Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Algorithms for Pattern Mining

# Relim & Relix

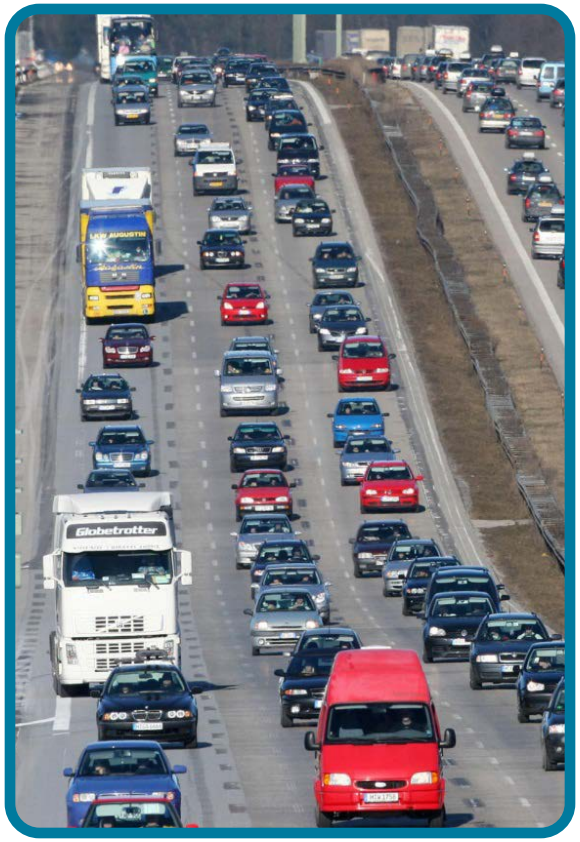
*- Final Presentation -*

Thomas Stening
Thorsten Papenbrock

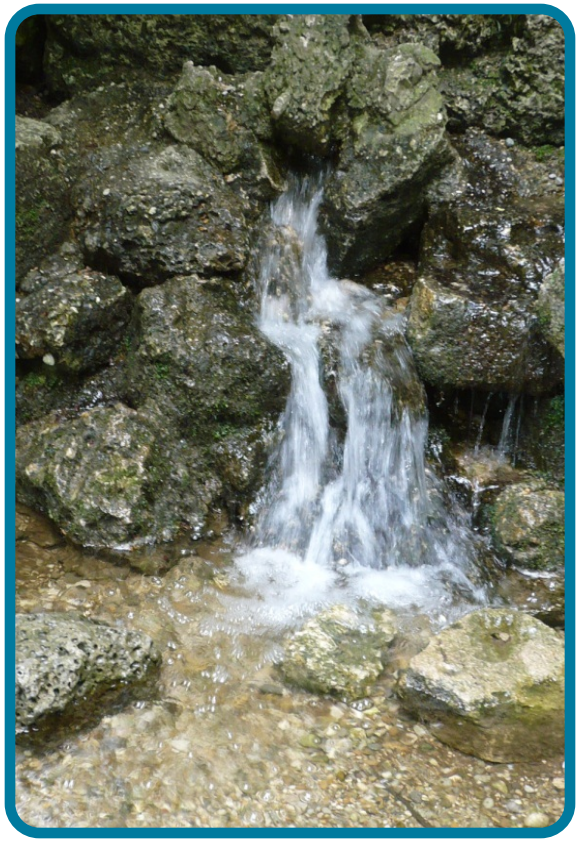
# Agenda



Fuzzy Pattern Mining



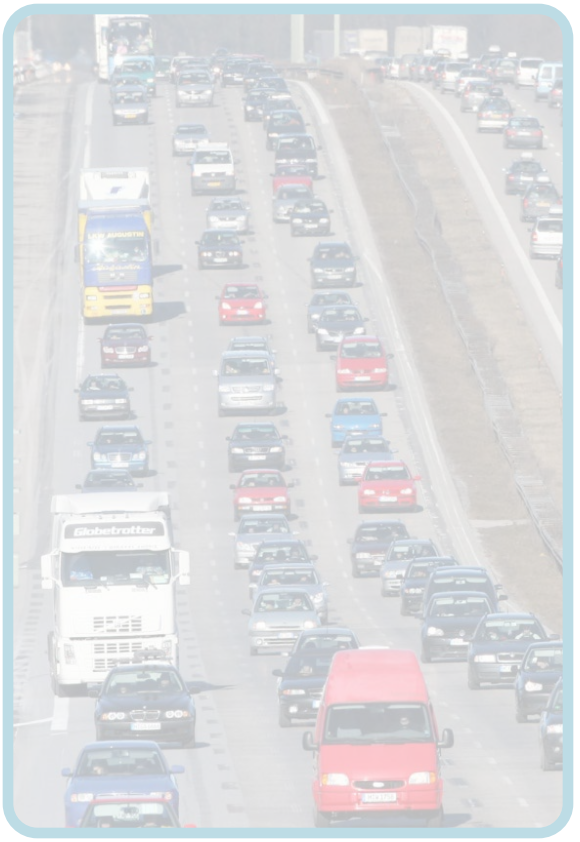
Parallelisation



Different Datasources

# Agenda





Fuzzy Pattern Mining

Parallelisation

Different Datasources

# Fuzzy Datamining – Use Case



**Medical Claim Analysis:**

- 113000 transactions (= number of patients)
- 46 different item values (= different claims)
- 610934 items (= number of claims)
  → ~5.4 items / transaction

**Questions:**

- Which claims occure often together?
- Are there any claims that correlate each other?

# Fuzzy Datamining – Motivation



~ 20% of all heard attacks go undiscovered (2003)

http://articles.mercola.com/sites/articles/archive/2003/01/01/heart-attacks-part-two.aspx

~ 39% of all HIV and Aids infections are not detected (2004)

www.ncbi.nlm.nih.gov/pubmed/15127908

~ 21% of different plasmodium infections are not found (2011)

http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0019010

Fuzzy Pattern Mining can be used to take these probabilities into account!

# Fuzzy Datamining – Motivation



- Items in transactions …
  - can get lost (claim was not detected or reported)
    - $LostProbability_i := \frac{\#not\ reported\ items\ i}{\#world\ population - \#reported\ items\ i}$
  - can be false (claim was wrongly diagnosed or pretended)
    - $AccuracyProbability_i := \frac{\#correct\ reported\ items\ i}{\#all\ reported\ items\ i}$
- Items have different probabilities for errors
- Find Frequent Item Sets although the data is incomplete

# Fuzzy Datamining – Example: HIV



- Statistics:
  - 7025 million people is the current world population
  - 22 million reported HIV infections worldwide
  - 36 million HIV infections are estimated by specialists
  - 0.2 million people positive tested on HIV in fact had no HIV

- Probabilities:

$$LostProbability_{HIV} = \frac{\#not\ reported\ HIV\ claims}{\#world\ population - \#reported\ items\ i} = \frac{36 - 22\ million}{7025 - 22\ million} = 0.2\%$$

$$AccuracyProbability_{HIV} = \frac{\#correct\ reported\ HIV\ claims}{\#all\ reported\ HIV\ claims} = \frac{22 - 0.2\ million}{22\ million} = 99\%$$

# Fuzzy Datamining – Relix



- Very similar algorithm to Relim
- Automatically …
  - inserts items to all transactions with respect to their *LostProbability*
  - reduces items occurrence probability with respect to their *AccuracyProbability*

# Relix – Datastructure Generation



1. Load transactions (in memory)

| |
|---|
| a d f |
| c d e |
| b d |
| a b c d |
| b c |
| a b d |
| b d e |
| b c e g |
| c d f |
| a b d |

# Relix – Datastructure Generation



1. Load transactions (in memory)
2. Count item frequencies

1. Iteration

| a d f |
|---|
| c d e |
| b d |
| a b c d |
| b c |
| a b d |
| b d e |
| b c e g |
| c d f |
| a b d |

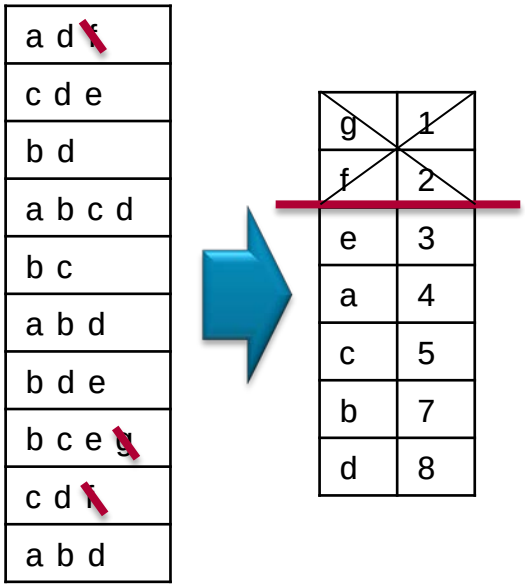
| g | 1 |
|---|---|
| f | 2 |
| e | 3 |
| a | 4 |
| c | 5 |
| b | 7 |
| d | 8 |

