}

SJSU SAN JOSÉ STATE UNIVERSITY

The GSP Mining Process

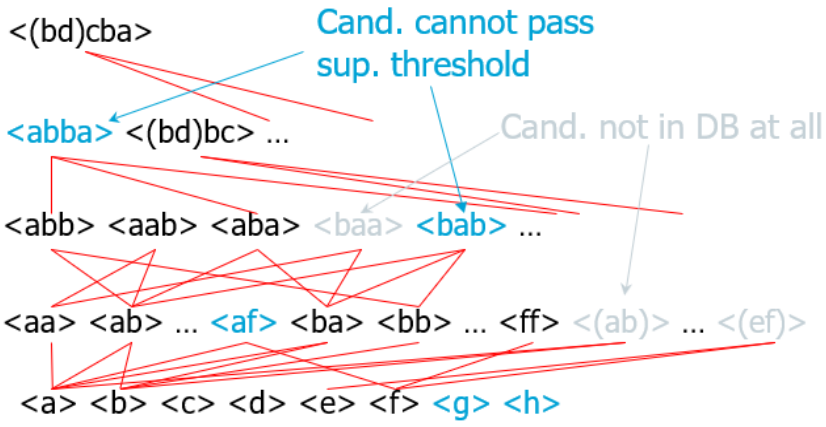
5th scan: 1 cand. 1 length-5 seq. pat.

4th scan: 8 cand. 7 length-4 seq. pat.

3rd scan: 46 cand. 20 length-3 seq. pat. (20 cand. not in DB at all)

2nd scan: 51 cand. 19 length-2 seq. pat. (10 cand. not in DB at all)

1st scan: 8 cand. 6 length-1 seq. pat.



SJSU SAN JOSÉ STATE UNIVERSITY

Drawbacks of GSP

- A huge set of candidate sequences generated (especially 2-item candidate sequence)
- Multiple scans of database needed. The length of each candidate grows by one at each database scan.
- Inefficient for mining long sequential patterns.
  - A long pattern grow up from short patterns
  - # of short patterns is exponential to the length of mined patterns.

SJSU SAN JOSÉ STATE UNIVERSITY

Bottlenecks of GSP

- A huge set of candidates could be generated
  - 1,000 frequent length-1 sequences generates huge number of length-2 candidates!
- Multiple scans of database in mining
- Breadth-first search
- Mining long sequential patterns
  - Needs an exponential number of short candidates
  - A length-100 sequential pattern needs ~10^{30} candidate sequences!
- Some of these can be addressed with PrefixSpan

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

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

SJSU SAN JOSÉ STATE UNIVERSITY

Sequential Pattern Mining

with PrefixSpan

31

SJSU SAN JOSÉ STATE UNIVERSITY

Prefix and Suffix

- Assume a pre-specified order on items, e.g., alphabetical order

  <a>, <aa>, <a(ab)>, <a(abc)> are prefixes of sequence <a(abc)(ac)d(cf)>

  Note: <a(ac)> is not a prefix of <a(abc)(ac)d(cf)>

- Given sequence <a(abc)(ac)d(cf)>:

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

- (_bc) means: the last element in the prefix together with (bc) form one element

32

32

SJSU SAN JOSÉ STATE UNIVERSITY

Prefix-based Projection

Given a sequence, α, let α' be subsequence of α:

- α' is called a projection of α with respect to prefix β, if and only if
  - α' has prefix β, and
  - α' is the maximum subsequence of α with prefix β

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

Example:

<(abc)(ac)d(cf)> is a projection of <a(abc)(ac)d(cf)> w.r.t. prefix <a>

<ad(cf)> is a projection of <a(abc)(ac)d(cf)> w.r.t. prefix <ad>

33

33

SJSU SAN JOSÉ STATE UNIVERSITY

Projected (Suffix) Database

- Let β be a sequential pattern, β-projected database is the collection of suffixes of projections of sequences in the database w.r.t. prefix β
- Examples:
  - <a>-projected database
    - <(abc)(ac)d(cf)>
    - <(_d)c(bc)(ae)>
    - <(_b)(df)cb>
    - <(_f)cbc>
  - <ab>-projected database
    - <(_c)(ac)d(cf)> (<a(bc)(ac)d(cf)> is the projection of <a(abc)(ac)d(cf)> w.r.t. prefix <ab>)
    - <(_c)(ae)> (<a(bc)(ae)> is the projection of <(ad)c(bc)(ae)> w.r.t. prefix <ab>)
    - <c> (<abc> is the projection of <eg(af)cbc> w.r.t prefix <ab>)

