Constrained Frequent Pattern Mining: A Pattern-Growth View

by Jian Pei and Jiawei Han
presentation by Rafal Rak
CMPUT 695 presentation

Scope

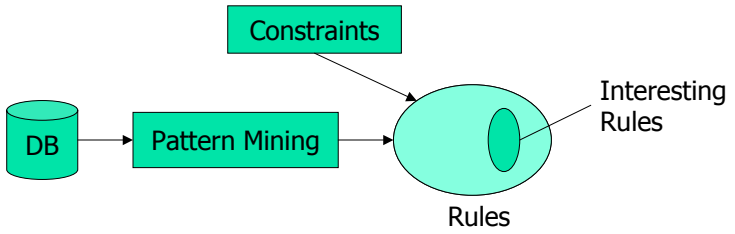
Constrained Frequent Pattern Mining:
A Pattern-Growth View

- Frequent Pattern Mining
- Constraints
- Pattern-Growth Approach

Outline

- Background
- Categories of Constraints
- Pattern-growth Method
  - Constrained Frequent Pattern Mining
  - Constrained Sequential Pattern Mining
- Conclusion

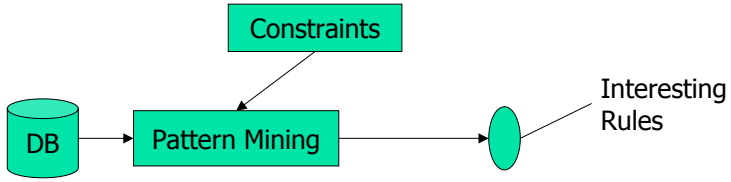
Pushing Constraints



Drawbacks

- Pattern Mining: inefficient
- Rules: ineffective

Pushing Constraints (2)



- Efficient and effective
- Feasible?

Apriori Approach

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Apriori anti-monotone property:
if a pattern is not frequent, its super-pattern can never be frequent

min(S)>15
min(df)<15
=> min(adf)<15
↓
anti-monotone

max(S)>35
max(df)<35
but max(adf)>35
↓
not anti-monotone

Categories of Constraints

- Application point of view
  - Item constraint
    e.g. dairy products in a grocery store
  - Length constraint
    e.g. at least 5 keywords in documents
  - Model-based constraint
    e.g. travel agency: after visiting Washington and NYC, what's next?
  - Aggregate constraint
    e.g. avg(price of items) > $100

Categories of Constraints (2)

- Properties point of view
  - Anti-monotone
    e.g. min(S)>v
  - Monotone
    e.g. max(S)>v
  - Succinct
  - Convertible Constraints

Anti-monotone Constraint

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

When an itemset violates the constraint, so does any of its superset.

min(S)>15

min(df)<15 => min(adf)<15

~~min(S)<15~~

min(af)>15, but ~~min(adf)<15~~

Monotone Constraint

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

When an itemset satisfies the constraint, so does any of its superset.

max(S)>35

max(af)>35 => max(adf)>35

~~max(S)<35~~

max(df)<35, but ~~max(adf)>35~~

Succinct Constraint

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

If it's possible to explicitly and precisely generate all the itemsets satisfying the constraint, then the constraint is succinct.

max(S)>15

itemsets containing: a, f, g

~~avg(S)<10~~

Convertible Anti-monotone Constraint

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

order

Item	Value
a	40
f	30
g	20
d	10
b	0
h	-10
c	-20
e	-30

A constraint is convertible anti-monotone if there is an order on items such that whenever an itemset satisfies the constraint, so does any of its prefix.

avg(S)>25

avg(afg)>25

=> avg(af)>25, avg(a)>25, avg(f)>25

~~avg(S)<32~~

avg(afg)<32, but ~~avg(af)>32~~

Convertible Monotone Constraint

A constraint is convertible monotone if there is an order on items such that whenever an itemset violates the constraint, so does any of its prefix.

