Mining Frequent Itemsets in a Stream

Toon Calders, TU/e

(joint work with Bart Goethals and Nele Dexters, UAntwerpen)

Outline

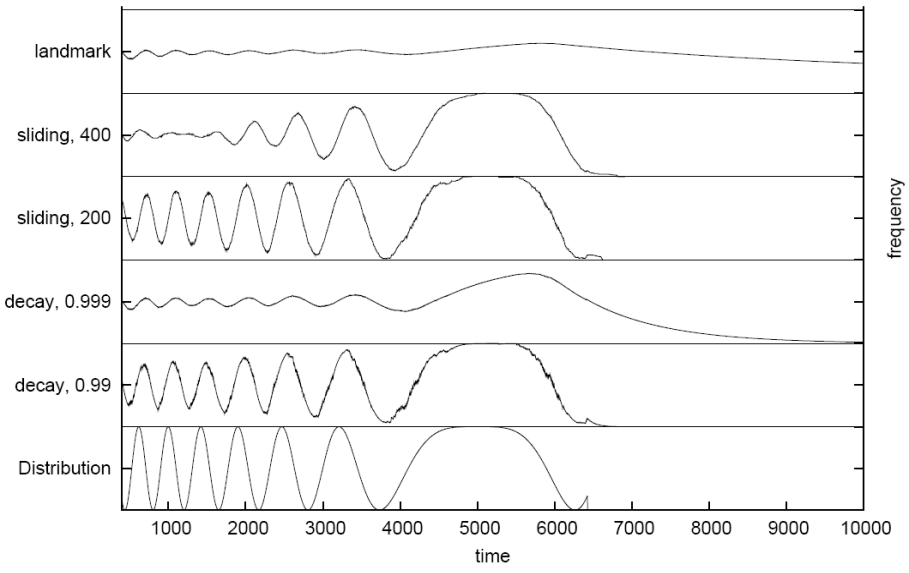
- Motivation
- Max-Frequency
- Algorithm
  - for one itemset
  - mining all Frequent Itemsets
- Experiments
- Conclusion

Motivation

- Model:
  - Every timestamp an itemset arrives
- Goal:
  - Find sets of items that frequently occur together
  - Take into account history,
  - Yet, recognize sudden bursts quickly

Motivation

- Most definitions of frequency rely heavily on the correct parameter settings
  - Sliding window length
  - Decay factor
  - …
- Correct parameter setting is hard
  - Can be different for different items (not to mention sets!)



Outline

- Motivation
- Max-Frequency
- Algorithm
  - for one itemset
  - mining all Frequent Itemsets
- Experiments
- Conclusion

Max-Frequency

Therefore, a new frequency measure:

$$mfreq(I, S) := \max_{k=1..|S|}(freq(I, last(k, S)))$$

Frequency is measured in the window where it is *maximal.*

Itemset gets the benefit of the doubt …

Example

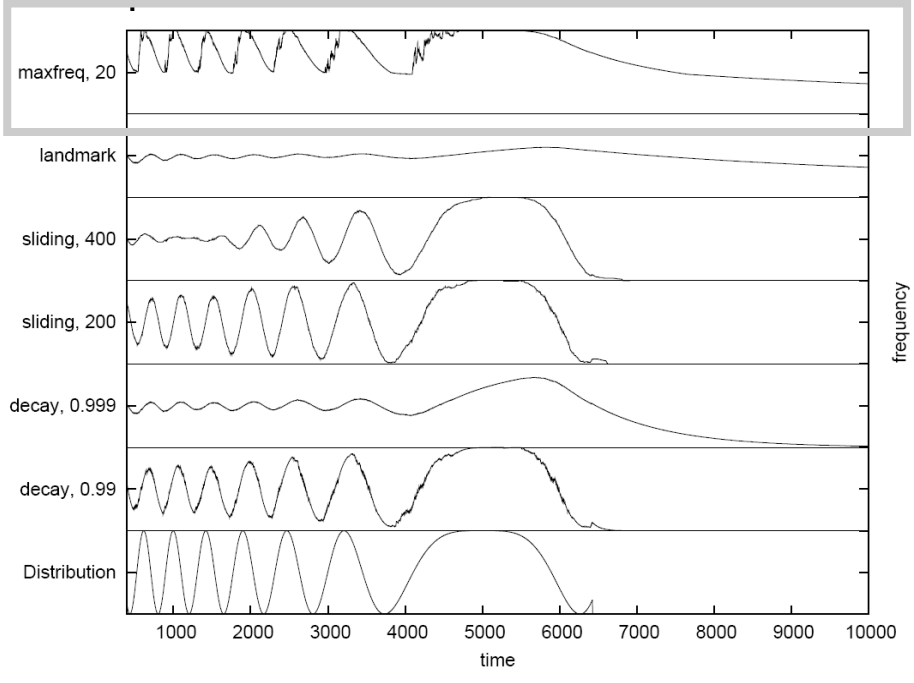
mfreq(a, ac abc ab ac ab bc)

