CS573 Data Privacy and Security

Secure Multiparty Computation –
Introduction to Privacy Preserving
Distributed Data Mining

Li Xiong

Slides credit: Chris Clifton, Purdue University; Murat Kantarcioglu, UT Dallas
Outline

• Overview
• Data partition
  – Horizontally partitioned
  – Vertically partitioned
• Privacy preserving Distributed Data Mining
• Approaches to preserve privacy
• Privacy preserving data mining toolkit
Overview

• What is Data Mining?
  – Extracting implicit un-obvious patterns and relationships from a warehoused of data sets.

• This information can be useful to increase the efficiency of the organization and aids future plans

• Can be done at an organizational level
  – By Establishing a data Warehouse
Motivation

• **Huge databases** exist in various applications
  – Medical data
  – Consumer purchase data
  – Census data
  – Communication and media-related data
  – Data gathered by government agencies

• **Can these data be utilized?**
  – For medical research
  – For improving customer service
  – For homeland security
Motivation

• Data sharing is necessary for full utilization
  • Pooling medical data can improve the quality of medical research

• The huge amount of data available means that it is possible to learn a lot of information about individuals from public data
  – Purchasing patterns
  – Family history
  – Medical data
  – ...
Horizontally Partitioned Data

• Data can be unioned to create the complete set

<table>
<thead>
<tr>
<th>key</th>
<th>X1...Xd</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td></td>
</tr>
<tr>
<td>k2</td>
<td></td>
</tr>
<tr>
<td>ki</td>
<td></td>
</tr>
</tbody>
</table>

Site 1

<table>
<thead>
<tr>
<th>key</th>
<th>X1...Xd</th>
</tr>
</thead>
<tbody>
<tr>
<td>K_{i+1}</td>
<td></td>
</tr>
<tr>
<td>k_{i+2}</td>
<td></td>
</tr>
<tr>
<td>kj</td>
<td></td>
</tr>
</tbody>
</table>

Site 2

<table>
<thead>
<tr>
<th>key</th>
<th>X1...Xd</th>
</tr>
</thead>
<tbody>
<tr>
<td>K_{m+1}</td>
<td></td>
</tr>
<tr>
<td>k_{m+2}</td>
<td></td>
</tr>
<tr>
<td>kn</td>
<td></td>
</tr>
</tbody>
</table>

Site r
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>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site 1</td>
<td>Site 2</td>
<td>...</td>
<td>Site r</td>
<td></td>
</tr>
</tbody>
</table>
Distributed Data Mining

• The setting:
  – Data is **distributed** at different sites
  – These sites may be third parties (e.g., hospitals, government bodies) or may be the individual him or herself
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)?
Privacy and Security Restrictions

• Individual Privacy
  – Nobody should know more about any entity after the data mining than they did before

• Organization Privacy
  – Protect knowledge about a collection of entities
    • Individual entity values may be known to all parties
    • Which entities are at which site may be secret
Privacy-Preserving Distributed Data Mining: Why?

• Data needed for data mining maybe distributed among parties
  – Credit card fraud data
• Inability to share data due to privacy reasons
  – HIPPAA
• Even partial results may need to be kept private
Approaches to preserve privacy

• Restrict Access to data (Protect Individual records)

• Protect both the data and its source:
  – Secure Multi-party computation (SMC)
  – Input Data Randomization

• There is no such one solution that fits all purposes
Secure computation and privacy

• Secure computation
  – Assume that there is a function that all parties wish to compute
  – Secure computation shows how to compute that function in the safest way possible
  – In particular, it guarantees minimal information leakage (the output only)

• Privacy
  – Does the function output itself reveal “sensitive information”, or
  – Should the parties agree to compute this function?
Secure Multi-Party Computation (SMC)

- The goal is computing a function $f(x_1, x_2, \ldots, x_n)$ without revealing $x_i$
- Semi-Honest Model
  - Parties follow the protocol
- Malicious Model
  - Parties may or may not follow the protocol
- We cannot do better than the existence of the third trusted party situation
- Generic SMC is too inefficient for PPDDM
Secure Multiparty Computation

- Basic cryptographic tools
  - Oblivious transfer
  - Random shares
  - 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)
But we aren’t done yet

• Circuit evaluation: Build a circuit that represents the computation
  – For all possible inputs
  – Impossibly large for typical data mining tasks

• Next step:
  – Efficient techniques for specialized tasks and computations
  – Tradeoff between security, efficiency, and accuracy
Secure computation tasks

• **Examples:**
  – Authentication protocols
  – Online payments
  – Auctions
  – Elections
  – Privacy preserving data mining
  – Essentially any task...
Application of SMC to Private Data Mining

• **Setting**
  – Data is *distributed* at different sites
  – These sites may be third parties (e.g., hospitals, government bodies) or individuals

• **Aim**
  – Compute the data mining algorithm on the data so that *nothing but the output is learned*
  – That is, carry out a *secure computation*
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
Primitive protocols

• Secure functions
  – Secure sum
  – Secure union
  – ...

