Privacy and anonymity in data publishing and (mobility) data mining

Fosca Giannotti, Dino Pedreschi, Franco Turini

Pisa KDD Laboratory
Università di Pisa and ISTI-CNR, Pisa, Italy

Dottorato di Ricerca in Informatica, Scuola Galilei

Università di Pisa. Giugno-Luglio 2009

Privacy-preserving data publishing: K-Anonymity

Data K-anonymity

- What is disclosed?
  - the data (modified somehow)
- What is hidden?
  - the real data
- How?
  - by transforming the data in such a way that it is not possible the re-identification of original database rows under a fixed anonymity threshold (individual privacy).

Why K-Anonymity?

- Several agencies, institutions, organizations make (sensitive) data involving people publicly available
  - termed microdata (vs. aggregated macrodata) used for analysis
  - often required and imposed by law
- To protect privacy microdata are sanitized
  - explicit identifiers (SSN, name, phone #) are removed
- Is this sufficient for preserving privacy? NO!
- Susceptible to link attacks
  - Attribute combinations, such as gender, age and postcode, uniquely identify some individuals

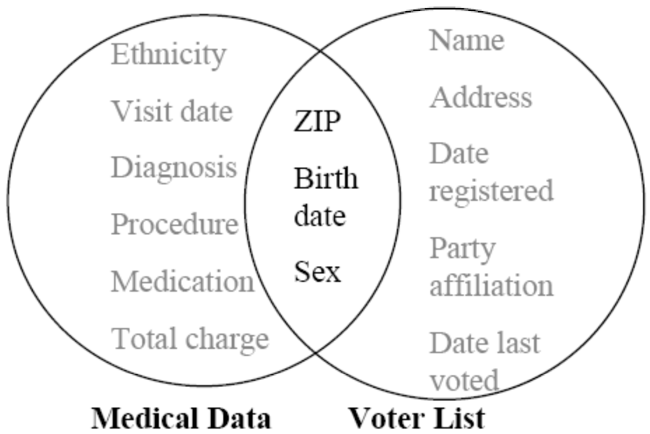
Unique Combination of attributes

Hospital Patient Data

DOB	Sex	Zipcode	Disease
1/21/76	Male	53715	Heart Disease
4/13/86	Female	53715	Hepatitis
2/28/76	Male	53703	Brochitis
1/21/76	Male	53703	Broken Arm
4/13/86	Female	53706	Flu
2/28/76	Female	53706	Hang Nail

Linking Attack

- Sweeney managed to re-identify the medical record of the governor of Massachussetts
  - MA collects and publishes sanitized medical data for state employees (microdata) left circle
  - voter registration list of MA (publicly available data) right circle



- looking for governor's record
- join the tables:
  - 6 people had his birth date
  - 3 were men
  - 1 in his zipcode
- regarding the US 1990 census data
  - 87% of the population are unique based on (zipcode, gender, dob)

Classification of Attributes

- **Key Attributes**:
  - Name, Address, Cell Phone
  - which can uniquely identify an individual directly
  - Always removed before release
- **Quasi-Identifiers**:
  - 5-digit ZIP code,Birth date, gender
  - A set of attributes that can be potentially linked with external information to re-identify entities
  - Suppressed or generalized
- **Sensitive Attribute:**
  - Medical record, wage, etc.
  - Always released directly. These attributes represent th information to be protected

Classification of Attributes: Example

Key Attribute	*Quasi-Identifier*			*Sensive Attribute*
Name	DOB	Gender	Zipcode	Disease
Andre	1/21/76	Male	53715	Heart Disease
Beth	4/13/86	Female	53715	Hepatitis
Carol	2/28/76	Male	53703	Brochitis
Dan	1/21/76	Male	53703	Broken Arm
Ellen	4/13/86	Female	53706	Flu
Eric	2/28/76	Female	53706	Hang Nail

K-Anonymity Protection Model

- PT: Private Table
- RT: Released Table
- QI: Quasi Identifier (Ai,...,Aj)
- (A1,A2,...,An): Attributes

Definition:

- Let RT(*A1*,...,*An*) be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy ***k*-anonymity** iff each sequence of values in RT[QI_{RT}] appears with at least *k* occurrences in RT[QI_{RT}].

K-Anonymity

- Proposed by Sweeney and Samarati
- k-anonymity: intuitively, hide each individual among k-1 others
  - each combination of values of QIs should appear at least k times in the released microdata
  - linking cannot be performed with confidence > 1/k
- How to achieve this?
  - Generalization: publish values **more general**, i.e., given a domain hierarchy, roll-up
  - Suppression: remove tuples, i.e., do not publish outliers. Often the number of suppressed tuples is bounded
- Privacy vs utility tradeoff
  - do not anonymize more than necessary
  - Minimize the distortion
- Complexity? Optimal anonymization (minimal distorsione) is NP-Hard!! [Meyerson and Williams PODS '04]

Example

	Race	Birth	Gender	ZIP	Problem
t1	Black	1965	m	0214*	short breath
t2	Black	1965	m	0214*	chest pain
t3	Black	1965	f	0213*	hypertension
t4	Black	1965	f	0213*	hypertension
t5	Black	1964	f	0213*	obesity
t6	Black	1964	f	0213*	chest pain
t7	White	1964	m	0213*	chest pain
t8	White	1964	m	0213*	obesity
t9	White	1964	m	0213*	short breath
t10	White	1967	m	0213*	chest pain
t11	White	1967	m	0213*	chest pain

Figure 2 Example of *k*-anonymity, where *k*=2 and QI={*Race*, *Birth*, *Gender*, *ZIP*}

Example

Release Table

	Race	Birth	Gender	ZIP	Problem
t1	Black	1965	m	0214*	short breath
t2	Black	1965	m	0214*	chest pain
t3	Black	1965	f	0213*	hypertension
t4	Black	1965	f	0213*	hypertension
t5	Black	1964	f	0213*	obesity
t6	Black	1964	f	0213*	chest pain
t7	White	1964	m	0213*	chest pain
t8	White	1964	m	0213*	obesity
t9	White	1964	m	0213*	short breath
t10	White	1967	m	0213*	chest pain
t11	White	1967	m	0213*	chest pain

External Data Source

Name	Birth	Gender	ZIP	Race
Andre	1964	m	02135	White
Beth	1964	f	55410	Black
Carol	1964	f	90210	White
Dan	1967	m	02174	White
Ellen	1968	f	02237	White

Suppose you have a external data table.

By linking these 2 tables, you still don't know Andre's problem.

Anonymization models/algs

- ***BOTTOM –UP: Incognito*** computes a k-minimal generalization [LeFevre SIGMOD '05] : A-Priori like method.
  - Uses a bottom-up breadth-first search of the domain generalization hierarchy
  - For each iteration ***i*** checks if each subset of quasi-identifiers of size ***i*** satisfies the k-anonymity property
  - Removing all the generalizations that do not satisfy it
  - Generates all possible k-anonymization full-domain generalizations of a given table
- ***TOP-DOWN: k-Optimize,*** *Bayardo and Agrawal*
  - *Assumes an ordering on QI attributes and discretizes them*
  - *Generates a tree corrsponding to the all possible generalization hirarchy. Such alg is optima wrt a certain cost metric*

K-anonymity Vulnerability

- **k-anonymity** does not provide privacy if:
  - Sensitive values in an equivalence class lack diversity
  - The attacker has background knowledge
- This leads to the l-Diversity model:

Lack diversity

Bob	
Zipcode	***Age***
47678	27

Background Knowledge
(Carl's brother has heart disease)

Carl	
Zipcode	***Age***
47673	36

A 3-anonymous patient table

Zipcode	Age	Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
4790*	≥40	Flu
4790*	≥40	Heart Disease
4790*	≥40	Cancer
476**	3*	Heart Disease
476**	3*	Cancer
476**	3*	Cancer

l-Diversity

- Principle
  - Each equivalence class has at least *l* well-represented sensitive values
- Distinct *l*-diversity
  - Each equivalence class has at least *l* distinct sensitive values
  - Probabilistic inference

…	Disease
	…
	HIV
	HIV
	…
	HIV
	pneumonia
	bronchitis
	…

10 records

8 records have HIV

2 records have other values

Limitations of *l*-Diversity

l-Diversity is insufficient to prevent attribute disclosure.

Similarity Attack

Bob	
Zip	***Age***
47678	27

A 3-diverse patient table

Zipcode	Age	Salary	Disease
476**	2*	20K	Gastric Ulcer
476**	2*	30K	Gastritis
476**	2*	40K	Stomach Cancer
4790*	≥40	50K	Gastritis
4790*	≥40	100K	Flu
4790*	≥40	70K	Bronchitis
476**	3*	60K	Bronchitis
476**	3*	80K	Pneumonia
476**	3*	90K	Stomach Cancer

Conclusion

1. Bob's salary is in [20k,40k], which is relative low.
2. Bob has some stomach-related disease.

l-Diversity does not consider semantic meanings of sensitive values

t-Closeness: A New Privacy Measure

- Adversarial belief

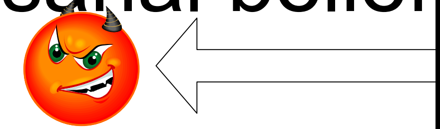
Belief	Knowledge
B_0	*External Knowledge*
B_1	Overall distribution Q of sensitive values

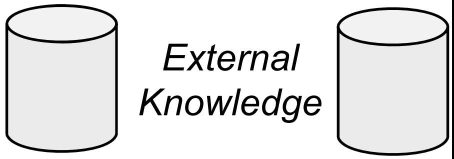
A completely generalized table

Age	Zipcode	……	Gender	Disease
*	*	……	*	Flu
*	*	……	*	Heart Disease
*	*	……	*	Cancer
.	.	……	.	.
.	.	……	.	.
.	.	……	.	.
*	*	……	*	Gastritis

t-Closeness: A New Privacy Measure

- Adversarial belief



Belief	Knowledge
B_0	
B_1	Overall distribution Q of sensitive values
B_2	Distribution P_i of sensitive values in each equi-class

A released table

Age	Zipcode	……	Gender	Disease
2*	479**	……	Male	Flu
2*	479**	……	Male	Heart Disease
2*	479**	……	Male	Cancer
. . .	. . .	…… …… ……	. . .	. . .
≥50	4766*	……	*	Gastritis

t-Closeness: A New Privacy Measure

- Adversarial belief

Belief	Knowledge
B_0	*External Knowledge*
B_1	Overall distribution Q of sensitive values
B_2	Distribution P_i of sensitive values in each equi-class

- Rationale
  - *Q* should be public information
  - Knowledge gain is separated:
    - About whole population (from B_0 to B_1)
    - About individuals (from B_1 to B_2)
  - We bound knowledge gain between B_1 and B_2
- Principle
  - The distance between *Q* and P_i is bounded by a threshold *t*.
  - *l*-diversity considers only P_i

Utility Measures

- Analysis dependent measures
  - **Query answering accuracy**: eg. How much aggregates such as SUM or COUNT differs from the computation on the original values
  - **Classification accuracy**: measuring the change of entropy during classification
  - **Distribution similarity**: how much the original distribution is preserved
- Data distortion measures
  - **Generalization height**: total number of generalization steps
  - **Discernability:minimizes** the dimension of avarage equivalence class: what is the effective minimal K introduced by the transformation

Pattern-Preserving k-Anonymization of sequences

Ruggero Pensa, Anna Monreale, Fabio Pinelli, Dino Pedreschi

ISTI-CNR & Computer Science Dept.