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

order

Item	Value
e	-30
c	-20
h	-10
b	0
d	10
g	20
f	30
a	40

avg(S)>0

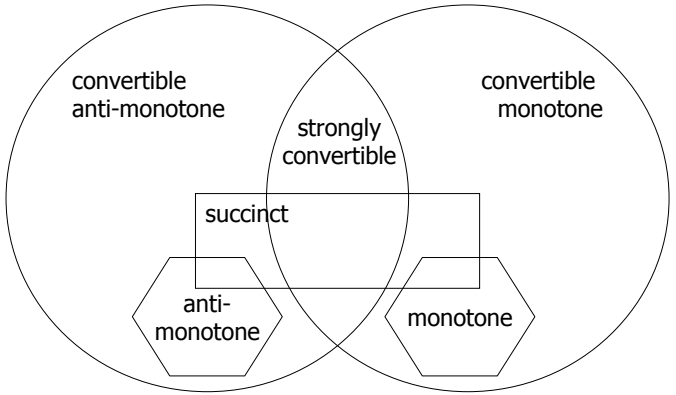
avg(ech)<0

=> avg(ec)<0, avg(e)<0, avg(c)<0

~~avg(S)<-22~~

avg(ech)>-22, but ~~avg(ec)<-22~~

Classification of Constraints



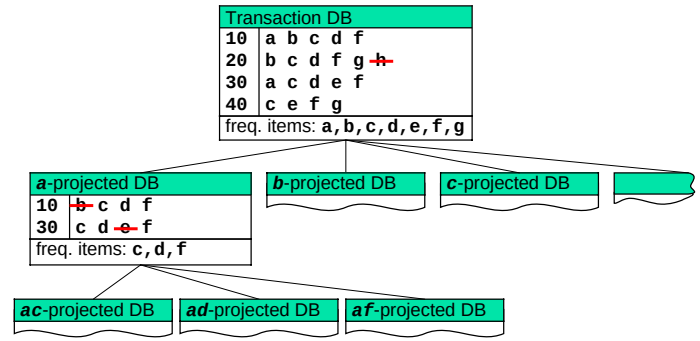
Outline

- Background
- Categories of Constraints
- Pattern-growth Method
  - Constrained Frequent Pattern Mining
  - Constrained Sequential Pattern Mining
- Conclusion

Pattern-growth Method

min_sup=2

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g



final patterns:
a,b,c,d,e,f,g,
ac,ad,af,bc,bd,bf,cd,ce,cf,cg,df,ef,fg,
acd,acdf,adf,bcd,bcf,cdf,cef,cfg,bdf,
bcdf

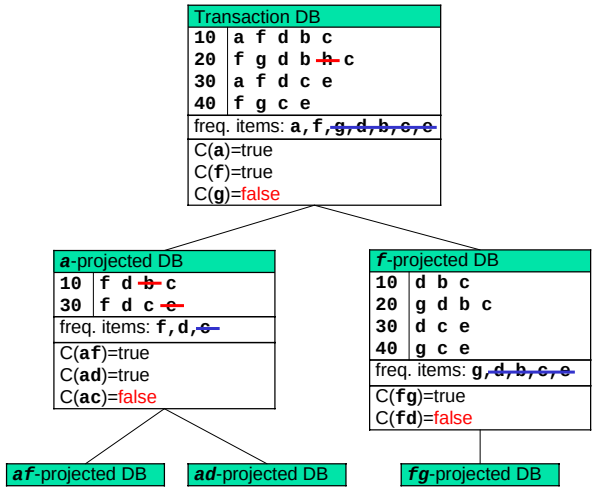
Constrained Frequent Pattern Mining

avg(S)≥25

min_sup=2

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
f	30
g	20
d	10
b	0
h	-10
c	-20
e	-30



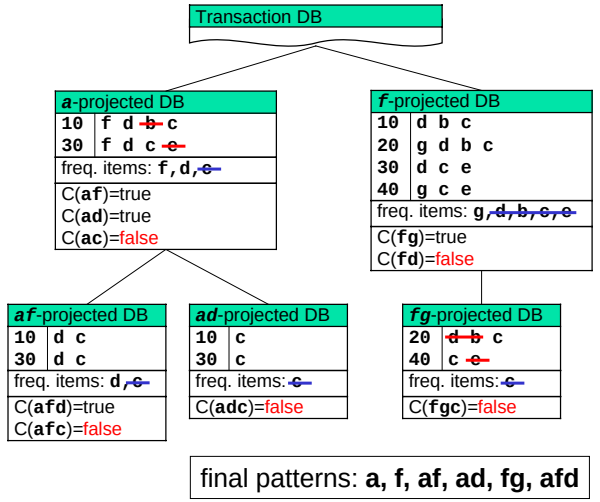
Constrained Frequent Pattern Mining

avg(S)≥25

min_sup=2

TID	Items
10	a,b,c,d,f
20	b,c,d,f,g,h
30	a,c,d,e,f
40	c,e,f,g

Item	Value
a	40
f	30
g	20
d	10
b	0
h	-10
c	-20
e	-30



final patterns: **a, f, af, ad, fg, afd**

Outline

- Background
- Categories of Constraints
- Pattern-growth Method
  - Constrained Frequent Pattern Mining
  - Constrained Sequential Pattern Mining
- Conclusion

Sequential Pattern Mining

Customer ID	Transaction Time	Items Bought
10	Nov. 10	a
	Nov. 13	bc
	Nov. 15	e
20	Oct. 30	e
	Nov. 3	ab
	Nov. 12	bc
	Nov. 13	d
	Nov. 14	d
30	Oct. 30	c
	Nov. 4	aef
	Nov. 13	abc
	Nov. 16	dd
40	Oct. 25	a
	Nov. 5	d
	Nov. 11	d
	Nov. 13	c
	Nov. 14	b

5-sequence

length = number of transactions

< e (ab) (bc) d d >

transactions in order

SID	Sequence
10	<a(bc)e>
20	<e(ab)(bc)dd>
30	<c(aef)(abc)dd>
40	<addcb>

Sequential Pattern Mining (2)

SID	Sequence
10	<a(bc)e>
20	<e(ab)(bc)dd>
30	<c(aef)(abc)dd>
40	<addcb>

Goal:

Find the complete set of sequential patterns w.r.t. a given sequence DB and a support threshold.