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

34

34

SJSU SAN JOSÉ STATE UNIVERSITY

Examples

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

Starting with <a>-projected database
- <(abc)(ac)d(cf)>
- <(_d)c(bc)(ae)>
- <(_b)(df)cb>
- <(_f)cbc>

<aa>-projected database
- <(_bc)(ac)d(cf)>
- <(_e)>

<ab>-projected database
- <(_c)(ac)d(cf)>
- <(_c)(ae)>
- <c>

<(ab)>-projected database
- <(_c)(ac)d(cf)>
- <(df)cb>

SJSU SAN JOSÉ STATE UNIVERSITY

Mining Sequential Patterns by Prefix Projections

- Step 1: Find length-1 sequential patterns.
  <a>, <b>, <c>, <d>, <e>, <f>
- Step 2: Divide search space. The complete set of sequential patterns can be partitioned into 6 subsets:
  The ones having prefix <a>
  The ones having prefix <b>
  …
  The ones having prefix <f>
- Step 3: Mine each subset recursively via corresponding projected databases.

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

SJSU SAN JOSÉ STATE UNIVERSITY

Finding Sequential Patterns with Prefix <a>

- Consider only projections w.r.t. <a>:
  - <a>-projected (suffix) database:
    <(abc)(ac)d(cf)>
    <(_d)c(bc)(ae)>
    <(_b)(df)cb>
    <(_f)cbc>

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

- Find all the length-2 sequential patterns having prefix <a>:
  <aa>, <ab>, <(ab)>, <ac>, <ad>, <af>
  - Further partition into 6 subsets:
    Having prefix <aa>
    …
    Having prefix <af>

Why are those 6 subsets?

SJSU SAN JOSÉ STATE UNIVERSITY

Finding Sequential Patterns with Prefix <a>

- By scanning the <a>-projected database once, its locally frequent items are identified as
  a : 2, b : 4, _b : 2, c : 4, d : 2, and f : 2.
- Thus, all the length-2 sequential patterns prefixed with <a> are found:
  <aa>: 2, <ab>: 4, <(ab)>: 2, <ac> : 4, <ad>: 2, and <af>: 2.

SJSU SAN JOSÉ STATE UNIVERSITY

Completeness of PrefixSpan

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

Length-1 sequential patterns
<a>, <b>, <c>, <d>, <e>, <f>

Having prefix <a>

Having prefix <b>

Having prefix <c>, …, <f>

<a>-projected database
<(abc)(ac)d(cf)>
<(_d)c(bc)(ae)>
<(_b)(df)cb>
<(_f)cbc>

Length-2 sequential patterns
<aa>, <ab>, <(ab)>,
<ac>, <ad>, <af>

<b>-projected database
<(_c)(ac)d(cf)>
<(_c)(ae)>
<(df)cb>
<c>

…

Having prefix <aa>

Having prefix <af>

<aa>-proj. db … <af>-proj. db … …

39

39

SJSU SAN JOSÉ STATE UNIVERSITY

Projected Databases & Sequential Patterns

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

prefix	projected (suffix) database	sequential patterns
$\langle a \rangle$	$\langle (abc)(ac)d(cf) \rangle$, $\langle (\_d)c(bc)(ae) \rangle$, $\langle (\_b)(df)cb \rangle$, $\langle (\_f)cbc \rangle$	$\langle a \rangle$, $\langle aa \rangle$, $\langle ab \rangle$, $\langle a(bc) \rangle$, $\langle a(bc)a \rangle$, $\langle aba \rangle$, $\langle abc \rangle$, $\langle (ab) \rangle$, $\langle (ab)c \rangle$, $\langle (ab)d \rangle$, $\langle (ab)f \rangle$, $\langle (ab)dc \rangle$, $\langle ac \rangle$, $\langle aca \rangle$, $\langle acb \rangle$, $\langle acc \rangle$, $\langle ad \rangle$, $\langle adc \rangle$, $\langle af \rangle$
$\langle b \rangle$	$\langle (\_c)(ac)d(cf) \rangle$, $\langle (\_c)(ae) \rangle$, $\langle (df)cb \rangle$, $\langle c \rangle$	$\langle b \rangle$, $\langle ba \rangle$, $\langle bc \rangle$, $\langle (bc) \rangle$, $\langle (bc)a \rangle$, $\langle bd \rangle$, $\langle bdc \rangle$, $\langle bf \rangle$
$\langle c \rangle$	$\langle (ac)d(cf) \rangle$, $\langle (bc)(ae) \rangle$, $\langle b \rangle$, $\langle bc \rangle$	$\langle c \rangle$, $\langle ca \rangle$, $\langle cb \rangle$, $\langle cc \rangle$
$\langle d \rangle$	$\langle (cf) \rangle$, $\langle c(bc)(ae) \rangle$, $\langle (\_f)cb \rangle$	$\langle d \rangle$, $\langle db \rangle$, $\langle dc \rangle$, $\langle dcb \rangle$
$\langle e \rangle$	$\langle (\_f)(ab)(df)cb \rangle$, $\langle (af)cbc \rangle$	$\langle e \rangle$, $\langle ea \rangle$, $\langle eab \rangle$, $\langle eac \rangle$, $\langle eacb \rangle$, $\langle eb \rangle$, $\langle ebc \rangle$, $\langle ec \rangle$, $\langle ecb \rangle$, $\langle ef \rangle$, $\langle efb \rangle$, $\langle efc \rangle$, $\langle efcb \rangle$.
$\langle f \rangle$	$\langle (ab)(df)cb \rangle$, $\langle cbc \rangle$	$\langle f \rangle$, $\langle fb \rangle$, $\langle fbc \rangle$, $\langle fc \rangle$, $\langle fcb \rangle$