ac bc ab ac ab **bc**	0
ac bc ab ac **ab bc**	1/2
ac bc ab **ac ab bc**	2/3
ac bc **ab ac ab bc**	3/4 ←
ac **bc ab ac ab bc**	3/5
ac bc ab ac ab bc	4/6

Properties of Max-Freq

+ Detects sudden bursts
+ Takes into account the past

- When target itemset arrives: sudden jump to a frequency of 1
+ Solution: minimal window length



Outline

- Motivation
- Max-Frequency
- Algorithm
  - for one itemset
  - mining all Frequent Itemsets
- Experiments
- Conclusion

Algorithm

1. How to do it for ***one*** itemset?
2. How to do it for a ***frequent*** itemset?
3. How to do it for ***all*** frequent itemsets?

Maintain a ***summary*** of the stream that allows to find the frequencies immediately.

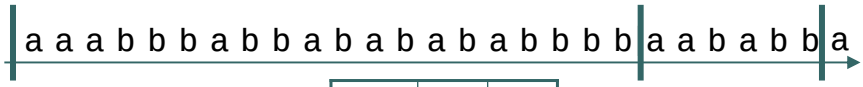
Properties (one itemset)

Checking all possible windows to find the maximal one: **infeasible**

BUT: not every point needs to be checked

↓

Only some special points = the borders



timestamp	1	21	27
# targets	8	3	1

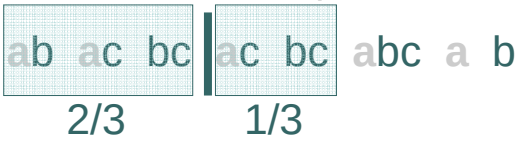
How to find a border?

- Target set a
- Is the marked position a border?

ab ac bc | ac bc abc a b

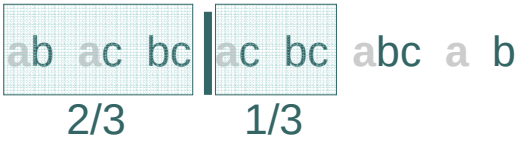
How to find a border?

- Target set a
- Is the marked position a border?



How to find a border?

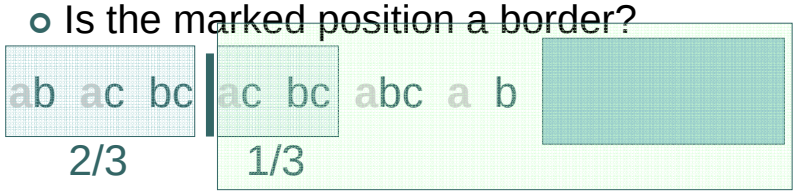
- Target set a
- Is the marked position a border?



NO

How to find a border?

- Target set a
- Is the marked position a border?

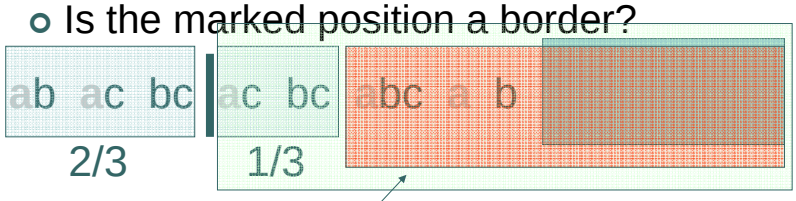


NO

> 2/3

How to find a border?

- Target set a
- Is the marked position a border?



