CS412 - Chapter 7 Review

Xiaolong Wang

December 5, 2015

Outline

1. Constraint
   - (Anti-)Monotonic/Succinct/Convertible definitions
   - Pruning when mining
2. Seuqnece
   - Sequence definition
   - GSP: Apriori-based mining
   - PrefixSpan: Devide-and-Conquer mining
3. Graph
   - GSPAN: Minimum DFS code and rightmost extension

Constraints

Mining only the patterns that satisfy constraints, which in turn often can be used for pruning.

- Anti-monotonic: satisfying pattern implies satisfying sub-patterns.
  - $max(S) \leq v$
- Monotonic: satisfying pattern implies satisfing super-patterns.
  - $min(S) \leq v$
- Succinct: satisfying patterns all based on one set A
  - $max(S) \leq v$ ($A = \{x|x \leq v\}$, then $S \subseteq A$)
- Convertible: constraints become anti-monotonic or/and monotonoic if properly ordering items.
  - $avg(S) \geq v$ (ordering items descendingly makes it anti-monotonoic)

Pattern Pruning with constraints

if a constraint only being convertible, it cannot be pushed deep into an Apriori mining algorithm

Example: $avg(S) \geq 10$

constraint being convertible anti-monotonoic.
$\{100, 5, 1\}$ satisfy constraint. But not $\{10, 1\}$.
However, without $\{10, 1\}$, Apriori cannot generate $\{100, 5, 1\}$

GSP and PrefixSpan

- Sequence: ordered lists of item(sets).

Example: $<$ beer, (cereal, milk), bacon $>$

first, buy beer;
then, buy cereal and milk at the same time;
last, bacon is bought.

- GSP: Apriori-based candidate generation.
  - candidate joining:

1st seq	2nd seq	joined seq
$< abc >$	$< bcd >$	$< abcd >$
$< (ab)(bc)c >$	$< b(bc)(cd) >$	$< (ab)(bc)(cd) >$
$< a >$	$< b >$	$< ab >$ or $< (ab) >$

  - length-100 sequence: it needs candidates

$$\sum_{i=1}^{100} \binom{100}{i} = 2^{100} - 1$$

GSP and PrefixSpan

PrefixSpan: Divide-and-Conquer mining based on prefix

- prefix $< a >$ and projected database $< b(ab)c >$

prefix	projection
$< ab >$	$< (ab)c >$
$< aa >$	$< (\_b)c >$
$< (ab) >$	$< c >$
$< ac >$	$NULL$

- physical- and pseudo-projection
  - pseudo- requires memory to hold the data, physical- cost more time to copy projections
  - combined memthod: load batch data, run in pseudo-projection in memory, and write physical projection when swapping.

GSPAN

- Minimum DFS code: minimum lexicographic DFS code; search has no redundancy
  - Edge: $(v_i, v_j, l(v_i), l(v_j), l(v_i, v_j))$
  - DFS code: list of edge tuples
    - extend new node and add forward edge
    - add backward edges
- Right-most extension: search is complete (no missing)
  - rightmost path and rightmost node
  - extension:
    - from RM node link backward edges;
    - extend forward edges and augment RM path
    - backtrack: use next available RM node on path

Q&A