Faster Association Rules for Multiple Relations

Siegfried Nijssen, Joost Kok

snijssen@liacs.nl, joost@liacs.nl

Universiteit Leiden

Leiden Institute of Advanced Computer Science

The Netherlands

Overview

- Basic problem: find "propositional" frequent item sets:
  - ▶ description;
  - ▶ algorithm;
- Extended problem: find "first order" frequent item sets:
  - ▶ description;
  - ▶ $\boxed{\text{algorithm;}}$
  - ▶ experiment.

Frequent item sets - problem

- Many companies have transaction databases:

Tid	products
id1	chips, donuts, cola
id2	chips, cheese, cola
id3	cheese, donuts, cola
id4	chips, donuts, cola, sugar
id5	chips
id6	cheese, cola

- they search for relations between items:
  - ▶ “$\boxed{\text{many}}$ people who buy chips and donuts, buy cola”
  - ▶ “cheese and cola are $\boxed{\text{frequently}}$ bought together”

Frequent item sets - problem

- Transform database into binary form:

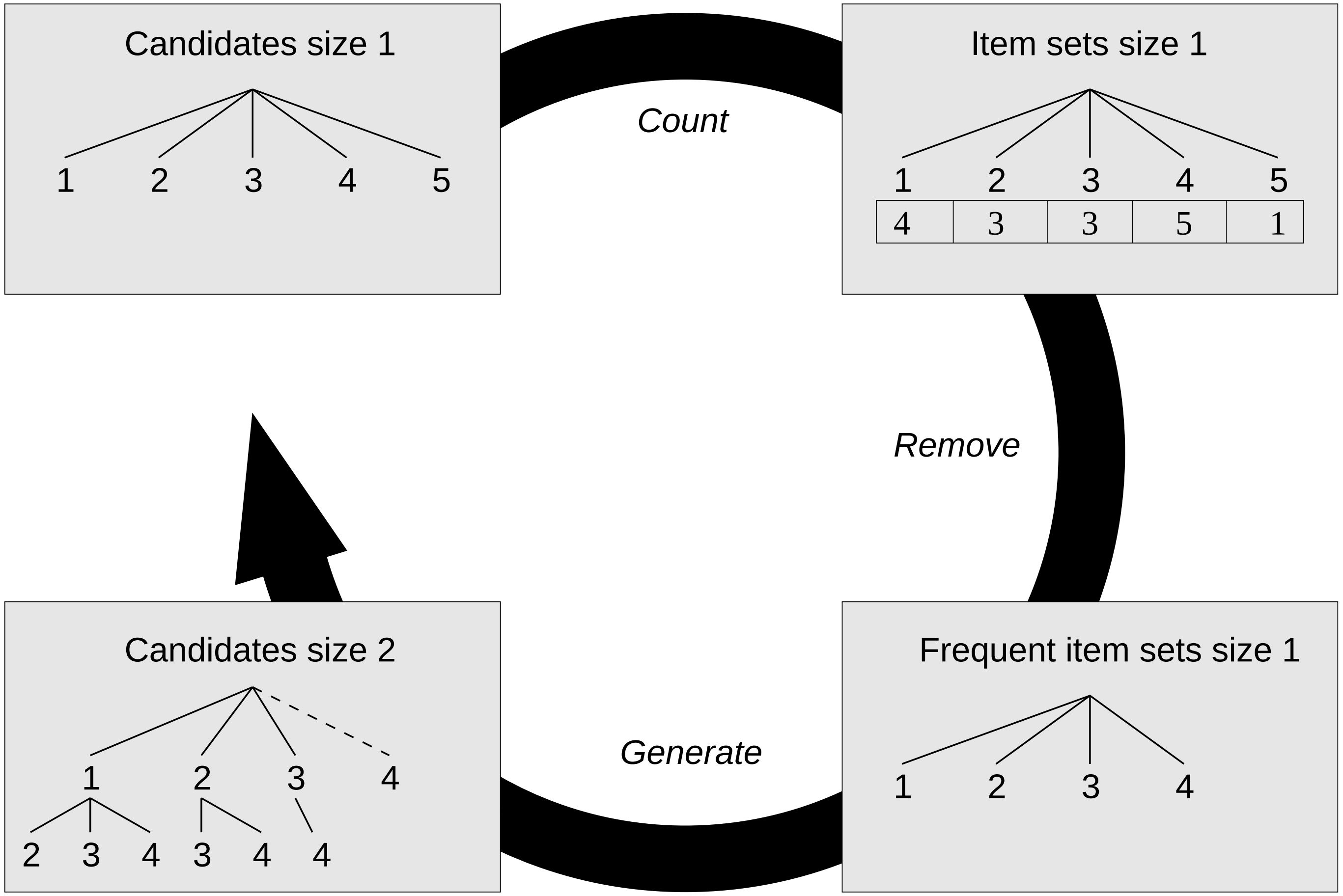
Tid	products
id1	chips, donuts, cola
id2	chips, cheese, cola
id3	cheese, donuts, cola
id4	chips, donuts, cola, sugar
id5	chips
id6	cheese, cola

