PRIVACY-PRESERVING MINING OF ASSOCIATION RULES FROM OUTSOURCED TRANSACTION DATABASES

F. Giannotti, L. Lakshmanan, A. Monreale, D. Pedreschi and W. Wang

Knowledge Discovery and Delivery Lab
(ISTI-CNR & Univ. Pisa)
www-kdd.isti.cnr.it

SEBD 2012 - Venice

Introduction

- Availability of large transactional database
- Data are an important resource for an organization if
  - Processed
  - Analyzed
  - Transformed in Knowledge by KDD techniques
- Mining the data requires
  - Computational resources
  - In-house expertise for data mining

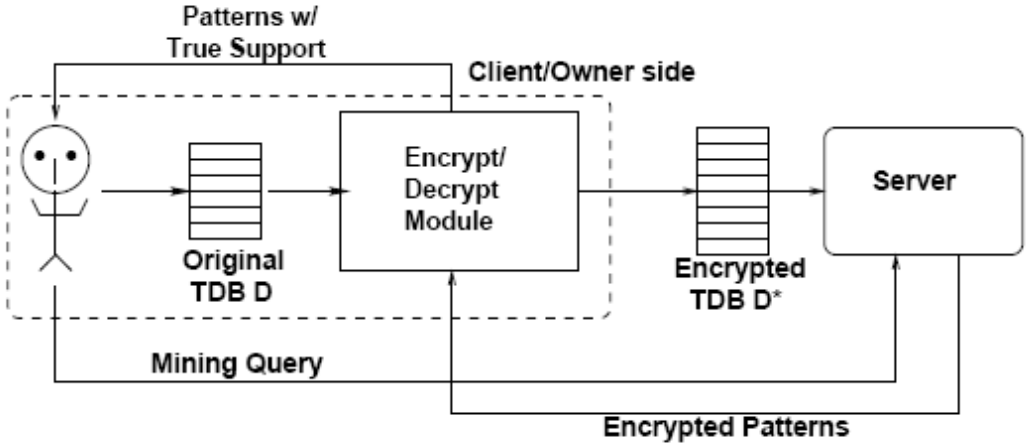
Privacy-Preserving Outsourcing of DM

- Organizations could do not posses
  - **in-house expertise** for doing data mining
  - **computing infrastructure** adequate
- **Solution:** Outsourcing of data mining to a service provider
  - specific human resources
  - technological resources
- The server has access to data of the owner
- Data owner has the property of both
  - **Data** can contain personal information about individuals
  - **Knowledge** extracted from data can provide competitive advantages

A valid solution

- A Privacy Model and so an Attack Model
- The Definition of an Encryption/Decryption Scheme
  - Efficient in time and space
  - Must allow the exact reconstruction of the query results

Framework Architecture



- The client encrypts its data using an encrypt/decrypt (ED) module
  - essentially treated as a "black box" from its perspective
- ED module transforms the input data into an encrypted database
- The server conducts data mining and sends the patterns to the client
- The ED module recovers the true identity of the returned patterns

Privacy Model

- **Adversary Knowledge:** attacker
  - knows the set of plain items and their true supports in D exactly
  - has access to the encrypted database D*

Item	Sup
Bread	5
Milk	3
Water	2
Beer	1
Eggs	1

K-Anonymous TDB
e_4 e_1
e_4
e_3
e_2
e_2 e_4
e_4 e_2
e_4 e_5
e_2 e_1
e_2 e_3
e_5

- **Attack Model**
  - ***Item-based attack:*** guessing the plain item corresponding to the cipher item e with probability *prob(e)*
  - ***Itemset-based attack:*** guessing the plain itemset corresponding to the cipher itemset E with probability *prob(E)*

Goal and Ideal Solution

- **Goal:** minimize the probabilities of crack of
  - an item $prob(e)$
  - an itemset (transaction or pattern) $prob(E)$

- **Ideal Solution:**
  - every cipher item should have as candidates all the items in D
  - every cipher itemset should have as candidates all the itemset with same size in D

- **Problem:** explosion in the computational effort required for mining patterns from D^*

K-Privacy

- **Our Solution:** use the well-known **k-anonymity** notion

Definition 1 (Item k-anonymity). Let D be a transaction database and D^* its encrypted version. We say D^* satisfies the property of *item k-anonymity* provided for every cipher item $e \in \mathscr{E}$, there are at least $k-1$ other distinct cipher items $e_1, ..., e_{k-1} \in \mathscr{E}$ such that $supp_{D^*}(e) = supp_{D^*}(e_i)$, $1 \leq i \leq k-1$. □