Pisa, Italy

Outline

- Motivations
- Analysis of Sequence Database and Privacy issue
- Our Framework
  - The problem
  - The Anonymization Algorithm
- Experiments on mobilty data
- Conclusions and Future Work

Motivation

- Availability of large amounts of sequential transaction data:
  - Web logs
  - GPS data
  - Clinical data
  - …
- An important and vital resource for an organization if
  - Processed
  - Analyzed
  - Transformed into information
- Many KDD (*Knowledge Discovery in Databases*) techniques to extract knowledge about citizens/users’ behavior

Privacy-Preserving Data Mining

- Data can contain personal sensitive information :
  - **Individual Privacy at risk**
- Need for new **privacy-preserving** data mining techniques
- Modifying the original data, so that
  - private data are protected
  - Analysis results are still useful
- Natural trade-off between privacy quantification and data utility

Analysis of Sequence Database

- Analysis of sequence data is a rising field in data mining
- User's actions stored with their timestamps
- **Spatio-temporal data** have a sequential nature
- Analyzing spatio-temporal data
  - Allows to extract sequential behavior of users
  - May reveal private information about a user
- Hiding personal identifiers may be insufficient
- Infrequent location sequences can be **harmful**

Mining Sequences - Example

Customer-sequence

CustId	Video sequence
1	{(C), (H)}
2	{(AB), (C), (DFG)}
3	{(CEG)}
4	{(C), (DG), (H)}
5	{(H)}

Sequential patterns with support > 0.25

{(C), (H)}

{(C), (DG)}

Formal Definition of a Subsequence

- A sequence $<a_1\ a_2 \ldots a_n>$ is contained in another sequence $<b_1\ b_2 \ldots b_m>$ ($m \geq n$) if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i1}$, …, $a_n \subseteq b_{in}$

Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {8} >	< {2} {3,5} >	Yes
< {1,2} {3,4} >	< {1} {2} >	No
< {2,4} {2,4} {2,5} >	< {2} {4} >	Yes

- The support of a subsequence w is defined as the fraction of data sequences that contain w
- A *sequential pattern* is a frequent subsequence (i.e., a subsequence whose support is ≥ *minsup*)

Sequential Frequent Patterns

Dataset: D

B C

A B C D

A B C D

B C E

B C D

Minimum support = 3 →

SFP (D): S

B

C

D

B C

B D

C D

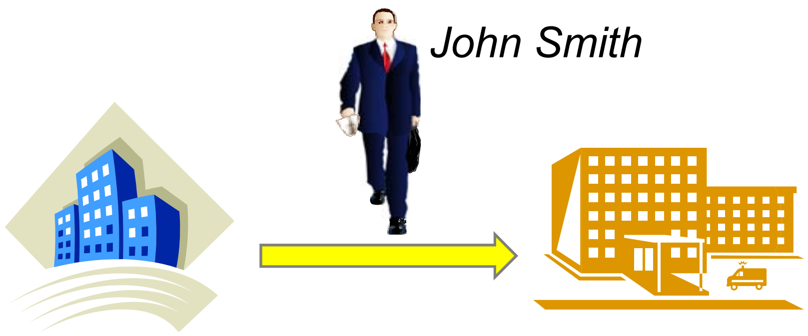
B C D

A : occurs only 2 times in D

C B: does not occur (order is important!)

Sequence Linking Attack

The Attacker knows:



*Commercial zone **A***

*Hospital **B***

+

***AB**CD*
CDAFB
FCHBL

A ⇒B

Infrequent Sequence in D

A subsequence can be both:

- ***Quasi-identifier***
- ***Private information***

***Attacker** can easily guess the sequence of locations crossed by Smith*

Countermeasure: k-Anonymous dataset

Definition 1 (**k-Harmful Sequence**). *Given a sequence dataset $\mathcal{D}$ and an anonymity threshold k, a sequence T is k-*harmful *(in $\mathcal{D}$) iff* $0 < supp_{\mathcal{D}}(T) < k$.

Definition 3 (**k-Anonymous Sequence Dataset**). *Given an anonymity threshold $k > 1$ and two sequence datasets $\mathcal{D}$ and $\mathcal{D}'$, we say that $\mathcal{D}'$ is a k-*anonymous version *of $\mathcal{D}$ iff each k-harmful sequence in $\mathcal{D}$ is not k-harmful in $\mathcal{D}'$.*

Dataset: D	***2-Anonymous: D'***
B C	*B C*
A B C D	*A B C D*
A B C D	*A B C D*
B C E	*B C*

Theorem 1. *Given a k-anonymous version $\mathcal{D}'$ of a sequence dataset $\mathcal{D}$, we have that, for any QI sequence T, $prob_{\mathcal{D}'}(T) \leq \frac{1}{k}$.*

Framework

- ***Anonymizes*** *dataset of sequences*

- *Preserves sequential pattern mining results*

- *Combines* ***k-anonymity*** *and* ***sequence hiding*** *methods*

- *Reformulates the anonymization problem as the problem of* ***hiding*** *k-harmful sequences*

Pattern-Preserving k-anonymization Problem

Definition 4 (optimal P2kA problem). *Given a sequence dataset $\mathcal{D}$, and an anonymity threshold $k > 1$, find a k-anonymous version $\mathcal{D}'$ of $\mathcal{D}$ such that the collection of all k-frequent patterns in $\mathcal{D}$ is preserved in $\mathcal{D}'$, i.e., the following two conditions hold:*

$$\mathcal{S}(\mathcal{D}', k) = \mathcal{S}(\mathcal{D}, k)$$
$$\forall T \in \mathcal{S}(\mathcal{D}', k) \; supp_{\mathcal{D}'}(T) = supp_{\mathcal{D}}(T).$$

Our approach assures:

- $\mathcal{D}'$ is k-anonymous
- $\mathcal{S}(\mathcal{D}', k) \subseteq \mathcal{S}(\mathcal{D}, k)$
- $\forall T \in \mathcal{S}(\mathcal{D}', k) \; supp_{\mathcal{D}'}(T) \simeq supp_{\mathcal{D}}(T)$

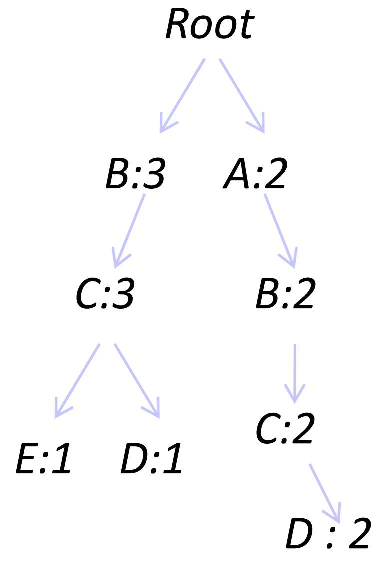
BF-2PkA Algorithm

- Based on a prefix-tree
- A 3-step approach
  - **Prefix Tree Construction**
  - **Prefix Tree Anonymization**
  - **Generation of anonymized sequences**

Running example: k = 2

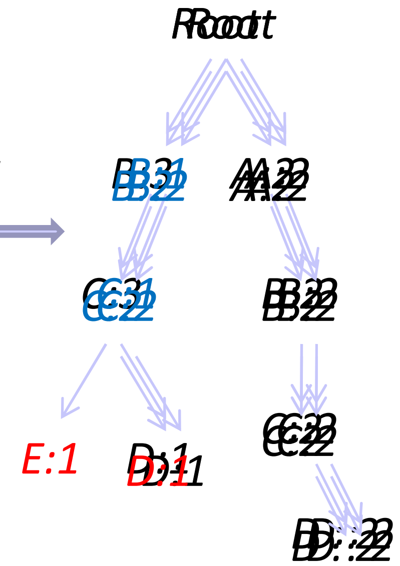
Dataset D
B C
A B C D
A B C D
B C E
B C D

Prefix Tree Construction



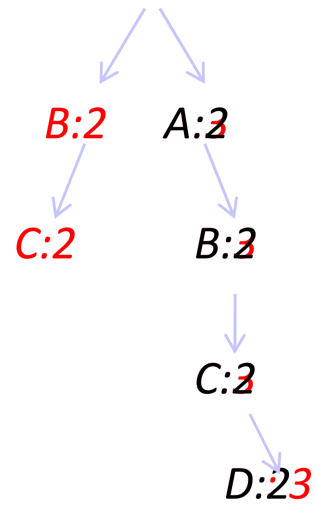
Tree Pruning

$\mathcal{L}_{cut}$
B C E : 1
B C D : 1



Tree Reconstruction

LCS:
1. *B C*
2. *B C D*

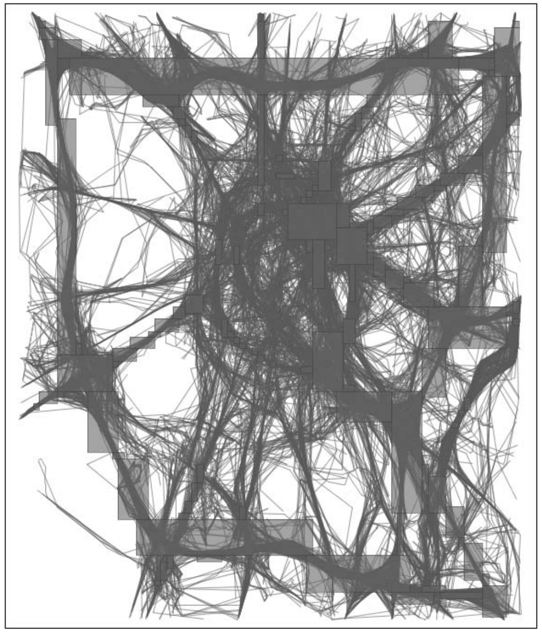


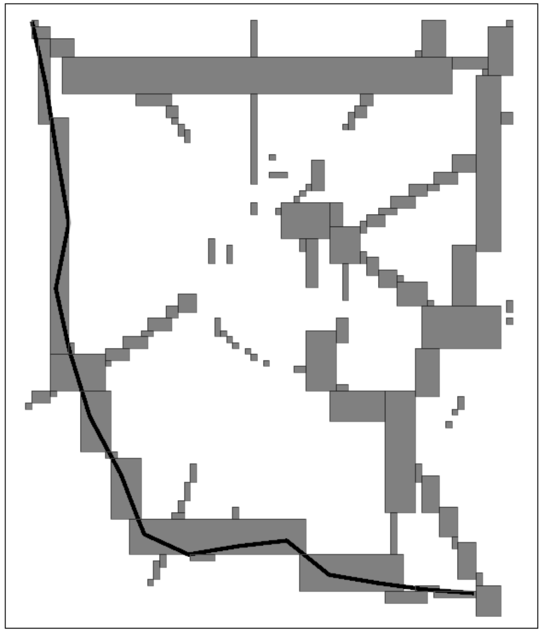
Generation of D'

