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)
  \( \rightarrow \approx 5.4 \) items / transaction

Questions:
- Which claims occur 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)

  \[
  \text{LostProbability}_i := \frac{\#\text{not reported items } i}{\#\text{world population} - \#\text{reported items } i}
  \]

- can be false (claim was wrongly diagnosed or pretended)

  \[
  \text{AccuracyProbability}_i := \frac{\#\text{correct reported items } i}{\#\text{all reported items } i}
  \]

- 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:

\[
\text{LostProbability}_{HIV} = \frac{\text{#not reported HIV claims}}{\text{#world population} - \text{#reported items}^i} = \frac{36 - 22 \text{ million}}{7025 - 22 \text{ million}} = 0.2% \\
\]

\[
\text{AccuracyProbability}_{HIV} = \frac{\text{#correct reported HIV claims}}{\text{#all reported HIV claims}} = \frac{22 - 0.2 \text{ million}}{22 \text{ million}} = 99% \\
\]
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*
1. Load transactions (in memory)
1. Load transactions (in memory)
2. Count item frequencies

1. Iteration
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 to the items frequency
5. Create Relim datastructure
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

Pattern Mining – Relim & Relix
| Thomas Stening, Thorsten Papenbrock | 10. July 2012
Relix – Recursive Tree Processing
Relix – Recursive Tree Processing

Side recursion, Prefix: e
Relix – Recursive Tree Processing

Side recursion,
Prefix: e
Relix – Recursive Tree Processing

Side recursion, Prefix: \( e \)

Main recursion

Relix – Recursive Tree Processing

Side recursion,
Prefix: a

**System:**
- DELL StudioXPS
- 64 bit Windows 7 Enterprise
- Intel Core i5 M520 2,40 GHz
- 4 GB RAM
Fuzzy Datamining – Performance

Time [seconds]

Support [%]

Relim AVG Time
Relix AVG Time

Support [%]

Fuzzy Datamining – Results

Results [#]

Support [%]

- Red line: Relim ResultCount
- Orange line: Relix ResultCount
Fuzzy Datamining – Frequent Sets

\[
\text{GoogleConfidence}(F) = \frac{\text{results}(F)}{\sum_{i=0}^{\#F} \text{results}(F_i) / \#F}
\]

- **GoogleConfidence(“Apfel”,”Birne”)**
  
  \[
  = \frac{\text{results(“Apfel Birne”)}}{\text{results(“Apfel”) + results(“Birne”)}}
  \]
  
  \[
  = \frac{2.970.000}{6.210.000 + 8.170.000} \times 2
  \]
  
  \[
  = 0.0413
  \]

- **GoogleConfidence(“Apfel”,”Auto”)**
  
  \[
  = \frac{\text{results(“Apfel Auto”)}}{\text{results(“Apfel”) + results(“Auto”)}}
  \]
  
  \[
  = \frac{2.720.000}{6.210.000 + 908.000.000} \times 2
  \]
  
  \[
  = 0.0059
  \]
Fuzzy Datamining – Frequent Sets

GoogleConfidence(F) = \frac{\sum results(F)}{\sum_{i=0}^{\#F} results(F_i) / \#F}

- **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]

