Mining All Non-Derivable Frequent Itemsets

Toon Calders

TU/e

Bart Goethals

ULB

Universiteit Antwerpen
Outline

- Frequent Itemset Mining in 2002
  - Pattern Explosion Problem
  - Condensed Representations
- Non-Derivable Itemsets  PKDD 2002
  - Quick Inclusion-Exclusion  KDID 2005
  - Depth-first NDI Mining  SDM 2005
- Recent Approaches Towards Non-Redundant Pattern Mining
**Association Rules**

<table>
<thead>
<tr>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>TID</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>200</td>
</tr>
<tr>
<td>300</td>
</tr>
<tr>
<td>400</td>
</tr>
</tbody>
</table>

**L₁**

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>2</td>
</tr>
<tr>
<td>{2}</td>
<td>3</td>
</tr>
<tr>
<td>{3}</td>
<td>3</td>
</tr>
<tr>
<td>{5}</td>
<td>3</td>
</tr>
</tbody>
</table>

**L₂**

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1 3}</td>
<td>2</td>
</tr>
<tr>
<td>{2 3}</td>
<td>2</td>
</tr>
<tr>
<td>{2 5}</td>
<td>3</td>
</tr>
<tr>
<td>{3 5}</td>
<td>2</td>
</tr>
</tbody>
</table>

**L₃**

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>{2 3 5}</td>
<td>2</td>
</tr>
</tbody>
</table>

Mining **association rules** between sets of items in large databases

*R Agrawal, T Imieliński, A Swami - ACM SIGMOD Record, 1993 - dl.acm.org*

Cited by 11735
Association rules gaining popularity

Literally hundreds of algorithms:
AIS, Apriori, AprioriTID, AprioriHybrid, FPGrowth, FPGrowth*, Eclat, dEclat, Pincer-search, ABS, DCI, kDCI, LCM, AIM, PIE, ARMOR, AFOPT, COFI, Patricia, MAXMINER, MAFIA, ...
Mushroom has 8124 transactions, and a transaction length of 23

Over 50 000 patterns

Over 10 000 000 patterns
Frequent itemset / Association rule mining
= find all itemsets / ARs satisfying thresholds

Many are redundant

- smoker → lung cancer
- smoker, bald → lung cancer
- pregnant → woman
- pregnant, smoker → woman, lung cancer
## Condensed Representations

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Number of frequent itemsets = 21
- Need a compact representation

**Discovering frequent closed itemsets for association rules**
N Pasquier, Y Bastide, R Taouil, L Lakhal - Database Theory—ICDT'99, 1999
Cited by 1089
Condensed Representations

- Condensed Representation: “Compressed” version of the collection of all frequent itemsets (usually a subset) that allows for lossless regeneration of the complete collection.
  - Closed Itemsets (Pasquier et al, ICDT 1999)
  - Free Itemsets (Boulicaut et al, PKDD 2000)
  - Disjunction-Free itemsets (Bykowski and Rigotti, PODS 2001)
Outline

- Frequent Itemset Mining in 2002
  - Pattern Explosion Problem
  - Condensed Representations
- Non-Derivable Itemsets  PKDD 2002
  - Quick Inclusion-Exclusion  KDID 2005
  - Depth-first NDI Mining  SDM 2005
- Recent Approaches Towards Non-Redundant Pattern Mining
2002

Mining All Non-Derivable Frequent Itemsets

Toon Calders
University of Antwerp
Belgium

Bart Goethals
University of Limburg
Belgium
Redundancies

- How do supports interact?

- What information about unknown supports can we derive from known supports?
  - Concise representation: only store relevant part of the supports
Redundancies

- Agrawal et al. (Monotonicity)
  - $\text{Supp}(AX) \leq \text{Supp}(A)$

- Lakhal et al. (Closed sets)
  - Boulicaut et al. (Free sets)
  - If $\text{Supp}(A) = \text{Supp}(AB)$
    - Then $\text{Supp}(AX) = \text{Supp}(AXB)$
Redundancies

- **Bayardo (MAXMINER)**
  - $\text{Supp}(\text{ABX}) \geq \text{Supp}(\text{AX}) - (\text{Supp}(\text{X}) - \text{Supp}(\text{BX}))$

