Knowledge Sanitization on the Web

Vasileios Kagklis\textsuperscript{1} Vassilios S. Verykios\textsuperscript{2}
Giannis Tzimas\textsuperscript{3} Athanasios K. Tsakalidis\textsuperscript{1}

\textsuperscript{1}Computer Engineering & Informatics Department, University of Patras
\textsuperscript{2}School of Science & Technology, Hellenic Open University
\textsuperscript{3}Computer & Informatics Engineering Department, Technological Educational Institute of Western Greece

Thessaloniki, GREECE
June 2014
Overview

1. Introduction
2. Background and Problem Formulation
3. A Taxonomy of FIH techniques
4. An Overview of LP-Based Techniques
5. Experimental Results
6. A Knowledge Sanitization Toolbox
7. Conclusions
8. References
Need for Privacy

- The widespread use of the Internet caused the rapid growth of data on the Web.

- As data on the Web grew larger in numbers, so did the perils due to the applications of data mining.

- Thus, the need for privacy preserving techniques related to data mining on the Web, became more essential.
A Failing-to-Preserve-Privacy Example

- AOL data release [4]
- Data in the form of 20,000,000 search keywords, for 650,000 users, for a period of 3 months.
- Intentional release for research purposes.
- Appropriate editing did not take place.
- The users were only identified by a unique numeric ID.
- Some clues from the search queries were enough for successfully tracking the identities of several users by their searches.
Privacy Preserving Data Mining (PPDM) [1, 2]

- Research area that investigates techniques to preserve the privacy of individual data and induced patterns.

- Looks into the interplay between data sharing and privacy violation.

- Data mining can violate privacy.

- Allow data mining while prohibiting leakage of sensitive information.
Taxonomy in PPDM

PPDM consists of several pillars:

- Input/Data/Individual Privacy
- Adversarial Privacy
- **Output/Knowledge/Collective Privacy**

We are going to focus on Output Privacy, also known as Knowledge Sanitization.
Knowledge Sanitization

Knowledge Sanitization [3] aims at concealing sensitive patterns included in the data.

It consists of a wide variety of different approaches.

**Frequent pattern** and association rule sanitization.

Sequence sanitization.

Classification rule sanitization.

Data stream sanitization.
Applications (1/2)

- Frequent patterns are widely used on the web.
- Product-selling (and other) websites use frequent basket analysis to:
  - discover similarities in purchasing habits among customers
  - make recommendations
- Some websites may sell those anonymously collected datasets to advertising companies.
- Web link and click stream analysis aims at:
  - the improvement of the structure of a website
  - improving of the navigation experience
  - the predictive web caching
Applications (2/2)

- Association rules derive from frequent itemsets.

- A powerful tool for discovering relationships hidden in large datasets.

- Association rule mining can be applied on web log files to profile the visitors’ behavior.

- Certain sanitization techniques must be applied in the cases mentioned.
Preliminaries (1/3)

- \( I = \{i_1, i_2, \ldots, i_n\} \): set of items.

- A subset \( X \subseteq I \) is an itemset.

- \( D = \{T_1, T_2, \ldots, T_m\} \): transactional database.

- Database \( D \) can be in binary format (\( |D| \times |I| \) matrix)
  - \( T_{kj} = 1 \), if \( k \)-th transaction contains \( j \)-th item.
  - \( T_{kj} = 0 \), otherwise.
Given an itemset $X$:
- $\sigma(X)$: number of supporting transactions, and
- $sup(X)$: fraction of supporting transactions

Itemset $X$ is **large** or **frequent** iff:
- $sup(X) \geq msup$, where $msup = \sigma_{min}/|D|$  
- or equiv. $\sigma(X) \geq \sigma_{min}$.

Otherwise, $X$ is **infrequent**.
Preliminaries (3/3)

- $F_\sigma$: set of all frequent itemsets in $D$, for $\sigma_{\text{min}} = \sigma$.

- We define the following borders of $F_\sigma$:
  - **Positive Border**: contains all maximally frequent itemsets in $D$.
  - **Negative Border**: contains all minimally infrequent itemsets in $D$.