# Relix – Datastructure Generation



1. Load transactions (in memory)
2. Count item frequencies
3. Delete all rare items from the transactions

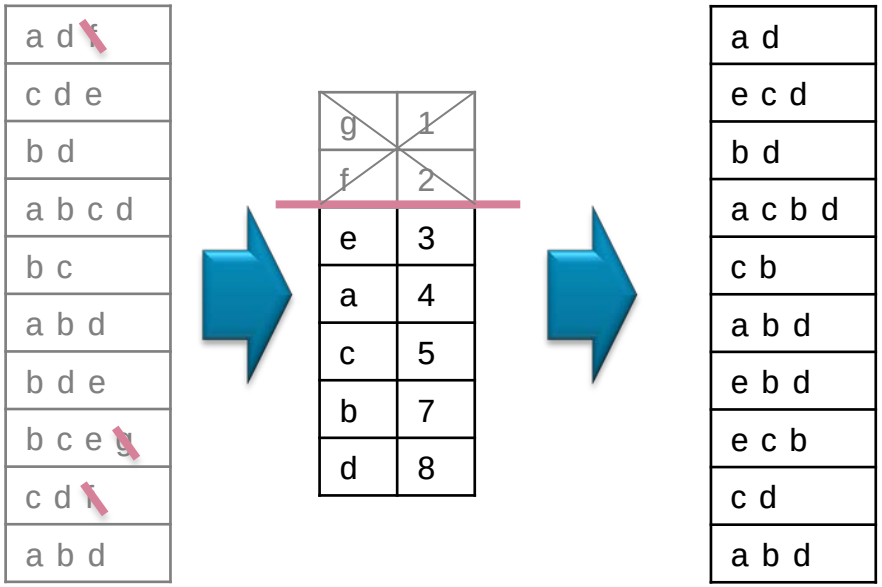
1. Iteration



# Relix – Datastructure Generation

1. Load transactions (in memory)
2. Count item frequencies
3. Delete all rare items from the transactions
4. Sort each transaction according the items frequency

1. Iteration

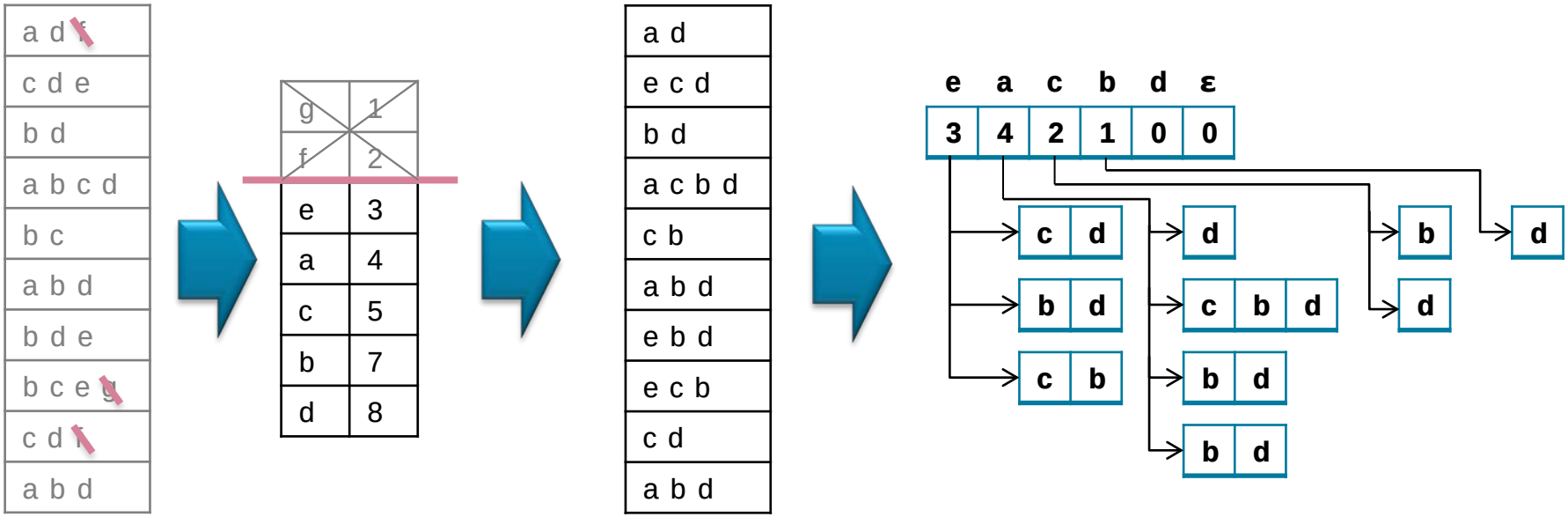


# Relix – Datastructure Generation

1. Load transactions (in memory)
2. Count item frequencies
3. Delete all rare items from the transactions
4. Sort each transaction according the items frequency
5. Create Relim datastructure

1. Iteration

2. Iteration

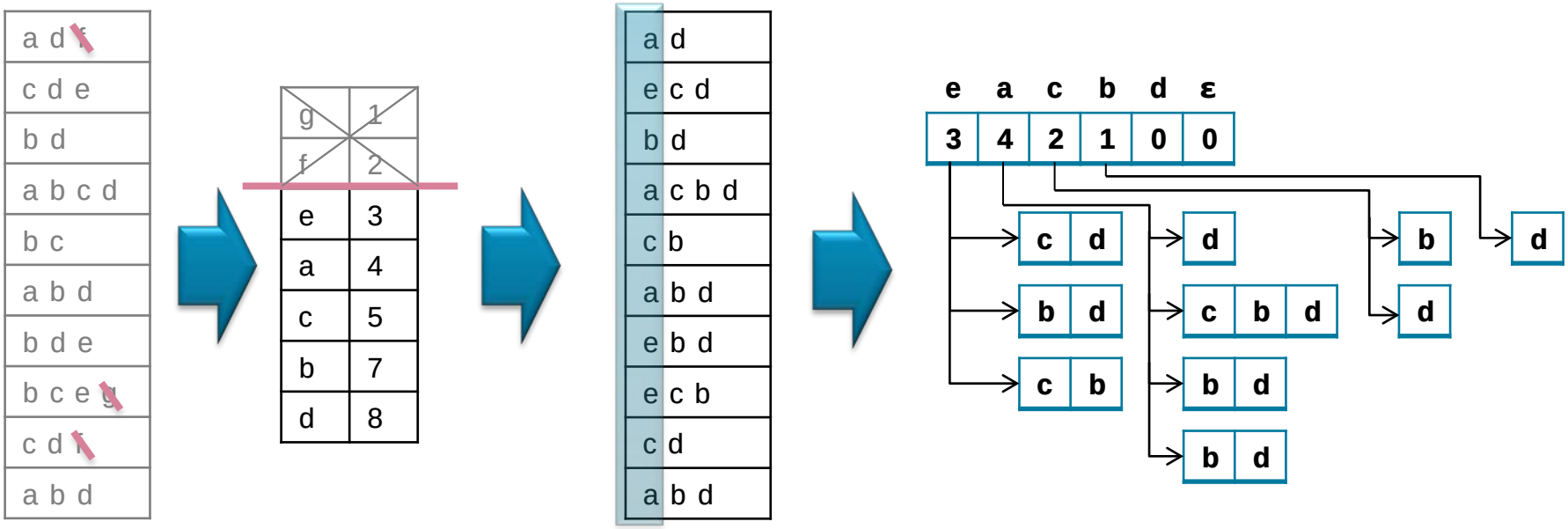


# Relix – Recursive Tree Processing

1. Load transactions (in memory)
2. Count item frequencies
3. Delete all rare items from the transactions
4. Sort each transaction according the items frequency
5. Create Relim datastructure