- **Bykowski, Rigotti (Disjunction-free sets)**
  - if $\text{Supp}(\text{ABC}) = \text{Supp}(\text{AB}) + \text{Supp}(\text{AC}) - \text{Supp}(\text{A})$
  - then $\text{Supp}(\text{ABCX}) = \text{Supp}(\text{ABX}) + \text{Supp}(\text{ACX}) - \text{Supp}(\text{AX})$

\[\text{drop (X, B)}\]
Theoretical study – general framework
  - Deduction Rules

Applicability in
  - Frequent Set Mining
  - Concise Representations
Outline

- Deduction Rules via Inclusion-Exclusion
- Derivable Itemsets
- NDI Algorithm
- Evaluation
- Conclusion & Further Work
I. The Inclusion – Exclusion Principle

\[ \text{supp}(\overline{abc}) = \text{supp}(a) - \text{supp}(ab) - \text{supp}(ac) + \text{supp}(abc) \]

\[ \geq 0 \]

\[ \text{supp}(abc) \geq \text{supp}(ab) + \text{supp}(ac) - \text{supp}(a) \]
Inclusion-exclusion principle:

\[
supp(\overline{abc}) = supp(\{} - supp(a) - supp(b) - supp(c) \\
+ supp(ab) + supp(ac) + supp(bc) \\
- supp(\overline{abc}) \geq 0
\]

\[
supp(ABC) \leq supp(AB) + supp(AC) + supp(BC) \\
- supp(A) - supp(B) - supp(C) \\
+ supp(\{} \}
\]
One more:

\[ \text{supp}(ab\bar{c}) = \text{supp}(ab) - \text{supp}(abc) \geq 0 \]
Complete Set for \( \text{supp}(abc) \)

0 \quad \text{supp}(abc) \geq 0

\begin{align*}
\text{supp}(abc) & \leq \text{supp}(ab) & \text{monotonicity} \\
\text{supp}(abc) & \leq \text{supp}(ac) \\
\text{supp}(abc) & \leq \text{supp}(bc)
\end{align*}

\begin{align*}
\text{supp}(abc) & \geq \text{supp}(ab) + \text{supp}(ac) - \text{supp}(a) & \text{free/closed} \\
\text{supp}(abc) & \geq \text{supp}(ab) + \text{supp}(bc) - \text{supp}(b) \\
\text{supp}(abc) & \geq \text{supp}(ac) + \text{supp}(bc) - \text{supp}(c)
\end{align*}

\begin{align*}
\text{supp}(abc) & \leq \text{supp}(ab) + \text{supp}(ac) + \text{supp}(bc) \\
& \quad - \text{supp}(a) - \text{supp}(b) - \text{supp}(c) + \text{supp}({})
\end{align*}

(PKDD03)
Example: Deduction Rules

\[ \text{supp}(\{\}) = 6 \]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: Deduction Rules

\[
\begin{array}{ccc}
  a & b & c \\
  0 & 0 & 1 \\
  0 & 1 & 1 \\
  0 & 1 & 0 \\
  1 & 1 & 0 \\
  1 & 0 & 0 \\
  1 & 1 & 1 \\
\end{array}
\]

\[
\begin{align*}
\text{supp(\{}\text{\})} &= 6 \\
\text{supp(a)} &= 3 \\
\text{supp(b)} &= 4 \\
\text{supp(c)} &= 3
\end{align*}
\]
Example: Deduction Rules

\[
\begin{align*}
\text{supp}({}) & = 6 & \text{supp}(ab) & = 2 \\
\text{supp}(a) & = 3 & \text{supp}(ac) & = 1 \\
\text{supp}(b) & = 4 & \text{supp}(bc) & = 2 \\
\text{supp}(c) & = 3
\end{align*}
\]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: Deduction Rules

\[
\begin{align*}
\text{supp}(\{\}) & = 6 \\
\text{supp}(\{\}) & = 2 \\
\text{supp}(\text{abc}) & \geq 0 \\
\text{supp}(\text{abc}) & \leq \text{supp}(\text{ab}) \\
\text{supp}(\text{abc}) & \leq \text{supp}(\text{ac}) \\
\text{supp}(\text{abc}) & \leq \text{supp}(\text{bc}) \\
\text{supp}(\text{abc}) & \geq \text{supp}(\text{ab}) + \text{supp}(\text{ac}) - \text{supp}(\text{a}) \\
\text{supp}(\text{abc}) & \geq \text{supp}(\text{ab}) + \text{supp}(\text{bc}) - \text{supp}(\text{b}) \\
\text{supp}(\text{abc}) & \geq \text{supp}(\text{ac}) + \text{supp}(\text{bc}) - \text{supp}(\text{c}) \\
\text{supp}(\text{abc}) & \leq \text{supp}(\text{ab}) + \text{supp}(\text{ac}) + \text{supp}(\text{bc}) - \text{supp}(\text{a}) - \text{supp}(\text{b}) - \text{supp}(\text{c}) + \text{supp}(\{\})
\end{align*}
\]
Example: Deduction Rules

