Top-k Frequent Itemsets via Differentially Private FP-trees

Jaewoo Lee and Chris Clifton
Department of Computer Science, Purdue University

Frequent Itemset Mining
- Find all itemsets whose support is above threshold \( \tau \)
- Frequent itemsets are aggregates over many individuals
- Releasing the exact result may reveal sensitive personal information

Differential Privacy
For all datasets \( D_1 \) and \( D_2 \) differing at most one element, \[ \Pr[M_f(D_1) = R] \leq e^\epsilon \]
- output of an algorithm is insensitive to the change of a single record
- each database access costs a privacy budget

Challenge
- Given a set of items \( I \), the size of search space is \( O(2^|I|) \)
- How to allocate privacy budget
- Smaller privacy budget implies less accurate answers
- The accuracy of algorithm is dependent on the number of queries

Our Approach
- (Phase 1) Frequent Itemset discovery
- (Phase 2) Noisy support derivation

Sparse Vector Technique
- A technique to avoid spending too much privacy budget on uninteresting queries
- Introduce a new randomness by perturbing the threshold

Algorithm 1
- \( \tilde{\tau} = \tau + \text{Lap}\left(\frac{\epsilon}{2}\right) \)
- \( \tilde{X} = \sigma(X) + \text{Lap}\left(\frac{\epsilon}{2}\right) \)
- If \( \tilde{X} \geq \tilde{\tau} \) (X is frequent) then, output 1
- Otherwise (X is infrequent), output 0
- The output of algorithm is a binary vector \( v = (v_1, v_2, \ldots, v_l) \)

Algorithm 2
- Each node monitors the count of a prefix
- Node count is initialized with a noise
- To get the correct count, child’s count needs to be added to its parent’s count
- (optional) post-processing can increase the accuracy

Performance Evaluation
- F-score = \( \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \)
- Relative error = \( \text{median}_X \left( \frac{|\sigma(X) - \tilde{\sigma}(X)|}{\sigma(X)} \right) \)
- the proposed method outperforms other two methods throughout all test datasets

| dataset         | \( |D| \) | \( |I| \) | max[|t|] | avg[|t|] |
|-----------------|--------|--------|---------|---------|
| mushroom (MUS)  | 8,124  | 119    | 23      | 23      |
| pumsb star (PUMSB) | 40,046 | 2,088  | 63      | 50.5    |
| retail (RET)    | 88,162 | 16,470 | 76      | 10.3    |

(a) mushroom (b) pumsb star (c) retail