Sequence Clustering

COMP 790-90 Research Seminar
Spring 2011

ApproxMAP

• Sequential Pattern Mining
• Support Framework
• Multiple Alignment Framework
• Evaluation
• Conclusion
Inherent Problems

- **Exact match**
  - A pattern gets support from a sequence in the database if and only if the pattern is exactly contained in the sequence
  - Often may not find general long patterns in the database
  - For example, many customers may share similar buying habits, but few of them follow an exactly same pattern

- **Mines complete set: Too many trivial patterns**
  - Given long sequences with noise
    - too expensive and too many patterns
  - Finding max / closed sequential patterns is non-trivial
    - In noisy environment, still too many max/close patterns

⇒ **Not Summarizing Trend**

Multiple Alignment

- line up the sequences to detect the trend
  - Find common patterns among strings
  - DNA / bio sequences

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>E</th>
<th>R</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>T</td>
<td>T</td>
<td>E</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>0</td>
<td>T</td>
<td>T</td>
<td>0</td>
<td>R</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>A</td>
<td>0</td>
<td>T</td>
<td>T</td>
<td>E</td>
<td>R</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>S</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
<td>T</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>A</td>
<td>0</td>
<td>T</td>
<td>T</td>
<td>E</td>
<td>R</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>
Edit Distance

✓ Pairwise Score = edit distance=\text{dist}(S_1, S_2)
  – Minimum # of ops required to change $S_1$ to $S_2$
  – Ops = \text{INDEL}(a) \text{ and/or } \text{REPLACE}(a,b)$

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>E</th>
<th>R</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>()</td>
<td>()</td>
<td>T</td>
<td>E</td>
<td>R</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>INDEL</td>
<td>INDEL</td>
<td></td>
<td></td>
<td>REPL</td>
<td></td>
</tr>
</tbody>
</table>

• Multiple Alignment Score
  ➢ $\sum PS(seq_i, seq_j)$ ($\forall 1 \leq i \leq N \text{ and } 1 \leq j \leq N$)
  ➢ Optimal alignment : minimum score

Weighted Sequence

• Weighted Sequence : profile
  ➢ Compress a set of aligned sequences into one sequence

<table>
<thead>
<tr>
<th>seq_1</th>
<th>(A)</th>
<th>(B)</th>
<th>(DE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq_2</td>
<td>(AE)</td>
<td>(H)</td>
<td>(BC)</td>
</tr>
<tr>
<td>seq_3</td>
<td>(A)</td>
<td>(BCG)</td>
<td>(D)</td>
</tr>
</tbody>
</table>
**Consensus Sequence**

- **strength**(*i, j*) = \# of occurrences of item *i* in position *j*  
  total # of sequences

- **Consensus itemset (j)**
  \{ *i_a* | \forall i_b \in (I \cup O) & strength(*i_a*, *j*) \geq min_strength \}

- **Consensus sequence : min_strength=2**
  - concatenation of the consensus itemsets for all positions excluding any null consensus itemsets

<table>
<thead>
<tr>
<th>seq₁</th>
<th>(A)</th>
<th>(B)</th>
<th>(DE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq₂</td>
<td>(AE)</td>
<td>(H)</td>
<td>(BC)</td>
</tr>
<tr>
<td>seq₃</td>
<td>(A)</td>
<td>(BCG)</td>
<td>(D)</td>
</tr>
<tr>
<td>Consensus Sequence</td>
<td>(A)</td>
<td>(BC)</td>
<td>(DE)</td>
</tr>
</tbody>
</table>

**Multiple Alignment Pattern Mining**

- **Given**
  - \( N \) sequences of sets,
  - Op costs (INDEL & REPLACE) for itemsets, and
  - Strength threshold for consensus sequences
    - can specify different levels for each partition

- **To**
  - (1) partition the \( N \) sequences into \( K \) sets of sequences such that the sum of the \( K \) multiple alignment scores is minimum, and
  - (2) find the optimal multiple alignment for each partition, and
  - (3) find the pattern consensus sequence and the variation consensus sequence for each partition
ApproxMAP
(Approximate Multiple Alignment Pattern mining)

• Exact solution: Too expensive!

