Motivation and Contribution

Motivation

- Sequence data are shared to enable mining (e.g., in marketing and healthcare).
- Sensitive patterns, which lead to intrusive inferences about individuals or leak confidential information about organizations, may be exposed.
- Existing methods use symbol deletion, which reduces symbol support.
- High utility loss in sequence mining and tasks based on itemset properties.

Contribution: The first, permutation-based approach to hiding sensitive sequential patterns.

- Study of the problem of avoiding side-effects
- PDPG, for generating permutations that avoid side-effects and have minimum distortion
- PH, for sanitizing transactions

Avoiding side-effects

Conditions for identifying candidates for lost and ghost, based on

- the shared symbols between a sensitive and a nonsensitive pattern.

<table>
<thead>
<tr>
<th>Swapping symbols in</th>
<th>s′</th>
<th>s′′</th>
<th>s′′′</th>
<th>d</th>
<th>π(s)</th>
<th>after π(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s \ s′, s′′</td>
<td>[a, b, c]</td>
<td>[a, c]</td>
<td>[a, c, b]</td>
<td>[a, c, b]</td>
<td>[a, c, b]</td>
<td></td>
</tr>
</tbody>
</table>

- the support of the nonsensitive pattern s′:
  - s′ becomes lost if it satisfies symbol-based conditions and \( \supp(s′) < \minSup + \Delta_{\supp}(s|\pi(s)) \)
  - s′ becomes ghost if it satisfies symbol-based conditions and \( \supp(s′) \geq \minSup - \Delta_{\supp}(s|\pi(s)) \)

NP-completeness

- Finding a permutation that prevents lost and avoids ghost patterns (pattern-constrained) is NP-complete (proof by reduction to SAT)

Identifying a satisfiable prefix of a permutation w.r.t. a candidate pattern

- Conditions based on distortion (Cayley distance)
- Swapping the remaining symbols of s must lead to preserving a candidate \( d^- \) for lost
- At least one swap of symbols of s must lead to preserving a \( d^+ \) for ghost

| t | s | d^- | d^+ | π(s|m) | t^* |
|---|---|-----|-----|------|-----|
| (0, 1, 2, 3) | [1, 2, 8] | [1, 2, 8] | [1, 2, 8] | [1, 2, 1, 3, 4] |

Permutation generation

PDPG

Aims at generating a pattern-constrained \( \pi(s|m) \) with min. Cayley distance (max. # cycles of elements). PDPG

- Random search over all possible solutions
- Extends \( \pi(1|m = 1) = \{0\} \) to \( \pi(s|m = |s|) \), iteratively
- Randomly selects the (satisfiable) permutations to extend, from both types of permutations w.r.t. their cycle structure
- The selection is made from the most likely type to produce a satisfiable \( \pi(s|m + 1) \)

Example

| \( \pi(s|m) \) | \( d^- \) | \( d^+ \) | \( t^* \) |
|----------|------|------|-----|
| [a, b, c] | [1, 2, 3, 4] | [2, 3, 4] | [0, 1, 2, 3, 4] |

RR-PDPG

Aims at generating a permutation that satisfies most candidates, when all candidates cannot be satisfied

- Halves the candidates and applies PDPG, until the generated permutation satisfies them
- Short candidates for ghost are the easiest to satisfy
- Long candidates for lost are the hardest

Experimental evaluation

- Side-effects (MSNBC)
- Lost freq. itemsets (BMS)
- KL-divergence (MSNBC)

Backgr. knowledge attacks

Attackers can discover a sensitive pattern s among nonsensitive patterns with the same support and/or do not consider certain permutations of s.

To prevent such attacks:

- Use all available permutations
- Use at least c different permutations, all having support at least λ