40

40

SJSU SAN JOSÉ STATE UNIVERSITY

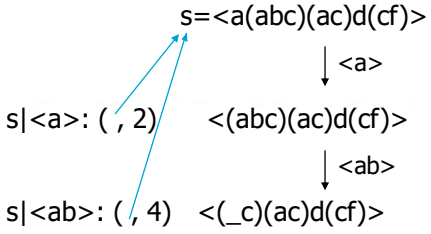
Efficiency of PrefixSpan

- No candidate sequence needs to be generated
- Projected databases keep shrinking
- Major cost of PrefixSpan: Constructing projected databases
- Can be improved by pseudo-projections (virtual projection in memory)

SJSU SAN JOSÉ STATE UNIVERSITY

Speed-up by Pseudo-Projection

- Major cost of PrefixSpan: projection
  - Postfixes of sequences often appear repeatedly in recursive projected databases
- When (projected) database can be held in main memory, use pointers to form projections:
  - Pointer to the sequence
  - Offset of the postfix



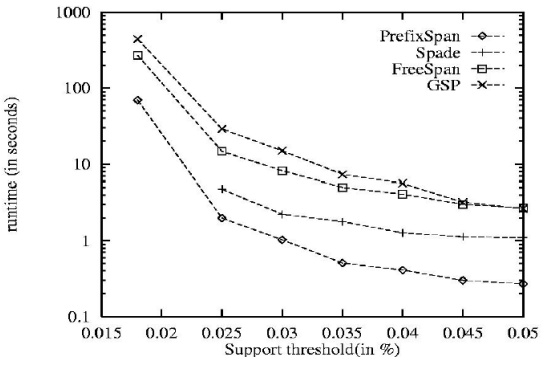
SJSU SAN JOSÉ STATE UNIVERSITY

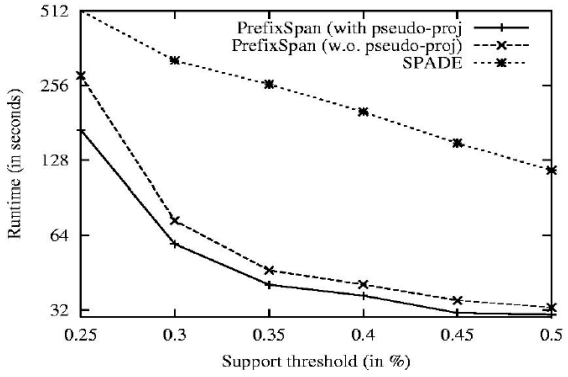
Pseudo-Projection vs. Physical Projection

- Pseudo-projection avoids physically copying postfixes
  - Efficient in running time and space when database can be held in main memory
- However, it is not efficient when database cannot fit in main memory
  - Disk-based random accessing is very costly
- Suggested Approach:
  - Integration of physical and pseudo-projection
  - Swapping to pseudo-projection when the data set fits in memory

SJSU SAN JOSÉ STATE UNIVERSITY

Performance Comparison





SJSU SAN JOSÉ STATE UNIVERSITY

Sequential Rules

45

SJSU SAN JOSÉ STATE UNIVERSITY

Sequential Pattern Implications

- <af> is a sequential pattern in the dataset:

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

- Does it imply that if someone buys <a>, he'll also buy <f> later?
- Not really, <a> implies <f> only 50% of the time! ➔ Need sequential rules!