- $S$: set of sensitive itemsets that the owner wants to conceal, i.e., force them to become infrequent in $D$. 
Frequent Itemset Extraction

For $\sigma_{\text{min}} = 3$, the set of frequent itemsets $F_\sigma$ is:

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
<th>$\sigma(a) = 7$</th>
<th>$\sigma(ab) = 4$</th>
<th>$\sigma(abc) = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>abcde</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>acd</td>
<td>$\sigma(b) = 6$</td>
<td>$\sigma(ac) = 5$</td>
<td>$\sigma(abd) = 3$</td>
</tr>
<tr>
<td>3</td>
<td>abdfg</td>
<td>$\sigma(c) = 7$</td>
<td>$\sigma(ad) = 6$</td>
<td>$\sigma(acd) = 4$</td>
</tr>
<tr>
<td>4</td>
<td>bcde</td>
<td>$\sigma(d) = 8$</td>
<td>$\sigma(bc) = 4$</td>
<td>$\sigma(bcd) = 3$</td>
</tr>
<tr>
<td>5</td>
<td>abd</td>
<td>$\sigma(e) = 3$</td>
<td>$\sigma(be) = 2$</td>
<td>$\sigma(cde) = 3$</td>
</tr>
<tr>
<td>6</td>
<td>bcdfh</td>
<td>$\sigma(f) = 2$</td>
<td>$\sigma(cd) = 6$</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>abcg</td>
<td>$\sigma(g) = 2$</td>
<td>$\sigma(de) = 3$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>acde</td>
<td>$\sigma(h) = 2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>acdh</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Initially:
- the **Positive Border**, $B^+(F_\sigma)$, is marked with yellow color, while
- the **Negative Border**, $B^-(F_\sigma)$, is marked with orange color
How does the hiding process affect the set of frequent itemsets?

Some of the frequent itemsets, i.e., the supersets of $S$ will be concealed as well.

This is due to the anti-monotonicity property of support: $X \subset Y \implies \sigma(X) \geq \sigma(Y)$.

Let $SS = \{X \in F_\sigma \mid \forall Y: Y \subseteq X \implies Y \in S\}$ be the set of non-sensitive itemsets and their supersets in $F_\sigma$.

The tentative set of frequent itemsets is defined as $\tilde{F}_\sigma = F_\sigma - SS$. 
Let $S = \{ab, bc, cd\}$. Then $\tilde{F}_\sigma = \{a, b, c, d, e, ac, ad, bd, ce, de\}$ and:

- the **Revised Positive Border**, $B^+ (\tilde{F}_\sigma)$, is marked with *yellow* color,
- the **sensitive itemsets** are marked with *blue* color and
- the **Revised Negative Border**, $B^- (\tilde{F}_\sigma)$, is marked with *orange* color, which also includes the sensitive itemsets.
Border Revision (4/4)

Why border revision?

- **Naive approach**: conceal without taking into account the non-sensitive frequent itemsets.
- **Better approach**: try to protect all non-sensitive frequent itemsets to avoid side effects.
- **Border based approach**: take into account only $B^+(\tilde{F}_\sigma)$.
  - Anti-monotonicity property of support.
  - $B^+(\tilde{F}_\sigma)$: maximal itemsets of $\tilde{F}_\sigma$.
- The last two approaches are equivalent, but the latter is computationally lighter.
Hiding Methodologies (1/2)

- **Heuristic distortion approaches**: rely on turning 1’s to 0’s and 0’s to 1’s in order to achieve hiding.

- **Heuristic blocking approaches**: make use of an unknown symbol to signify the absence of a specific value.

- **Probabilistic distortion approaches**: apply a probabilistic model in order to distort the data.
Hiding Methodologies (2/2)

- **Database reconstruction approaches**: the non-sensitive knowledge is transformed to a database that is built from scratch.

- **Inverse frequent itemset mining**: has as its goal to create a database that corresponds to a certain set of useful and interesting patterns.