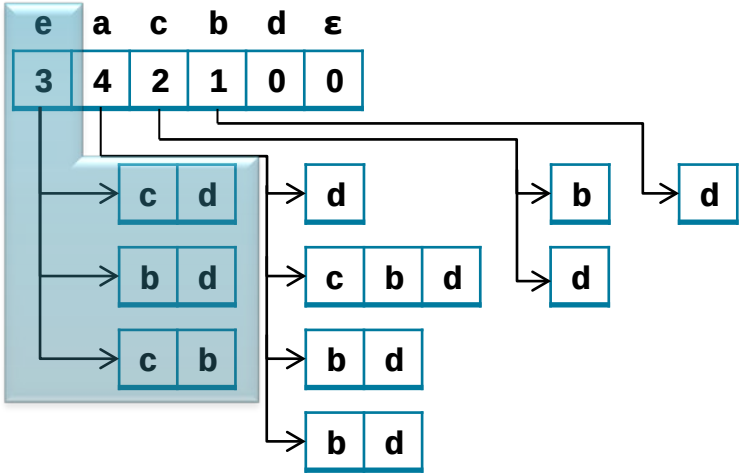
1. Iteration

2. Iteration

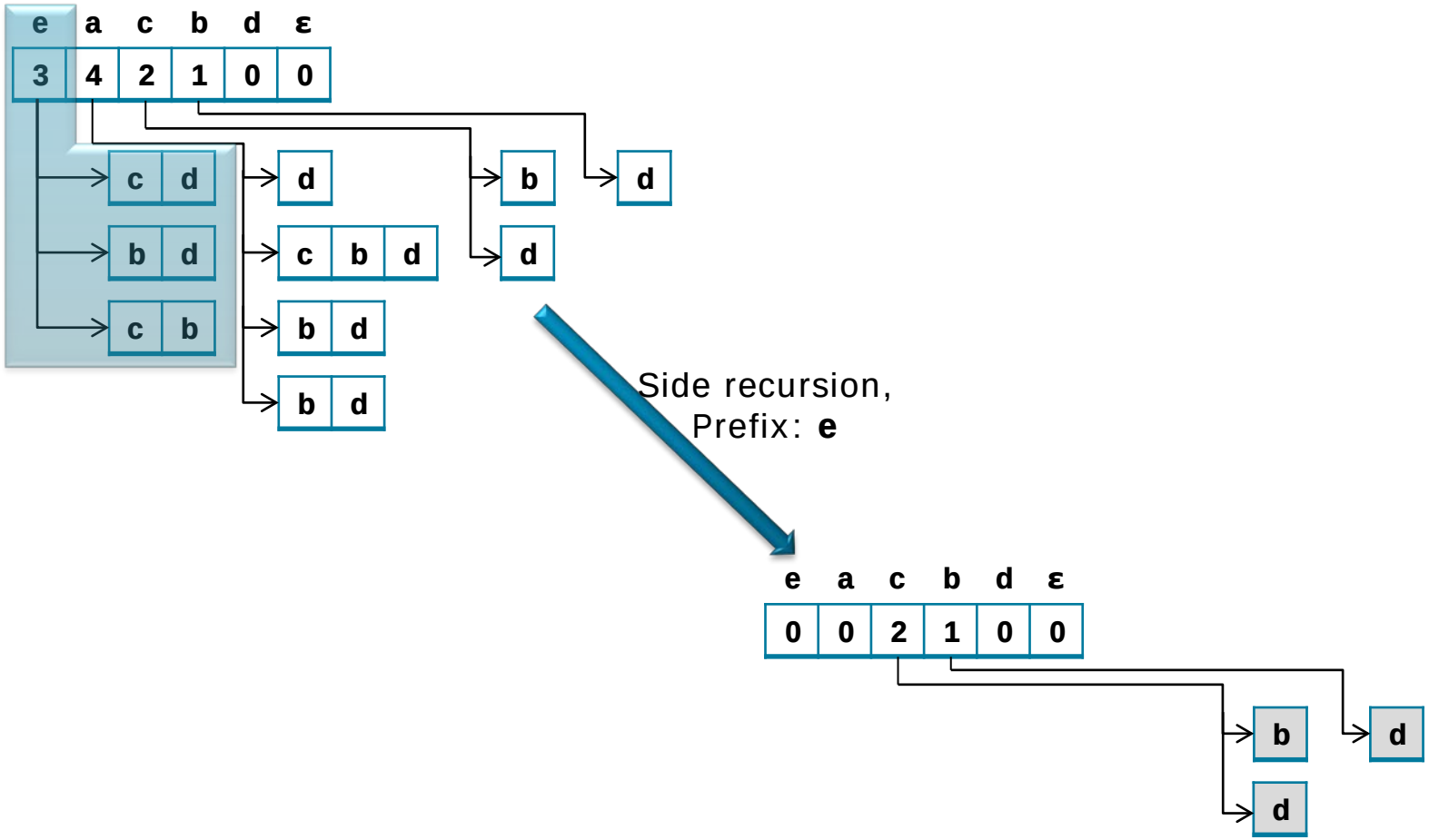


# Relix – Recursive Tree Processing

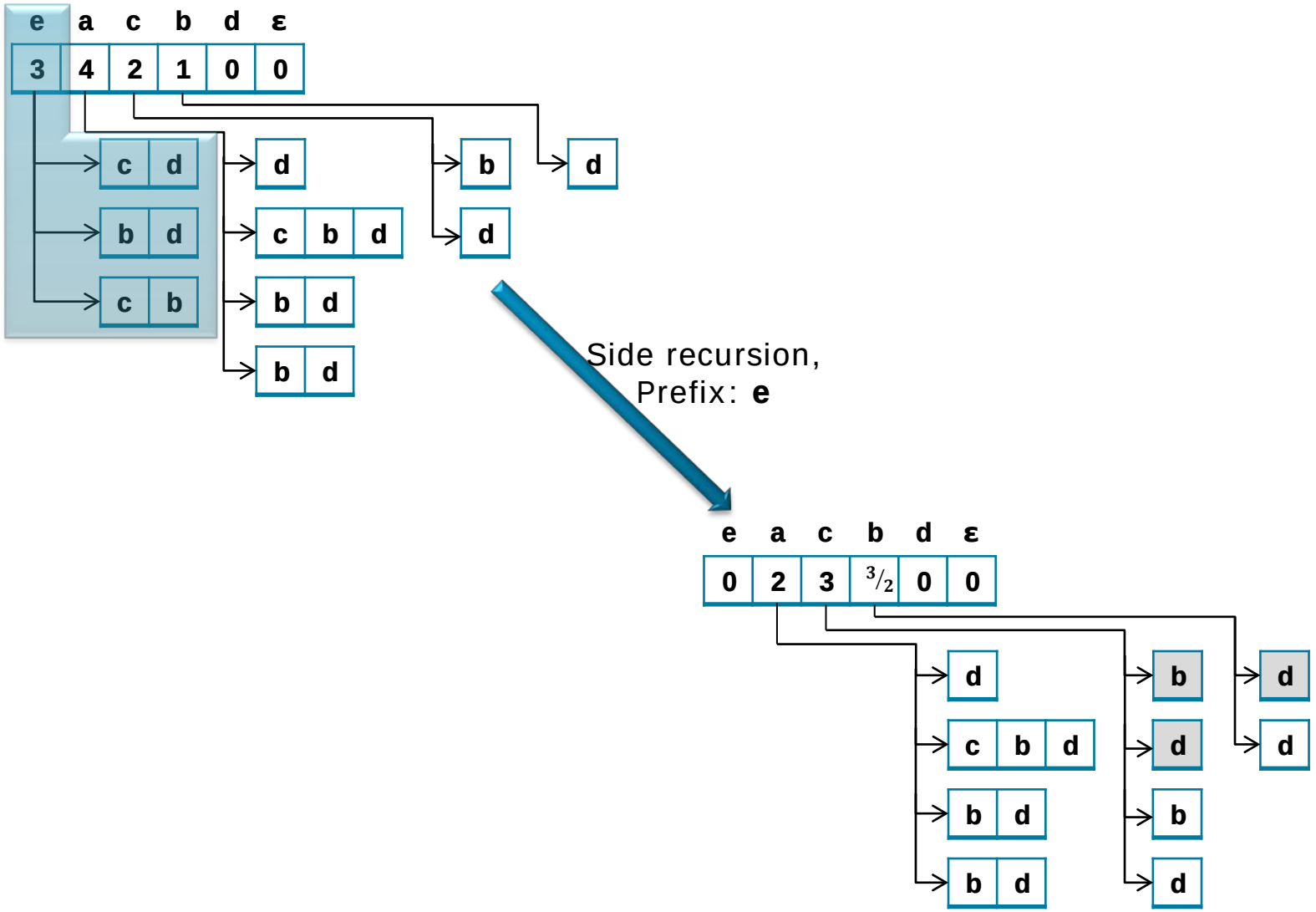




# Relix – Recursive Tree Processing

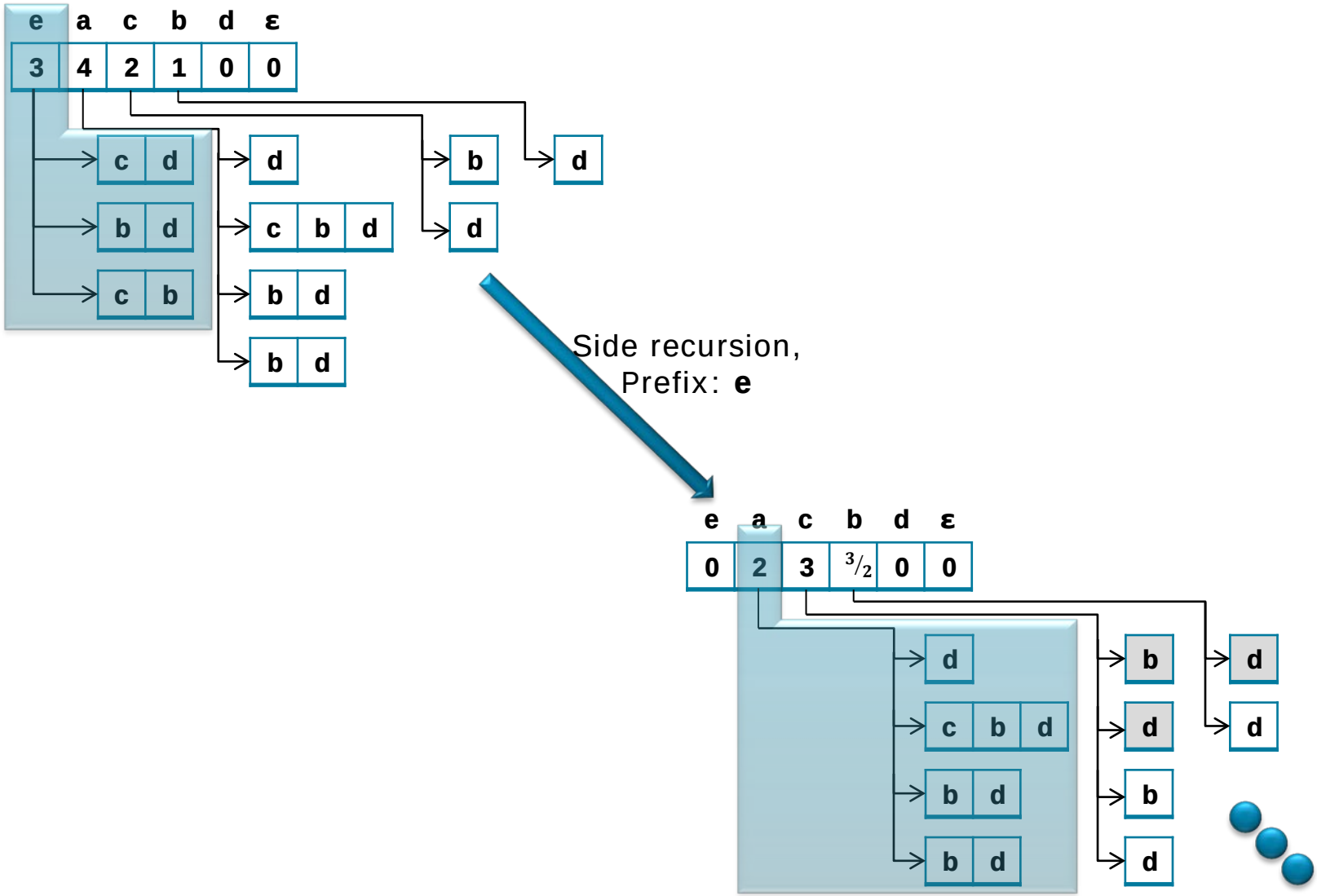


# Relix – Recursive Tree Processing

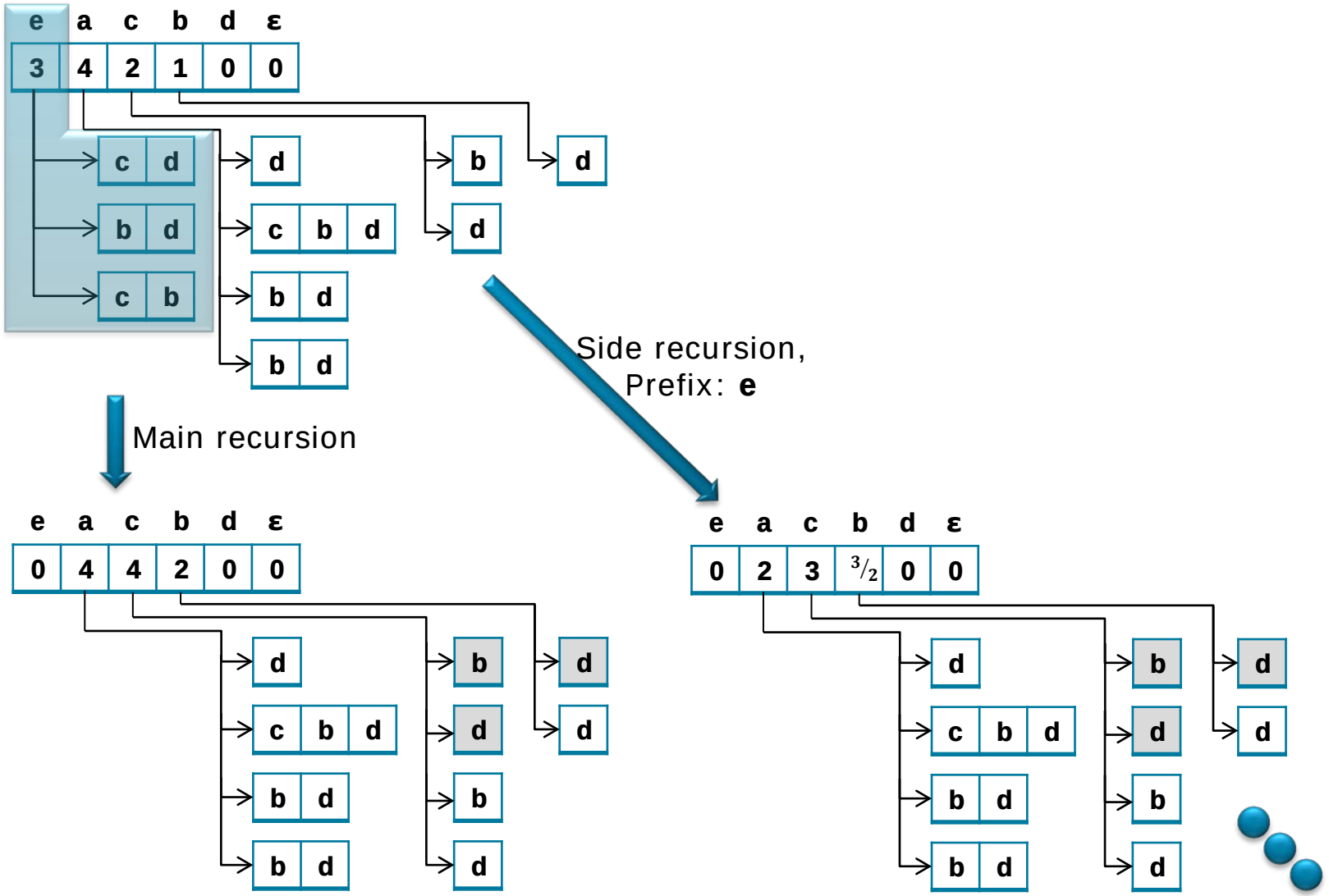