46

SJSU SAN JOSÉ STATE UNIVERSITY

Sequential Rules

Two main types of sequential rules

- Standard Sequential rules
- Partially-ordered Sequential rules

SJSU SAN JOSÉ STATE UNIVERSITY

Standard Sequential Rules

Standard Sequential Rules: Rules of the form $X \rightarrow Y$, where X and Y are sequential patterns.

Example: <a(bc)> → <de>

- Two thresholds must be set by the user:
  - minimum support > 0
  - minimum confidence > 0
- Support: how many sequences contain the rule
- Confidence: how many sequences contain the rule divided by how many sequences contain its antecedent

SJSU SAN JOSÉ STATE UNIVERSITY

Example: Standard Sequential Rule

- Consider the standard sequential rule: <ab> → <f>

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

Support: 1 sequence (25%)

Confidence: 1 / 4 = 0.25 (25%)

SJSU SAN JOSÉ STATE UNIVERSITY

Order Matters?

- Notice that some sequential rules that are very similar but have only some small ordering variations.
- Example: <ab> → <f> and <ba> → <f>

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

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

Rule	Support	Confidence
<ab> → <f>	25%	25%
<ba> → <f>	25%	50%

SJSU SAN JOSÉ STATE UNIVERSITY

Partially-Ordered Sequential Rules

Partially-Ordered Sequential Rules: Rules of the form $X \rightarrow Y$, where X and Y are itemsets that are unordered.

Example: {a,b} → {f}

- If we observe a and b (in any order), they will be followed by f

SJSU SAN JOSÉ STATE UNIVERSITY

Partially-Ordered Sequential Rules

- This type of rule is often more interesting because it can summarize many standard sequential rules
- Example:

$\{Vivaldi\}, \{Mozart\}, \{Handel\} \Rightarrow \{Berlioz$
$\{Mozart\}, \{Vivaldi\}, \{Handel\} \Rightarrow \{Berlioz\},$
$\{Handel\}, \{Vivaldi\}, \{Mozart\} \Rightarrow \{Berlioz\},$
$\{Handel, Vivaldi\}, \{Mozart\} \Rightarrow \{Berlioz\},$
$\{Handel\}, \{Vivaldi, Mozart\} \Rightarrow \{Berlioz\},$
$\{Handel, Vivaldi, Mozart\} \Rightarrow \{Berlioz\}.$

Standard sequential rules

$\{Mozart, Vivaldi, Handel\} \Rightarrow \{Berlioz\}$

Partially-ordered sequential rules

SJSU SAN JOSÉ STATE UNIVERSITY

Partially-Ordered Sequential Rules

- A partially-ordered sequential rule X → Y is a relationship between two disjoint and non-empty itemsets X , Y .
- A sequential rule X → Y has two properties:
  - Support: #of sequences where X occurs before Y, divided by the number of sequences.
  - Confidence: # of sequences where X occurs before Y, divided by the number of sequences where X occurs.
- Need to find all valid rules, rules with a support and confidence not less than user-defined thresholds minSup and minConf

SJSU SAN JOSÉ STATE UNIVERSITY

Example: Mining Partially-Ordered Sequential Rules

- Consider the following sequence database with minSup= 0.5 and minConf= 0.5

ID	Sequences
seq1	{*a*, *b*},{*c*},{*f*},{*g*},{*e*}
seq2	{*a*, *d*},{*c*},{*b*},{*a*, *b*, *e*, *f*}
seq3	{*a*},{*b*},{*f*},{*e*}
seq4	{*b*},{*f*, *g*}

ID	Rule	Support	Confidence
r1	{*a*, *b*, *c*}⇒{*e*}	0.5	1.0
r2	{*a*}→{*c*, *e*, *f*}	0.5	0.66
r3	{*a*, *b*} →{*e*, *f*}	0.75	1.0
r4	{*b*} →{*e*, *f*}	0.75	0.75
r5	{*a*} →{*e*, *f*}	0.75	1.0
r6	{*c*} →{*f*}	0.5	1.0
r7	{*a*}→{*b*}	0.5	0.66
…	…	…	…

SJSU SAN JOSÉ STATE UNIVERSITY

Summary

- Basic concepts: sequential patterns etc
- Scalable sequential pattern mining methods
  - GSP
  - PrefixSpan
- Sequential Rules: Standard and Partially-Ordered