Dataset D'
B C
A B C D
A B C D
B C
A B C D

Experiments on Mobility Data

- *Dataset of GPS trajectories of cars from the European project GeoPKDD (road network of Milan)*
- *Each trajectory is translated into a sequence of regions of interest*





Experiments: Similarity metrics

- Two metrics:
  - **SupSim:** measures the similarity of patterns in terms of support

$$SupSim = \frac{1}{\left|\hat{\mathcal{S}}(\sigma)\right|} \sum_{s \in \hat{\mathcal{S}}(\sigma)} \frac{\min\{supp_{\mathcal{D}'}(s), supp_{\mathcal{D}}(s)\}}{\max\{supp_{\mathcal{D}'}(s), supp_{\mathcal{D}}(s)\}}$$

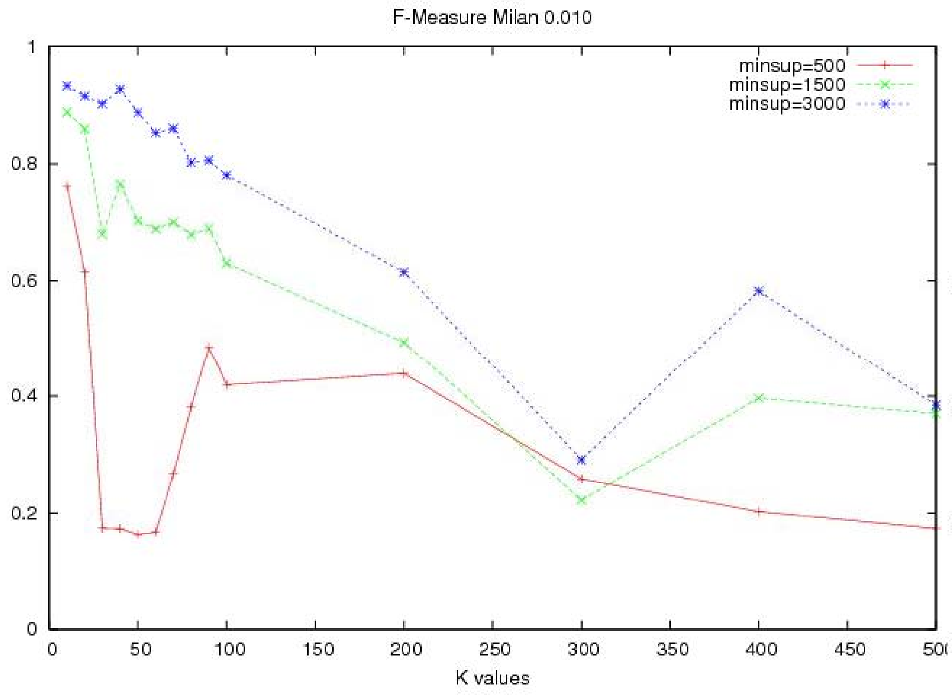
$$\hat{\mathcal{S}}(\sigma) \; = \; \mathcal{S}(\mathcal{D}', \sigma) \cap \mathcal{S}(\mathcal{D}, \sigma)$$

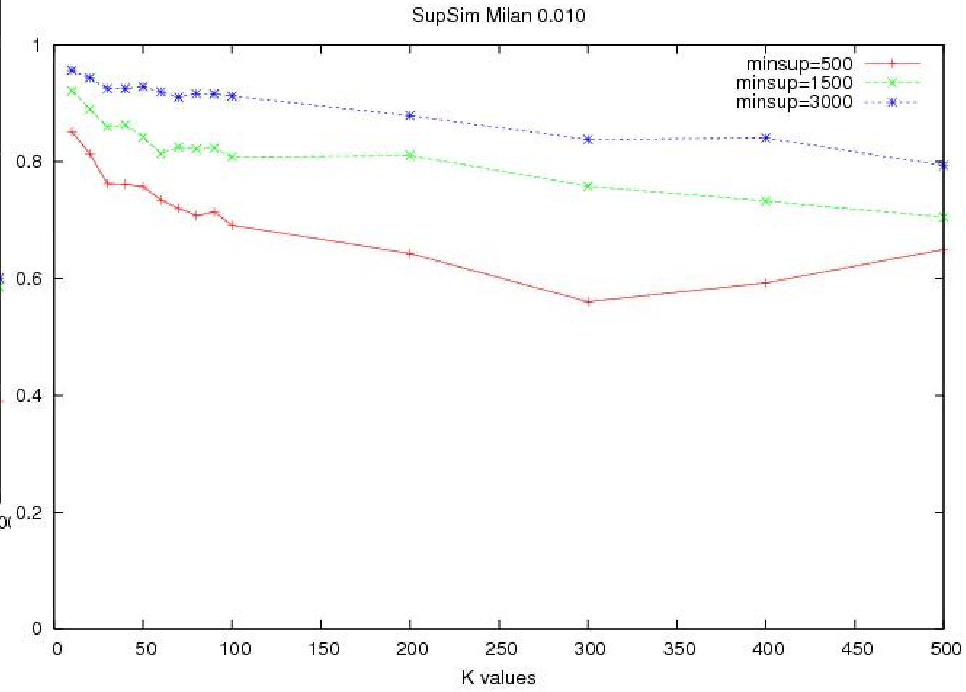
  - **F-Measure*:*** measures the similarity of patterns in terms of number of patterns

*F-Measure=2(Precision*Recall)/(Precision+Recall)*

Experiments: Sparse Data

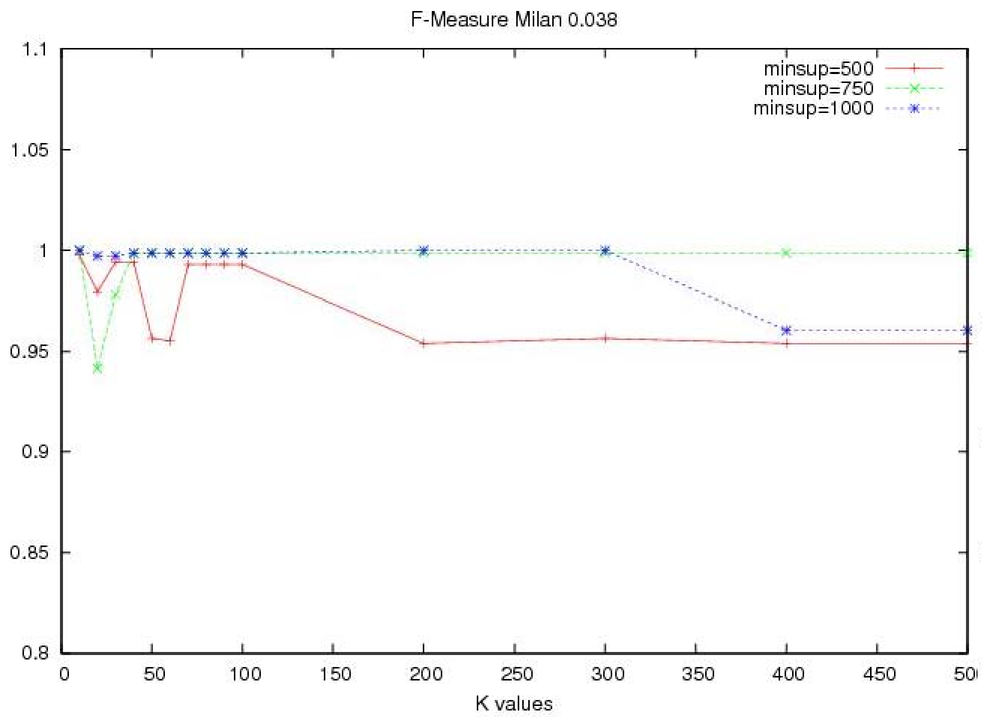
- The anonymization tends to prune more sequences
- Some frequent sequential patterns in ***D*** are missing in ***D'***

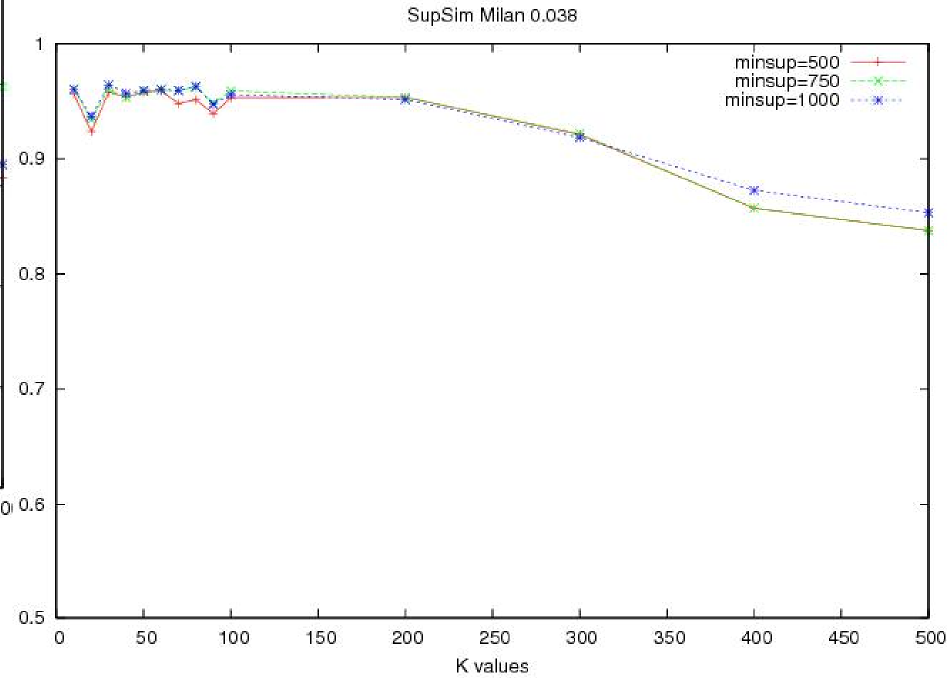




Experiments: higher density threshold

- The **collections** of patterns before and after the anonymization **are similar**





Privacy-preserving data publishing: Data Randomization, Perturbation and Obfuscation

Warner, S., Randomized response: a survey technique for eliminating evasive answer bias.

JASA, March 1965, 63-69.

RANDOMIZED RESPONSE: A SURVEY TECHNIQUE FOR ELIMINATING EVASIVE ANSWER BIAS

STANLEY L. WARNER
Claremont Graduate School

For various reasons individuals in a sample survey may prefer not to confide to the interviewer the correct answers to certain questions. In such cases the individuals may elect not to reply at all or to reply with incorrect answers. The resulting evasive answer bias is ordinarily difficult to assess. In this paper it is argued that such bias is potentially removable through allowing the interviewee to maintain privacy through the device of randomizing his response. A randomized response method for estimating a population proportion is presented as an example. Unbiased maximum likelihood estimates are obtained and their mean square errors are compared with the mean square errors of conventional estimates under various assumptions about the underlying population.