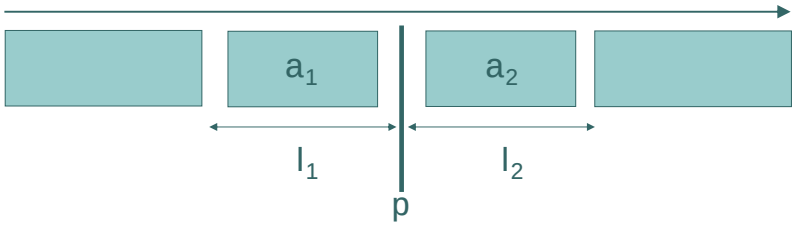
NO

> 2/3

even bigger

How to find the borders?

- This is true in general:



If $a_1/l_1 \geq a_2/l_2$, position p is never the border again!

Very powerful pruning criterion!

The summary

- Summary only keeps counts for the borders.

ab	ac	bc	ac	bc	abc	a	b

1	6
3	2

The summary

- Summary only keeps counts for the borders.

ab	ac	bc	ac	bc	abc	a	b

1	6
3	2

- Frequencies always increasing
  - Thus: max-frequency in last cell
- Block with largest frequency before border p_i = always block from p_{i-1}

Updating the Summary

- When a new itemset arrives, the summary is updated.
  - borders need to be checked again

ab	ac	bc	ac	bc	abc	a	b	T

Updating the Summary

- When a new itemset arrives, the summary is updated.
  - borders need to be checked again

  - no new « before » - blocks
  - only one new « after » - block
  - maximal block before: always previous border

Updating the Summary

- When a new itemset arrives, the summary is updated.
  - borders need to be checked again

  - no new « before » - blocks
  - only one new « after » - block
  - maximal block before: always previous border

Updating the Summary

- The new position is a border if and only if it contains the target itemset.

1	6	9
3	2	1

1
5

Summary: the Summary

- Only keep entries for borders
- Get Max-frequency = access last cell only
- Update summary:
  - if target: add new entry
  - if non-target: check borders
    - only one check required: still in ascending order?
    - most recent border always drops first
    - no need to check at every timestamp

Mining Frequent Itemsets

- Only interested in itemsets that are frequent.
- We can throw away any border with a frequency lower than the minimal frequency.

ab ac bc ac bc | abc a b | ab

6	9
2	1

minfeq = 2/3

Mining All Frequent Itemsets

- We only need to maintain the summaries for the frequent itemsets
- Can still be a lot, though …
  - every subset of the most recent transaction …
  - minimal window length reduces this problem
- FUTURE WORK: reduce this number; rely, e.g., on approximate counts

Outline

- Motivation
- Max-Frequency
- Algorithm
  - for one itemset
  - mining all Frequent Itemsets
- Experiments
- Conclusion

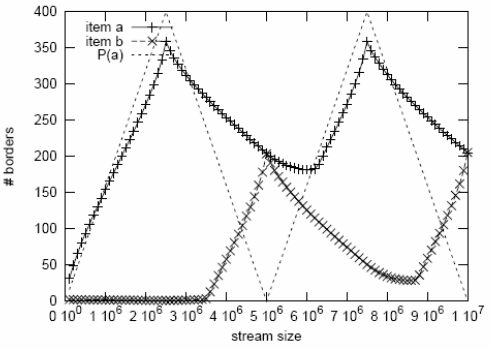
Experiments

- Size of the summaries
  - number of borders for random data
  - average, maximal number of borders in real-life data
- Theoretical worst case

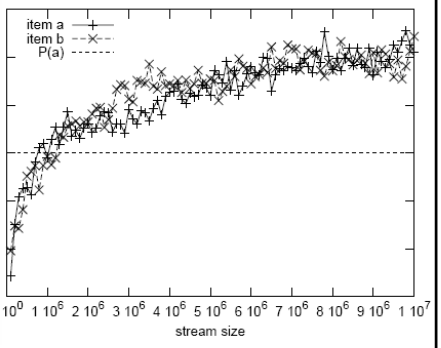
$$N = \left(\frac{\pi^2 L}{2}\right)^{2/3} \frac{3}{\pi^2}$$