# Relix – Recursive Tree Processing



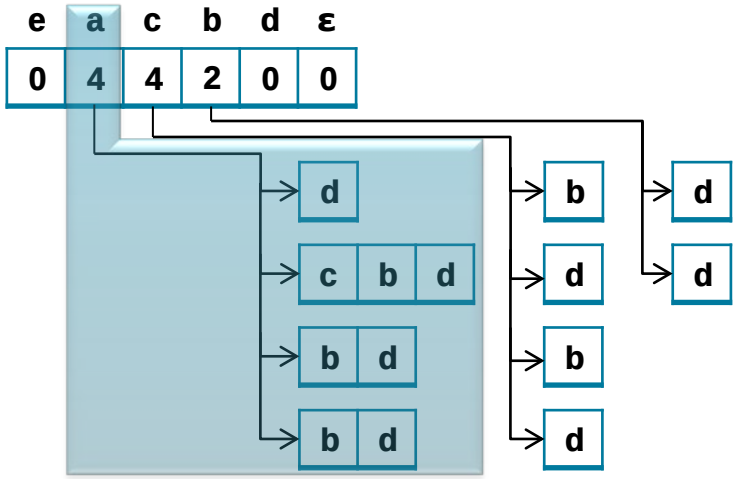


# Relix – Recursive Tree Processing



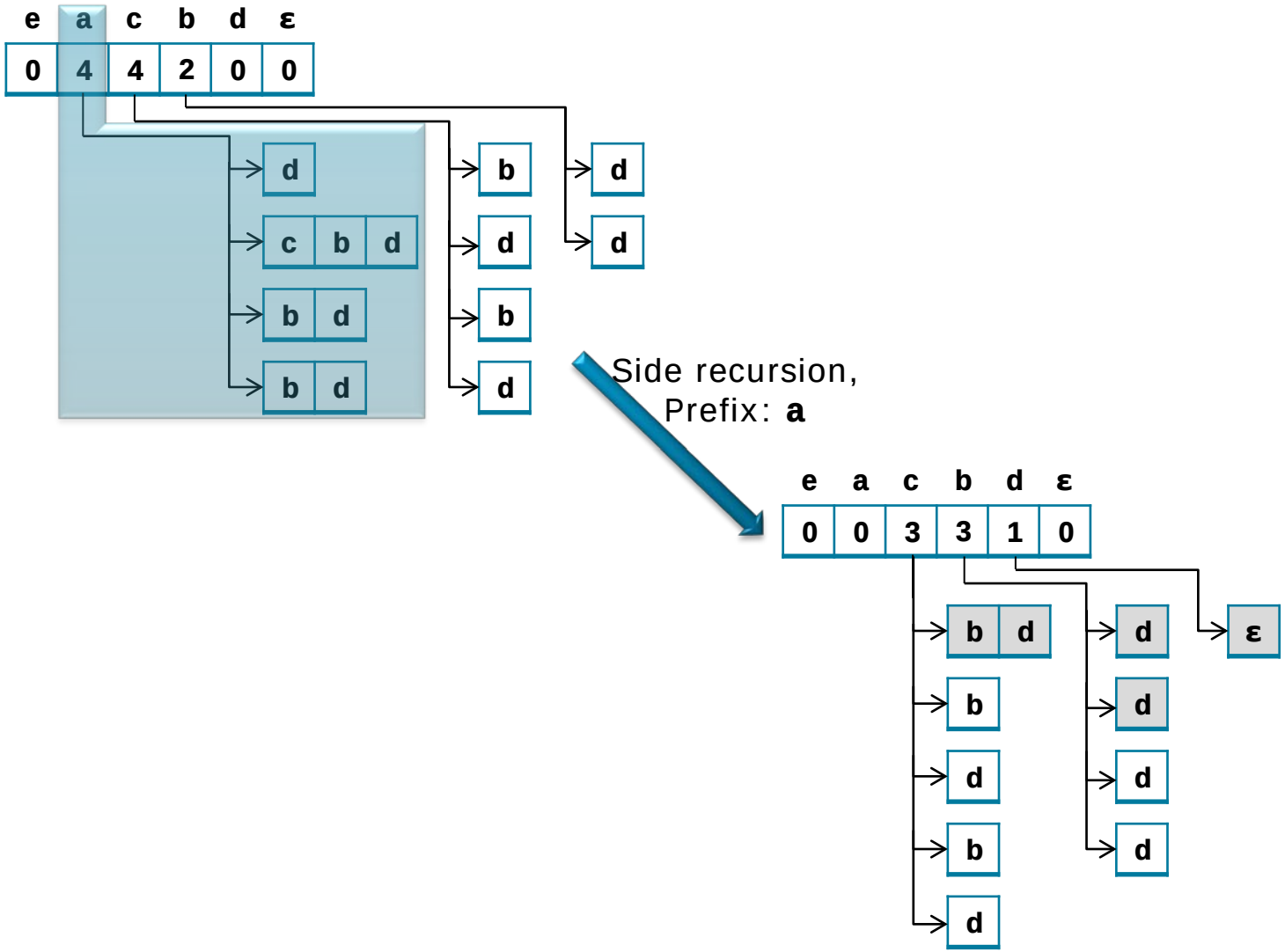
# Relix – Recursive Tree Processing





# Relix – Recursive Tree Processing





# Fuzzy Datamining – Performance

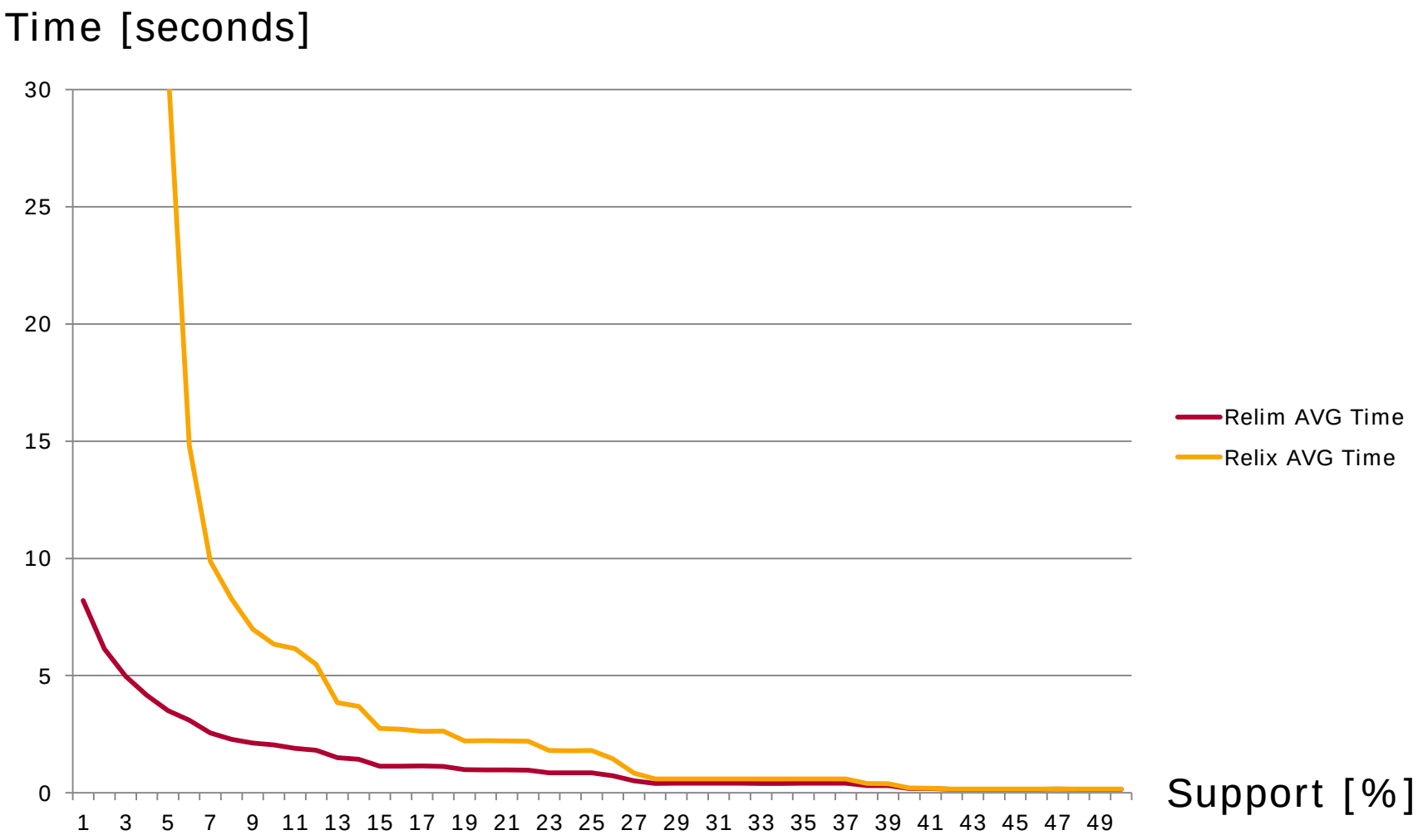


