**CHARM: An Efficient Algorithm for Closed Itemset Mining**

Authors: Mohammed J. Zaki and Ching-Jui Hsiao
Presenter: Junfeng Wu

---

**Introductions**

When we are mining association rules in a database, a **huge number** of frequent patterns (itemsets) will be generated.

- Database: \{(1,2,3,4),(1,2,3,4,5,6)\}
- Minimum support = 50%
- 63 frequent itemsets
  \[\{(1),(2),(3),(4),(5),(6),(1,2),(1,3),\ldots,(1,2,3,4,5,6)\}\]

---

**Introductions**

Closed frequent itemsets are **non-redundant** representations of all frequent itemsets.

Mining association rules on closed frequent itemsets is a much easier task.

In the previous database, the number of closed frequent itemsets is only 2, \((1,2,3,4)\) and \((1,2,3,4,5,6)\).
Closed frequent itemsets

- A frequent itemset \( X \) is closed if and only if there is no itemset \( Y \) such that
  - \( Y \) subsumes \( X \)
  - every transaction that contains \( X \) also contains \( Y \)

Database: \( \{(1,2,3,4),(1,2,3,4,5,6)\} \)
Itemset \( (1,2) \) is not a closed itemset.
Itemset \( (1,2,3,4) \) is a closed itemset.

Example Database

<table>
<thead>
<tr>
<th>Itemsets</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100%</td>
</tr>
<tr>
<td>A,C,D,T</td>
<td>95%</td>
</tr>
<tr>
<td>A,C,D,W</td>
<td>85%</td>
</tr>
<tr>
<td>A,C</td>
<td>75%</td>
</tr>
<tr>
<td>A</td>
<td>65%</td>
</tr>
<tr>
<td>C</td>
<td>55%</td>
</tr>
<tr>
<td>D</td>
<td>45%</td>
</tr>
<tr>
<td>T</td>
<td>35%</td>
</tr>
<tr>
<td>W</td>
<td>25%</td>
</tr>
</tbody>
</table>

Distinct Database Items

<table>
<thead>
<tr>
<th>Jane Austen</th>
<th>Agatha Christie</th>
<th>Sir Arthur Conan Doyle</th>
<th>Mark Twain</th>
<th>P.G. Wodehouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
<td>D</td>
<td>T</td>
<td>W</td>
</tr>
</tbody>
</table>

DATABASE

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
<th>Support</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A,C,T,W</td>
<td>100%</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>C,D,W</td>
<td>83%</td>
<td>W,C</td>
</tr>
<tr>
<td>3</td>
<td>A,C,T,W</td>
<td>67%</td>
<td>A,D,T,AC,AW,CD,CT,ACW</td>
</tr>
<tr>
<td>4</td>
<td>A,C,D,W</td>
<td>50%</td>
<td>AT,D,W,TW,ACT,ATW,CDW,C TW,ACTW</td>
</tr>
<tr>
<td>5</td>
<td>A,C,D,T,W</td>
<td>45%</td>
<td>AT,D,W,TW,ACT,ATW,CDW,C TW,ACTW</td>
</tr>
<tr>
<td>6</td>
<td>C,D,T</td>
<td>35%</td>
<td>AT,D,W,ACT,ATW,CDW,C TW,ACTW</td>
</tr>
</tbody>
</table>

Horizontal/Vertical format database

- Horizontal format database
  - Each record is a set of items.
  - Each record is assigned a distinct number named transaction id.
- Vertical format database
  - Each record is a set of transaction id about an item.
  - This item occurs in these transactions.
Notations

Given an itemset \( X \), \( t(X) \) is the set of all tids that contains \( X \).
For example: \( t(ACW) = 1345 \)

Given a tidset \( Y \), \( i(Y) \) is the set of all common items to all the tids in \( Y \).
For example: \( i(12) = CW \)

Given an itemset \( X \), \( c(X) \) is the smallest closed set that contains \( X \).
For example: \( c(A) = c(C) = c(W) = ACW \)

Itemset-Tidset Search Tree (IT-tree)

- Each node in the IT-tree is an itemset-tidset pair, \( X \times t(X) \).
  For example: \( AT \times 135 \)

- All the children of node \( X \) share the same prefix \( X \) and belong to an equivalence class

Example of IT-tree

```
\( \emptyset \)

A 1345
AC 1345
ACD 45
ACDTW 135
ACDTW 5

AD 45
ACT 135
ACD 45
ACDTW 45

AT 135
ADT 5
ATW 135
ACDTW 45

AW 1345
CD 2456
CDT 45
CDTW 5

CT 1356
CID 2456
CDTW 5

CW 12345
CD 2456
CDT 45
CDTW 5

DT 56
DW 245
DTW 5

T 1356
T 12345

W 12345
```

Theorem 1

Let \( X \times t(X)_1 \) and \( X \times t(X)_2 \) be any two members of a class \( \langle p \rangle \), with \( X \leq X_j \), where \( f \) is a total order. The following four properties hold:

1. If \( t(X)_1 = t(X)_2 \), then \( c(X)_1 = c(X)_2 = c(X \cup X_j) \)
2. If \( t(X)_1 \subset t(X)_2 \), then \( c(X)_1 \neq c(X)_2 \) but \( c(X)_2 = c(X \cup X_j) \)
3. If \( t(X)_1 \supset t(X)_2 \), then \( c(X)_1 \neq c(X)_2 \) but \( c(X)_1 = c(X \cup X_j) \)
4. If \( t(X)_1 \neq t(X)_2 \), then \( c(X)_1 \neq c(X)_2 \neq c(X \cup X_j) \)
CHARM algorithm