\[ \text{supp}(\{\}) = 6 \quad \text{supp}(ab) = 2 \]
\[ \text{supp}(a) = 3 \quad \text{supp}(ac) = 1 \]
\[ \text{supp}(b) = 4 \quad \text{supp}(bc) = 2 \]
\[ \text{supp}(c) = 3 \]

\[ \text{supp}(abc) \geq 0 \]
\[ \text{supp}(abc) \leq 2 \]
\[ \text{supp}(abc) \leq 1 \]
\[ \text{supp}(abc) \leq 2 \]
\[ \text{supp}(abc) \geq 2 + 1 - 3 = 0 \]
\[ \text{supp}(abc) \geq 2 + 2 - 4 = 0 \]
\[ \text{supp}(abc) \geq 1 + 2 - 3 = 0 \]
\[ \text{supp}(abc) \leq 2 + 1 + 2 - 3 - 4 - 3 + 6 = 1 \]

Hence, \( \text{supp}(abc) \in [0,1] \)
Main Theorem

Theorem
Given: Supp(I) for all I ⊂ J
The deduction rules are sound, complete, and non-redundant for deducing upper and lower bounds on Supp(J).

- Complete: if bounds are [L,U], then
  - Exists consistent D_L s.t. supp(J,D) = L
  - Exists consistent D_U s.t. supp(J,D) = U
Example: Completeness

- Lower bound: 0
- Upper bound: 1

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$\text{supp}({}) = 6 \quad \text{supp}(ab) = 2$

$\text{supp}(a) = 3 \quad \text{supp}(ac) = 1$

$\text{supp}(b) = 4 \quad \text{supp}(bc) = 2$

$\text{supp}(c) = 3$
Example: Completeness

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

- Lower bound: 0
- **Upper bound: 1**

**DU =**

- \(\text{supp} \{\} = 6\)
- \(\text{supp}(ab) = 2\)
- \(\text{supp}(a) = 3\)
- \(\text{supp}(ac) = 1\)
- \(\text{supp}(b) = 4\)
- \(\text{supp}(bc) = 2\)
- \(\text{supp}(c) = 3\)
**Example: Completeness**

\[ D_L = \]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Lower bound:** 0
- **Upper bound:** 1

\[ \text{supp}\{\} = 6 \quad \text{supp}(ab) = 2 \]
\[ \text{supp}(a) = 3 \quad \text{supp}(ac) = 1 \]
\[ \text{supp}(b) = 4 \quad \text{supp}(bc) = 2 \]
\[ \text{supp}(c) = 3 \]
II. Derivable Itemsets

Given: Supp(I) for all $I \subseteq J$
Lower bound on $\text{Supp}(J) = l$
Upper bound on $\text{Supp}(J) = u$

- **Without** counting: $\text{Supp}(J) \in [l, u]$
- $J$ is a **derivable itemset** (DI) iff $l = u$
  - We know $\text{Supp}(J)$ **exactly** without counting!
Derivable Itemsets

J derivable itemset:
- No need to **count** supp(J)
- No need to **store** supp(J)
  - We can use the deduction rules

- Concise representation:
  \[
  C = \{ (J, \text{supp}(J)) \mid \text{J not derivable from } \text{supp}(I), I \subset J \}\]
## Example: Derivable Itemset

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- \( \text{supp}\{\}\) = 6
- \( \text{supp}\{ab\}\) = 2
- \( \text{supp}\{a\}\) = 3
- \( \text{supp}\{ac\}\) = 1
- \( \text{supp}\{b\}\) = 4
- \( \text{supp}\{bc\}\) = 3
- \( \text{supp}\{c\}\) = 3
Example: Derivable Itemset