**System:**

- DELL StudioXPS
- 64 bit Windows 7 Enterprise
- Intel Core i5 M520 2,40 GHz
- 4 GB RAM

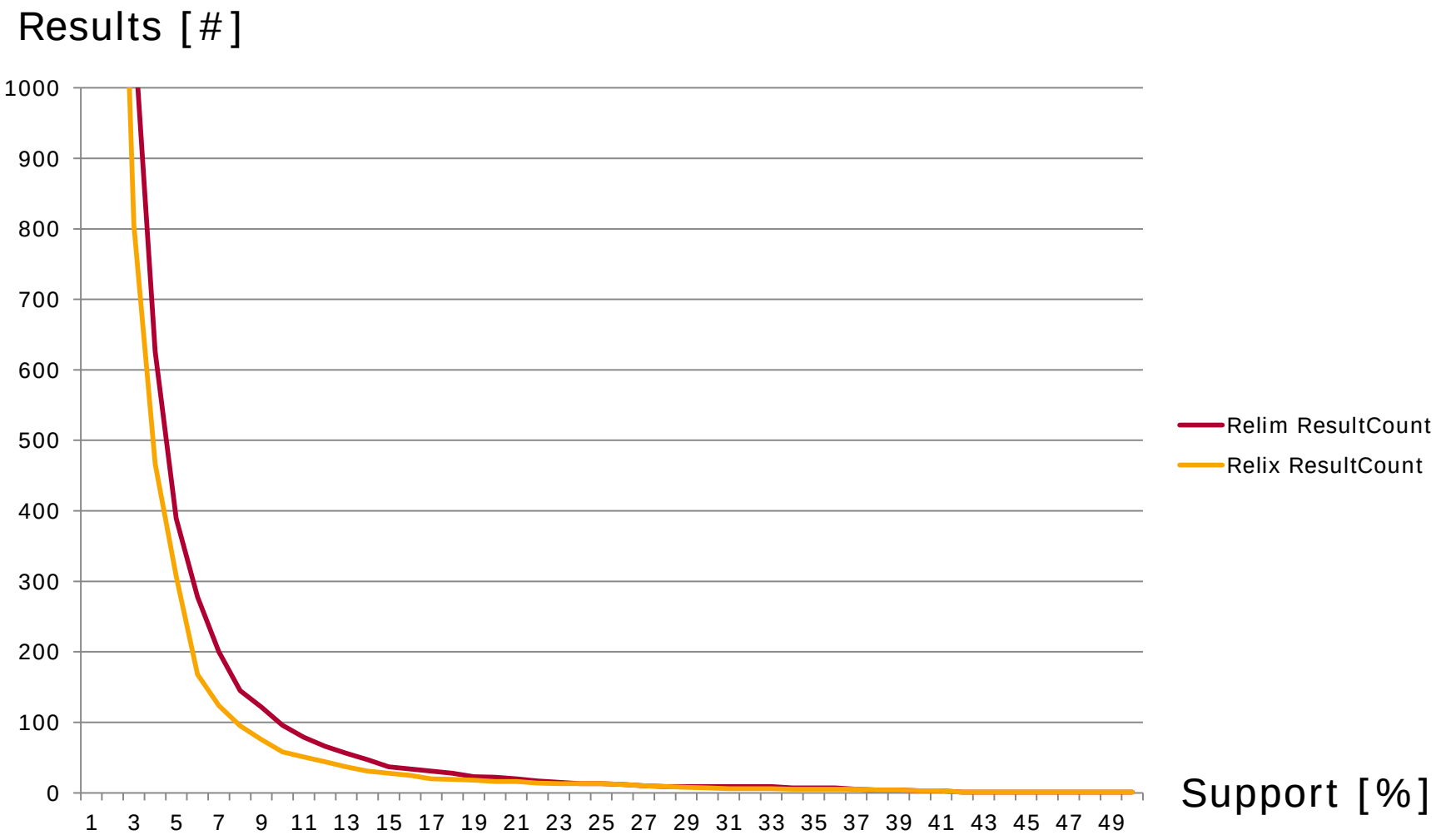
# Fuzzy Datamining – Performance





# Fuzzy Datamining – Results





# Fuzzy Datamining – Frequent Sets



$$\text{GoogleConfidence(F)} = \frac{results(F)}{\frac{\sum_{i=0}^{\#F} results(Fi)}{\#F}}$$