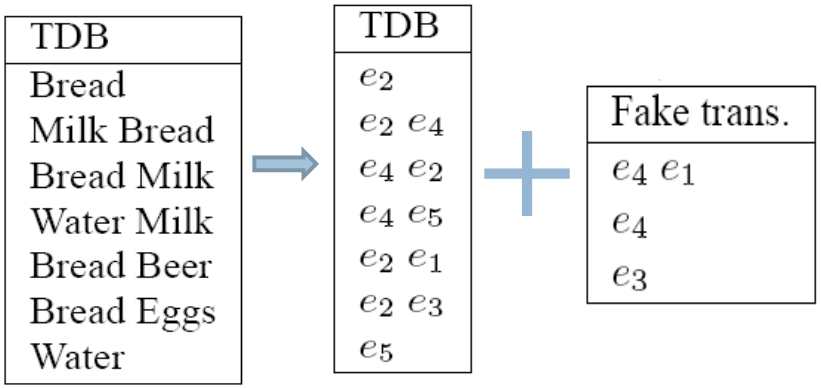
Definition 2 (k–Privacy). Given a database D and its encrypted version D^*, we say D^* is *k-private* if:

(1) for each cipher item $e \in D^*$, $prob(e) \leq 1/k$; and

(2) for each cipher itemset E with support $supp_{D^*}(E) > 0$, $prob(E) \leq 1/k$. □

Encryption and Decryption

- **Encryption:**
  - Replacing each plain item in D by a 1-1 substitution cipher
  - **K-Grouping**: for each item **e** there are at least others *k-1* enciphered items with same support
  - Adding fake transactions



- **Decryption:** A Synopsis allows computing the actual support of every pattern

RobFrugal: k-Private Grouping Method

- The idea: obtaining ***Robust k-groups*** unsupported in D
- RobFrugal Grouping
  - Given the TDB D and its item support table in decreasing order of support:
    - **Step1:** grouping together cipher items into groups of k adjacent items Obtaining **G = (G_1, ...,G_m)** (Frugal Grouping)
    - **Step2:** modifying the groups of G by swapping operations, until no group of items is supported in D

TDB
e_2
e_2 e_4
e_4 e_2
e_4 e_5
e_2 e_1
e_2 e_3
e_5

Item	Support
e_2	5
e_4	3
e_5	2
e_1	1
e_3	1

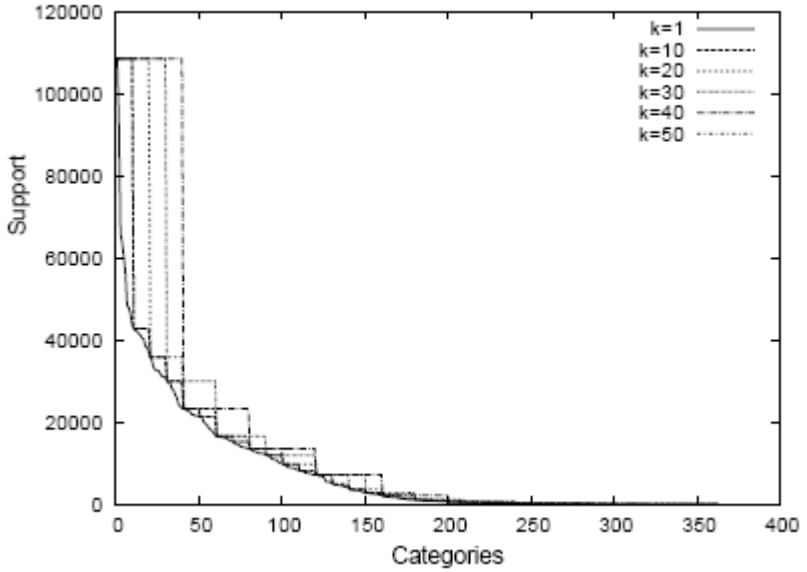
Item	Support
e_2	5
e_5	2
e_4	3
e_1	1
e_3	1

How create Fake Transactions?