\[
\begin{align*}
\text{supp}(abc) & \geq 0 \\
\text{supp}(abc) & \leq \text{supp}(ab) \\
\text{supp}(abc) & \leq \text{supp}(ac) \\
\text{supp}(abc) & \leq \text{supp}(bc) \\
\text{supp}(abc) & \geq \text{supp}(ab) + \text{supp}(ac) - \text{supp}(a) \\
\text{supp}(abc) & \geq \text{supp}(ab) + \text{supp}(bc) - \text{supp}(b) \\
\text{supp}(abc) & \geq \text{supp}(ac) + \text{supp}(bc) - \text{supp}(c) \\
\text{supp}(abc) & \leq \text{supp}(ab) + \text{supp}(ac) + \text{supp}(bc) - \text{supp}(a) - \text{supp}(b) - \text{supp}(c) + \text{supp}(\emptyset)
\end{align*}
\]

\[
\begin{align*}
\text{supp}(\emptyset) & = 6 \\
\text{supp}(a) & = 3 \\
\text{supp}(b) & = 4 \\
\text{supp}(c) & = 3 \\
\text{supp}(ab) & = 2 \\
\text{supp}(ac) & = 1 \\
\text{supp}(bc) & = 3
\end{align*}
\]
Example: Derivable Itemset

\[
\begin{align*}
\text{supp}(\{\}) &= 6 \\
\text{supp}(ab) &= 2 \\
\text{supp}(a) &= 3 \\
\text{supp}(ac) &= 1 \\
\text{supp}(b) &= 4 \\
\text{supp}(bc) &= 3 \\
\text{supp}(c) &= 3
\end{align*}
\]

\[
\begin{align*}
\text{supp}(abc) &\geq 0 \\
\text{supp}(abc) &\leq 2 \\
\text{supp}(abc) &\leq 1 \\
\text{supp}(abc) &\leq 2 \\
\text{supp}(abc) &\geq 2 + 1 - 3 = 0 \\
\text{supp}(abc) &\geq 2 + 3 - 4 = 1 \\
\text{supp}(abc) &\geq 1 + 3 - 3 = 1 \\
\text{supp}(abc) &\leq 2 + 1 + 3 - 3 - 4 - 3 + 6 = 2
\end{align*}
\]

Hence, \( \text{supp}(abc) \in [1,1] \)
Derivable Itemsets

Theorem (Monotonicity)
If $J \subseteq K$, $J$ derivable, then $K$ derivable.

Hence, being derivable is anti-monotone !!

Theorem (Halving)
The width of the interval for $J \cup \{A\}$ is at most $\text{half}$ the size of the interval for $J$
Hence, any NDI has length at most $\log(|D|)$ !!!
Non-derivability is monotone
- Levelwise search, apriori-like
- Also depth-first version (SDM 2005)

Only evaluate itemset if
- All its subsets are frequent
- Bounds cannot derive support exactly
Lemma
If bound on I is \([l,u]\) and \(\text{Supp}(I)\) equals either \(l\) or \(u\), then all supersets of \(I\) are derivable

- **Apriori-style algorithm**
  - Itemset of length \(k\) is a candidate if: all subsets
    - are frequent
    - are non-derivable
    - lower nor upper bound equal actual support
  - bounds cannot derive support exactly
Computing the bounds is expensive if done brute-force

Exploit that many of sums share terms:

\[
\overline{abc}d = ab - abc - abd + abcd
\]

\[
\overline{abcd} = a - ab - ac - ad + abc + abd + acd - abcd
\]
Quick Inclusion-Exclusion
KDID 2005

\[ \text{support}(\overline{a}G) = \text{support}(G) - \text{support}(aG) \]

<table>
<thead>
<tr>
<th></th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
<td>{c</td>
<td>b</td>
<td>bc</td>
<td>a</td>
<td>ac</td>
<td>ab</td>
<td>abc</td>
<td></td>
</tr>
<tr>
<td>[c</td>
<td>{c</td>
<td>b\bar{c}</td>
<td>bc</td>
<td>a\bar{c}</td>
<td>ac</td>
<td>ab\bar{c}</td>
<td>abc</td>
<td></td>
</tr>
<tr>
<td>[b\bar{c}</td>
<td>\bar{b}c</td>
<td>b\bar{c}</td>
<td>bc</td>
<td>abc</td>
<td>\bar{a}bc</td>
<td>ab\bar{c}</td>
<td>abc</td>
<td></td>
</tr>
<tr>
<td>[\bar{a}bc</td>
<td>\bar{a}bc</td>
<td>\bar{a}b\bar{c}</td>
<td>\bar{a}bc</td>
<td>abc</td>
<td>\bar{a}bc</td>
<td>ab\bar{c}</td>
<td>abc</td>
<td></td>
</tr>
</tbody>
</table>
Quick Inclusion-Exclusion
KDID 2005
Evaluation - Empirically