2. A RANDOM RESPONSE MODEL FOR PROPORTIONS

Suppose that every person in a population belongs to either Group A or Group B and it is required to estimate by survey the proportion belonging to Group A. A simple random sample of n people is drawn with replacement from the population and provisions made for each person to be interviewed. Before the interviews, each interviewer is furnished with an identical spinner with a face marked so that the spinner points to the letter A with probability p and to the letter B with probability $(1-p)$. Then, in each interview, the interviewee is asked to spin the spinner unobserved by the interviewer and report only whether or not the spinner points to the letter representing the group to which the interviewee belongs. That is, the interviewee is required only to say yes or no according to whether or not the spinner points to the correct group; he does not report the group to which the spinner points. Under the assumption that these yes and no reports are made truthfully, maximum likelihood estimates of the true population proportion are straightforward.

Let

π = the true probability of A in the population,

p = the probability that the spinner points to A, and

$$X_i = \begin{cases} 1 & \text{if the } i\text{th sample element says yes} \\ 0 & \text{if the } i\text{th sample element says no.} \end{cases}$$

Then

$$P(X_i = 1) = \pi p + (1 - \pi)(1 - p),$$

$$P(X_i = 0) = (1 - \pi)p + \pi(1 - p),$$

Estimating π

- $P(X=1) = \pi p + (1 - \pi)(1 - p)$
  - Solving for π
- $\pi = [P(X=1) - (1 - p)] \ / \ (2p - 1)$
  - $P(X=1)$ estimated by $n1/n$
- $\pi = [(n1 / n) - (1 - p)] \ / \ (2p - 1)$
- What happens with $p=1$?
- What happens with $p=1/2$?

and arranging the indexing of the sample so that the first n_1 report "yes" while the second $(n-n_1)$ report "no," the likelihood of the sample is

$$L = [\pi p + (1-\pi)(1-p)]^{n_1}[(1-\pi)p + \pi(1-p)]^{n-n_1}. \tag{1}$$

The log of the likelihood is

$$\begin{aligned} \log L = n_1 \log\,[\pi p + (1-\pi)(1-p)] \\ + (n-n_1)\log\,[(1-\pi)p + \pi(1-p)], \end{aligned} \tag{2}$$

and necessary conditions on π for a maximum are

$$\frac{(n-n_1)(2p-1)}{(1-\pi)p + \pi(1-p)} = \frac{n_1(2p-1)}{\pi p + (1-\pi)(1-p)}$$

or

$$\pi p + (1-\pi)(1-p) = \frac{n_1}{n}\cdot \tag{3}$$

Then, supposing $p \neq 1/2$, the maximum likelihood estimate of π is

$$\hat{\pi} = \frac{p-1}{2p-1} + \frac{n_1}{(2p-1)n} \cdot$$

TABLE 1. COMPARISON OF RANDOMIZED AND REGULAR ESTIMATES FOR TRUE PROBABILITY OF $A=.6$ AND $n=1000$

Regular Estimates			Mean Square Error Randomized / Mean Square Error Regular			
Probability of Truth T_a	Probability of Truth T_b	Bias	$p=.6$	$p=.7$	$p=.8$	$p=.9$
.95	1.00	−.03	5.45	1.36	.60	.33
.90	1.00	−.06	1.62	.40	.18	.10
.70	1.00	−.18	.19	.05	.02	.01
.50	1.00	−.30	.07	.02	.01	.00
1.00	.95	.02	9.82	2.44	1.08	.60
1.00	.90	.04	3.41	.85	.37	.21
1.00	.70	.12	.43	.11	.05	.03
1.00	.50	.20	.16	.04	.02	.01
.95	.95	−.01	18.25	4.54	2.00	1.11
.90	.90	−.02	9.70	2.41	1.06	.59
.70	.70	−.06	1.62	.40	.18	.10
.50	.50	−.10	.61	.15	.07	.04

Data Perturbation and Obfuscation

- What is disclosed?
  - the data (modified somehow)
- What is hidden?
  - the real data
- How?
  - by perturbating the data in such a way that it is not possible the identification of original database rows (individual privacy), but it is still possible to extract **valid** knowledge (models and patterns).

  - A.K.A. ***"distribution reconstruction"***

Data Perturbation and Obfuscation

- R. Agrawal and R. Srikant. Privacy-preserving data mining. In Proceedings of SIGMOD 2000.
- D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In Proceedings of PODS, 2001.
- W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In Proceedings of SIGKDD 2003.
- A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In Proceedings of PODS 2003.
- A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In Proceedings of SIGKDD 2002.
- K. Liu, H. Kargupta, and J. Ryan. Random Projection-based Multiplicative Perturbation for Privacy Preserving Distributed Data Mining. IEEE Transactions on Knowledge and Data Engineering (TKDE), VOL. 18, NO. 1.
- K. Liu, C. Giannella and H. Kargupta. An Attacker's View of Distance Preserving Maps for Privacy Preserving Data Mining. In Proceedings of PKDD'06

Data Perturbation and Obfuscation

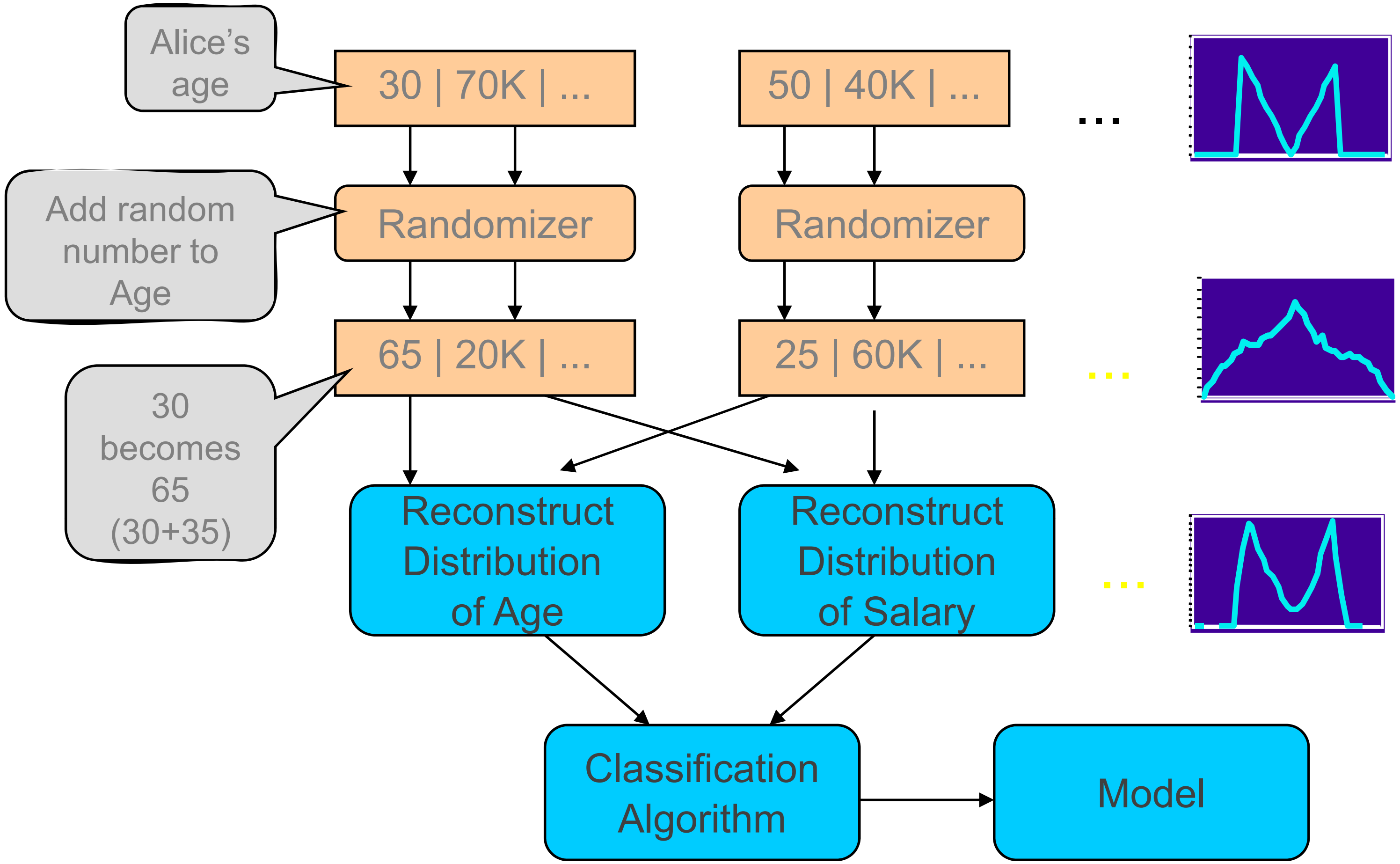
- This approach can be instantiated to association rules as follows:
  - *D* source database;
  - *R* a set of association rules that can be mined from *D;*

  - <u>Problem:</u> define two algorithms *P* and M_P such that
    - *P(D) = D'* where *D'* is a database that do not disclose any information on singular rows of *D*;
    - $M_P(D') = R$

Agrawal and Srikant '00

- Assume users are willing to
  - Give true values of certain fields
  - Give modified values of certain fields
- Practicality
  - 17% refuse to provide data at all
  - 56% are willing, as long as privacy is maintained
  - 27% are willing, with mild concern about privacy
- Perturb Data with Value Distortion
  - User provides x_i+r instead of x_i
  - r is a random value
    - Uniform, uniform distribution between $[-\alpha, \alpha]$
    - Gaussian, normal distribution with $\mu = 0, \sigma$

Randomization Approach Overview



Preserving Data Privacy (1)

- **Value-Class Membership**
  - *Discretization*: values for an attribute are discretized into intervals
    - Intervals need not be of equal width.
    - Use the interval covering the data in computation, rather than the data itself.
  - Example:
    - Perhaps Adam doesn't want people to know he makes $4000/year.
      - Maybe he's more comfortable saying he makes between $0 - $20,000 per year.
  - The most often used method for hiding individual values.

Preserving Data Privacy (2)

- **Value Distortion**
  - Instead of using the actual data x_i
  - Use $x_i + r$, where r is a random value from a distribution.
    - **Uniform Distribution**
      - r is uniformly distributed between $[-\alpha, +\alpha]$
      - Average r is 0.
    - **Gaussian Distribution**
      - r has a normal distribution
      - Mean $\mu(r)$ is 0.
      - *Standard_deviation(r)* is σ

What do we mean by "private?"

W = width of intervals in discretization

	Confidence		
	50%	95%	99.9%
Discretization	$0.5 \times W$	$0.95 \times W$	$0.999 \times W$
Uniform	$0.5 \times 2\alpha$	$0.95 \times 2\alpha$	$0.999 \times 2\alpha$
Gaussian	$1.34 \times \sigma$	$3.92 \times \sigma$	$6.8 \times \sigma$

Table 1: Privacy Metrics

- If we can estimate with *c%* confidence
  - The value *x* lies within the interval $[x_1, x_2]$
  - *Privacy* = $(x_2 - x_1)$, the size of the range.
- If we want very high privacy
  - $2\alpha > W$
  - Value distortion methods (Uniform, Gaussian) provide more privacy than discretization at higher confidence levels.