$\Rightarrow$

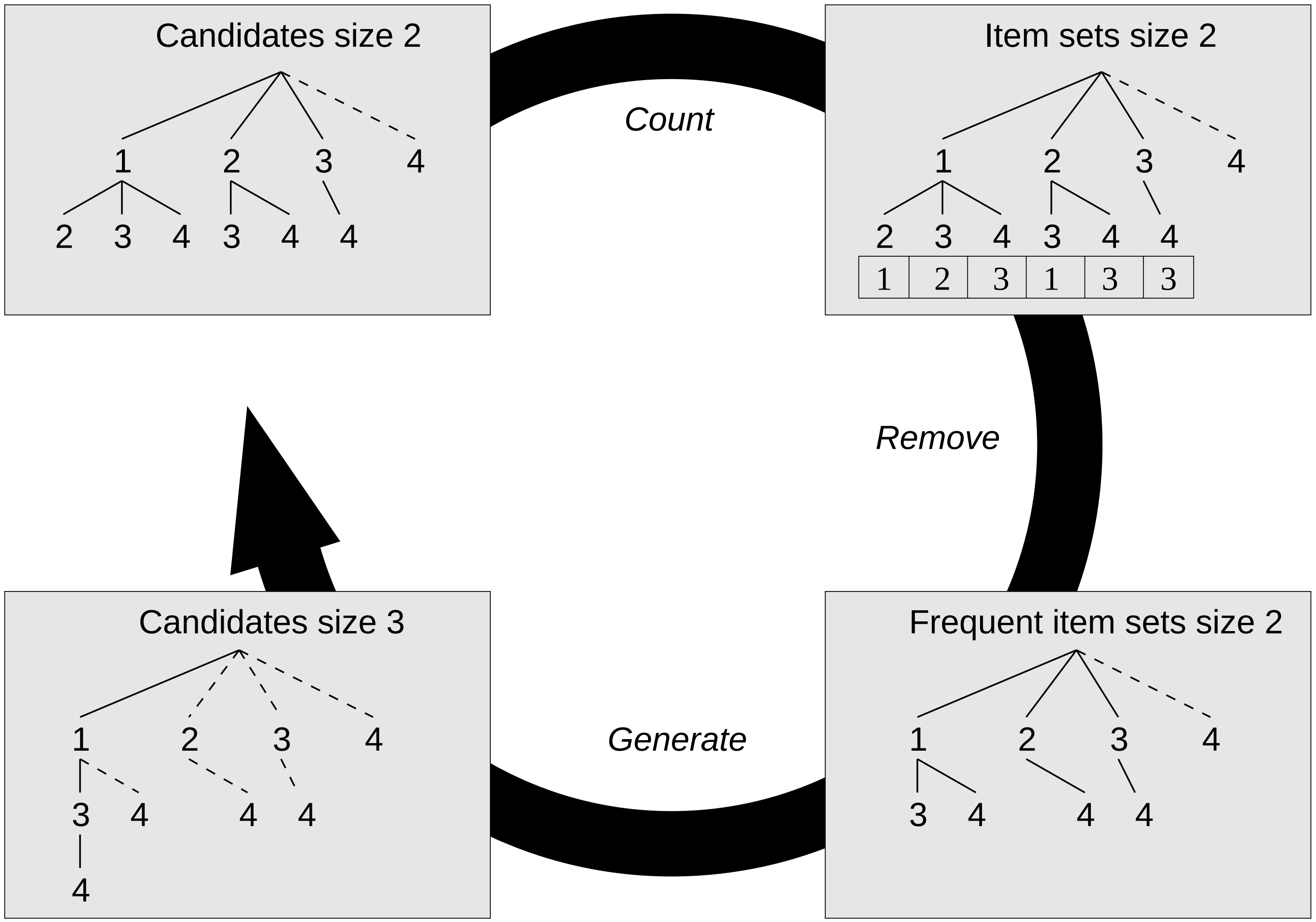
	1	2	3	4	5
1	1	0	1	1	0
2	1	1	0	1	0
3	0	1	1	1	0
4	1	0	1	1	1
5	1	0	0	0	0
6	0	1	0	1	0

