Secure Multiparty Computation – Applications for Privacy Preserving Distributed Data Mining

Li Xiong

CS573 Data Privacy and Security

Slides credit: Chris Clifton, Purdue University; Murat Kantarcioglu, UT Dallas
Distributed Data Mining

- Government / public agencies. Example:
  - The Centers for Disease Control want to identify disease outbreaks
  - Insurance companies have data on disease incidents, seriousness, patient background, etc.
  - But can/should they release this information?

- Industry Collaborations / Trade Groups. Example:
  - An industry trade group may want to identify best practices to help members
  - But some practices are trade secrets
  - How do we provide “commodity” results to all (Manufacturing using chemical supplies from supplier X have high failure rates), while still preserving secrets (manufacturing process Y gives low failure rates)?
Classification

- Data partition
  - Horizontally partitioned
  - Vertically partitioned

- Techniques
  - Data perturbation
  - Secure Multi-party Computation protocols

- Mining applications
  - Decision tree
  - Bayes classifier
  - ...
Horizontally Partitioned Data

- Data can be unioned to create the complete set
Vertically Partitioned Data

Data can be joined to create the complete set

<table>
<thead>
<tr>
<th>key</th>
<th>X1...Xi</th>
<th>Xi+1...Xj</th>
<th>...</th>
<th>Xm+1...Xd</th>
</tr>
</thead>
</table>

Site 1

Site 2

...  

Site r
Secure Multiparty Computation

- Goal: Compute function when each party has some of the inputs
- Secure
  - Can be simulated by ideal model - nobody knows anything but their own input and the results
  - Formally: \( \exists \) polynomial time \( S \) such that \( \{S(x,f(x,y))\} \equiv \{\text{View}(x,y)\} \)
- Semi-Honest model: follow protocol, but remember intermediate exchanges
- Malicious: “cheat” to find something out
Secure Multiparty Computation

*It can be done!*

- Basic cryptographic tools
  - Oblivious transfer
  - Oblivious circuit evaluation
- Yao’s Millionaire’s problem *(Yao ’86)*
  - Secure computation possible if function can be represented as a circuit
- Works for multiple parties as well *(Goldreich, Micali, and Wigderson ’87)*
Why aren’t we done?

- Secure Multiparty Computation is possible
  - But is it practical?
- Circuit evaluation: Build a circuit that represents the computation
  - For all possible inputs
  - Impossibly large for typical data mining tasks
- The next step: *Efficient* techniques for specialized tasks and computations
Outline

- Privacy preserving two-party decision tree mining (Lindell & Pinkas ’00)
- Privacy preserving distributed data mining toolkit (Clifton ’02)
  - Secure sum
  - Secure union
- Association Rule Mining on Horizontally Partitioned Data (Kantarcioglu ‘04)

Privacy preserving data mining, Lindell, 2000
Tools for Privacy Preserving Data Mining, Clifton, 2002
Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data, Kantarcioglu, 2004
Decision Tree Construction (*Lindell & Pinkas ’00*)

- Two-party horizontal partitioning
  - Each site has same schema
  - Attribute set known
  - Individual entities private
- Learn a decision tree classifier
  - ID3
- Essentially ID3 meeting Secure Multiparty Computation Definitions
- Semi-honest model
A Decision Tree for “buys_computer”

- **age?**
  - **<=30**
  - **31..40**
  - **>40**
    - **student?**
      - **no**
      - **yes**
    - **yes**
    - **credit rating?**
      - **excellent**
        - **no**
        - **yes**
      - **fair**
        - **yes**
ID3 Algorithm for Decision Tree Induction

- Greedy algorithm - tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - A test attribute is selected that “best” separate the data into partitions - information gain
  - Samples are partitioned recursively based on selected attributes
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left
Privacy Preserving ID3

Input:
- R – the set of attributes
- C – the class attribute
- T – the set of transactions

Step 1: *If R is empty, return a leaf-node with the class value with the most transactions in T*

- Set of attributes is public
  - Both know if R is empty
- Use secure protocol for majority voting
  - Yao’s protocol
    - Inputs ($|T_1(c_1)|$, $|T_2(c_1)|$, $|T_1(c_L)|$, $|T_2(c_L)|$)
    - Output $i$ where $|T_1(c_i)| + |T_2(c_i)|$ is largest
Privacy Preserving ID3

Step 2: If $T$ consists of transactions which have all the same value $c$ for the class attribute, return a leaf node with the value $c$

- Represent having more than one class (in the transaction set), by a fixed symbol different from $c_i$,
- Force the parties to input either this fixed symbol or $c_i$
- Check equality to decide if at leaf node for class $c_i$
- Various approaches for equality checking
  - Yao’86
  - Fagin, Naor ’96
  - Naor, Pinkas ‘01
Privacy Preserving ID3