Reconstructing Original Distribution From Distorted Values (1)

- Original data values: $x_1, x_2, \ldots, x_n$
- Random variable distortion: $y_1, y_2, \ldots, y_n$
- Distorted samples: $x_1+y_1, x_2+y_2, \ldots, x_n+y_n$
- F_Y: The Cumulative Distribution Function (CDF) of random distortion variables y_i
- F_X: The CDF of original data values x_i

Reconstructing Original Distribution From Distorted Values (2)

- **The Reconstruction Problem**
  - Given
    - F_Y
    - distorted samples $(x_1+y_1, \ldots, x_n+y_n)$
  - Estimate F_X

Reconstruction Algorithm (1)

(1) $f_X^0 :=$ Uniform distribution
(2) $j := 0$ // Iteration number
repeat
(3) $$f_X^{j+1}(a) := \frac{1}{n} \sum_{i=1}^{n} \frac{f_Y(w_i - a)\, f_X^j(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)\, f_X^j(z)\, dz}$$
(4) $j := j + 1$
until (stopping criterion met)

<u>How it works</u> (incremental refinement of F_X) :

1. *The f(x, 0) initialized to uniform distribution*
2. *For j=0 until stopping, do*
3. *Find f(x, j+1) as a function of f(x, j) and F_Y*
4. *When loop stops, f(x) estimates F_X*

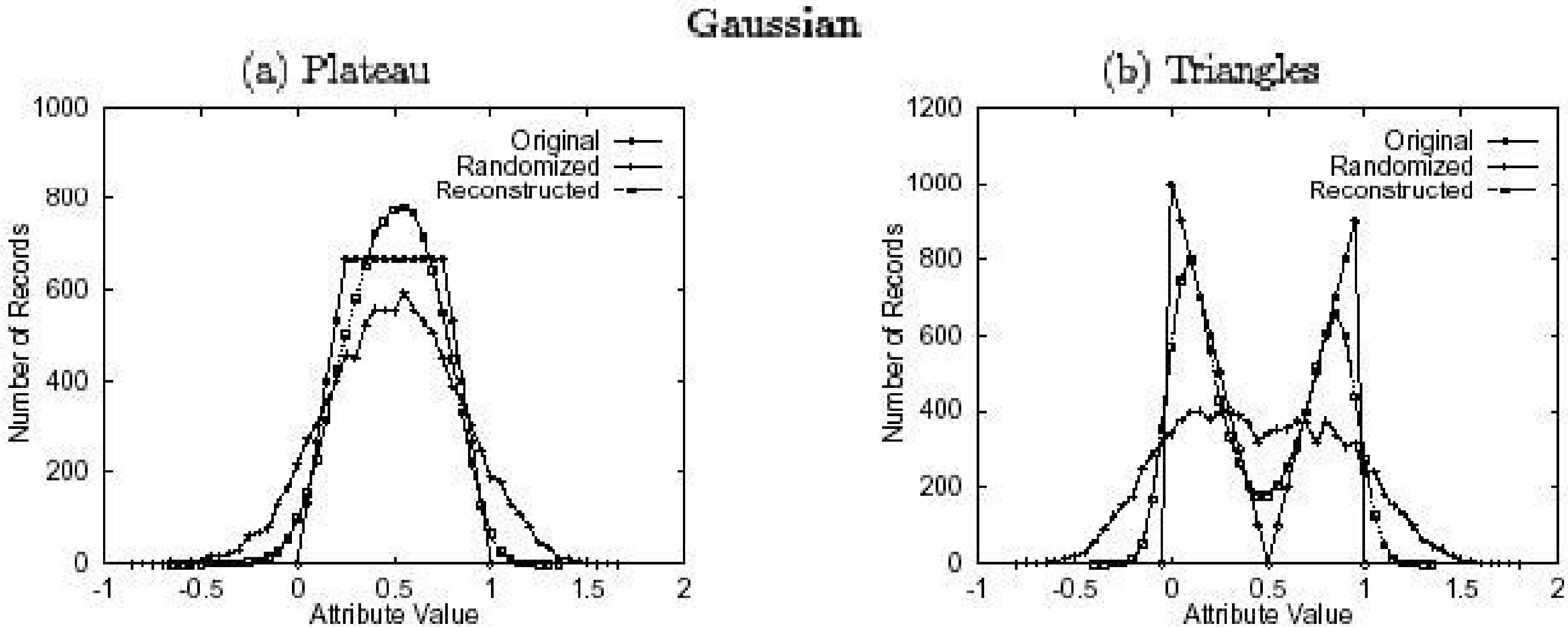
Reconstruction Algorithm (2)

(1) $f_X^0 :=$ Uniform distribution
(2) $j := 0$ // Iteration number
repeat
(3) $f_X^{j+1}(a) := \frac{1}{n} \sum_{i=1}^{n} \frac{f_Y(w_i - a) f_X^j(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z) f_X^j(z) dz}$
(4) $j := j + 1$
until (stopping criterion met)

Stopping Criterion

- *Compare successive estimates f(x, j).*
- *Stop when difference between successive estimates very small.*

Distribution Reconstruction Results (1)

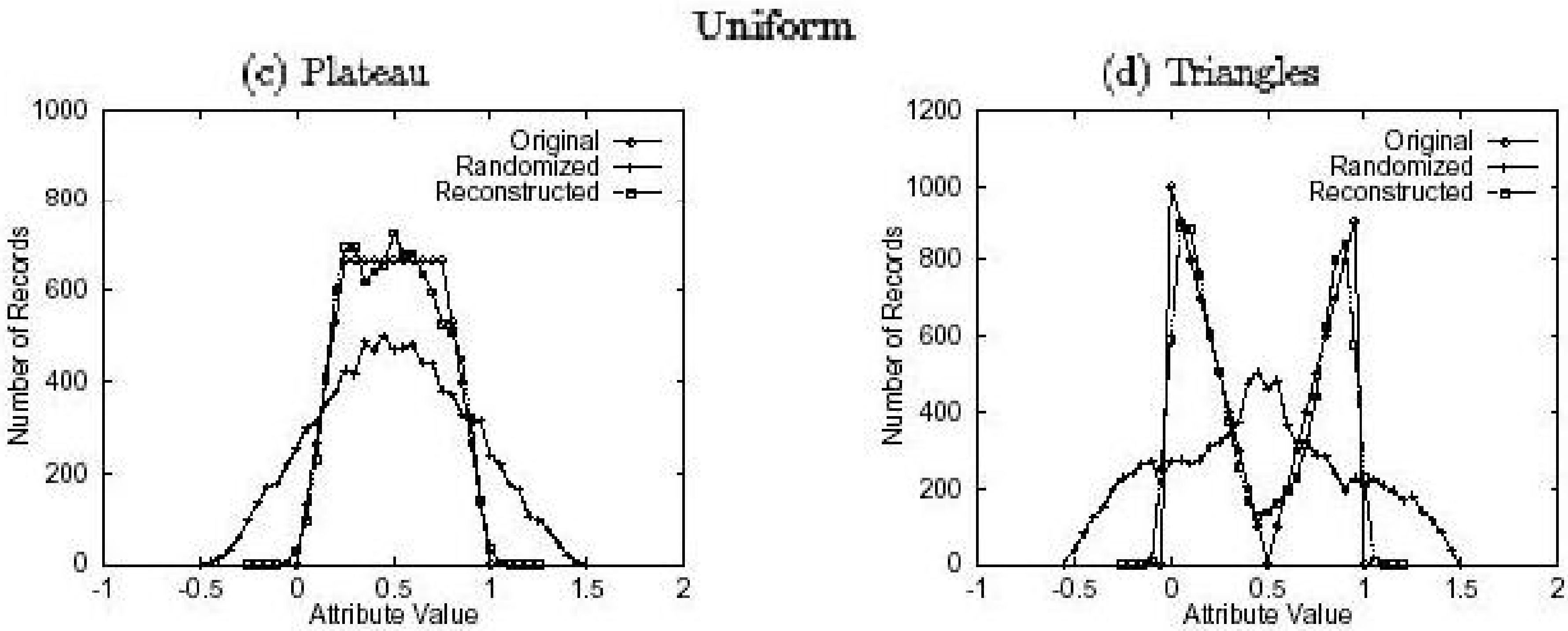


Original = original distribution

Randomized = effect of randomization on original dist.

Reconstructed = reconstructed distribution

Distribution Reconstruction Results (2)



Original = original distribution

Randomized = effect of randomization on original dist.

Reconstructed = reconstructed distribution

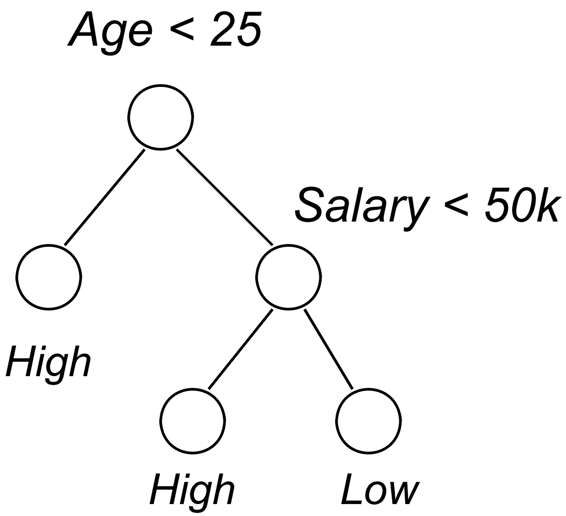
Summary of Reconstruction Experiments

- Authors are able to reconstruct
  - Original shape of data
  - Almost same aggregate distribution

- This can be done even when randomized data distribution looks nothing like the original.