*F*: Freqent Set

$F_i$: i-tes item in F

*results(F)*: Google query count for F

*results($F_i$)*: Google query count for $F_i$

GoogleConfidence(”Apfel”,”Birne”)

$$= \frac{results(\text{"}Apfel\ Birne\text{"})}{\frac{results(\text{"}Apfel\text{"}) + results(\text{"}Birne\text{"})}{2}}$$

$$= \frac{2.970.000}{\frac{6.210.000 + 8.170.000}{2}}$$

$$= 0{,}0413$$

GoogleConfidence(”Apfel”,”Auto”)

$$= \frac{results(\text{"}Apfel\ Auto\text{"})}{\frac{results(\text{"}Apfel\text{"}) + results(\text{"}Auto\text{"})}{2}}$$

$$= \frac{2.720.000}{\frac{6.210.000 + 908.000.000}{2}}$$

$$= 0{,}0059$$

# Fuzzy Datamining – Frequent Sets



$$\text{GoogleConfidence(F)} = \frac{results(F)}{\frac{\sum_{i=0}^{\#F} results(Fi)}{\#F}}$$

*F*: Freqent Set

$F_i$: i-tes item in F

*results(F)*: Google query count for F

*results($F_i$)*: Google query count for $F_i$

| Relim Results |
|---|
| [All other trauma,Arthropathies] |
| [Skin and autoimmune disorders,Other metabolic] |
| [Gastrointestinal bleeding,Arthropathies] |
| [Skin and autoimmune disorders,Arthropathies] |
| [Acute respiratory,Other metabolic] |
| [Other neurological,Acute respiratory] |
| [Acute respiratory,Arthropathies] |
| [Other neurological,Other metabolic] |
| [Other neurological,Arthropathies] |
| [Other metabolic,Arthropathies] |

| Relix Results |
|---|
| [Skin and autoimmune disorders,Acute respiratory] |
| [Acute respiratory,Other metabolic] |
| [Skin and autoimmune disorders,Other neurological] |
| [All other infections,Acute respiratory] |
| [Skin and autoimmune disorders,Arthropathies] |
| [Other neurological,Acute respiratory] |
| [Acute respiratory,Arthropathies] |
| [Other neurological,Other metabolic] |
| [Other neurological,Arthropathies] |
| [Other metabolic,Arthropathies] |

$$\sum GoogleConfidence = 0{,}3592$$

$$\sum GoogleConfidence = 0{,}6433$$

# Fuzzy Datamining – Association Rules



Relim:

| Association rule | Conf. | Lift |
|---|---|---|
| [Miscellaneous cardiac,Skin and autoimmune disorders] --> [Other neurological,Other metabolic,Miscellaneous 2] | 0.50 | 2.76 |
| [Other neurological,Other metabolic,Miscellaneous 2] --> [Miscellaneous cardiac,Skin and autoimmune disorders] | 0.28 | 2.76 |
| [Miscellaneous cardiac,Other neurological,Miscellaneous 2] --> [Skin and autoimmune disorders,Other metabolic] | 0.40 | 2.73 |
| [Skin and autoimmune disorders,Other metabolic] --> [Miscellaneous cardiac,Other neurological,Miscellaneous 2] | 0.34 | 2.73 |
| [Miscellaneous cardiac,Arthropathies,Miscellaneous 2] --> [Skin and autoimmune disorders,Other metabolic] | 0.40 | 2.72 |
| [Skin and autoimmune disorders,Other metabolic] --> [Miscellaneous cardiac,Arthropathies,Miscellaneous 2] | 0.38 | 2.72 |
| [Miscellaneous cardiac,Skin and autoimmune disorders,Miscellaneous 2] --> [Other neurological,Other metabolic] | 0.55 | 2.71 |

Relix:

| Association rule | Conf. | Lift |
|---|---|---|
| [Other neurological,Other metabolic] --> [Miscellaneous cardiac,Miscellaneous 2] | 0.42 | 2.65 |
| [Miscellaneous cardiac,Miscellaneous 2] --> [Other neurological,Other metabolic] | 0.37 | 2.65 |
| [Miscellaneous cardiac,Miscellaneous 2] --> [Other metabolic,Arthropathies] | 0.41 | 2.64 |
| [Other metabolic,Arthropathies] --> [Miscellaneous cardiac,Miscellaneous 2] | 0.42 | 2.64 |
| [Other metabolic,Miscellaneous 2] --> [Miscellaneous cardiac,Arthropathies] | 0.30 | 2.62 |
| [Miscellaneous cardiac,Arthropathies] --> [Other metabolic,Miscellaneous 2] | 0.57 | 2.62 |
| [Miscellaneous cardiac,Other neurological,Miscellaneous 2] --> [Other metabolic] | 0.72 | 2.44 |

# Fuzzy Datamining – Probabilities



**Problem:**

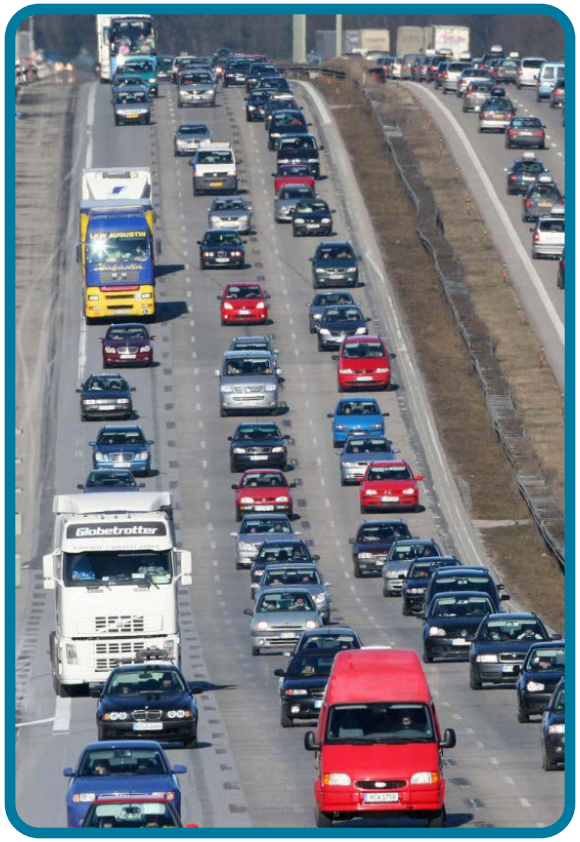
- LostProbabilities and AccuracyProbabilies do not exist in the claims dataset
- Probabilities are based on our estimations
  - Comparison does only compare the quality of our estimations

**Argument:**

- Results show that Fuzzy Datamining produces different results
- Missing probabilities can be measured