CHARM \( (D, \text{min}_{\sup}) \):
1. \[ |P| = \{X_i \times t(X_i) : X_i \in D \land \sigma(X_i) \geq \text{min}_{\sup} \} \]
2. \( \text{CHARM-Extend} \ (|P|, C = \{\}) \)
3. \( \text{return } C / \text{all closed sets} \)

CHARM-Extend \( (|P|, C) \):
4. for each \( X_i \times t(X_i) \) in \( |P| \)
5. \( |P| = \emptyset \) and \( X = X_i \)
6. for each \( X_j \times t(X_j) \) in \( |P| \), with \( X_j \geq X_i \)
7. \( X = X \cup X_j \) and \( Y = t(X_i) \cap t(X_j) \)
8. \( \text{CHARM-Property} \ (|P|, |P|) \)
9. \( \text{if } (|P| \neq \emptyset ) \) then \( \text{CHARM-Extend} \ (|P|, C) \)
10. delete \( |P| \)
11. \( C = C \cup X \) / if \( X \) is not subsumed

How does CHARM work?

Subsumption Checking

Before add a set \( X \) to the current set of closed set, we need check if \( X \) is subsumed by some closed sets.

- Comparing \( X \) with all closed set is expensive.

Solution: using hash function to retrieve relevant closed sets

Hash function

\[ h(X) = \sum_{T \in t(X)} T \]

The sum of the tids in the tidset of an itemset

Assumption: itemsets with the same hash key have different supports.
Complexity issues

Comparing two itemset’s tidsets becomes a time consuming task when tidset gets very large.

Keeping all tids of itemsets in memory needs lots of space.

Solution: using diffsets

Diffset and Tidset

Let \( m(X_i) \) and \( m(X_j) \) denote the number of mismatches in the diffsets \( d(X_i) \) and \( d(X_j) \)

For example: \( X_i=D, X_j=T \), then \( d(X_i)=2456, d(X_j)=1356 \), \( m(X_i)=|(13)|=2, m(X_j)=|(24)|=2 \)

\[
\begin{align*}
m(X_i) = 0 \quad \text{and} \quad m(X_j) = 0, \quad &\text{then} \quad d(X_i) = d(X_j) \quad \text{or} \quad t(X_i) = t(X_j) \\
m(X_i) > 0 \quad \text{and} \quad m(X_j) = 0, \quad &\text{then} \quad d(X_i) \supset d(X_j) \quad \text{or} \quad t(X_i) \supset t(X_j) \\
m(X_i) = 0 \quad \text{and} \quad m(X_j) > 0, \quad &\text{then} \quad d(X_i) \subset d(X_j) \quad \text{or} \quad t(X_i) \subset t(X_j) \\
m(X_i) > 0 \quad \text{and} \quad m(X_j) > 0, \quad &\text{then} \quad d(X_i) \neq d(X_j) \quad \text{or} \quad t(X_i) \neq t(X_j)
\end{align*}
\]

CHARM using diffsets
Performance study

Datasets

<table>
<thead>
<tr>
<th>Database</th>
<th># Items</th>
<th>Avg. Length</th>
<th>Std. Dev.</th>
<th># Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>chess</td>
<td>76</td>
<td>37</td>
<td>0</td>
<td>3,196</td>
</tr>
<tr>
<td>connect</td>
<td>130</td>
<td>43</td>
<td>0</td>
<td>67,557</td>
</tr>
<tr>
<td>mushroom</td>
<td>120</td>
<td>23</td>
<td>0</td>
<td>8,124</td>
</tr>
<tr>
<td>punch*</td>
<td>7117</td>
<td>50</td>
<td>2</td>
<td>49,046</td>
</tr>
<tr>
<td>punch</td>
<td>7117</td>
<td>74</td>
<td>0</td>
<td>49,046</td>
</tr>
<tr>
<td>gazelle</td>
<td>498</td>
<td>2.5</td>
<td>4.9</td>
<td>59,604</td>
</tr>
<tr>
<td>T1014D100K</td>
<td>1000</td>
<td>10</td>
<td>3.7</td>
<td>100,000</td>
</tr>
<tr>
<td>T4016D100K</td>
<td>1000</td>
<td>40</td>
<td>8.5</td>
<td>100,000</td>
</tr>
</tbody>
</table>

Performance study

Performance study
Scalability

Linear increasing in the running time with increasing number of transactions at a giving support.

Memory usage

The memory usage is 50 times smaller by using diffsets than using tidsets.

<table>
<thead>
<tr>
<th>DB</th>
<th>50%</th>
<th>20%</th>
<th>DB</th>
<th>0.1%</th>
<th>0.05%</th>
<th>DB</th>
<th>1%</th>
<th>0.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect</td>
<td>0.68MB</td>
<td>1.17MB</td>
<td>gazelle</td>
<td>0.15MB</td>
<td>1.24MB</td>
<td>T40H10D100K</td>
<td>0.39MB</td>
<td>0.52MB</td>
</tr>
</tbody>
</table>

Conclusion

- Advantage of CHARM
  - Faster than other algorithm at low support threshold
  - Faster than other algorithm on a database with very long closed patterns
- Disadvantage of CHARM
  - Slower than Closet when most of closed sets are 2-itemset

Comments

- Strength
  - The ideas in the paper are intuitive.
  - The authors first introduced an efficient data structure (IT-tree) for closed itemset mining.
  - The authors demonstrated the algorithm on various datasets.
  - The experimental studies are convincing.
- Weakness
  - The algorithm requires the conversion of database from horizontal format to vertical format.
- Follow-up
  - Closet+ (Wang et al, 2003) beats CHARM one year later.
THANK YOU!

Questions or comments?