Decision-Tree Classifiers w/ Perturbed Data

CREDIT RISK



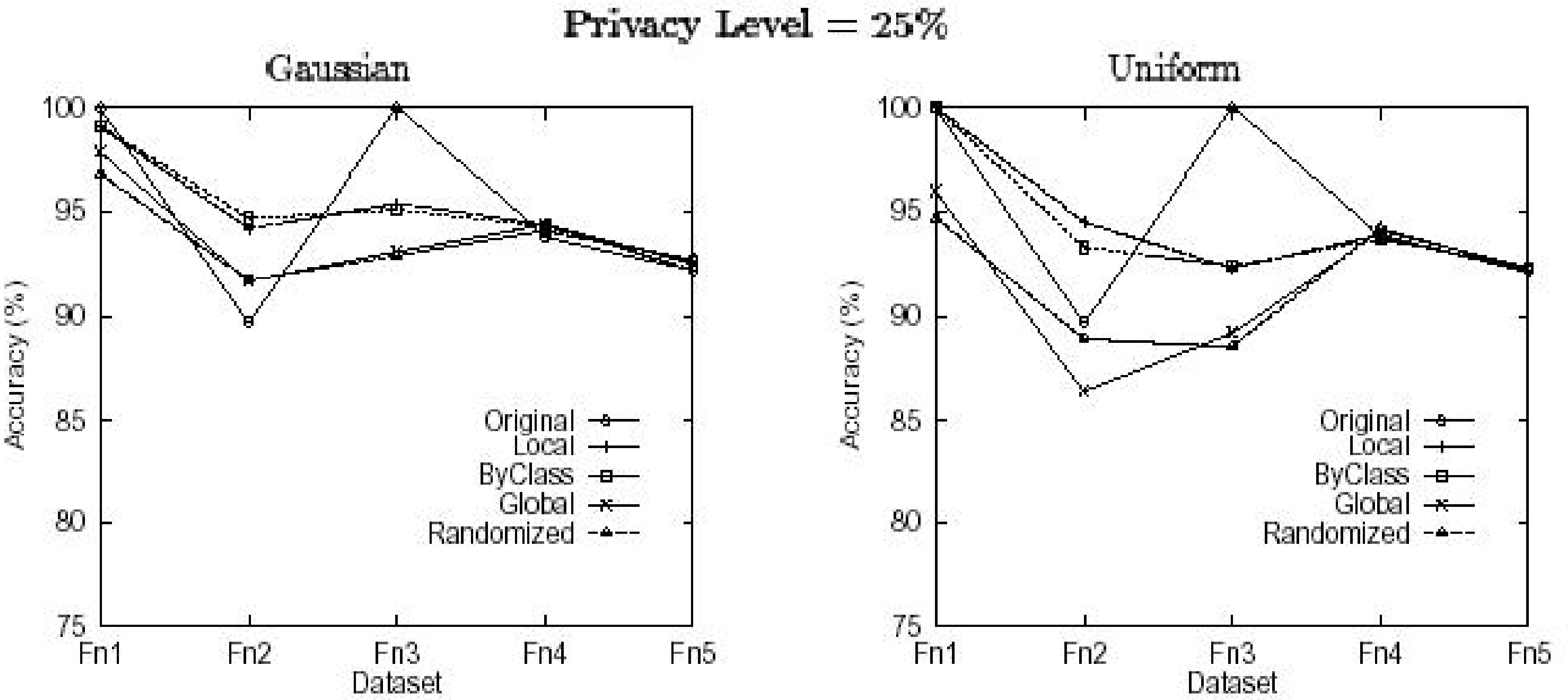
When/how to recover original distributions in order to build tree?

- ***Global -*** *for each attribute, reconstruct original distribution before building tree*
- ***ByClass –*** *for each attribute, split the training data into classes, and reconstruct distributions separately for each class; then build tree*
- ***Local –*** *like* ***ByClass****, reconstruct distribution separately for each class, but do this reconstruction while building decision tree*

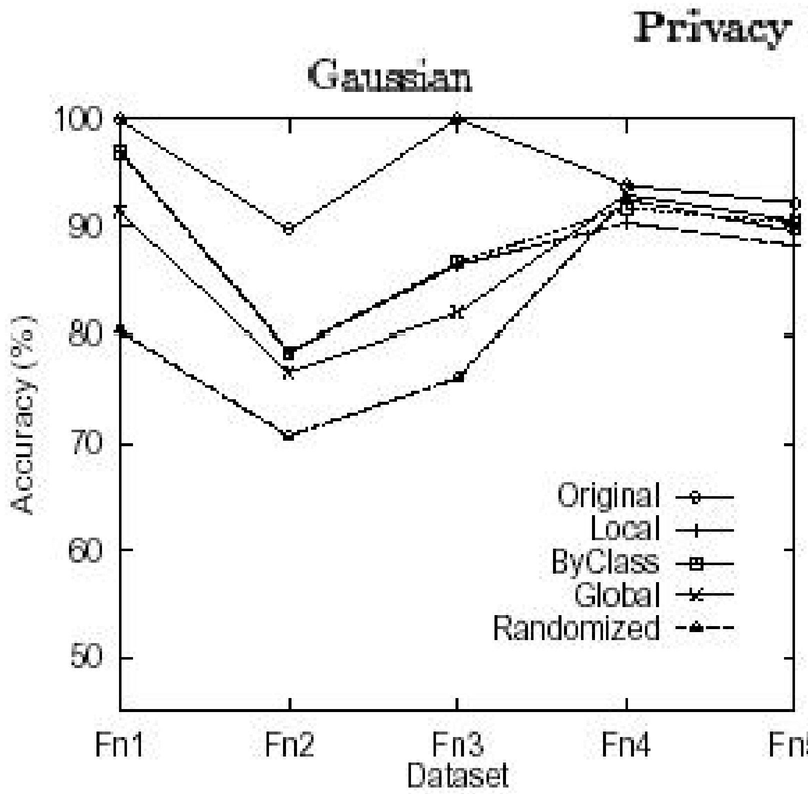
Experimental Results – Classification w/ Perturbed Data

- Compare **Global**, **ByClass**, **Local** algorithms against control series:
  - **Original –** result of classification of unperturbed training data
  - **Randomized –** result of classification on perturbed data with no correction
  -
- Run on five classification functions Fn1 through Fn5. (classify data into groups based on attributes)

Results – Classification Accuracy (1)



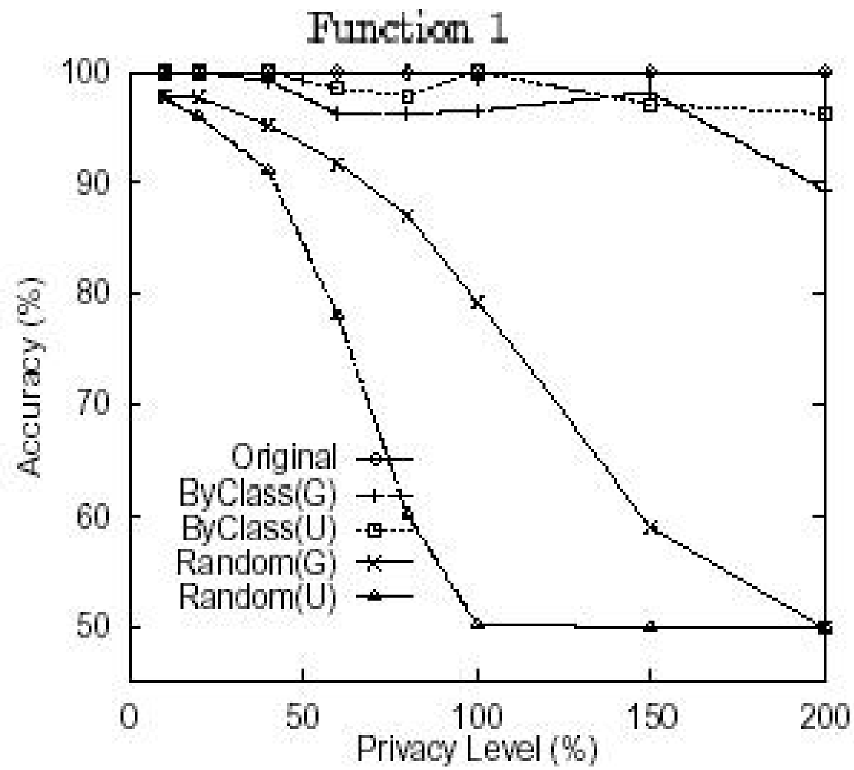
Results – Classification Accuracy (2)

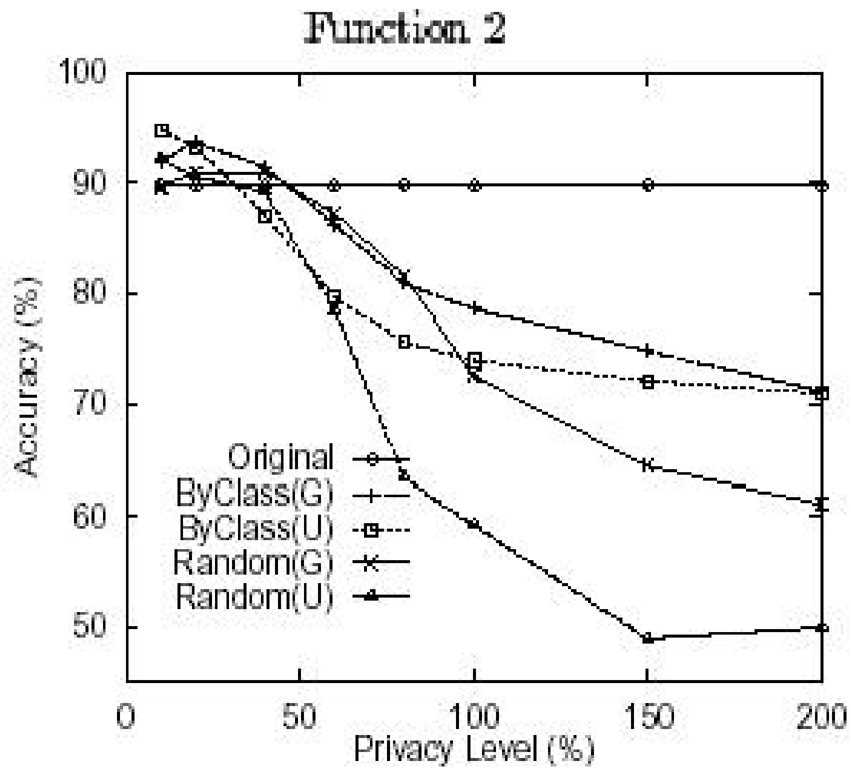


Experimental Results – Varying Privacy

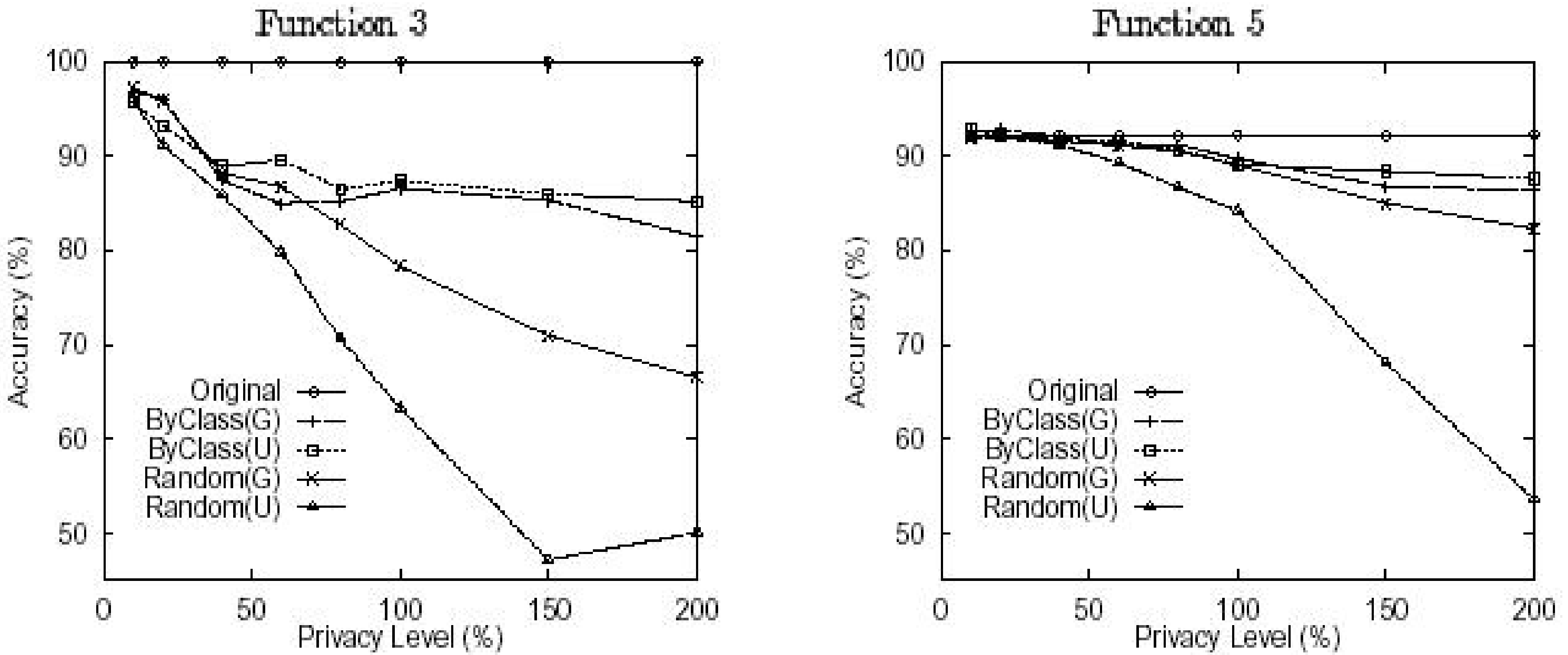
- Using **ByClass** algorithm on each classification function (except Fn4)
  - Vary privacy level from 10% - 200%
  - Show
    - Original – unperturbed data
    - ByClass(G) – ByClass with Gaussian perturbation
    - ByClass(U) – ByClass with Uniform perturbation
    - Random(G) – uncorrected data with Gaussian perturbation
    - Random(U) – uncorrected data with Uniform perturbation