- search for:
  - ▶ all unique subsets of products which
  - ▶ occur in more transactions than a predefined threshold.

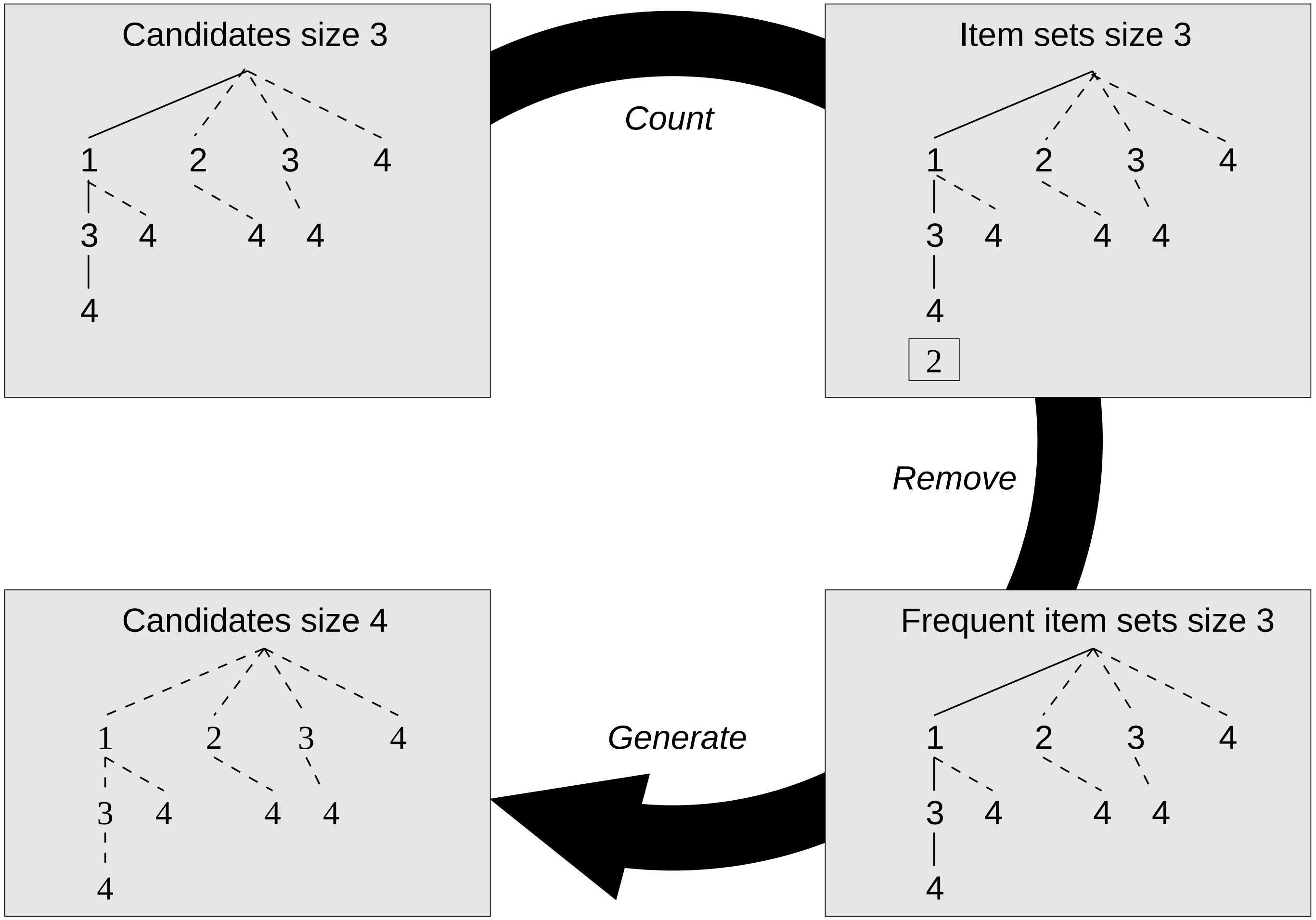
Frequent item sets - an algorithm



Frequent item sets - an algorithm



Frequent item sets - an algorithm



- In practice: more databases

Sales receipts

Tid	shop	products
id1	A	chips, donuts, cola
id2	B	chips, cheese, cola
id3	A	cheese, donuts, cola
id4	C	chips, donuts, cola, sugar
id5	B	chips
id6	A	cheese, cola

Shop data

shop	has	clients from
A	parking,fresh bread dep.	a,b
B	parking	b
C	fresh bread dep.	c

Area data

area	income	mean age
a	low	low
b	high	medium
c	low	high

Frequent queries - problem

- Frequent item sets $\Rightarrow$ frequent queries:
  - " $\boxed{\text{many}}$ donuts are bought in shops with a fresh department and clients from poor areas"