- Step 3: (a) *Determine the attribute that best classifies the transactions in T, let it be A*

  \[ Entropy(S) = - \sum_{v \in \text{label}(S)} P(v) \log P(v) = - \sum_{v \in \text{label}(S)} \frac{n_v}{n} \log \frac{n_v}{n} \]

  - Essentially done by securely computing \( x^*(\ln x) \) where \( x \) is the sum of values from the two parties

- P1 and P2, i.e., \( x_1 \) and \( x_2 \), respectively

- Step 3: (b,c) *Recursively call ID3 for the remaining attributes on the transaction sets \( T(a_1), \ldots, T(a_m) \) where \( a_1, \ldots, a_m \) are the values of the attribute A*

  - Since the results of 3(a) and the attribute values are public, both parties can individually partition the database and prepare their inputs for the recursive calls
Security proof tools

- Real/ideal model: the real model can be simulated in the ideal model
  - Key idea – Show that whatever can be computed by a party participating in the protocol can be computed based on its input and output only
  - ∃ polynomial time $S$ such that $\{S(x,f(x,y))\} \equiv \{\text{View}(x,y)\}$
Security proof tools

- Composition theorem
  - if a protocol is secure in the hybrid model where the protocol uses a trusted party that computes the (sub) functionalities, and we replace the calls to the trusted party by calls to secure protocols, then the resulting protocol is secure
  - Prove that component protocols are secure, then prove that the combined protocol is secure
Secure Sub-protocols for PPDDM

- In general, PPDDM protocols depend on few common sub-protocols.

- Those common sub-protocols could be re-used to implement PPDDM protocols.
Privacy preserving data mining toolkit (Clifton ‘02)

- Many different data mining techniques often perform similar computations at various stages (e.g., computing sum, counting the number of items)

- Toolkit
  - simple computations – sum, union, intersection …
  - assemble them to solve specific mining tasks – association rule mining, bayes classifier, …

- The protocols may not be truly secure but more efficient than traditional SMC methods

Tools for Privacy Preserving Data Mining, Clifton, 2002
Toolkit

- Secure functions
  - Secure sum
  - Secure union
  - ...

- Applications
  - Association rule mining for horizontally partitioned data
  - ...

Secure Sum

- Leading site: \( (v_1 + R) \mod n \)
- Site l receives:

\[
V = R + \sum_{j=1}^{l-1} v_j \mod n
\]

- Leading site: \( (V-R) \mod n \)
Secure Sum

Site 3
12+7
7
19
19-R=2!
R=17

Site 2
-5
17-5

Site 1
0
R+0
Secure sum - security

- Does not reveal the real number
- Is it secure?
- Site can collude!

Each site can divide the number into shares, and run the algorithm multiple times with permutated nodes
Secure Union

- Commutative encryption
  - For any set of permutated keys and a message $M$
    \[ E_{K_{i_1}} \ldots E_{K_{i_n}} (M) \ldots = E_{K_{j_1}} \ldots E_{K_{j_n}} (M) \ldots \]
  - For any set of permutated keys and message $M_1$ and $M_2$
    \[ \Pr(E_{K_{i_1}} \ldots E_{K_{i_n}} (M_1) \ldots = E_{K_{j_1}} \ldots E_{K_{j_n}} (M_2) \ldots) < \epsilon \]

- Secure union
  - Each site encrypt its items and items from other site, remove duplicates, and decrypt
Secure union
Secure Union Security

- Does not reveal which item belongs to which site
- Is it secure under the definition of secure multi-party computation?
- It reveals the number of items that are common in the sites!
- Revealing innocuous information leakage allows a more efficient algorithm than a fully secure algorithm
Association Rules Mining

- Assume data is horizontally partitioned
  - Each site has complete information on a set of entities
  - Same attributes at each site
- If goal is to avoid disclosing entities, problem is easy
- Basic idea: Two-Phase Algorithm
  - First phase: Compute candidate rules
    - Frequent globally $\Rightarrow$ frequent at some site
  - Second phase: Compute frequency of candidates
Association Rules in Horizontally Partitioned Data

A & B \Rightarrow C

Data Mining Combiner

Combined results

Request for local bound-tightening analysis

A & B \Rightarrow C 4%

Local Data Mining

Local Data

Local Data Mining

Local Data

Local Data Mining

Local Data
Association Rule Mining: Horizontal Partitioning

What if we do not want to reveal which rule is supported at which site, the support count of each rule, or database sizes?