Secure Sum

• Distributed data mining algorithms frequently calculate the sum of values from individual sites
• Suppose we have s sites 1, ..., s
• Site \( l \) has an integer \( v_l \)
• The sites want to know the value of

\[
f (v_1, ..., v_s) = \sum_{l=1}^{s} v_l
\]

• Easy:
  – One site is designated the master site, numbered 1
  – Site \( l \) send \( v_l \) to party 1 (\( 2 \leq l \leq s \))
  – Site 1 computes \( f (v_1, ..., v_s) = \sum_{l=1}^{s} v_l \) and broadcasts it
Secure Sum

• What they don’t like about this:
  – Site 1 now knows everyone’s values

• Privacy constraint:
  – Site $l$ does not wish to reveal $v_l$
Secure Sum II

• Suppose we have $s$ sites $1, \ldots, s$
• Site $l$ has an integer $v_l$
• The sites want to know the value of
  \[ f (v_1, \ldots, v_s) = v_1 + \ldots + v_s \]

• Assume that the value $v = \sum_{l=1}^{s} v_l$ to be computed is known to lie in the range $[0..n]$
Secure Sum II

• Site 1:
  – generates a random number $R$, uniformly chosen from $[0..n]$
  – adds $R$ to its local value $v_1$, and sends $R + v_1 \mod n$ to site 2

• For $l = 2 \ldots s - 1$
  – Site $l$ receives $V = R + \sum_{j=1}^{l-1} v_j \mod n$
  – Site $l$ then computes
    • $V = R + \sum_{j=1}^{l} v_j \mod n = (v_l + V) \mod n$
    – Pass it to site $l + 1$

• Site $s$ performs the above step, and sends the result to site 1

• Site 1, knowing $R$, can subtract $R$ to get the actual result:
  \[(V - R) \mod n\]
Secure Sum II

Site 3
12 + 7
7

Site 2
-5
17 - 5

Site 1
0
R + 0

19
19 - R = 2!
R = 17
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

- Useful in DM where each party needs to give rules, frequent itemsets, etc., without revealing the owner
- Can be evaluated using SMC methods if the domain of the items is small
- Each party creates a binary vector where 1 in the $i^{th}$ entry represents that the party has the $i^{th}$ item
- After this point, a simple circuit that or’s the corresponding vectors can be built and it can be securely evaluated using general SM circuit evaluation protocols
- However, in data mining the domain of the items is usually large
Secure Union

• Consider k parties $P_1, ..., P_k$ having local sets $S_1, ..., S_k$, we wish to securely compute

• $U = S_1 \cup S_2 \cup \cdots \cup S_k$

• Such that each party only knows $U$ and nothing else

• Key: Commutative Encryption $E_a(E_b(x)) = E_b(E_a(x))$ — (decryption function has the same property)

• Multiple encryption and decryption operations can be performed over a value without any restriction about the order of these operations
Secure Union

• Global Union Set $U$
• Each site:
  – Encrypts its items
  – Creates an array $M[k]$ and adds it to $U$
• Upon receiving $U$ a party should encrypt all items in $U$ that it did not encrypt before
• In the end: all parties are encrypted with all keys $K_1, ..., K_k$
• Remove the duplicates:
  – Identical plain text will result the same cipher text regardless of the order of the use of encryption keys
• Decryption $U$:
  – Done by all parties in any order

Slide credit: Privacy Preserving Data Mining, Moheeb Rajab, Johns Hopkins University
Secure Union

1
ABC

2
ABD

3
ABC

E_1(E_2(E_3(ABC)))
E_1(E_2(E_3(ABD)))

E_3(ABC)
E_3(ABD)

E_2(E_3(E_3(ABO))))

E_3(E_8(E_2(ABD))))
U = \{E3(E2(E1(A))), E3(E2(C)), E3(A)\}

U = \{E2(E1(A)), E2(C)\}

U = \{E3(E2(E1(A))), E1(E3(E2(C))), E1(E3(A))\}

U = \{E3(E2(E1(A))), E1(E3(E2(C))), E2(E1(E3(A)))\}
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
Privacy-Preserving Distributed Association Rule Mining

• Exchanging support counts is enough for mining association rules

• We do not want to reveal
  – which rule is supported (or not) at which site
  – the support count of each rule
  – the database sizes
  – e.g. Hospitals may not want to reveal procedures with high mortality rates
  – e.g. Companies may not want to reveal the traces of intrusions
Overview of the Method

1. Find the union of the locally large candidate itemsets securely

2. After the local pruning, compute the globally supported large itemsets securely

3. Check the confidence of the potential rules securely
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.
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.
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
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
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
- Learned models are accurate
- Efficient/specific cryptographic solutions for many distributed data mining problems are developed
- Mainly semi-honest assumption (i.e. parties follow the protocols)
- Malicious model is also explored recently
- Many SMC based PPDM algorithms share common sub-protocols (e.g. dot product, summation, etc.)
Drawbacks for SMC Based PPDDM

- Drawbacks:
  - Still not efficient enough for very large datasets. (e.g. petabyte sized datasets ??)
  - Semi-honest model may not be realistic
  - Malicious model is even slower