- Comparison with other condensed representations
- Influence of Rule Depth
Comparison

(a) BMS-Webview-1

(DAMI 2007)
Comparison

(e) Connect-4

(DAMI 2007)
Comparison

(f) PUMSB  (DAMI 2007)
Influence of Rule Depth

![Graph showing the influence of rule depth on the number of frequent NDIs in the Mushroom dataset (DAMI 2007).]
Evaluation

- Number of frequent NDIs considerable smaller than number of frequent itemsets
- Most work is done by rules of limited depth
- Algorithm is efficient
  - Calculating NDI + deducing DIs often outperforms Apriori
V. Conclusion

- Deduction rules for support
  - Sound, complete, non-redundant
- NDI as a concise representation
- Efficient algorithm for finding NDI
  - NDI is monotone
  - Intervals at least halve each iteration
- Theoretical and empirical evaluation
  - Sometimes better, sometimes worse than other condensed representations
Outline

- Frequent Itemset Mining in 2002
  - Pattern Explosion Problem
  - Condensed Representations
- Non-Derivable Itemsets PKDD 2002
  - Quick Inclusion-Exclusion KDID 2005
  - Depth-first NDI Mining SDM 2005
- Recent Approaches Towards Non-Redundant Pattern Mining
**Illustrative Example: Tiles**

- **Tile** = rectangle of 1’s in a 0-1 matrix
  - We can reorder rows and columns

```
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
```

```
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
```

```
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
```

- **Matrix Reordering Result:**

```
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
```

```
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
0 1 1 0 0 0 1 1 0 0 0 1 1 1 1
```

```
0 0 0 0 0 0 1 1 1 1 1 1
0 0 0 0 0 0 1 1 1 1 1 1
0 0 0 0 0 0 1 1 1 1 1 1
0 0 0 0 0 0 1 1 1 1 1 1
```
Suppose we found a 4x3 tile; area = 12
“Is this significant in a database with 20 tuples, 6 items and all items have a probability of 0.3?”

Approach:
- Characterize the distribution of maximal area of tiles in “random” data
- Compute how likely it is to have a tile of size 16 or more = p-value
Characterizing the distribution is not easy
Therefore: Simulation
- Sample over all databases
- Compute empirical p-value

20 tuples, 6 items; all items are independent and have probability 0.3
Null-model expresses the prior believe of the user

p-values can be computed for every itemset
  - How *extreme* is the observed support given our prior believe?

Select null-model
- Compute p-values for the itemsets
- Rank itemsets according to p-value
Model I: Swap Randomization

1100 ← 2
0110 ← 2
1110 ← 3
0001 ← 1
↑↑↑↑↑
2321

- Uniform over all databases with:
  - the number of rows and columns
  - same row and column marginals

Assessing data mining results via swap randomization
A Gionis, H Mannila, T Mielikäinen… - ACM Transactions on …, 2007
### Swap Randomization

#### Dataset $D_1$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Significant!!**

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Dataset $D_2$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Trivial**

**Buying a lot**

**Buying nothing**

---

**Figure 2:** Examples of two 0–1 datasets, $D_1$ and $D_2$. In both cases we are interested in the correlation between columns (attributes) $X$ and $Y$. The significance of the correlation result might depend on the overall context of the dataset.
Observation:

- When we do have a solution, it is easy to create a new solution.

By swapping we can reach every other database that satisfies the row and column totals.

We can use this observation to generate other databases!
“Swap graph” = Markov Chain
After a while, the distribution will converge to a unique stationary distribution (under conditions)
- Independent of starting point

Under some conditions the stationary distribution of a Markov Chain is uniform
- All nodes are connected
- The chain is reversible
- All nodes have the same degree (in general *not true*)
Figure 4: Convergence: $x$-axis: the number of steps ($\times$ the number of 1's in the data); $y$-axis: the number of frequent itemsets in the sampled datasets, divided by the number of frequent itemsets in the original dataset.
Swap Randomization: Experiments

(a) original kosarak
(b) swapped kosarak
(c) permuted kosarak

(d) original retail
(e) swapped retail
(f) permuted retail

(g) original paleo
(h) swapped paleo
(i) permuted paleo
II. MaxEntropy Based Models

- Given a set of frequency-constraints
- Null-model: usually based upon maximal entropy
  - Database = distribution (Nikolaj Tatti)
    Constraints must be satisfied
    Of these, select the (unique) MaxEnt distribution
    → one database

- Distribution over databases (Tijl De Bie)
  Constraints must be satisfied in expectation
  Select the MaxEnt distribution
  → Distribution over databases
  → Depending on the type of constraint: analytical solution
II. MaxEntropy Based Models

- Why MaxEntropy? **principle of maximum entropy**
  - if nothing known about a distribution except that it belongs to a certain class, pick distribution with the largest entropy. Maximizing entropy minimizes the amount of prior information built into the distribution.