- **Linear programming-based hiding techniques**: formulate a hiding problem as a linear program, the solution of which helps to accomplish the concealing.
Linear Programming-Based Techniques

- Transform the problem into a linear program.
- The various types of constraints play a different role, depending on the formulation.
- The solution indicates the transactions to be sanitized or the exact items to be removed from each transaction.
The LP Hiding Techniques

- Max-Accuracy
- Coefficient-Based Max-Accuracy
- Inline
- Hybrid
The Max-Accuracy Algorithm [5]
Basic Features

- Each transaction is modeled by a corresponding binary variable.
- For each sensitive itemset in $S$, a constraint is built.
- If a sensitive itemset is contained in a transaction, then the corresponding constraint contains the corresponding binary variable.
- Size of the linear program: $|D|$ variables and $|S|$ constraints.
- The solution will determine which transactions need to be sanitized.
- Sanitization process on specified transactions follows.
Define parameters $a_{iy}$ to be 1 if transaction $T_i \in D$ supports itemset $y \in S$ (sensitive itemsets) and 0 otherwise. Variables $x_i$ will be set to 1 if transaction $T_i$ needs to be sanitized and 0 otherwise, depending on the solution of the linear program.

\[
\begin{align*}
\text{minimize} & \quad \sum_{\forall i: \ T_i \in D} x_i \\
\text{subject to} & \quad \sum_{\forall i: \ T_i \in D} a_{iy} x_i \geq (\sigma_y - \sigma_{y,\min} + 1), \quad \forall y \in S \\
& \quad x_i \in \{0, 1\} \quad \forall i : T_i \in D.
\end{align*}
\]
Objective Function: the minimum number of transactions should be sanitized.

Constraints: a sensitive itemset \( y \) needs to be hidden from at least \( (\sigma_y - \sigma_{y_{\text{min}}} + 1) \) transactions, in order to become infrequent.

Obviously, the side effects that will be introduced are not taken into account.
The Data Hiding Algorithm

for transactions $T_i \in D$ such that $T_i$ is to be sanitized do

identify set of sensitive itemsets $S_i$ supported by transaction $T_i$

while $S_i \neq \emptyset$ do

calculate $f_j = |\{k \in S_i | j \in k\}|, \forall$ item $j \in S_i$

remove item $j^* = \arg\max_j \{f_j\}$

update $S_i = S_i - \{k \in S_i | j^* \in k\}$

end while

end for
The Data Hiding Algorithm Explained

- Variables set to 1 in the solution of the linear program indicate-mark transactions for sanitization.
- The sensitive itemsets $S_i \subseteq S$ supported by a marked transaction are identified.
- Item $j^*$ that appears in most itemsets in $S_i$ is eliminated.
- Itemsets in $S_i$ also containing $j^*$ are removed from $S_i$.
- The process is repeated until $S_i$ is left empty.
- If only one sensitive itemset is supported, then an item is removed randomly.
An Example (1/3)

- Let the transaction database $D$, the set of sensitive itemsets $S = \{ab, bc, cd\}$ and $\sigma_{min} = 3$.

- $ab \rightarrow$ Tid set $\{1, 3, 5, 7\}$.

- $bc \rightarrow$ Tid set $\{1, 4, 6, 7\}$.

- $cd \rightarrow$ Tid set $\{1, 2, 4, 6, 8, 9\}$.

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>abcde</td>
</tr>
<tr>
<td>2</td>
<td>acd</td>
</tr>
<tr>
<td>3</td>
<td>abdfg</td>
</tr>
<tr>
<td>4</td>
<td>bcde</td>
</tr>
<tr>
<td>5</td>
<td>abd</td>
</tr>
<tr>
<td>6</td>
<td>bcdfh</td>
</tr>
<tr>
<td>7</td>
<td>abcg</td>
</tr>
<tr>
<td>8</td>
<td>acde</td>
</tr>
<tr>
<td>9</td>
<td>acdh</td>
</tr>
</tbody>
</table>
An Example (2/3)

Constraint Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$T_4$</th>
<th>$T_5$</th>
<th>$T_6$</th>
<th>$T_7$</th>
<th>$T_8$</th>
<th>$T_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ab$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$bc$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$cd$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\text{minimize } x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9
\]

\[
\text{subject to } \begin{cases} 
ab : x_1 + x_3 + x_5 + x_7 \geq 2 \\
bc : x_1 + x_4 + x_6 + x_7 \geq 2 \\
\text{cd : } x_1 + x_2 + x_4 + x_6 + x_8 + x_9 \geq 4 
\end{cases}
\]
The optimal solution is $x_1 = x_2 = x_7 = x_8 = x_9 = 1$, while $x_3 = x_4 = x_5 = x_6 = 0$.

