An Architecture for Privacy Preserving Mining of Client Information

Jaideep Vaidya
Purdue University
jsvaidya@cs.purdue.edu

This is joint work with Murat Kantarcioglu

CERIAS

Indiana Center for Database Systems

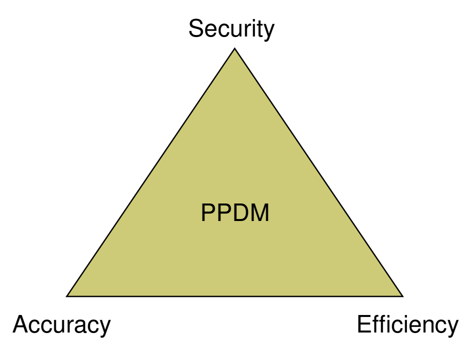
What is Privacy Preserving Data Mining?

- Term appeared in 2000:
  - Agrawal and Srikant, SIGMOD
    - Added noise to data before delivery to the data miner
    - Technique to reduce impact of noise learning a decision tree
  - Lindell and Pinkas, CRYPTO
    - Two parties, each with a portion of the data
    - Learn a decision tree without sharing data
- *Different Concepts of Privacy!*

Related Work

- Perturbation Approaches
  - Agrawal & Srikant, SIGMOD 2000
  - Agrawal & Aggarwal, SIGMOD 2001
  - Evfimievski et al, SIGKDD 2002
  - Rizvi & Haritsa, VLDB 2002
- SMC approaches
  - Lindell & Pinkas, CRYPTO 2000
  - Kantarcioglu & Clifton, DMKD 2002
  - Vaidya & Clifton, SIGKDD 2002
  - Du & Atallah, NSPW 2001

Motivation



Improving any one aspect typically degrades the other two

Motivating Example

- Assume that an attribute Y is perturbed by uniform random variable with range [-2,2].
- If we see $Y'_i = Y_i + r = 5$, then $Y_i \in [3,7]$
- Assume after reconstruction of the distribution(The basic assumption of all perturbation techniques is that we *can* reconstruct distributions),

$$\Pr\{3 \le Y \le 4\} \approx 0$$

- This implies $Y_i \in [4,7]$

Motivating Example (Cont.)

- Even worse, assume that

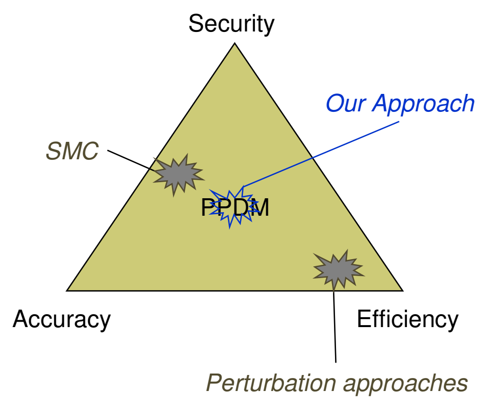
$$\Pr\{6 \le Y \le 7 \mid 0.5 \le T \le 1.0\} \approx 0.9$$

$$\Pr\{0.5 \le T_i \le 1.0\} \approx 0.9$$

- Therefore we could infer that

$$\Pr\{6 \le Y_i \le 7\} \approx 0.8$$

Motivation



Motivation

- Perfect Privacy *is* achievable without compromising on Accuracy
- Users do not want to be permanently online (to engage in some complex protocol)
- Outside parties can be used as long as there are strict bounds on what information they receive and what operations they are allowed to do

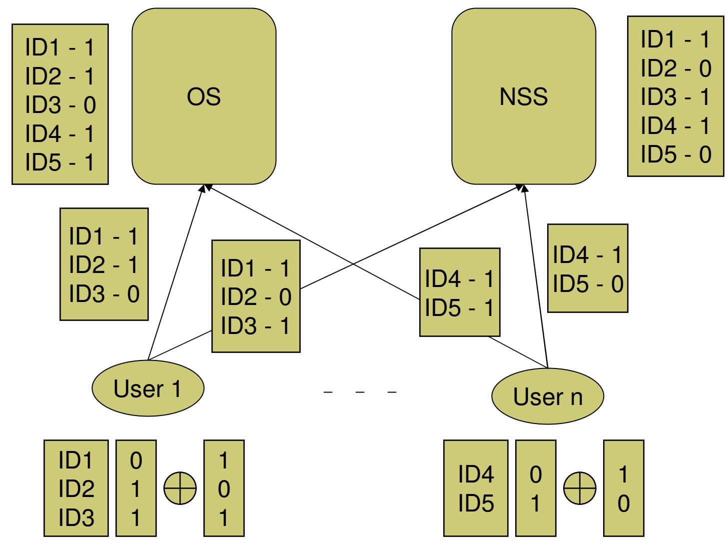
Key Insight

- Consider using *non-colluding, untrusted, semi-honest* third parties to carry out computation
- Non-colluding
  – Should not collude with any of the original users or any of the other parties
- Untrusted
  – Throughout the process, should never gain access to any information (in the clear), as long as the first assumption (non-colluding) holds true
- Semi-honest
  – All parties correctly follow the protocol, but are then free to use whatever information they see during the execution of the protocols in any way
  – Required to guarantee accuracy of result
  – *Even if a party is malicious, privacy is preserved!*