# Agenda





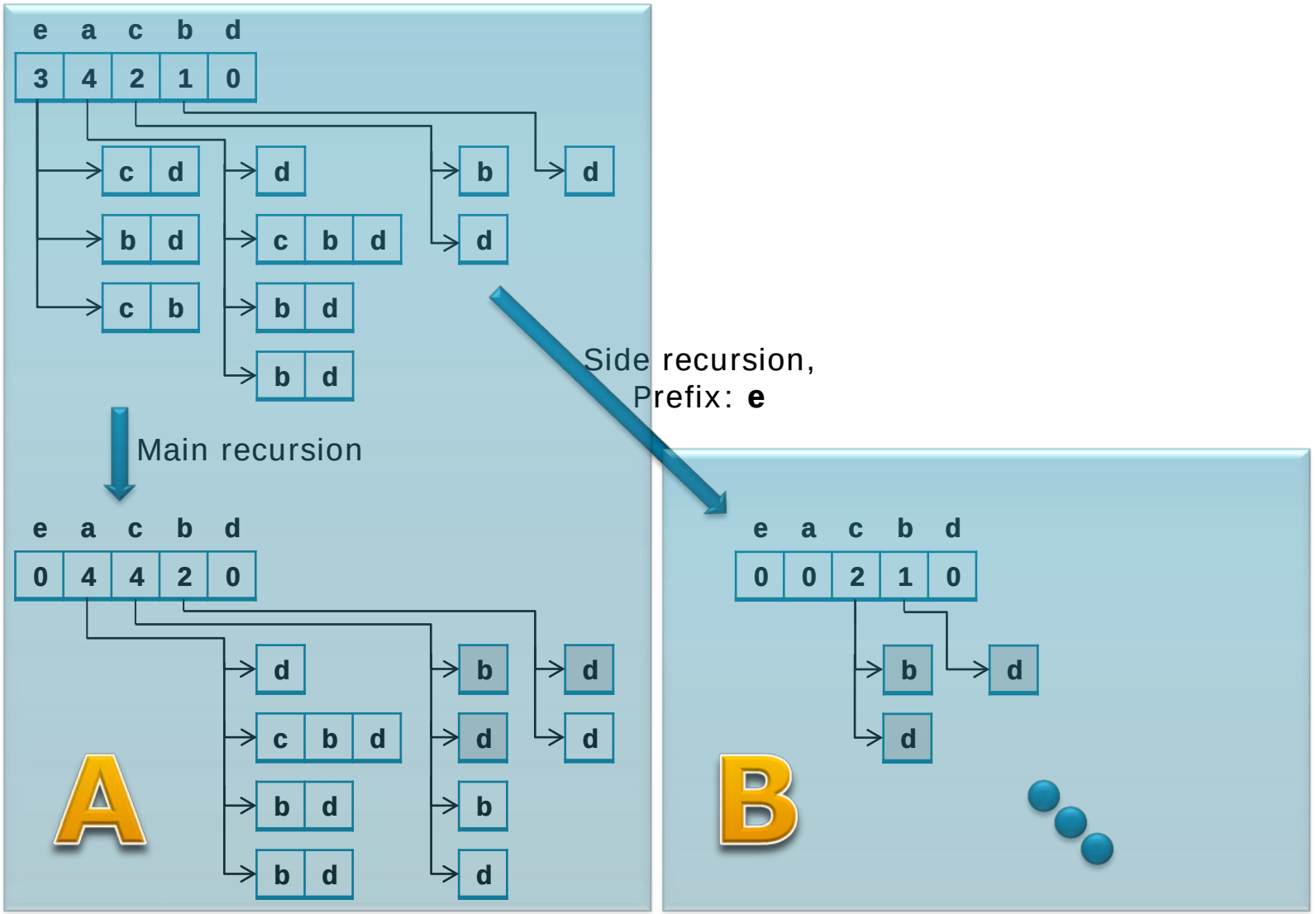

Fuzzy Pattern Mining

Parallelisation

Different Datasources

# Parallelisation – Algorithm





# Parallelisation – Task Brevity



**Problem:**

- Runtime for many tasks is low
- Thread scheduling and context switches dominate execution time

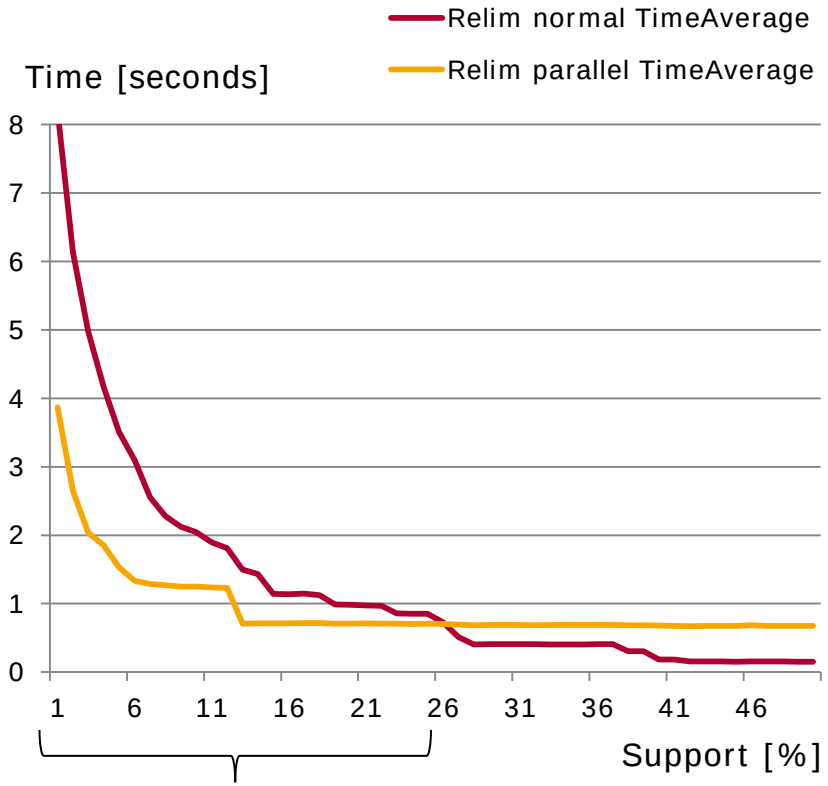
**Solution:**

- Maximum amount of threads is limited to 16
- New branches are created only if free threads are available; otherwise the current thread does the recursive step itself
  - Average runtime increases
  - Scheduling and context switch overhead decrease