Summary of the sanitization process:

<table>
<thead>
<tr>
<th>Tid</th>
<th>Transaction</th>
<th>S.I. supported</th>
<th>Victim Items</th>
<th>Sanitized</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>abcde</td>
<td>cd, bc, ab</td>
<td>c, a</td>
<td>bde</td>
</tr>
<tr>
<td>2</td>
<td>acd</td>
<td>cd, ac</td>
<td>c</td>
<td>ad</td>
</tr>
<tr>
<td>7</td>
<td>abcg</td>
<td>ab</td>
<td>b</td>
<td>acg</td>
</tr>
<tr>
<td>8</td>
<td>acde</td>
<td>cd</td>
<td>c</td>
<td>ade</td>
</tr>
<tr>
<td>9</td>
<td>acdh</td>
<td>cd</td>
<td>c</td>
<td>adh</td>
</tr>
</tbody>
</table>
The Coefficient-Based Max-Accuracy Algorithm [6]
Basic Features

- An improved version of the Max-Accuracy algorithm.
- The algorithm introduces proper coefficients for each variable, that corresponds to a transaction.
- As a result, the transactions that are going to be sanitized are selected more accurately.
- Size of the linear program: $|D|$ variables and $|S|$ constraints.
- The solution will determine which transactions need to be sanitized.
- The very same sanitization process as in Max-Accuracy is used.
Calculating the Coefficients

The coefficients $c_m, \forall m \in \{1, \ldots, |D|\}$, are calculated as follows:

- The coefficient $c_m$ is initialized to zero.
- Let $S_i$ be the set of all sensitive itemsets supported by $T_j$. The item $i_k$ that is supported by most of the itemsets in $S_j$ is selected.
- The number of non-sensitive frequent itemsets that are both supported by $T_j$ and contain $i_k$ is added to $c_m$.
- A sensitive itemset $y$ is removed from $S_j$, if after removing item $i_k$ itemset $y$ stops being supported by the current transaction $T_j$.
- The process is done repeatedly, until $S_j$ is left empty.
The Formulation

Simply put, the coefficient of a transaction is the number of affected non-sensitive frequent itemsets given the transaction is sanitized. The formulation is almost the same as in the Max-Accuracy. Only the objective function changes:

\[
\begin{align*}
\text{minimize} & \quad \sum_{\forall i: \ T_i \in D} c_i x_i \\
\text{subject to} & \quad \sum_{\forall i: \ T_i \in D} a_{iy} x_i \geq (\sigma_y - \sigma_{\text{min}} + 1), \ \forall y \in S \\
& \quad x_i \in \{0, 1\} \quad \forall i: \ T_i \in D.
\end{align*}
\]
Consider the same transaction database $D$, sensitive itemsets $S = \{ab, bc, cd\}$ and $\sigma_{min} = 3$ as in the previous example. The coefficients must be first calculated.

<table>
<thead>
<tr>
<th>Tid</th>
<th>Trans.</th>
<th>Victim Items</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>abcde</td>
<td>c, a</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>acd</td>
<td>c</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>abdfg</td>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>bcde</td>
<td>c</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>abd</td>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>bcd fh</td>
<td>c</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>abc g</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>acde</td>
<td>c</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>acdh</td>
<td>c</td>
<td>3</td>
</tr>
</tbody>
</table>
An Example (2/3)

\[
\begin{align*}
\text{minimize} & \quad 11x_1 + 3x_2 + 3x_3 + 4x_4 + 3x_5 \\
& \quad + 2x_6 + 1x_7 + 5x_8 + 3x_9 \\
\text{subject to} & \quad \begin{cases}
ab : x_1 + x_3 + x_5 + x_7 \geq 2 \\
bc : x_1 + x_4 + x_6 + x_7 \geq 2 \\
cd : x_1 + x_2 + x_4 + x_6 + x_8 + x_9 \geq 4
\end{cases}
\end{align*}
\]
An Example (3/3)

- The optimal solution is
  \[ x_2 = x_4 = x_5 = x_6 = x_7 = x_9 = 1, \text{ while} \]
  \[ x_1 = x_3 = x_8 = 0. \]

- Summary of the sanitization process:

<table>
<thead>
<tr>
<th>Tid</th>
<th>Trans.</th>
<th>S.I. supported</th>
<th>Victim Items</th>
<th>Sanitized</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>abd</td>
<td>ab</td>
<td>a : 1, b : 1</td>
<td>ad</td>
</tr>
<tr>
<td>4</td>
<td>bcde</td>
<td>bc, cd</td>
<td>b : 1, c : 2, d : 1</td>
<td>bde</td>
</tr>
<tr>
<td>5</td>
<td>abd</td>
<td>ab</td>
<td>a : 1, b : 1</td>
<td>bd</td>
</tr>
<tr>
<td>6</td>
<td>bcdfh</td>
<td>bc, cd</td>
<td>b : 1, c : 2, d : 1</td>
<td>bdfh</td>
</tr>
<tr>
<td>7</td>
<td>abcg</td>
<td>ab, bc</td>
<td>a : 1, b : 2, c : 1</td>
<td>acg</td>
</tr>
<tr>
<td>9</td>
<td>abdh</td>
<td>ab</td>
<td>a : 1, b : 1</td>
<td>adh</td>
</tr>
</tbody>
</table>
The Inline Algorithm [7]
Basic Features (1/2)

- Database must first be transformed into a $|D| \times |I|$ binary array with elements:

$$b_{kj} = \begin{cases} 1, & \text{if item } i_j \in T_k \\ 0, & \text{otherwise} \end{cases}$$

- $b_{kj}$ values participating in the sensitive itemsets are substituted in all transactions with $u_{kj}$ variables, which participate in the linear program’s formulation.
Basic Features (2/2)

- For the two previous algorithms, the solution determines the transactions to be sanitized. Then sanitization follows.

- For the Inline algorithm the solution of the linear program specifies which items must be removed and from which transactions.

- This is a more exact database distortion approach [8].
The Formulation

\[
\begin{align*}
\text{maximize} & \quad \sum_{u_{kj} \in U} u_{kj} \\
\text{subject to} & \quad \left\{ \begin{array}{l}
\sum_{T_k \in D\{X\}} \left( \prod_{i_j \in X} u_{kj} \right) < \sigma_{\text{min}}, \forall X \in S \\
\sum_{T_k \in D\{R\}} \left( \prod_{i_j \in R} u_{kj} \right) \geq \sigma_{\text{min}}, \forall R \in V
\end{array} \right.
\end{align*}
\]

where \(V = \{X \in B^+(\tilde{F}) \mid X \cap I^S \neq \emptyset\}\) and \(I^S\) is the set of items contained by itemsets in \(S\).
The Formulation Explained

- **Objective Function**: maximize the number of variables with value equal to 1. In other words, remove the fewest items.

- A sensitive itemset will get concealed if:
  \[ \sum_{T_k \in D\{X\}} \left( \prod_{i,j \in X} u_{kj} \right) < \sigma_{\text{min}}, \forall X \in S. \]

- Non-sensitive frequent itemsets will remain frequent if:
  \[ \sum_{T_k \in D\{R\}} \left( \prod_{i,j \in R} u_{kj} \right) \geq \sigma_{\text{min}}, \forall R \in V, \]
  where \( V = \{ X \in B^+(\tilde{F}) | X \cap I^S \neq \emptyset \} \).
Constraint Degree Reduction (CDR)

Linear programs cannot contain products. Products occurring in the inequalities of the formulation must be "linearized".

\[
\sum_{T_k \in D\{F\}} \psi_k \leq \sigma_{\min}, \psi_k = \prod_{i \in F} u_{kj} = u_{kF_1} \times \ldots \times u_{kF_{|F|}}
\]

with

\[
\forall k \quad \begin{cases} 
\psi_k \leq u_{kF_1} \\
\psi_k \leq u_{kF_2} \\
\vdots \\
\psi_k \leq u_{kF_{|F|}} \\
\psi_k \geq u_{kF_1} + u_{kF_2} + \ldots + u_{kF_{|F|}} - |I| + 1, \text{ where } |I| = \#\text{vars in product}
\end{cases}
\]

and

\[
\sum_k \psi_k \leq \sigma_{\min}
\]

where \( \psi_k \in \{0, 1\} \).
Dealing with Infeasibilities

- The formulation of the Inline algorithm might give an infeasible solution.
- The problem is relaxed until it becomes solvable.
- Only inequalities from the set $V (B^+ (\tilde{F}_\sigma))$ are removed.
- A constraint involving maximal size and minimum support itemsets in $V$ is removed each time.
- The formulation with only the constrains in $S$ has always a feasible solution.
An Example (1/4)

