1. MOTIVATION & CONTRIBUTIONS

Knowledge hiding aims to publish data to support data mining, such as web log mining, while concealing sensitive knowledge patterns (e.g., business secrets) [1].

- Simply attempting to preserve the support of all nonsensitive sequences [2] may hamper data utility, because many nonsensitive frequent sequences are lost (side-effects).
- We re-define the problem of sequential pattern hiding to capture both side-effects and the distortion. This allows more useful data for sequential mining to be produced.
- We design two novel algorithms: one to minimize distortion, and another that focuses on preventing side-effects. Both of the algorithms outperform the state-of-the-art [2] in terms of data utility and are an order of magnitude more efficient.

2. SEQUENCE HIDING PROBLEM & STRATEGIES

**Sequence Hiding Problem** Given a sequence database \( D \), a minimum support threshold \( \nu \), and a set of sensitive sequences \( S \) (selected by data owners among the frequent ones), construct a new, sanitized database \( D^* \), from \( D \) such that:

(1) \( supp(D') > \nu \) for each sensitive sequence \( s \in S \),

(2) \( D^* \supseteq D \), and

(3) \( distance(D, D^*) = \min \) where \( T \in S \) supports in \( D \).

The first goal of this NP-hard problem has to be accomplished, but an algorithm needs to prioritize between goals (2) and (3).

**Matching graph for a sensitive sequence** Given a transaction \( T \in D \) and a sensitive sequence \( s \in S \), a matching graph \( G_{S,T} \) is a multi-partite graph such that:

- each position \( i \) of \( s \) corresponds to a different part (layer) \( i \) in the graph that carries the symbol of \( s \) in the position as a label - the layers are positioned consecutively in \( G_{S,T} \) in the same order as the symbols of \( s \),

- each position \( i \) of \( T \) is a node in each layer of the graph that has the same label as the corresponding symbol in \( s \).

In \( G_{S,T} \), the node carries a label equal to the position of the corresponding symbol in \( T \). (c) an edge between two nodes \( u,v \) exists in the graph only if (i) \( u \) and \( v \) are in adjacent layers, with \( u \) preceding \( v \) in \( G_{s,T} \), and (ii) \( u \) and \( v \) belong to a path, which is a sequence of nodes \( u_1,u_2,\ldots,u_k \) for which each node \( u_i \) is in the \( i \)-th layer of \( G_{S,T} \) and \( label(u_i) = \bar{label}(u_{i+1}) \) - this is called a complete path, and (d) isolated nodes are removed from \( G_{S,T} \).

**Selecting transactions to sanitize** Transactions that do not support sensitive sequences are detected by \( \psi \), which also orders the remaining transactions in descending order of cost and selects the fewest transactions that suffice to hide the sensitive sequences for sanitization.

\[\psi(D) = \sum \text{cost}\cdot|D'|/|D| \]

**Distance Based Sequence Hiding (DBSH)**

**Objective** Ensure that no sensitive sequence can be mined at support threshold \( \nu \), while minimizing \( \text{distance}(D, D^*) \).

**Sketch** For each transaction \( T \) that needs sanitization, and each set of sensitive sequences, it deletes an approximately minimal set of positions to sanitize \( T \).

**Cost** \( \text{cost}(D) = |T'|/|T| \) time in the worst case, where \( r \) is the largest sensitive sequence.

3. EXPERIMENTAL RESULTS

**Evaluation in terms of Side-effects (%)**

**Evaluation in terms of Distance (%)**

**Evaluation in terms of Runtime (sec)**

**Impact of parameter \( r \) (TRUCKS)**

**REFERENCES**


Revisiting Sequential Pattern Hiding to Enhance Utility

Aris Gkoulalas-Divanis

Grigoris Loukides

Vanderbilt University, gloukides@acm.org

IBM Research Zurich, agd@zurich.ibm.com