# Parallelisation – Evaluation

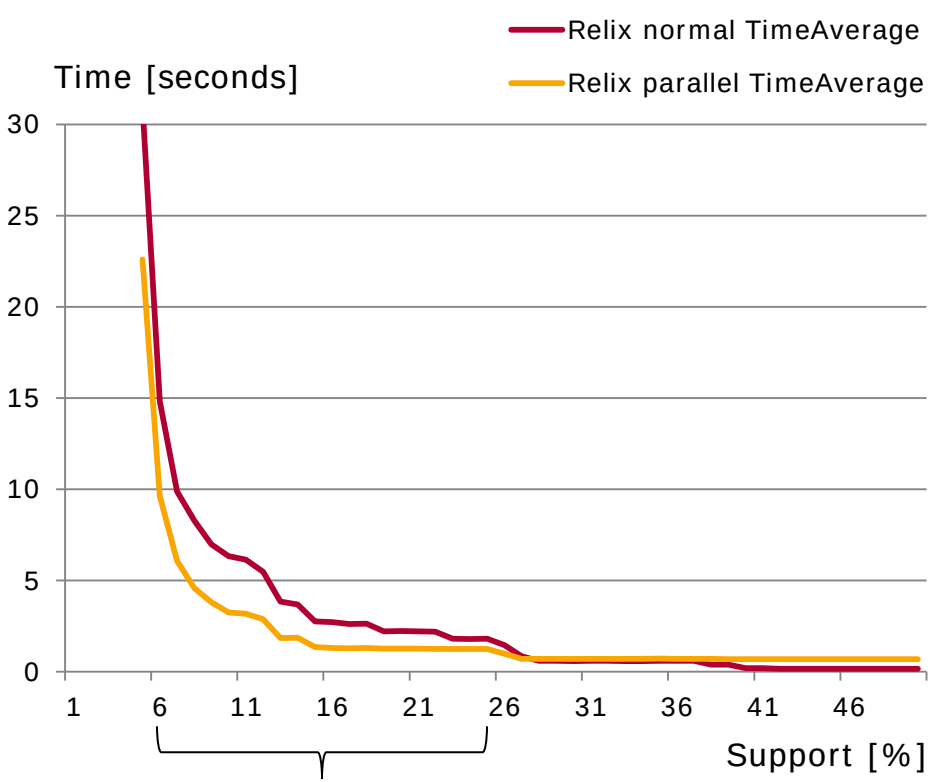


## Relim




Runtime reduction ~40%

## Relix

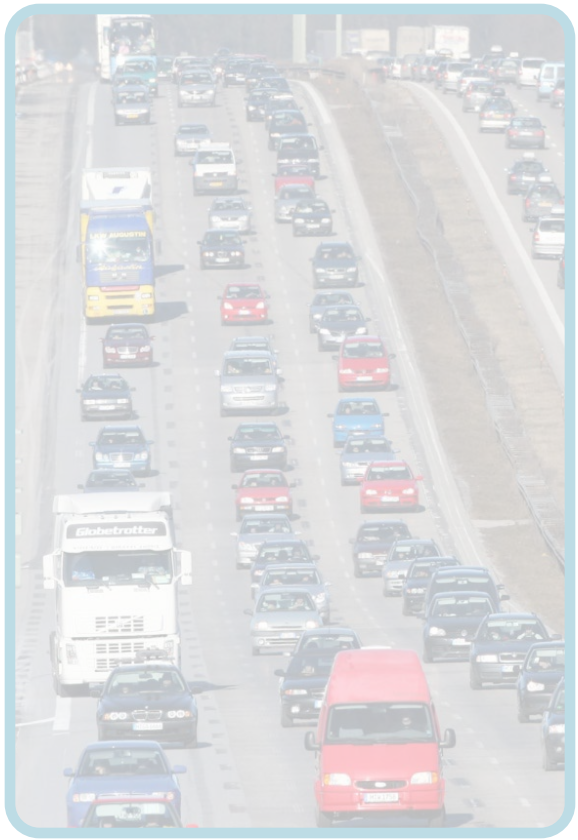


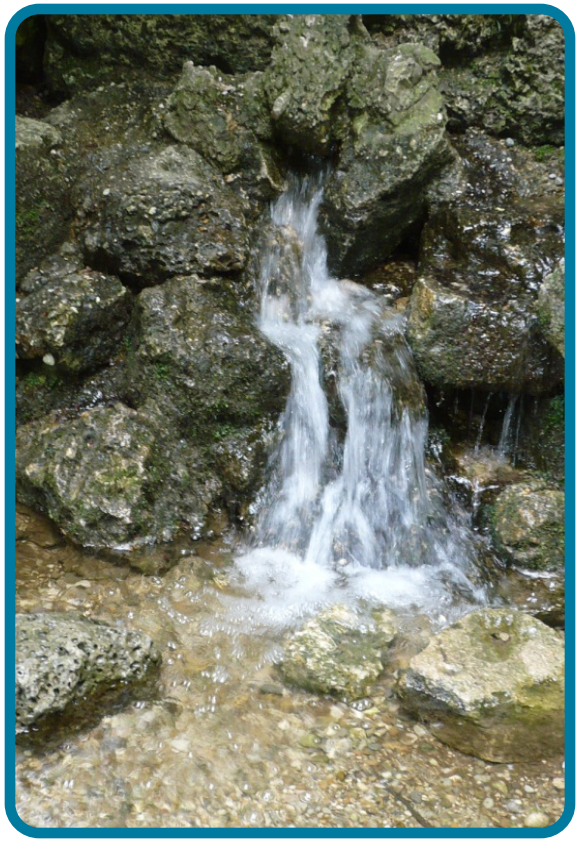

Runtime reduction ~43%

# Agenda









Fuzzy Pattern Mining

Parallelisation

Different Datasources

# Different Datasources – Statistics



| Claims | |
|---|---|
| #Transactions | 113000 |
| #Items | 610934 |
| #Item values | 46 |
| Ø Items / Transaction | 5.41 |
| [MB] Memory | 3.64 |

| FB-Likes | |
|---|---|
| #Transactions | - |
| #Items | - |
| #Item values | - |
| Ø Items / Transaction | - |
| [MB] Memory | - |

| Netflix | |
|---|---|
| #Transactions | 463615 |
| #Items | 23168214 |
| #Item values | 17755 |
| Ø Items / Transaction | 49.98 |
| [MB] Memory | 97.48 |

| Twitter | |
|---|---|
| #Transactions | 389894 |
| #Items | 938723 |
| #Item values | 253982 |
| Ø Items / Transaction | 2.41 |
| [MB] Memory | 7.50 |

# Different Datasources – Performance



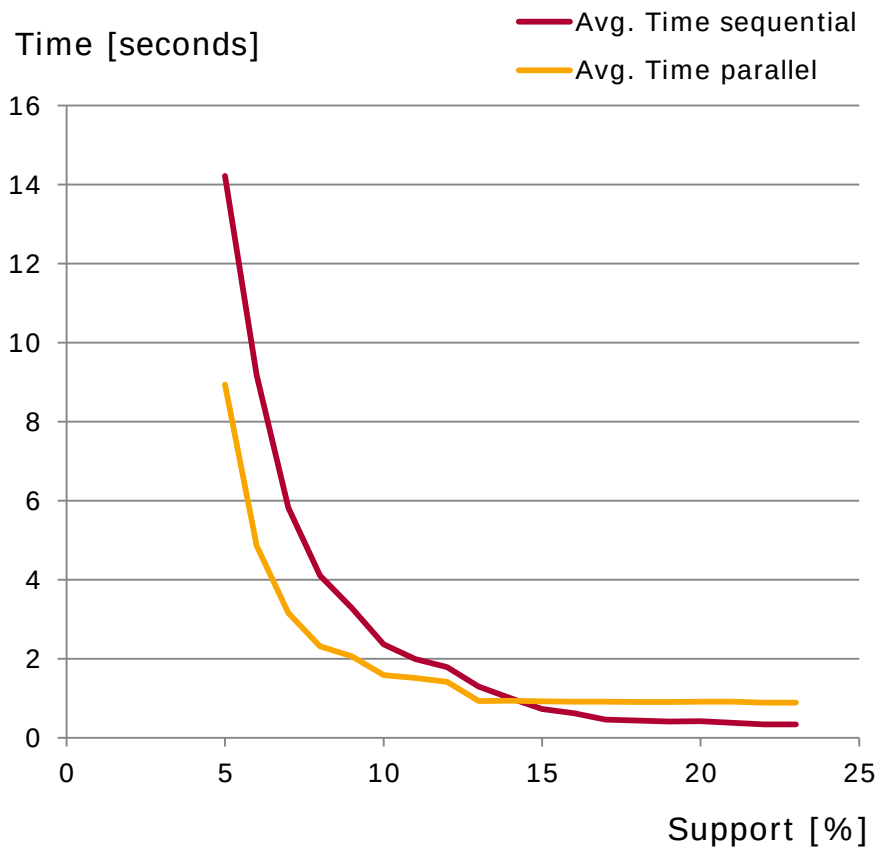
**System:**

- DELL StudioXPS
- 64 bit Windows 7 Enterprise
- Intel Core i5 M520 2,40 GHz
- 4 GB RAM

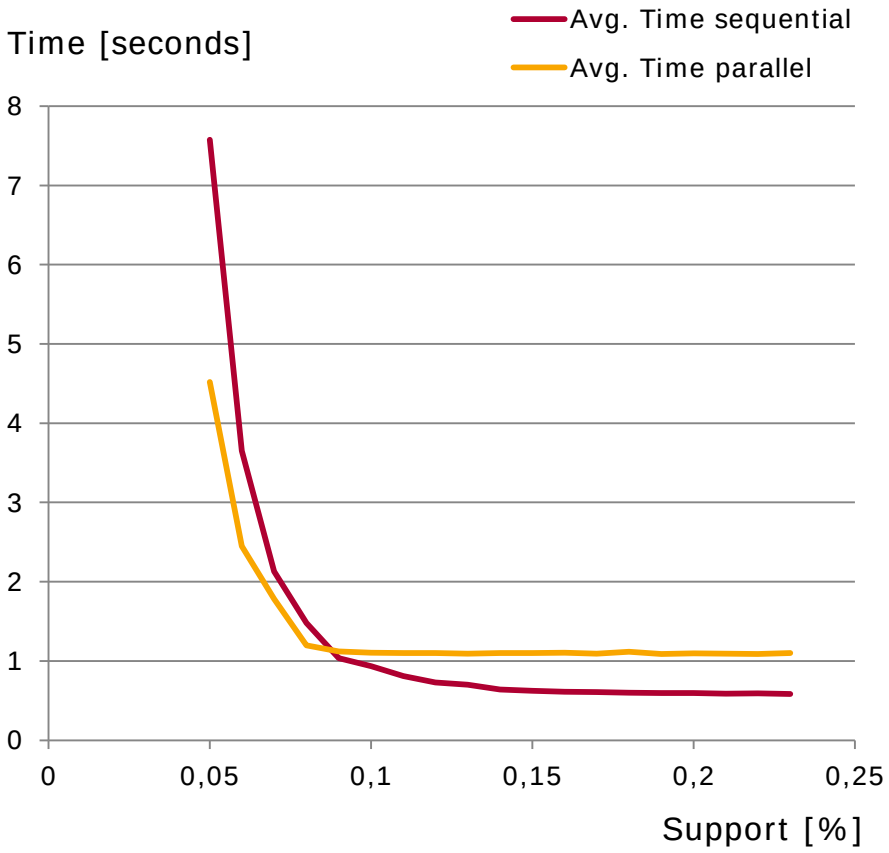
# Different Datasources – Performance



## Netflix




## Twitter



# Conclusion




## Fuzzy Pattern Mining

- Delivers better results for some Use Cases
- Performs worse than non Fuzzy Algorithms
- Introduces Accuracy Probability

## Parallelisation

- Reduces runtime by ~40% on Intel i5
- Performs worse than non parallelised version in sub-second runtimes

## Different Datasources

- Require different min support thresholds

# Sources



**Paper:** (Links Stand 28.06.12)

- "Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination", Christian Borgelt, http://www.borgelt.net/relim.html
- "Mining Fuzzy Frequent Item Sets", Xiaomeng Wang, Christian Borgelt, and Rudolf Kruse, http://www.borgelt.net/relix.html

**Bilder:** (Links Stand 28.06.12)

- http://regrounding.files.wordpress.com/2011/07/doctor-talking-to-patient.jpg
- http://upload.wikimedia.org/wikipedia/commons/a/a2/2009-07-15Muenchen_Nymphenburgerpark_Kronprinzengarten_Quelle.JPG
- http://www.wiwo.de/images/autobahn-stau/6483278/2.jpg
- https://naturfotografen-forum.de/data/media/7/Margeriten%2520unscharf%2520klein::Kevin_Winterhoff_rembrandt_unscharf_van_picasso_monet_gemaelde_gogh_hundertwasser_chagall.jpg
- http://images.fastcompany.com/upload/inline-sweat-the-small-details.jpg