- Let the transaction database $D$, the set of sensitive itemsets $S = \{ab\}$ and $\sigma_{min} = 2$.
- $F_{\sigma} = \{a, b, c, d, ab, ac, ad, cd, acd\}$.
- $S = \{ab\}$, and $SS = \{ab\}$.
- $\tilde{F}_{\sigma} = F_{\sigma} - SS = \{a, b, c, d, ac, ad, cd, acd\}$.
- $B^+(\tilde{F}_{\sigma}) = \{b, acd\}$.

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$ac$</td>
</tr>
<tr>
<td>2</td>
<td>$acd$</td>
</tr>
<tr>
<td>3</td>
<td>$cd$</td>
</tr>
<tr>
<td>4</td>
<td>$b$</td>
</tr>
<tr>
<td>5</td>
<td>$abcd$</td>
</tr>
<tr>
<td>6</td>
<td>$d$</td>
</tr>
<tr>
<td>7</td>
<td>$c$</td>
</tr>
<tr>
<td>8</td>
<td>$ab$</td>
</tr>
</tbody>
</table>
An Example (2/4)

The database is converted into a binary array and the 1 values of sensitive itemsets (contained in transactions) are replaced with variables:

<table>
<thead>
<tr>
<th>Tid</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>u51</td>
<td>u52</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>u81</td>
<td>u82</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Hiding itemset $S = \{ab\}$

- Itemsets in $V$ must remain frequent:
  - $b: \ 1 + u_{52} + u_{82} \geq \sigma_{min}$
  - $acd: \ 1 + u_{51} \geq \sigma_{min}$

- Itemsets in $S$ must become infrequent:
  - $ab: \ u_{51}u_{52} + u_{81}u_{82} < \sigma_{min}$

Application of CDR for $\{ab\}$:

\[
\begin{align*}
\psi_1 & \leq u_{51} & \psi_2 & \leq u_{81} \\
\psi_1 & \leq u_{52} & \psi_2 & \leq u_{82} \\
\psi_1 & \geq u_{51} + u_{52} - 1 & \psi_2 & \geq u_{81} + u_{82} - 1 \\
\psi_1 + \psi_2 & < \sigma_{min}
\end{align*}
\]
The Hybrid Algorithm [7]
The solution of the previous algorithms determines from which transactions and/or which specific items should be extracted.

The Hybrid algorithm creates an extension of the original database with synthetically generated transactions.

The goal is to fix the contents of the extension so that to control the support of sensitive and non-sensitive itemsets.
Basic Features (2/2)

- Extension of database $D_X$: must contain the minimum sufficient number of transactions.

- Minimum size: $Q = \lceil (\sigma(X_M)/msup) - |D| \rceil + 1$, where $X_M \in S$ such that $\sigma(X_M) \geq \sigma(X), \forall X \in S - X_M$.

- Theoretically, this size seems to be sufficient. Practically, this is not always the case. $\implies$ Use of safety margin $SM$, i.e., $SM$ more transactions in $D_X$.

- The extension $D_X$ is a $|Q + SM| \times |I|$ array that initially contains only variables. The solution of the linear program gives a value to each variable and the transactions are formed.
The Formulation

\[
\begin{align*}
\text{minimize} & \quad \sum_{q\in[1,Q+SM],m\in[1,|D|]} u_qm \\
\text{subject to} & \quad \begin{cases} 
Q+SM \sum_{q=1}^{Q+SM} (\prod_{i_m\in X} u_qm) < \text{thr}, & \forall X \in B^- (\tilde{F}_\sigma) \\
Q+SM \sum_{q=1}^{Q+SM} (\prod_{i_m\in X} u_qm) \geq \text{thr}, & \forall X \in B^+ (\tilde{F}_\sigma) \\
\forall T_q \in D_X : \sum_{i_m \in I} u_qm \geq 1 
\end{cases} \\
\text{where} \quad \text{thr} = \text{msup} \ast (|D| + Q + SM) - \sigma(X)
\end{align*}
\]
The Formulation Explained