\[ \sum GoogleConfidence = 0.3592 \] \quad \sum GoogleConfidence = 0.6433
<table>
<thead>
<tr>
<th>Association rule</th>
<th>Conf.</th>
<th>Lift</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Miscellaneous cardiac, Skin and autoimmune disorders] --&gt; [Other neurological, Other metabolic, Miscellaneous 2]</td>
<td>0.50</td>
<td>2.76</td>
</tr>
<tr>
<td>[Other neurological, Other metabolic, Miscellaneous 2] --&gt; [Miscellaneous cardiac, Skin and autoimmune disorders]</td>
<td>0.28</td>
<td>2.76</td>
</tr>
<tr>
<td>[Miscellaneous cardiac, Other neurological, Miscellaneous 2] --&gt; [Skin and autoimmune disorders, Other metabolic]</td>
<td>0.40</td>
<td>2.73</td>
</tr>
<tr>
<td>[Skin and autoimmune disorders, Other metabolic] --&gt; [Miscellaneous cardiac, Other neurological, Miscellaneous 2]</td>
<td>0.34</td>
<td>2.73</td>
</tr>
<tr>
<td>[Miscellaneous cardiac, Arthropathies, Miscellaneous 2] --&gt; [Skin and autoimmune disorders, Other metabolic]</td>
<td>0.40</td>
<td>2.72</td>
</tr>
<tr>
<td>[Skin and autoimmune disorders, Other metabolic] --&gt; [Miscellaneous cardiac, Arthropathies, Miscellaneous 2]</td>
<td>0.38</td>
<td>2.72</td>
</tr>
<tr>
<td>[Miscellaneous cardiac, Skin and autoimmune disorders, Miscellaneous 2] --&gt; [Other neurological, Other metabolic]</td>
<td>0.55</td>
<td>2.71</td>
</tr>
<tr>
<td>[Other neurological, Other metabolic] --&gt; [Miscellaneous cardiac, Miscellaneous 2]</td>
<td>0.42</td>
<td>2.65</td>
</tr>
<tr>
<td>[Miscellaneous cardiac, Miscellaneous 2] --&gt; [Other neurological, Other metabolic]</td>
<td>0.37</td>
<td>2.65</td>
</tr>
<tr>
<td>[Miscellaneous cardiac, Miscellaneous 2] --&gt; [Other metabolic, Arthropathies]</td>
<td>0.41</td>
<td>2.64</td>
</tr>
<tr>
<td>[Other metabolic, Arthropathies] --&gt; [Miscellaneous cardiac, Miscellaneous 2]</td>
<td>0.42</td>
<td>2.64</td>
</tr>
<tr>
<td>[Other metabolic, Miscellaneous 2] --&gt; [Miscellaneous cardiac, Arthropathies]</td>
<td>0.30</td>
<td>2.62</td>
</tr>
<tr>
<td>[Miscellaneous cardiac, Arthropathies] --&gt; [Other metabolic, Miscellaneous 2]</td>
<td>0.57</td>
<td>2.62</td>
</tr>
<tr>
<td>[Miscellaneous cardiac, Other neurological, Miscellaneous 2] --&gt; [Other metabolic]</td>
<td>0.72</td>
<td>2.44</td>
</tr>
</tbody>
</table>
Problem:
• LostProbabilities and AccuracyProbabilities 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
Parallelisation – Algorithm

Main recursion

Side recursion, Prefix: e

**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**

- Time [seconds]
- Support [%]

**Relix**

- Time [seconds]
- Support [%]

**Runtime reduction ~40%**

**Runtime reduction ~43%**
Agenda

- Fuzzy Pattern Mining
- Parallelisation
- Different Datasources
### Different Datasources – Statistics

<table>
<thead>
<tr>
<th></th>
<th>Claims</th>
<th>Facebook Likes</th>
<th>Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#Transactions</td>
<td>#Transactions</td>
<td>#Transactions</td>
</tr>
<tr>
<td></td>
<td>113000</td>
<td></td>
<td>389894</td>
</tr>
<tr>
<td>#Items</td>
<td>610934</td>
<td></td>
<td>938723</td>
</tr>
<tr>
<td>#Item values</td>
<td>46</td>
<td></td>
<td>253982</td>
</tr>
<tr>
<td>Ø Items / Transaction</td>
<td>5.41</td>
<td></td>
<td>2.41</td>
</tr>
<tr>
<td>[MB] Memory</td>
<td>3.64</td>
<td></td>
<td>7.50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Netflix</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#Transactions</td>
<td>463615</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#Items</td>
<td>23168214</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#Item values</td>
<td>17755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ø Items / Transaction</td>
<td>49.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[MB] Memory</td>
<td>97.48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

![Graph: Netflix](image)

Twitter

![Graph: Twitter](image)
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)
- “Mining Fuzzy Frequent Item Sets”, Xiaomeng Wang, Christian Borgelt, and Rudolf Kruse, [http://www.borgelt.net/relix.html](http://www.borgelt.net/relix.html)

**Bilder:** (Links Stand 28.06.12)
- [http://www.wiwo.de/images/autobahn-stau/6483278/2.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](https://naturfotografen-forum.de/data/media/7/Margeriten%2520unscharf%2520klein::Kevin_Winterhoff_rembrandt_unscharf_van_picasso_monet_gemaelde_gogh_hundertwasser_chagall.jpg)