- \( H(X) = -\Sigma_a p(X=a) \log(p(X=a)) \)
  - \(-\log(p(X=a))\) denotes space required to encode a, given an optimal Shannon encoding for the distribution \( p \); characterizes the *information content of a*
  - \( p(X=a) \) denotes the probability that event \( X=a \) occurs
  - \( H(X) = \) average information content
Database = Distribution

- $P(\text{transaction } T) = \left| \{ \text{tid} \mid (\text{tid},T) \in D \} \right| / |D|$
- Entropy(D) = $- \sum_{T \in D} P(T) \log(P(T))$
- Given a set of constraints $C = \{ \text{supp}(J_i) = s_i \}$
  - the null-model is the distribution D that satisfies $\text{supp}(J_i,D) = s_i$
  - and maximizes the entropy
- Comes down to optimizing a convex function under linear constraints
Database = Distribution

- Scoring an itemset J:
  - Compare empirical distribution with the MaxEnt model
    - E.g., KL-divergence
  - Compare support under null-model with true support
  - Compute p-value of the support of J under the null-model

- Itemsets to summarize the database
  - Select the itemset that minimizes KL-divergence between MaxEnt distribution and true dataset
Distribution over Databases

- Original database is $n \times m$
- Consider all 0-1 databases of size $n \times m$
- Every database has a probability
  - distribution over databases

\[ E(\text{supp}(J)) = \sum_D P(D) \text{supp}(J,D) \]

- Given a set of constraints
  \[ C = \{ \text{supp}(J_i) = s_i \} \]
  - the null-model is the distribution $D$ that satisfies
    \[ E(\text{supp}(J_i)) = s_i \]
    and maximizes the entropy
Depending on the type of constraints it can be solved; e.g.,
- density of a given tile
- row and column marginals
- Anything expressible as a linear constraint in the variables $D[i,j]$

Does not work for frequency constraints!
- $freq(ab) = 5 \Rightarrow D[1,a]*D[1,b] + D[2,a]*D[2,b] + \ldots = 5$
III. Minimal Description Length

- A good model helps us to compress the data and is compact
  - Let $L(M)$ be the description length of the model,
  - Let $L(D|M)$ be the size of the data when compressed by the model

- Find set of patterns (model $M$) that minimizes:
  $L(M) + L(D|M)$

- Explicit trade-off; adding a pattern:
  - Increases $L(M)$,
  - Decreases $L(D|M)$
III. Minimal Description Length

- Rank itemsets according to how well they can be used to compress the dataset
  - Property of a set of patterns

- The “Krimp” algorithm was the first to use this paradigm in itemset mining
  - Assumes a seed set of patterns
  - A subset of these patterns is selected to form the “code book”
  - The best codebook is the one that gives the best compression

**Krimp: mining itemsets that compress**
J Vreeken, M van Leeuwen, A Siebes - Data Mining and Knowledge ..., 2011
III. Minimal Description Length

Figures of Vreeken et al.
III. Minimal Description Length

Fig. 4  Krimp in action

Figure of Vreeken et al.
Frequent pattern mining introduced in 1993
- Hundreds of algorithms
- Pattern explosion problem

Early 2000: condensed representations
- NDI was one of them

Contributions of PKDD 2002 paper:
- Sound, complete, non-redundant rules
- Unifying framework (PKDD 2003)
- Competitive condensed representations
More recent approaches:

- Based on Minimal Description Length Principle:
  - Pattern collection → Model
  - Good models allow for compression
  - Heuristic method for selecting “best” model

- Based on statistics
  - Null-model expresses “expectation”
  - Measure how “surprising” a pattern is w.r.t. the expectation
  - Update null model when accepting a pattern
Future?

- Make these approaches more practical
  - Work only on toy-examples

- Extend to other pattern domains
  - Sequences, graphs, dynamic graphs

Please, please, please, STOP making new algorithms for mining all frequent itemsets ...