- Hospitals want to participate in a medical study
- But rules only occurring at one hospital may be a result of bad practices
Privacy-preserving Association rule mining for horizontally partitioned data
(Kantarcioğlu’04)

- Find the union of the locally large candidate itemsets securely
- After the local pruning, compute the globally supported large itemsets securely
- At the end check the confidence of the potential rules securely
Securely Computing Candidates

- Compute local candidate set
- Using secure union!
Computing Candidate Sets

1

\[ E_1(E_2(E_3(ABC))) \]
\[ E_1(E_2(E_3(ABD))) \]

2

\[ E_2(E_3(E_3(ABC))) \]
\[ E_2(E_3(E_3(ABD))) \]

3

\[ E_3(E_3(E_3(ABC))) \]
\[ E_3(E_3(E_3(ABD))) \]

ABC

ABD
Compute Which Candidates Are Globally Supported?

Goal: To check whether

\[ X_{\text{sup}} \geq s^* \sum_{i=1}^{n} |DB_i| \]  \hspace{1cm} (1)

\[ \sum_{i=1}^{n} X_{\text{sup},i} \geq \sum_{i=1}^{n} s^* |DB_i| \]  \hspace{1cm} (2)

\[ \sum_{i=1}^{n} (X_{\text{sup},i} - s^* |DB_i|) \geq 0 \]  \hspace{1cm} (3)

Note that checking inequality (1) is equivalent to checking inequality (3)
Securely compute sum then check if sum $\geq 0$

Is this a good approach?

Sum is disclosed!

Securely compute Sum - R
Securely compare sum $\geq R$?
  - Use oblivious transfer
Computing Frequent: Is $ABC \geq 5\%$?

$ABC: 12 + 18 - 0.05 \times 300$

$ABC: 19$

$ABC: 17 + 9 - 0.05 \times 200$

$ABC: 12$

$ABC: 17 + 5 - 0.05 \times 100$

$ABC: 17$

$ABC: 17 + 9 - 0.05 \times 200$

$ABC: 12$

$ABC: R + \text{count-freq.} \times \text{DBSize}$

$ABC: R = 17$

$ABC: YES!$

$ABC: 19 \geq R?$

$ABC: YES!$
Computing Confidence

Checking confidence can be done by the previous protocol. Note that checking confidence for $X \Rightarrow Y$

$$\frac{\{X \cup Y\}.\sup}{X.\sup} \geq c \implies \sum_{i=1}^{n} XY.\sup_i \geq c$$

$$\sum_{i=1}^{n} X.\sup_i$$

$$\Rightarrow \sum_{i=1}^{n} (XY.\sup_i - c \times X.\sup_i) \geq 0$$
Secure Functionalities

- **Secure Comparison**: Comparing two integers without revealing the integer values.

- **Secure Polynomial Evaluation**: Party A has polynomial \( P(x) \) and Part B has a value \( b \), the goal is to calculate \( P(b) \) without revealing \( P(x) \) or \( b \).

- **Secure Set Intersection**: Party A has set \( S_A \) and Party B has set \( S_B \), the goal is to calculate \( S_A \cap S_B \) without revealing anything else.
Secure Functionalities Used

- **Secure Set Union:** Party A has set $S_A$ and Party B has set $S_B$, the goal is to calculate $S_A \cup S_B$ without revealing anything else.

- **Secure Dot Product:** Party A has a vector $X$ and Party B has a vector $Y$. The goal is to calculate $X.Y$ without revealing anything else.
Specific Secure Tools

- Secure Sum
- Secure Comparison
- Secure Union
- Secure Logarithm
- Secure Poly. Evaluation

Data Mining on Horizontally Partitioned Data

- Association Rule Mining
- Decision Trees
- EM Clustering
- Naïve Bayes Classifier

Data Mining on Horizontally Partitioned Data
Specific Secure Tools

- Secure Comparison
- Secure Set Intersection
- Secure Dot Product
- Secure Logarithm
- Secure Poly. Evaluation

Data Mining on Vertically Partitioned Data

- Association Rule Mining
- Decision Trees
- K-means Clustering
- Naïve Bayes Classifier
- Outlier Detection
Summary of SMC Based PPDDM

- Mainly used for distributed data mining.
- Provably secure under some assumptions.
- Efficient/specific cryptographic solutions for many distributed data mining problems are developed.
- Mainly semi-honest assumption (i.e. parties follow the protocols)
Drawbacks for SMC Based PPDDM

- Drawbacks:
  - Still not efficient enough for very large datasets
  - Semi-honest model may not be realistic
  - Malicious model is even slower
Possible New Directions

- New models that can trade-off better between efficiency and security
- Game theoretic / incentive issues in PPDM
- Combining anonymization and cryptographic techniques for PPDM
References

- Privacy preserving data mining, Lindell, 2000
- Tools for Privacy Preserving Data Mining, Clifton, 2002
- Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data, Kantarcioglu, 2004