- Output of the grouping step is a **Noise table**

Item	Support	Noise
e_2	5	0
e_5	2	3
e_4	3	0
e_1	1	2
e_3	1	2

Item	Noise
e_5	3
e_3	2
e_1	2



$\{e_5\}$

$\{e_5, e_3\}$ $\{e_5, e_1\}$

$\{e_5, e_3, e_1\}$

$L > L_{max}$

Synopsis in client-side

- The noise table provides a compact *synopsis*
  - used for decryption to compute the true support of a pattern
  - represents the fake transactions
- Hash table created with a *minimal perfect hash function*

Item	Noise
e_5	3
e_3	2
e_1	2

Fake Trans.
$\{e_5\}$
$\{e_1\}$ $\{e_1\}$
$\{e_5, e_3\}$ $\{e_5, e_3\}$

e_5 = item
1 = $\{e_5\}$ occurs once
2 = $\{e_5,e_3\}$ occurs 2 times

	Table1
0	$\langle e_5, 1, 2\rangle$
1	$\langle e_3, 2, 0\rangle$

	Table2
0	$\langle e_1, 2, 0\rangle$

Decryption: How to use the synopsis?

- The client receives frequent patterns mined over D*
- Synopsis allows computing the actual support of every pattern

Item	Support	Noise
e_2	5	0
e_5	2	3
e_4	3	0
e_1	1	2
e_3	1	2

	Table1
0	$\langle e_5, 1, 2 \rangle$
1	$\langle e_3, 2, 0 \rangle$

	Table2
0	$\langle e_1, 2, 0 \rangle$

Fake Trans.
$\{e_5\}$
$\{e_1\}$ $\{e_1\}$
$\{e_5, e_3\}$ $\{e_5, e_3\}$

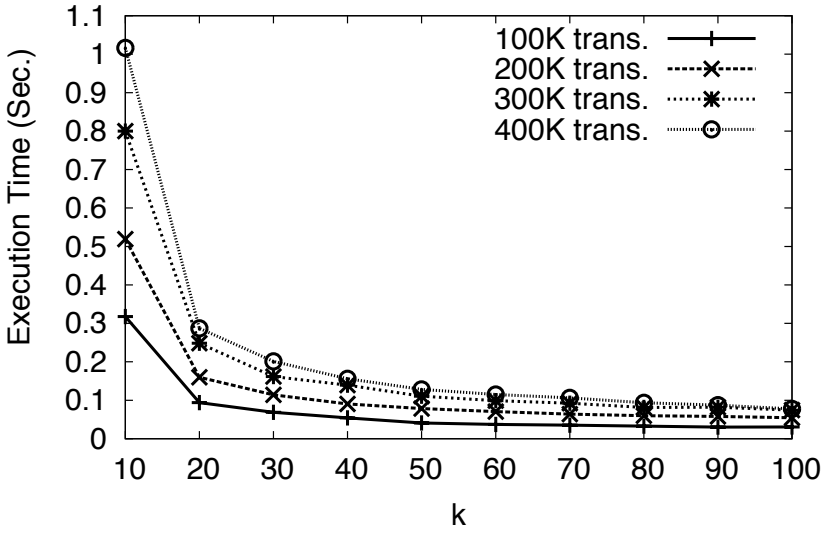
- $RS(\{\mathbf{e_5}\}) = supp_{D^*} - supp_{D^* \setminus D} = 5 - (1 + 2) = 2$
- $RS(\{\mathbf{e_5}, \mathbf{e_3}\}) = supp_{D^*} - supp_{D^* \setminus D} = 2 - (2 + 0) = 0$