- Let $D' = D \cup D_X$.
- Objective Function: minimize the number of variables that will be set to 1.
- An itemset will be **frequent** in $D'$ iff:
  \[
  \sum_{q=1}^{Q+SM} (\prod_{i_m \in X} u_{qm}) \geq m_{sup} \times (|D| + Q + SM) - \sigma(X)
  \]
- An itemset will be **infrequent** in $D'$ iff:
  \[
  \sum_{q=1}^{Q+SM} (\prod_{i_m \in X} u_{qm}) < m_{sup} \times (|D| + Q + SM) - \sigma(X)
  \]
- Empty transactions are not allowed:
  \[
  \forall T_q \in D_X : \sum_{i_m \in I} u_{qm} \geq 1
  \]
Linear programs cannot contain products. Products occurring in the inequalities of the formulation must be "linearized".

Replace

$$\sum_{T_k \in D\{F\}} \psi_k \leq \sigma_{\min}, \psi_k = \prod_{i,j \in F} u_{kj} = u_{kF_1} \times \ldots \times u_{kF_{|F|}}$$

with

$$\forall k \begin{cases} 
\psi_k \leq u_{kF_1} \\
\psi_k \leq u_{kF_2} \\
\vdots \\
\psi_k \leq u_{kF_{|F|}} \\
\psi_k \geq u_{kF_1} + u_{kF_2} + \ldots + u_{kF_{|F|}} - |I| + 1 
\end{cases}$$

and

$$\sum_k \psi_k \leq \sigma_{\min}$$

where $$\psi_k \in \{0, 1\}$$.
An Example (1/2)

<table>
<thead>
<tr>
<th>Tid</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>$u_{11}$</td>
<td>$u_{12}$</td>
<td>$u_{13}$</td>
<td>$u_{14}$</td>
<td>$u_{15}$</td>
<td>$u_{16}$</td>
</tr>
<tr>
<td>12</td>
<td>$u_{21}$</td>
<td>$u_{22}$</td>
<td>$u_{23}$</td>
<td>$u_{24}$</td>
<td>$u_{25}$</td>
<td>$u_{26}$</td>
</tr>
<tr>
<td>13</td>
<td>$u_{31}$</td>
<td>$u_{32}$</td>
<td>$u_{32}$</td>
<td>$u_{34}$</td>
<td>$u_{35}$</td>
<td>$u_{36}$</td>
</tr>
<tr>
<td>14</td>
<td>$u_{41}$</td>
<td>$u_{42}$</td>
<td>$u_{43}$</td>
<td>$u_{44}$</td>
<td>$u_{45}$</td>
<td>$u_{46}$</td>
</tr>
</tbody>
</table>

Let transaction database $D$, 
$S = \{e, ae, bc\}$ and $\sigma_{min} = 3$.

$\sigma(e) = 3$, $\sigma(ae) = 4$, 
$\sigma(bc) = 4$.

The extension $D_x$ has size 
$Q = \lfloor (4/0.3) - 10 \rfloor + 1 = \lfloor 3.33 \rfloor + 1 = 4$ and initially contains variables.
Due to the large number, the constraints are omitted.

In the extended database support for the itemsets in $S$ changes.

When $|D| = 10$, then $\text{sup}(e) = \frac{3}{10}$ and thus indeed $\sigma(e) = 3$.

$|D| = 14$, $\text{sup}(e) = \frac{3}{14} \Rightarrow \sigma(e) = \frac{30}{14} < \sigma_{\text{min}}$
Datasets used

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Trans.</th>
<th># Items</th>
<th>Avg. Len.</th>
<th>$\sigma_{min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampled</td>
<td>500</td>
<td>34</td>
<td>11.12</td>
<td>100</td>
</tr>
<tr>
<td>BMS-1</td>
<td>59602</td>
<td>497</td>
<td>2.50</td>
<td>42</td>
</tr>
<tr>
<td>Mushroom</td>
<td>8124</td>
<td>119</td>
<td>23.00</td>
<td>1625</td>
</tr>
</tbody>
</table>

- Real datasets used for evaluation are available in the FIMI repository [9].
- **Sampled**: sampled version of Mushroom dataset.
- **BMS1**: stream data collected from the Blue Martini Software, Inc. [10].
- **Mushroom**: created by Roberto Bayardo (University of California, Irvine) [11].
The evaluation process had 3 phases and for each phase one of datasets was used.