Results – Accuracy vs. Privacy (1)





Results – Accuracy vs. Privacy (2)



Note: *Function 4 skipped because almost same results as Function 5.*

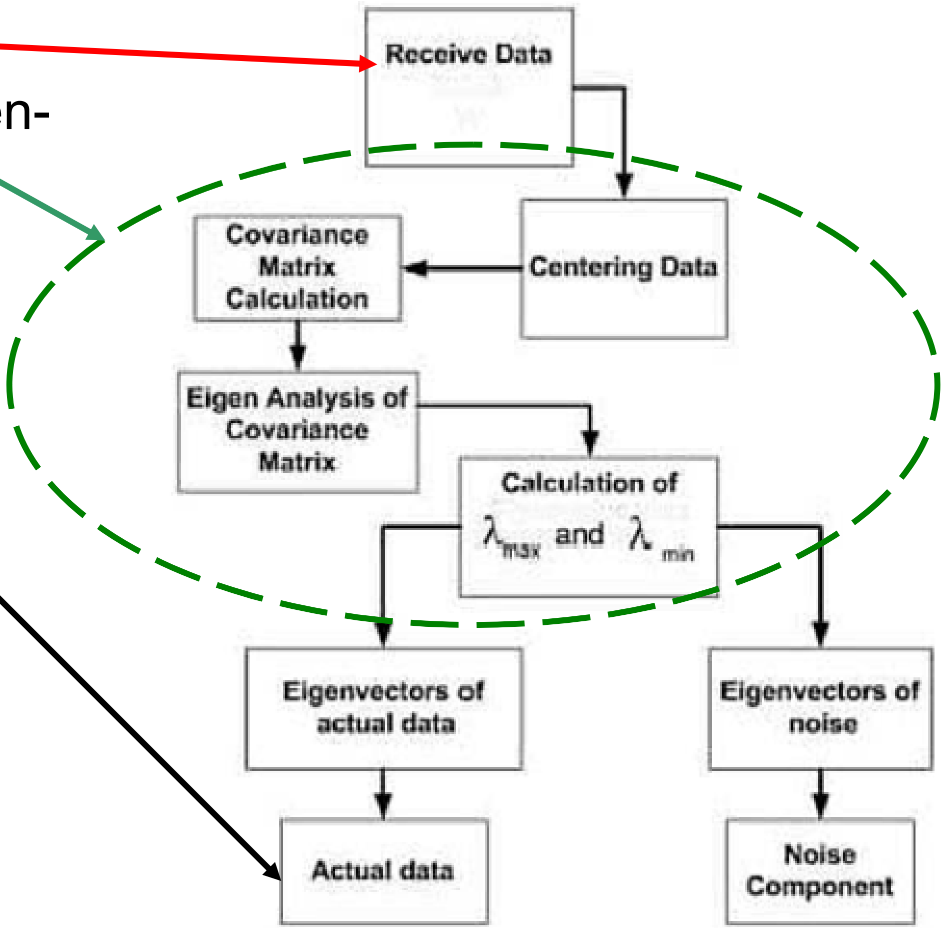
Vulnerability

- In many cases, the original data can be accurately estimated from the perturbed data using **spectral filter** designed based on **random matrix**

- **Main Idea**: Use eigen-values properties of noise to filter

Spectral Filtering

- U+V data
- Decomposition of eigen-values of noise and original data
- Recovered data



Decomposing eigen-values: separating data from noise (1)

Let U and V be the m x n data and noise matrices

P the perturbed matrix $U_P = U+V$

Covariance matrix of

$U_P = U_P{}^T U_P = (U+V)^T (U+V) = U^T U + V^T U + U^T V + U^T U$

Since signal and noise are uncorrelated in random perturbation, for large no. of observations: $V^T U \sim 0$ and $U^T V \sim 0$, therefore

$$U_P{}^T U_P = U^T U + V^T V$$

Since the above 3 matrices are correlation matrices, they are ***symmetric and positive semi-definite****, therefore, we can perform eigen decomposition:*

$$U^T U = Q_u \Lambda_u Q_u^T,$$
$$U_p^T U_p = Q_p \Lambda_p Q_p^T, \text{ and}$$
$$V^T V = Q_v \Lambda_v Q_v^T,$$

Decomposing eigen-values: separating data from noise (2)

$$\Lambda_p \approx \Lambda_u + \Lambda_v.$$

Wigner's law: *Describes distribution of eigen values for normal random matrices:*

- *eigen values for noise component V stick in a thin range given by* $\boldsymbol{\lambda}_{min}$ *and* $\boldsymbol{\lambda}_{max}$ ***(show example next page)*** *with high probability.*
- *Allows us to compute* $\boldsymbol{\lambda}_{min}$ *and* $\boldsymbol{\lambda}_{max}$*.*

Giving us the following algorithm:

Solution!

1. *Find a large no. of eigen values of the perturbed data P.*
2. *Separate all eigen values inside* $\boldsymbol{\lambda}_{min}$ *and* $\boldsymbol{\lambda}_{max}$ *and save row indices* I_V
3. *Take the remaining eigen indices to get the "peturbed" but not noise eigens coming from true data U: save their row indices* I_U
4. *Break perturbed eigenvector matrix* Q_P *into* $A_U = Q_P(I_U)$, $A_V = Q_P(I_V)$.
5. *Estimate true data as projection :* $\hat{U} = U_p A_u A_u^T.$

Related Work: Statistical Databases

- **Data Perturbation:**
  - replace the original database by a sample from the same distribution (e.g. [LST83][LCL85][Rei84])
  - sample the result of a query (e.g. [Den80])
  - swap values between records (e.g. [Den82])
  - add noise to the query result (e.g. [Bec80])
  - add noise to the values (e.g. [TYW84][War65])
- **Synthetic Techniques:**
  - Full Synthetic: generate a dataset that is completely new
  - Partially synthetic: produce a dataset, where the original data and synthetic data are mixed.
  - Synthetic and Original data have the same analitical properties

Privacy-aware Knowledge Publishing

The Purpose

- We want to publish data mining results
- We DON'T want to release information related to few people, that can help to trace single individuals
- We don't want to specify any other information

Privacy-aware Knowledge Sharing

- What is disclosed?
  - the intentional knowledge (i.e. rules/patterns/models)
- What is hidden?
  - the data source
- The central question:

  "do the data mining results themselves violate privacy?"

- Focus on **individual privacy**: the individuals whose data are stored in the source database being mined.

Privacy-aware Knowledge Sharing

- M. Kantarcioglu, J. Jin, and C. Clifton. When do data mining results violate privacy? In Proceedings of the tenth ACM SIGKDD, 2004.
- S. R. M. Oliveira, O. R. Zaiane, and Y. Saygin. Secure association rule sharing. In Proc.of the 8th PAKDD, 2004.
- P. Fule and J. F. Roddick. Detecting privacy and ethical sensitivity in data mining results. In Proc. of the 27° conference on Australasian computer science, 2004.
- Atzori, Bonchi, Giannotti, Pedreschi. K-anonymous patterns. In PKDD and ICDM 2005, The VLDB Journal (accepted for publication).
- A. Friedman, A. Schuster and R. Wolff. *k*-Anonymous Decision Tree Induction. In Proc. of PKDD 2006.

An Example in Medical Domain

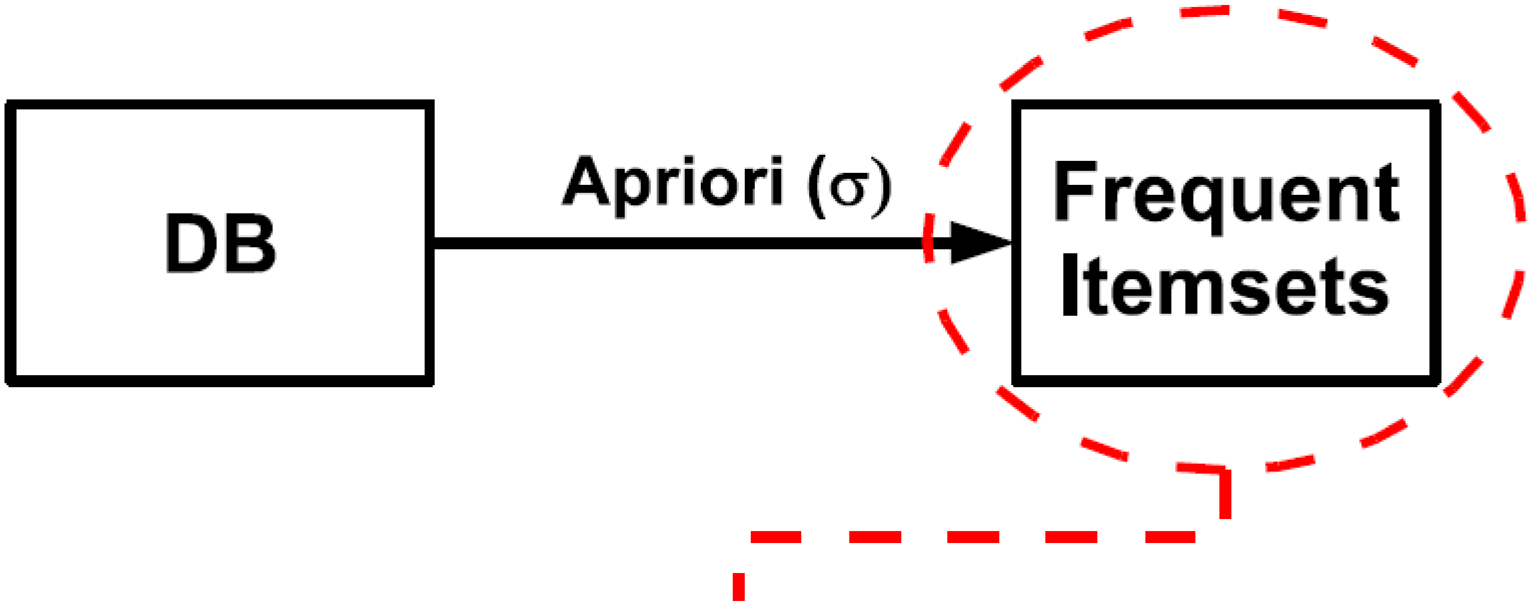
Example

- Suppose Dr. Gregory House conduces both usual hospital activities and research
- He has a big database with all sensitive information about his patients
- Playing with Data Mining, he discovered interesting trends about patologies in his patient data

Question

Can Dr. House publish his discoveries to third persons without offending the privacy of his patients?