The Architecture

- Use three sites with the properties defined earlier:
- Originating Site (OS)
  – Site that collects share of the information from all clients, and will learn the final result of the data mining process
- Non-Colluding Storage Site (NSS)
  – Used for storing shared part of user information
- Processing Site (PS)
  – Used to do data mining efficiently

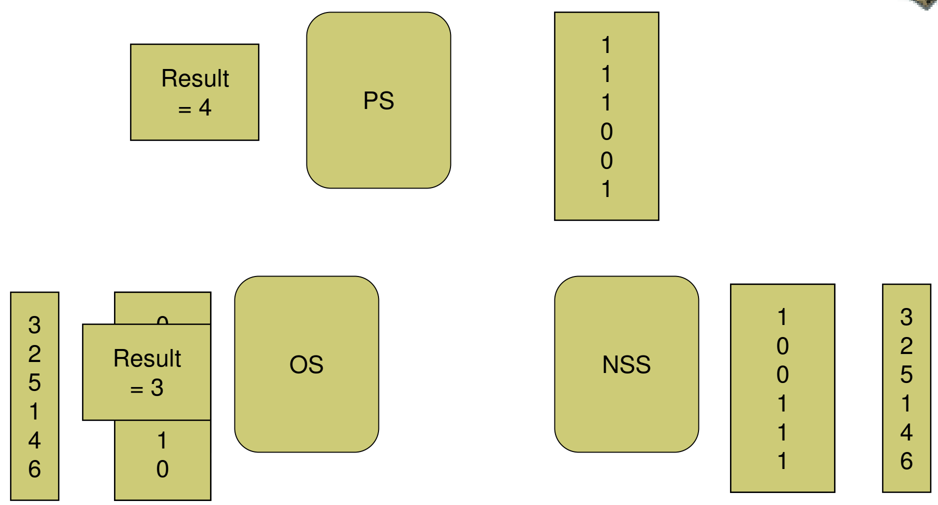
Finding Frequent Itemsets



Interlude

- For our protocol,
- total number of transactions, $n = 5$
- number of fake transactions to add (fraction of total), *epsilon* = 0.2
- epsilon*n = 5*0.2 = 1
- Originating Site decides to make some of the fake transactions supporting the itemset, while some don't (it knows the exact count)

Finding Frequent Itemsets



Doing Secure Data Mining

- Once the support count of an itemset has been calculated, the process for finding association rules securely is well known
- Other data mining algorithms become easily possible by modifying the process

Communication Cost:

- For each *k*-itemset at least $O(n \cdot k)$ bits must be transferred for the exact result.
  - The absolute minimum in any equivalently secure mechanism is the (boolean) database size ($C_1 \cdot n$)
- Assume that:
  - the number of candidate *k*-itemsets is C_k .
  - The largest candidate itemset is of size *m*
- Total communication cost for the association rule mining would be $O\left(\sum_{i=1}^{m} C_i \cdot n \cdot i \cdot (1+\varepsilon)\right)$

Security Analysis

- NSS view: The NSS only gets to see random numbers. Thus, it does not learn anything.
- OS view: OS learns the support count of the itemsets but does not learn which user supports any particular itemset. Essentially,
  $\forall i, j \ \Pr\{\text{user}_i \text{ supports X}\} = \Pr\{\text{user}_j \text{ supports X}\}$

Security Analysis (Cont.)

- PS learns an upper bound on the support count but it does not know for which itemset. (Ordering of the attributes randomized)
- Because of the addition of fake items and random ordering, it has no way of correlating the itemsets to any particular user.

Security Analysis

As long as the three sites (OS, NSS and PS) do not collude with each other, they do not learn anything

Benefits of the framework

- Perfect individual privacy is achieved
- Users do not have to stay online for a complicated protocol. Once they have split their information among the storage sites, they are done

Future Work

- An extremely efficient way of generating one-itemsets securely is possible. Using this instead of the general method, will lead to great savings in communication
- Sampling should be done to further lower communication cost and increase efficiency

Conclusion

- Privacy and Efficiency are both important for Secure Data Mining. Compromising on either is not practical
- A framework for privacy preserving data mining has been suggested
- Need to implement and evaluate true efficiency, after including improvements such as sampling