• Approximation Method
  ➢ Group: \(O(kN) + O(N^2L^2I)\)
    ➢ partition by Clustering (k-NN)
    ➢ distance metric
  ➢ Compress: \(O(nL^2)\)
    ➢ multiple alignment (greedy)
  ➢ Summarize: \(O(1)\)
    ➢ Pattern and Variation Consensus Sequence
  ➢ Time Complexity: \(O(N^2L^2I)\)

Multiple Alignment: Weighted Sequence

<table>
<thead>
<tr>
<th>seq1</th>
<th>(A)</th>
<th>(B)</th>
<th>(DE)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>seq2</td>
<td>(AE)</td>
<td>(H)</td>
<td>(B)</td>
<td>(D)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>seq1</td>
<td>(A)</td>
<td>(BCG)</td>
<td>(D)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Evaluation Method: Criteria & Datasets**

- **Criteria**
  - **Recoverability**: max patterns
    - degree of the underlying patterns in DB detected
    - \( \sum E(F_B) \times [\max_{res\,pat\,P}(|B\otimes P|) / E(L_B)] \)
    - Cutoff so that \( 0 \leq R \leq 1 \)
  - # of spurious patterns
  - # of redundant patterns
  - Degree of extraneous items in the patterns
    - total # of extraneous items in P / total # of items in P

- **Datasets**
  - Random data: Independence between and across itemsets
  - Patterned data: IBM synthetic data (Agrawal and Srikant)
  - Robustness w.r.t. noise: alpha (Yang – SIGMOD 2002)
  - Robustness w.r.t. random sequences (outliers)

---

**Evaluation : Comparison**

<table>
<thead>
<tr>
<th></th>
<th>ApproxMAP</th>
<th>Support Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Data</td>
<td>No patterns with more than 1 item returned</td>
<td>Lots of spurious patterns</td>
</tr>
<tr>
<td>Patterned Data</td>
<td>k=6 &amp; MinStrgh=30%</td>
<td>MinSup=5%</td>
</tr>
<tr>
<td>10 patterns</td>
<td>Recoverability: 92.5%</td>
<td>Recoverability: 91.6%</td>
</tr>
<tr>
<td>embedded into 1000 seqs</td>
<td>10 patterns returned</td>
<td>253,924 patterns returned</td>
</tr>
<tr>
<td></td>
<td>2 redundant patterns</td>
<td>247,266 redundant patterns</td>
</tr>
<tr>
<td></td>
<td>0 spurious patterns</td>
<td>6,648 spurious patterns</td>
</tr>
<tr>
<td></td>
<td>0 extraneous items</td>
<td>93,043=5.2% extraneous items</td>
</tr>
<tr>
<td>Noise</td>
<td>Robust</td>
<td>Not Robust</td>
</tr>
<tr>
<td></td>
<td>Recoverability degrades fast</td>
<td></td>
</tr>
<tr>
<td>Outliers</td>
<td>Robust</td>
<td>Robust</td>
</tr>
</tbody>
</table>
Robustness w.r.t. noise

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Results : Scalability

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL
Evaluation: Real data

- Successfully applied **ApproxMAP** to sequence of monthly social welfare services given to clients in North Carolina
- Found interpretable and useful patterns that revealed information from the data

Conclusion: why does it work well?

- Robust on random & weak patterned noise
  - Noises can almost never be aligned to generate patterns, so they are ignored
  - If some alignment is possible, the pattern is detected
- Very good at organizing sequences
  - when there are “enough” sequences with a certain pattern, they are clustered & aligned
  - When aligning, we start with the sequences with the least noise and add on those with progressively more noise
  - This builds a center of mass to which those sequences with lots of noise can attach to
- Long sequence data that are not random have unique signatures
Conclusion

- Works very well with market basket data
  - High dimensional
  - Sparse
  - Massive outliers
- Scales reasonably well
  - Scales very well w.r.t # of patterns
  - $k$ : scales very well = $O(1)$
  - $DB$ : scales reasonably well = $O(N^2 L^2 I)$
  - Less than 1 minute for $N=1000$ on Intel Pentium