    Or: (for many T) $\exists$ `S,A products(T,donuts), shop(T,S), has(S,freshdep), client(S,A), income(A,low)`
  - " cheese and cola are $\boxed{\text{frequently}}$ bought together in shops with a parking.
    Or: (for many T) $\exists$ `S products(T,donuts), products(T,cola), shop(T,S), has(S,parking)`

Frequent queries - problem

- Transform database into logic facts:

  product(id1,chips).
  product(id1,donuts).
  shop(id1,A).
  has(A,parking).
  ...

- search for:
  - rules which are true for more examples than a threshold;
  - rules which confirm to a *bias* and are unique by θ-subsumption.

Frequent queries - problem

- Bias can be expressed using *modes*:

product(+T,chips)
product(+T,donuts)
shop(+T,-S)
has(+S,parking)
has(+S,freshdep)

Bias

T	S	query
T		
T		`product(T,donuts)`
T		`product(T,donuts),product(T,chips)`

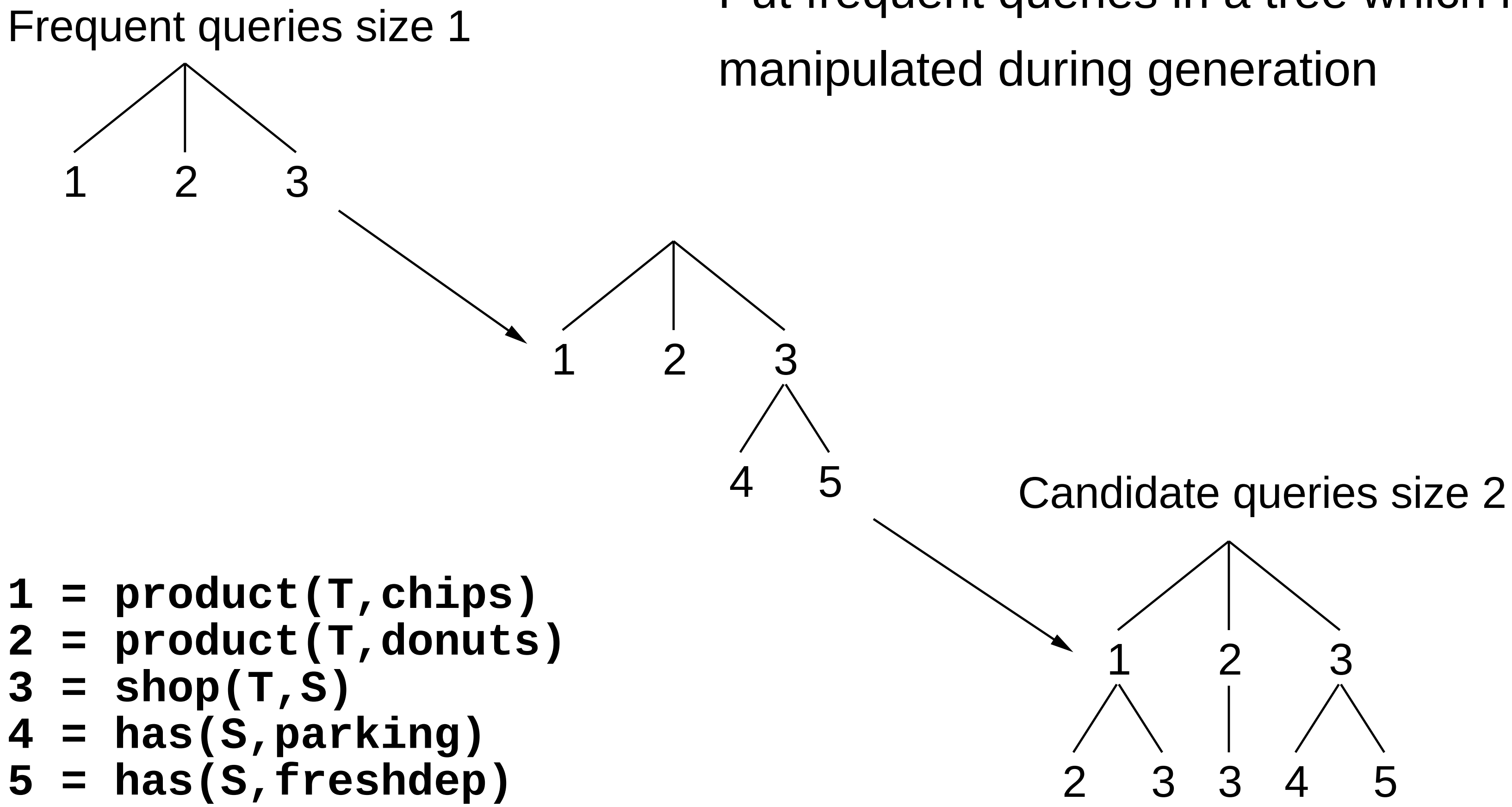
T	S	query
T		
T	S	`shop(T,S)`
T	S	`shop(T,S),has(S,parking)`