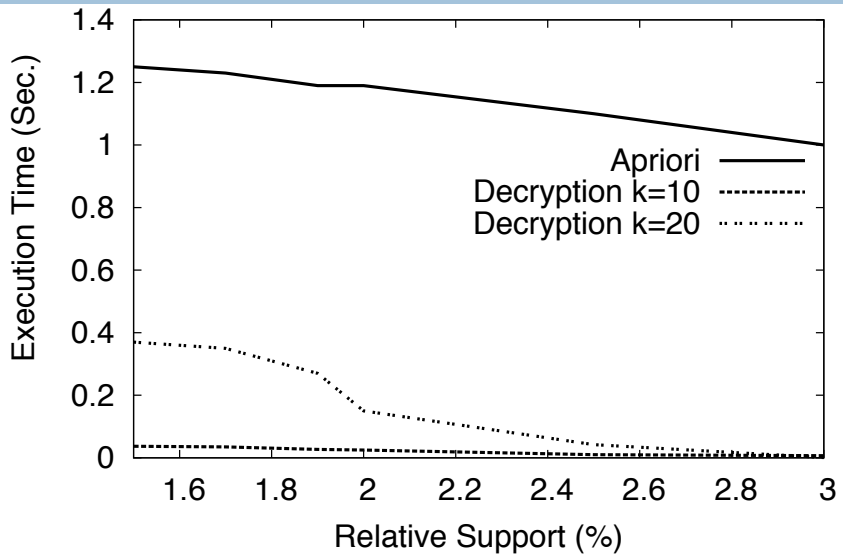
Complexity

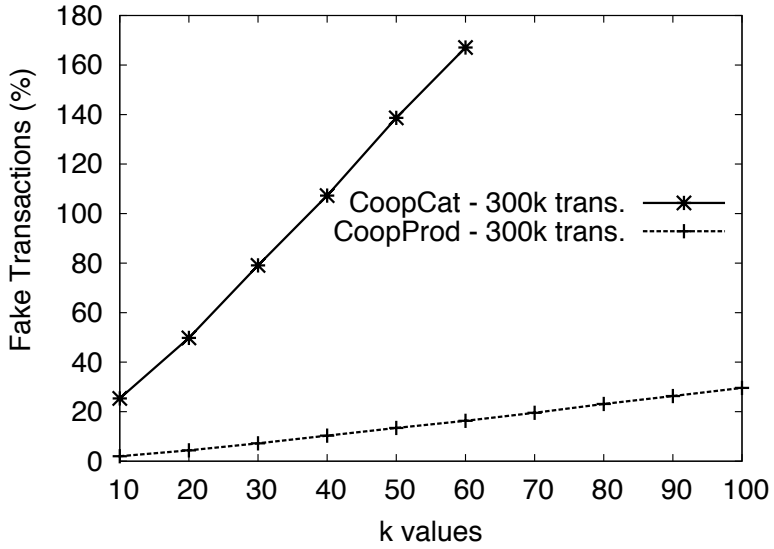
- **Encryption** by RobFrugal scheme requires
  - $O(n)$ space
  - $O(n^2)$ time

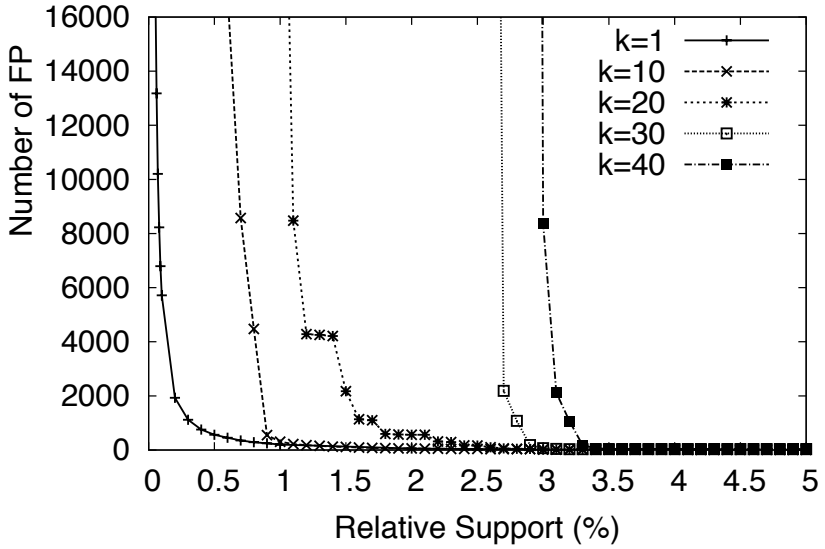
- **Decryption** requires $O(|E|)$ for each pattern E

Client and Server Overhead: Coop Data









Privacy Analysis

- Item-based attack
  - **RobFrugal** guarantees the k-privacy against the item-based attack (***prob(e) <= 1/k***)
- Itemset-based attack
  - **RobFrugal** guarantees the k-privacy against the itemset-based attack (***prob(E) <= 1/k***)
- On Coop dataset for k=10 we have:
  - 5% of transactions have exactly a crack probability 1/10
  - 95% of transactions have a probability strictly smaller than 1/10
  - 90% have a probability strictly smaller than 1/100
  - No single transaction contains any pattern consisting exactly of the items in a group created by RobFrugal

Conclusion

- An Enryption/Decryption Schema for privacy-preserving outsourcing of association rules mining
- Preliminary experiments on large real database
- Issues to be addressed:
  - Complexity Analysis
  - Privacy analysis to prove that the crack probability can be controlled
  - Strategy for incrementally maintaining the synopsis

Thank You!