Privacy Preserving Association Rule Mining in Vertically Partitioned Data

Jaideep Vaidya and Chris Clifton

CERIAS

Indiana Center for Database Systems

The Problem

Medical Records

JSV	Brain Tumor	Diabetic
CWC	No Tumor	Non-Diabetic

Cell Phones with Li/Ion batteries lead to brain tumors in diabetics

Cell Phone

JSV	5210	Li/Ion
CWC	None	None

CWC doesn't have a cell phone

JSV has diabetes

- Centralized Data Mining often infeasible
- Multiple entities want to find association rules from combined data
- Unwilling to reveal proprietary information
  - Ford / Firestone (trade secrets)
  - General Manufacturer / Bulk Consumer
- Legal restrictions on sharing information
  - Health Service Providers / Insurance Companies
  - FBI / IRS

The data model

- Two parties – Alice (A) and Bob (B)
- Same set of entities (data cleansing, join assumed done)
- A has p attributes, $A_1 \ldots A_p$
- B has q attributes, $B_1 \ldots B_q$
- Total number of transactions, *n*
- Support Threshold, *k*

Basic idea

- Find out if itemset $\{A_1, B_1\}$ is frequent (Support of $\{A_1, B_1\} \geq k$?)
  - For binary data, support = $|A_i$ _ $B_i|$
  - *Boolean AND* is replaced by normal (arithmetic) multiplication.
- Thus,

$$Support = \sum_{i=1}^{n} A_i \times B_i \quad \text{(scalar dot product)}$$

- For finding out if an arbitrary (shared) itemset is frequent:
  - create a vector on each side, which consists of the product of all attribute vectors on that side (contained in the itemset)
  - Compute product of vectors
- E.g., to find out if $\{A_1, A_3, A_5, B_2, B_3\}$ is frequent,
  - A forms the vector $X = \prod A_1\, A_3\, A_5$
  - B forms the vector $Y = \prod B_2\, B_3$
  - they securely compute the dot product of X and Y.

The Solution

The overall algorithm

$L_1 = \{\text{large 1-itemsets}\}$
for (k=2; $L_{k-1} \neq \phi$; k++)
  C_k = apriori-gen(L_{k-1});
  for all candidates $c \in C_k$ do begin
  if all the attributes in c are entirely at A or B
    that party independently calculates *c.count*
  else
    let A have l of the attributes and B have the remaining m attributes
    construct $\vec{X}$ on A's side and $\vec{Y}$ on B's side where $\vec{X} = \prod_{i=1}^{l} \vec{A_i}$ and $\vec{Y} = \prod_{i=1}^{m} \vec{B_i}$
    compute $c.count = \vec{X}.\vec{Y} = \sum_{i=1}^{n} x_i * y_i$
  endif
  $L_k = L_k \cup c | c.count \geq minsup$
  end
end
Answer $= \cup_k L_k$

The Component Protocol (Simplified Version)

- A generates n/2 randoms, $R_1 \ldots R_{n/2}$
- A sends the following n values to B

$$\begin{aligned}
&\langle x_1 + a_{1,1} * R_1 + a_{1,2} * R_2 + \cdots + a_{1,n/2} * R_{n/2} \rangle\\
&\langle x_2 + a_{2,1} * R_1 + a_{2,2} * R_2 + \cdots + a_{2,n/2} * R_{n/2} \rangle\\
&\vdots\\
&\langle x_n + a_{n,1} * R_1 + a_{n,2} * R_2 + \cdots + a_{n,n/2} * R_{n/2} \rangle
\end{aligned}$$

- The $(n^2/2)$ $a_{i,j}$ values are known to both A and B

- B multiplies each value he gets with the corresponding y value he has and adds all of them up to get a sum S, which he sends to A.

$$S = \left[\begin{aligned}
&y_1 * \{x_1 + (a_{1,1} * R_1 + a_{1,2} * R_2 + \cdots + a_{1,n/2} * R_{n/2})\}\\
&+ y_2 * \{x_2 + (a_{2,1} * R_1 + a_{2,2} * R_2 + \cdots + a_{2,n/2} * R_{n/2})\}\\
&\vdots\\
&+ y_n * \{x_n + (a_{n,1} * R_1 + a_{n,2} * R_2 + \cdots + a_{n,n/2} * R_{n/2})\}
\end{aligned}\right]$$

- Group the $x_i * y_i$ terms, and expand the equations

$$\begin{aligned}
S = {}& x_1 * y_1 + x_2 * y_2 + \cdots + x_n * y_n\\
&+ (a_{1,1} * y_1 * R_1 + a_{1,2} * y_1 * R_2 + \cdots + a_{1,n/2} * y_1 * R_{n/2})\\
&+ (a_{2,1} * y_2 * R_1 + a_{2,2} * y_2 * R_2 + \cdots + a_{2,n/2} * y_2 * R_{n/2})\\
&\vdots\\
&+ (a_{n,1} * y_n * R_1 + a_{n,2} * y_n * R_2 + \cdots + a_{n,n/2} * y_n * R_{n/2})
\end{aligned}$$

$$\sum_{i=1}^{n} x_i * y_i$$

Grouping components vertically and factoring out R_i

$$\begin{aligned}
S = {}& \sum_{i=1}^{n} x_i * y_i\\
&+ R_1 * (a_{1,1} * y_1 + a_{2,1} * y_2 + \cdots + a_{n,1} * y_n)\\
&+ R_2 * (a_{1,2} * y_1 + a_{2,2} * y_2 + \cdots + a_{n,2} * y_n)\\
&\vdots\\
&+ R_{n/2} * (a_{1,n/2} * y_1 + a_{2,n/2} * y_2 + \cdots + a_{n,n/2} * y_n)
\end{aligned}$$

- A already knows $R_1 \ldots R_{n/2}$
- Now, if B sends these n/2 values to A,
- *A can remove the baggage and get the scalar product*

Security and Communication Analysis

- For Component Protocol,
  - A sends to B n+1 equations in 3n/2 unknowns
  - B sends to A n/2+1 equations in n unknowns
  - Total communication ≅ 3n/2 values (3 messages)
- Security is symmetric
- Based on revealing less equations than the number of unknowns – possible solutions infinite!
- Everything revealed *only* when about half the values are (externally) revealed
- For every sub-itemset, hidden data values sent only *once*.
- Thus, communication cost is comparable to I/O cost of a-priori. Loosely – Communication Cost = O(a-priori I/O cost), 1.5< k <2.5

Limitations

- Protocol works only in Honest-But-Curious model
  - Either party can "doctor" their input and find out specific information:
  - Can be fixed by allowing approximations on both sides (Thus neither party can get specific information, but the final result will be approximate)
- {0,1} data causes problems with security

Work Completed

- Extended to multiple parties
- Resistant to collusion
- Solutions for clustering and classification also developed

Prior/Related Work

- Privacy Preserving Data Mining
  - Data Perturbation, Secure Multiparty Computation Approaches
- Distributed Data Mining
  - Operating on both Horizontally and Vertically partitioned data
  - Meta learning approaches
- Secure Multiparty Computation
  - General proofs, Some specific problems solved

PURDUE UNIVERSITY

e-Enterprise Center