Examples

Frequent queries - solutions

- Farmer target: $\boxed{\text{more efficiency}}$ by
  - $\boxed{\text{changing query generation;}}$
  - reimplementing the algorithm.

- Put frequent queries in a tree which is manipulated during generation




```
1 = product(T,chips)
2 = product(T,donuts)
3 = shop(T,S)
4 = has(S,parking)
5 = has(S,freshdep)
```

Frequent queries - solutions

- The output of Farmer is equal to Warmr if:
  - no functions in a bias;
  - no two modes for the same predicate:

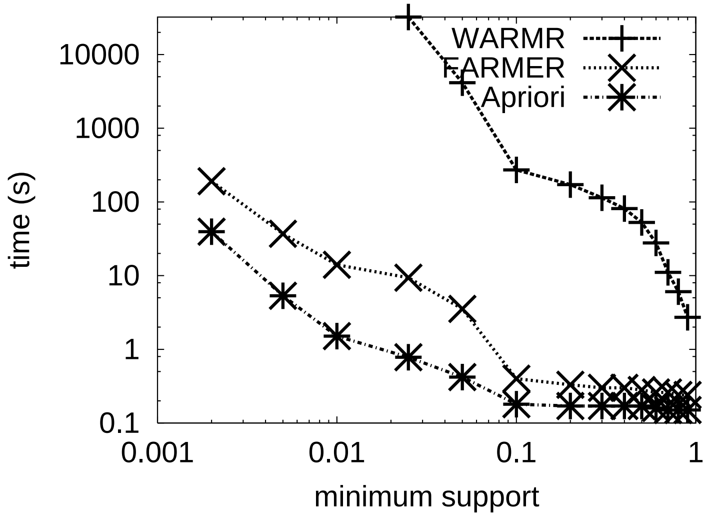
    has(+S,-X)
    has(+S,freshdep) $\Rightarrow$ `shop(T,S),has(S,X),has(S,freshdep).`

  - repetition of mode usage not allowed:

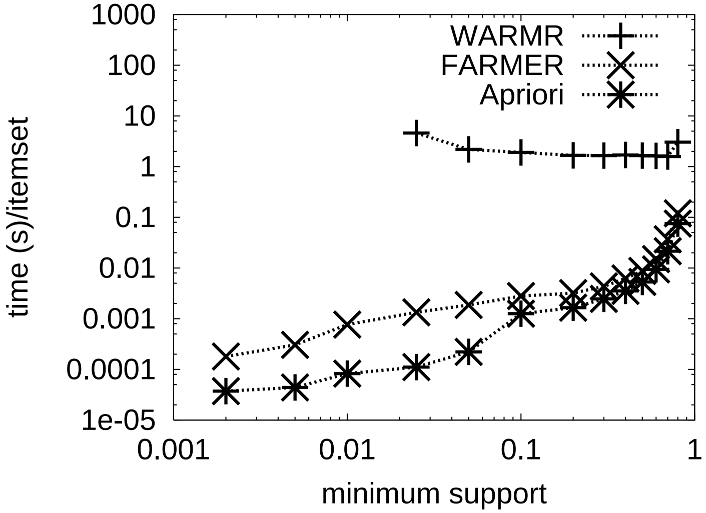
    shop(+T,-S)
    has(+S,freshdep) $\Rightarrow$
    has(+S,parking)

    $\texttt{shop(T,S}_1\texttt{),has(S}_1\texttt{,parking),}$
    $\quad\texttt{shop(T,S}_2\texttt{),has(S}_2\texttt{,freshdep).}$
    $\texttt{shop(T,S}_1\texttt{),has(S}_1\texttt{,freshdep),}$
    $\quad\texttt{shop(T,S}_2\texttt{),has(S}_2\texttt{,parking).}$

- Experimental results:
  - ▶ Apriori, Farmer, Warmr on frequent item sets



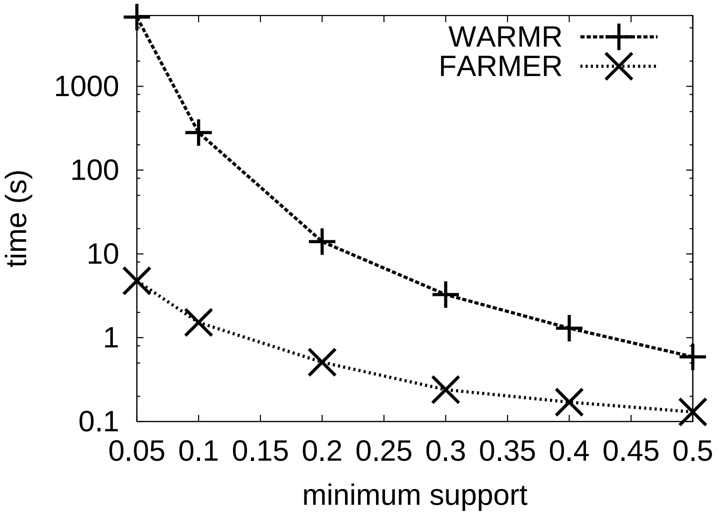
Itemsets Time



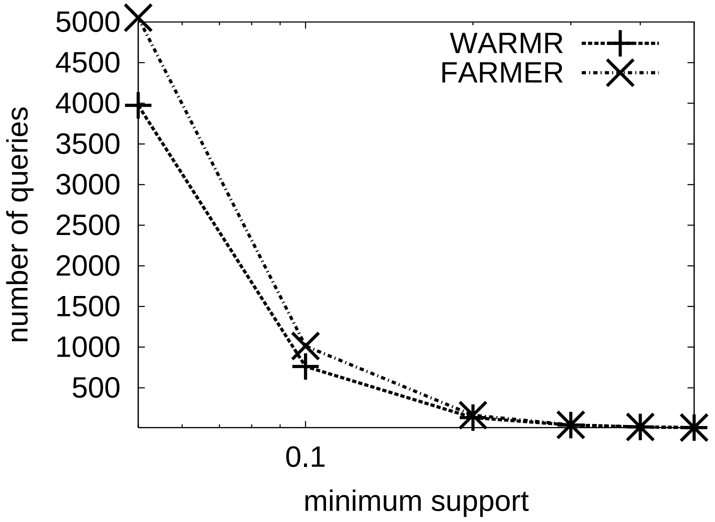
Time/Itemset

Frequent queries - experiments

- Farmer and Warmr on the Bongard dataset



Time



#Queries

Conclusions

- Queries can be found efficiently for usable simplified biases;

- the queries should be processed further to filter out uninteresting ones.

`http://www.liacs.nl/~snijssen/farmer`