Different hiding scenarios were selected with various number/size of sensitive itemsets to hide.

Experiments were conducted several times with different sets of sensitive itemsets (the same set for all algorithms each time).

Phase 1: Sample dataset, Phase 2: BMS1 dataset, Phase 3: Mushroom dataset

At the end of each phase, the slowest algorithm is eliminated.
Experiments were conducted with a toolbox written in Python.

Linear programming techniques use the CPLEX [12] interface for Python.

More about the toolbox in the next slides.
Experimental Results - Phase 1 (1/2)

- Figure with runtime in seconds for each hiding scenario with the Sample dataset.
  
- Max-Accuracy and Coefficient-Based Max-Accuracy have much lower execution time.
  
- Inline and Hybrid have larger time complexity.
  
- But what about the side effects?
Figure with side effects for each hiding scenario with the Sample dataset.

- Inline and Hybrid introduce almost 0 side effects.
- But time is important. Very important!
- For the next phase the slowest algorithm is eliminated, which is Hybrid.
Figure with runtime in seconds for each hiding scenario with the BMS1 dataset.

- Inline again has much larger time complexity than the other two algorithms.

- Let’s see what happens with the side effects.
Experimental Results - Phase 2 (2/2)

- Figure with side effects for each hiding scenario with the BMS1 dataset
- Inline again has much fewer side effects than the other two algorithms.
- Again, the algorithm with the highest time complexity is eliminated, i.e. the Inline algorithm.
Experimental Results - Phase 3 (1/2)

- Figure with runtime in seconds for each hiding scenario with the Mushroom dataset.
- Max-Accuracy and Coefficient-Based Max-Accuracy have a good scalability.
- What happens with the side effects?
**Figure with side effects for each hiding scenario with the Mushroom dataset.**

**Time complexity is linear, but they introduce quite a few side effects.**
### Qualitative Comparison

A qualitative comparison of the algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Execution Time</th>
<th>Scalability</th>
<th>Side Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max-Accuracy</td>
<td>Very Fast</td>
<td>Very Good</td>
<td>Moderate</td>
</tr>
<tr>
<td>Coeff.-Based Max-Accuracy</td>
<td>Fast</td>
<td>Good</td>
<td>Moderate-Good</td>
</tr>
<tr>
<td>Inline</td>
<td>Slow</td>
<td>Bad</td>
<td>Very Good</td>
</tr>
<tr>
<td>Hybrid</td>
<td>Slow</td>
<td>Very Bad</td>
<td>Very Good</td>
</tr>
</tbody>
</table>
Toolbox Interface (2/3)
Toolbox Interface (3/3)
Conclusions

- Max-Accuracy and Coefficient-Based Max-Accuracy: scalable, while introducing numerous side effects

- Inline and Hybrid: few side effects, but with bad scalability

- An optimal LP-based algorithm remains yet to be found
References I

  Privacy-preserving data mining
  *SIGMOD 29*: 439-450

  Privacy preserving data mining
  *CRYPTO ’00 Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology* 1880: 36-54

- Johnsten, Tom and Raghavan, Vijay V. (2002)
  A Methodology for Hiding Knowledge in Databases
  *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining* 14: 9-17
References II

Michael Barbaro and Tom Zeller
A Face Is Exposed for AOL Searcher No. 4417749
The New York Times

Maximizing accuracy of shared databases when concealing sensitive patterns
Information Systems Research, 16(3):256-270

Coefficient-based exact approach for frequent itemset hiding
eKNOW2014: The Sixth International Conference on Information, Process, and Knowledge Management, 124-130
A. Gkoulalas-Divanis and V. S. Verykios (2009)
Hiding sensitive knowledge without side effects
Knowledge Information Systems, 20(3):263-299

V. S. Verykios (2013)
Association rule hiding methods
WIREs Data Mining Knowledge Discovery, 3(1):28-36

Frequent Itemset Mining Dataset Repository
http://fimi.ua.ac.be/data/
KDD-Cup 2000 Organizers’ Report: Peeling the Onion
SIGKDD Explor. Newsl., 2(2):86-93

Bayardo, Jr., Roberto J. (1998)
Efficiently Mining Long Patterns from Databases
SIGMOD, 27(2):85-93

IBM ILOG CPLEX User’s Manual v12.6