<(ab)d> is a subsequence of:

- <e(ab)(bc)dd>
- <c(aef)(abc)dd>
- but not <c(aef)(bc)dd>

Given a support threshold = 2, <(ab)d> is a sequential pattern.

More Constraints

- Regular expression constraint

  e.g. web click stream starting from Yahoo's home page and reaching hotels in NYC: „Travel (New York | New York City) (Hotels | Motels)"
- Duration constraint

  e.g. long-term investment patterns based on the duration of 1 year between the first and the last items
- Gap constraint

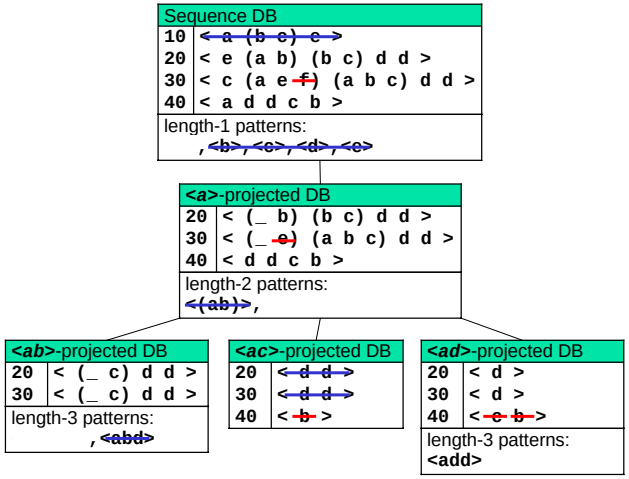
  e.g. basketball players regularly (every week) on the field, i.e., gap < 2 weeks

Constrained Sequential Pattern Mining

<a+{bb|(bc)d|dd}>

min_sup = 2

SID	Sequence
10	<a(bc)e>
20	<e(ab)(bc)dd>
30	<c(aef)(abc)dd>
40	<addcb>

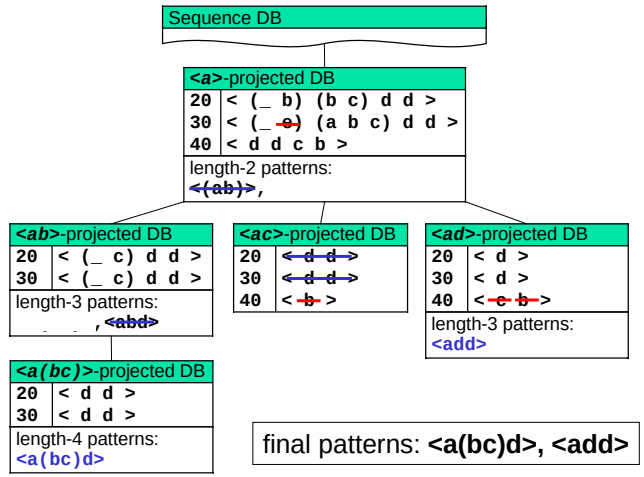


Constrained Sequential Pattern Mining

<a+{bb|(bc)d|dd}>

min_sup = 2

SID	Sequence
10	<a(bc)e>
20	<e(ab)(bc)dd>
30	<c(aef)(abc)dd>
40	<addcb>



final patterns: **<a(bc)d>, <add>**

Conclusion

- Pushing constraints deep into the mining process is efficient and effective
- Constraints can be classified
- Pattern-growth approach is more powerful then the Apriori-based one w.r.t. „tough” constraints
- Pattern-growth methods: FP-growth, PrefixSpan
- „Tough” constraints can be also applied to depth-first algorithms: First Search, MAFIA, CHARM
- Pattern-growth approach can be extended to structured pattern mining (trees, graphs), classification, clustering, outlier analysis

References

1. J. Pei, J. Han. Constrained Frequent Pattern Mining: A Pattern-growth View. *ACM SIGKDD Explorations (Special Issue on Constrained Data Mining)*, 2002.
2. J. Pei, J. Han, and L.V.S. Lakshmanan. Mining Frequent Itemsets with Convertible Constraints. *Proc. 2001 Int. Conf. Data Engineering*, 2001.
3. J. Pei, J. Han, and W. Wang. Constraint-based Sequential Pattern Mining in Large Databases. *Proc. 11th Int. Conf. on Information and Knowledge Management (CIKM'02)*, 2002.
4. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. *Proc. 2001 Int. Conf. Data Engineering (ICDE'01)*, 2001.

Thank you!

Questions?