Mining Closed Sequential Patterns in Large Datasets

Presenter: Ildar Nurgaliev
Lab: Dainfos

Main idea

Instead of mining the complete set of frequent subsequences we mine frequent *closed subsequences*

Benefits

- can mine really long sequences
- produce significantly less number of discovered frequent sequences

Preliminary Concepts

Sequence

- items: $I = \{i_1, i_2, ..., i_m\}$
- itemset (t_i): $t_i \subseteq I$
- sequence (ordered list): $s = \langle t_1, t_2, ..., t_m \rangle$
- size $|$s$|$: number of itemsets in s
- length l(s): $l(s) = \sum_{i=1}^{n} |t_i|$

Preliminary Concepts

α sub-sequence of β OR β super-sequence of α (contains)

- $\alpha = \langle \alpha_1, \alpha_2, ..., \alpha_m \rangle$
- $\beta = \langle \beta_1, \beta_2, ..., \beta_m \rangle$
- $\alpha \sqsubseteq \beta$ (if $\alpha \neq \beta$, written as $\alpha \sqsubset \beta$)
- iff $\exists i_1, i_2, ..., i_m$, such that
  $1 \leq i_1 < i_2 < ... < i_m \leq n$ and
  $\alpha_1 \subseteq \beta_i, \alpha_2 \subseteq \beta_{i_2}, ..., \alpha_m \subseteq \beta_{i_m}$
- β absorbs α: if β contains α and their *support* are the same

Preliminary Concepts
Support

- $D = \{s_1, s_2, ..., s_n\}$: sequence database
- each s associated with id (id of s_i is i)
- $|D|$: number of s in D
- $support(\alpha)$: number of s in D which contain α
  $support(\alpha) = |\{s|s \in D \text{ and } \alpha \sqsubseteq s\}|$
- min_sup: minimum support threshold

Preliminary Concepts

Frequent sequential pattern (FS) and closed FS (CS)

- FS: includes all s of *support(s)* $\leq$ *min_sup*
- $CS = \{\alpha | \alpha \in \; FS \; and \; \nexists \beta \in \text{FS}$
  such that $\alpha \sqsubseteq \beta \; and \; support(\alpha) = support(\beta)\}$
- *closed sequence mining*: find CS above *min_sup*
- database containment relation $D \sqsubseteq D'$:
  if $\exists$ an injective function $f : D \rightarrow D'$, s.t.
  $\forall s \in D, s \sqsubseteq f(s)$

Preliminary Concepts

Item extension

- Given: $s = \langle t_1, \ldots, t_m \rangle$ and item α
- $s \diamond \alpha$: concatenation (I-Step or S-Step)
- $s \diamond_i \alpha = \langle t_1, \ldots, t_m \cup \{\alpha\} \rangle$ if $\forall k \in r_m, k < \alpha$
  Example: $\langle(\alpha e)\rangle$ is I-Step extension of $\langle(\alpha)\rangle$
- $s \diamond_s \alpha = \langle t_1, \ldots, t_m, \{\alpha\} \rangle$
  Example: $\langle(\alpha)(c)\rangle$ is S-Step extension of $\langle(\alpha)\rangle$

Preliminary Concepts

Sequnce extension

- Given: $s = \langle t_1, ..., t_m \rangle$ and $p = \langle t'_1, ..., t'_n \rangle$
- $s \diamond p$: concatenation (itemset-extension or sequence-extension)
- $s \diamond_i p = \langle t_1, ..., t_m \cup t'_1, ..., t'_n \rangle$ if $\forall k \in t_m, j \in t'_1, k < j$
- $s \diamond_s p = \langle t_1, ..., t_m, t'_1, ..., t'_n \rangle$
- $s' = p \diamond s$: p - prefix and s - suffix of s'
  Example: $\langle (e)(\alpha) \rangle$ is prefix of $\langle (e)(abf)(bde) \rangle$ and $\langle (bf)(bde) \rangle$ is its suffix

Preliminary Concepts

s-projected database (*physical projection* and *pseudo projection*)

- $D_s = \{p | s' \in D, s' = r \diamond p$ s.t. r is minimum prefix containing s $(s \sqsubseteq r$ and $\nexists r', s \sqsubseteq r' \sqsubset r)\}$
  p can be empty

Seq ID.	Sequence
0	$\langle (af)(d)(e)(a) \rangle$
1	$\langle (e)(a)(b) \rangle$
2	$\langle (e)(abf)(bde) \rangle$

Example

- $D_{\langle (\alpha f) \rangle} = \{\langle (d)(e)(\alpha) \rangle, \langle (bde) \rangle\}$
- $D_{\langle (e)(\alpha) \rangle} = \{\$, \langle (b) \rangle, \langle (\_bf)(bde) \rangle\}$

Lexicographic Sequence Tree

Set Lexicographic Order

- Let $t = \{i_1, i_2, ..., i_k\}, t' = \{j_1, j_2, ..., j_l\}$, where $i_1 \leq ... \leq i_k$ and $j_1 \leq ... \leq j_l$
- $t < t'$ iff *either* of the following is true:
    1. $0 \leq h \leq min\{k, l\}$, we have $i_r = j_r$ for $r < h$, and $i_h < j_h$
    2. $k < l$, and $i_1 = j_1, i_2 = j_2, ..., i_k = j_k$

Example: $(a, f) < (b, f), (a, b) < (a, b, c)$ and $(a, b, c) < (b, c)$

Lexicographic Sequence Tree

Sequence Lexicographic Order

i. if $s' = s \diamond p$, then $s < s'$
ii. if $s = \alpha \diamond_i p$ and $s' = \alpha \diamond_s p'$, no matter what is order relation between p and p' is, $s < s'$
iii. if $s = \alpha \diamond_i p$ and $s' = \alpha \diamond_i p'$, $p < p'$ indicated $s < s'$
iv. $s = \alpha \diamond_s p$ and $s' = \alpha \diamond_s p'$, $p < p'$ indicates $s < s'$

Example: $\langle (a, b) \rangle < \langle (a, b)(a) \rangle$; $\langle (a, b) \rangle < \langle (a)(a) \rangle$