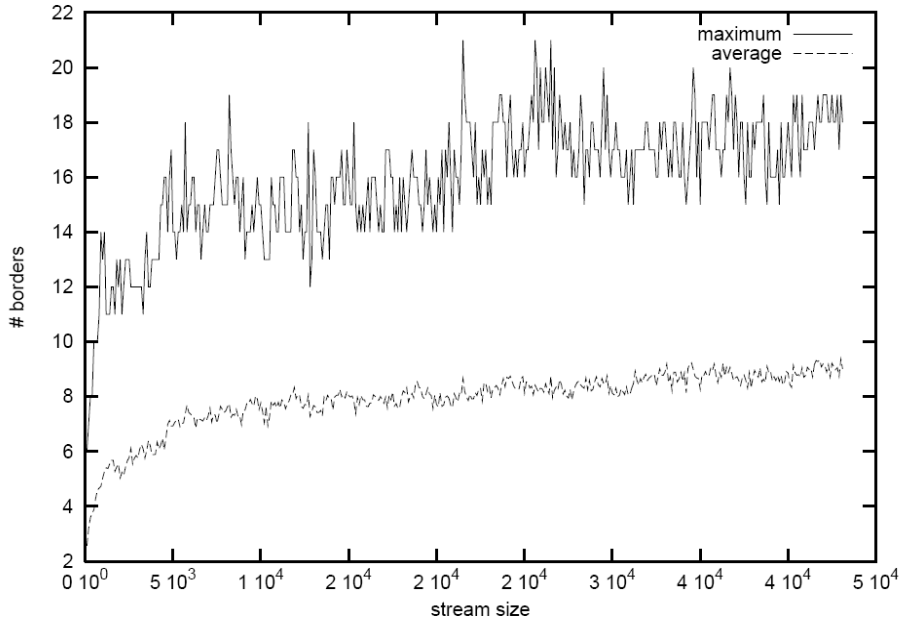
Experiments

Twin Peaks distribution



Uniform Distribution





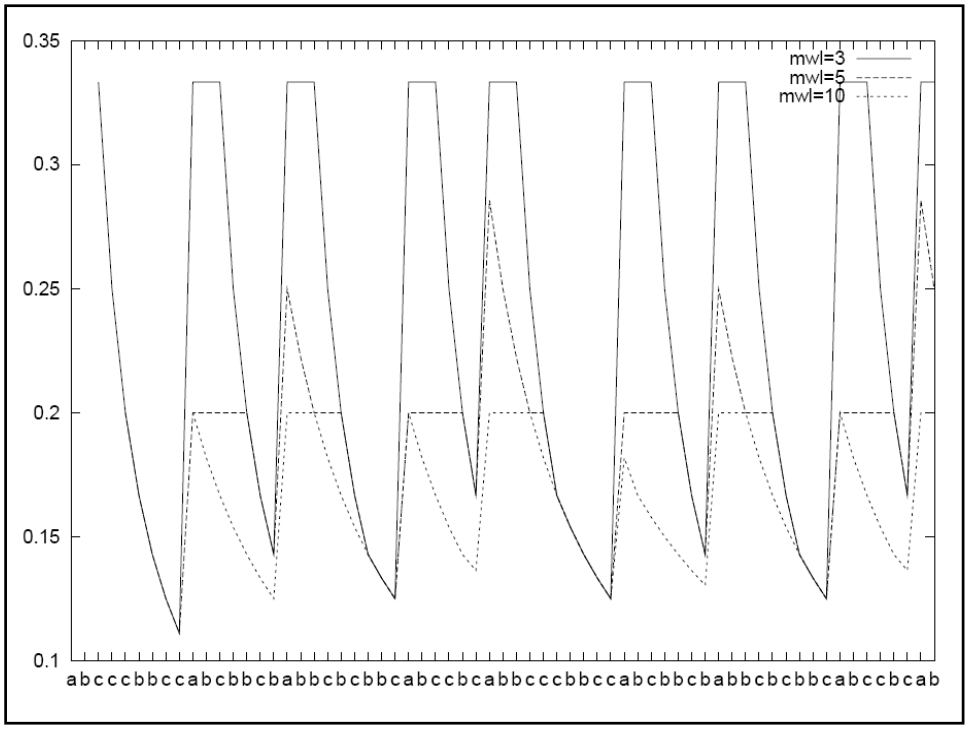
Outline

- Motivation
- Max-Frequency
- Algorithm
  - for one itemset
  - mining all Frequent Itemsets
- Experiments
- Conclusion

Conclusions

- New frequency measure
- Summary for one itemset
  - small
  - easy to maintain
  - only few updates
- Mining all frequent itemsets
  - only need summary for frequent itemsets

Thank you !


0.35
0.3
0.25
0.2
0.15
0.1
mwl=3
mwl=5
mwl=10
abcccbbccabcbbcbabbcbcbbcabccbcabbcccbbccabcbbcbabbcbcbbcabccbcab