An Example in Medical Domain



Does this set of itemsets violate the anonymity of individuals in DB?

Privacy-aware Knowledge Sharing

- Association Rules can be dangerous…

Example

$$a_1 \wedge a_2 \wedge a_3 \Rightarrow a_4 \quad [sup = 80,\ conf = 98.7\%]$$

$$sup(\{a_1, a_2, a_3\}) = \frac{sup(\{a_1, a_2, a_3, a_4\})}{conf} \approx \frac{80}{0.987} = 81.05$$

In other words, we know that there is just one individual for which the pattern $a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4$ holds.

- How to solve this kind of problems?

Now we know that ….

Fact

- *Even if we mine with a high support value, we can infer patterns holding in the original database which are not intentionally released*
- *They can regards very few individuals*
- *The support value of such patterns can be inferred without accessing the database*

What is a k-anonymous pattern?

Definition (Anonymous Pattern)

Given a database $\mathcal{D}$ and an anonymity threshold k, a pattern p is said to be k*-anonymous* if $sup_{\mathcal{D}}(p) \geq k$ or $sup_{\mathcal{D}}(p) = 0$.

Definition (Inference Channel)

An Inference Channel is any set of itemsets from which it is possible to infer that a pattern p is not k-anonymous.

We are interested in inference channels that are made of frequent itemsets.

Example

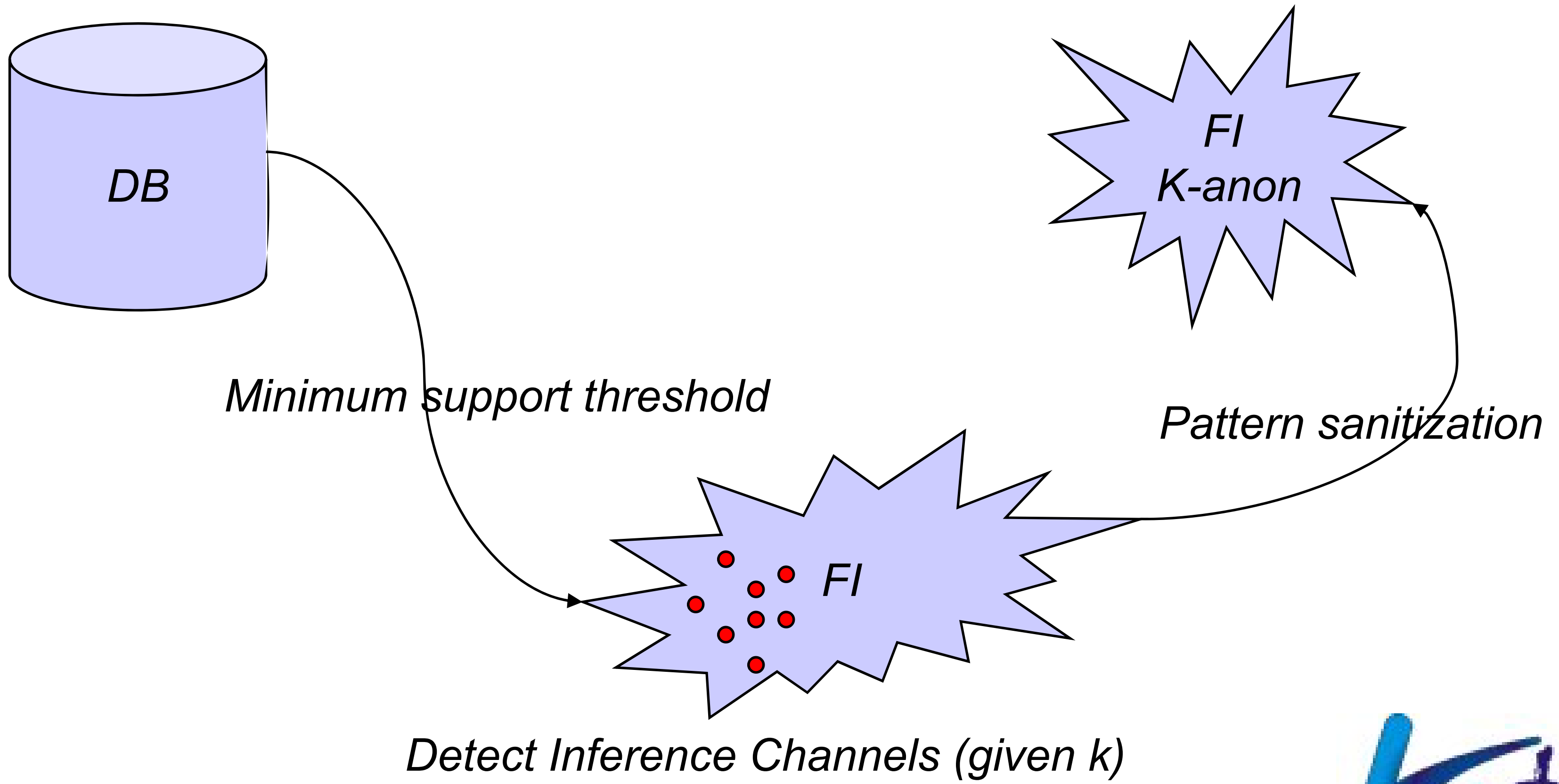
T1	a	b	c	d	e	f	g	h
T2	a	b	c	d	e		g	
T3	a	b	c	d	e			
T4	a	b	c	d	e	f	g	
T5	a	b	c	d	e			
T6	a	b	c	d	e			
T7	a	b		d	e			
T8	a				e	f	g	
T9			c	d	e	f	g	
T10			c	d	e			
T11			c	d	e	f	g	h
T12	a	b				f	g	

$p = a \wedge b \wedge \neg c \wedge \neg d \wedge \neg e$

$I = ab$

$J = abcde$

The scenario



Reduce the number of Patterns to check

Theorem

$\forall p \in \mathcal{P}at(\mathcal{I}) : 0 < sup_{\mathcal{D}}(p) < k \;.\; \exists I \subseteq J \in 2^{\mathcal{I}} : \mathcal{C}_I^J.$

- Translation: we can prune the search space by looking for Inference Channels regarding only conjunctive patterns.
- This property makes possible to have a (Naïve) Inference Channel Detector Algorithm

Distributed Privacy Preserving Data Mining

Distributed Privacy Preserving Data Mining

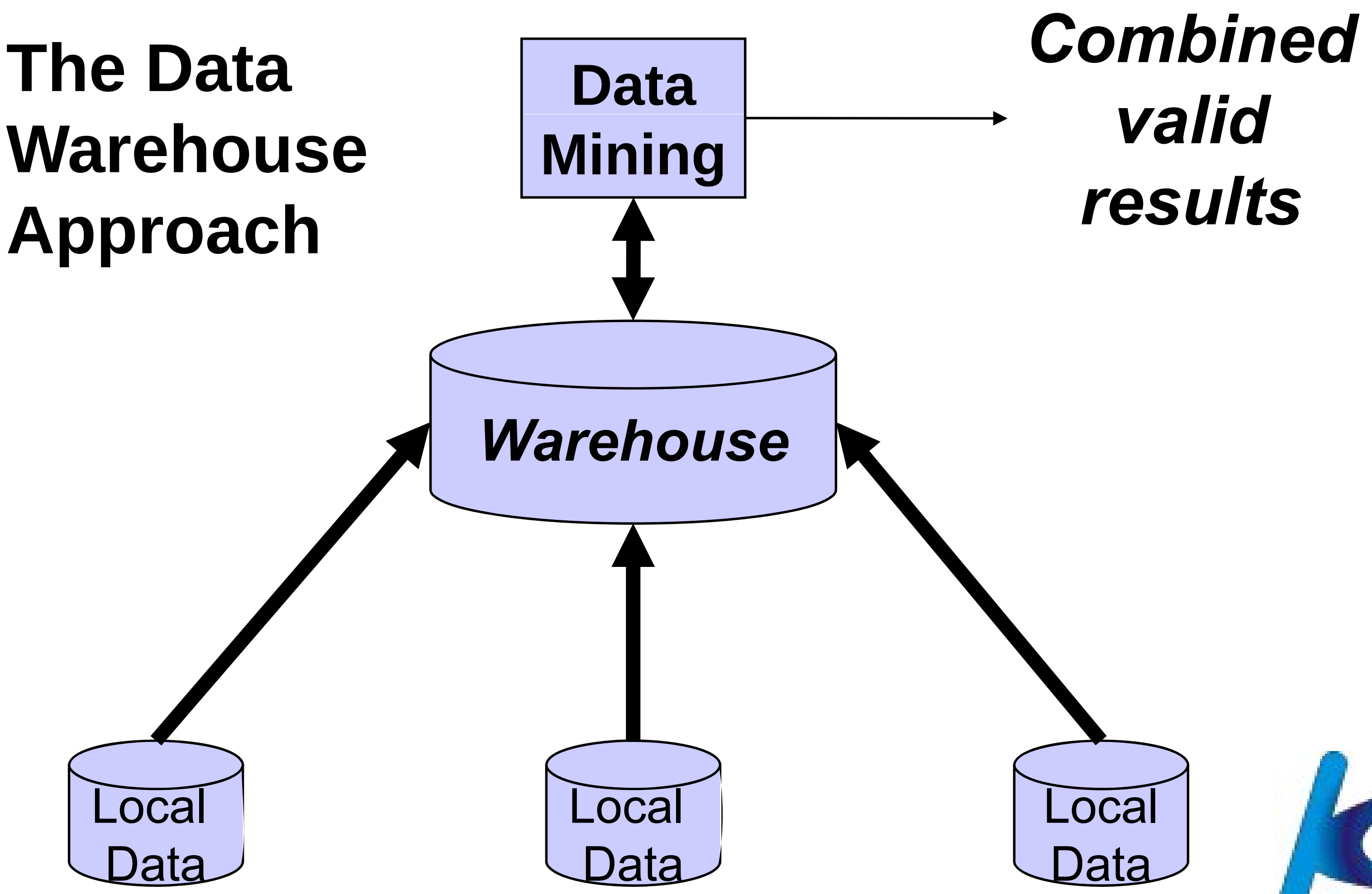
- Objective?
  - computing a valid mining model from several distributed datasets, where each party owing a dataset does not communicate its data to the other parties involved in the computation.
- How?
  - cryptographic techniques

- A.K.A. "*Secure Multiparty Computation*"

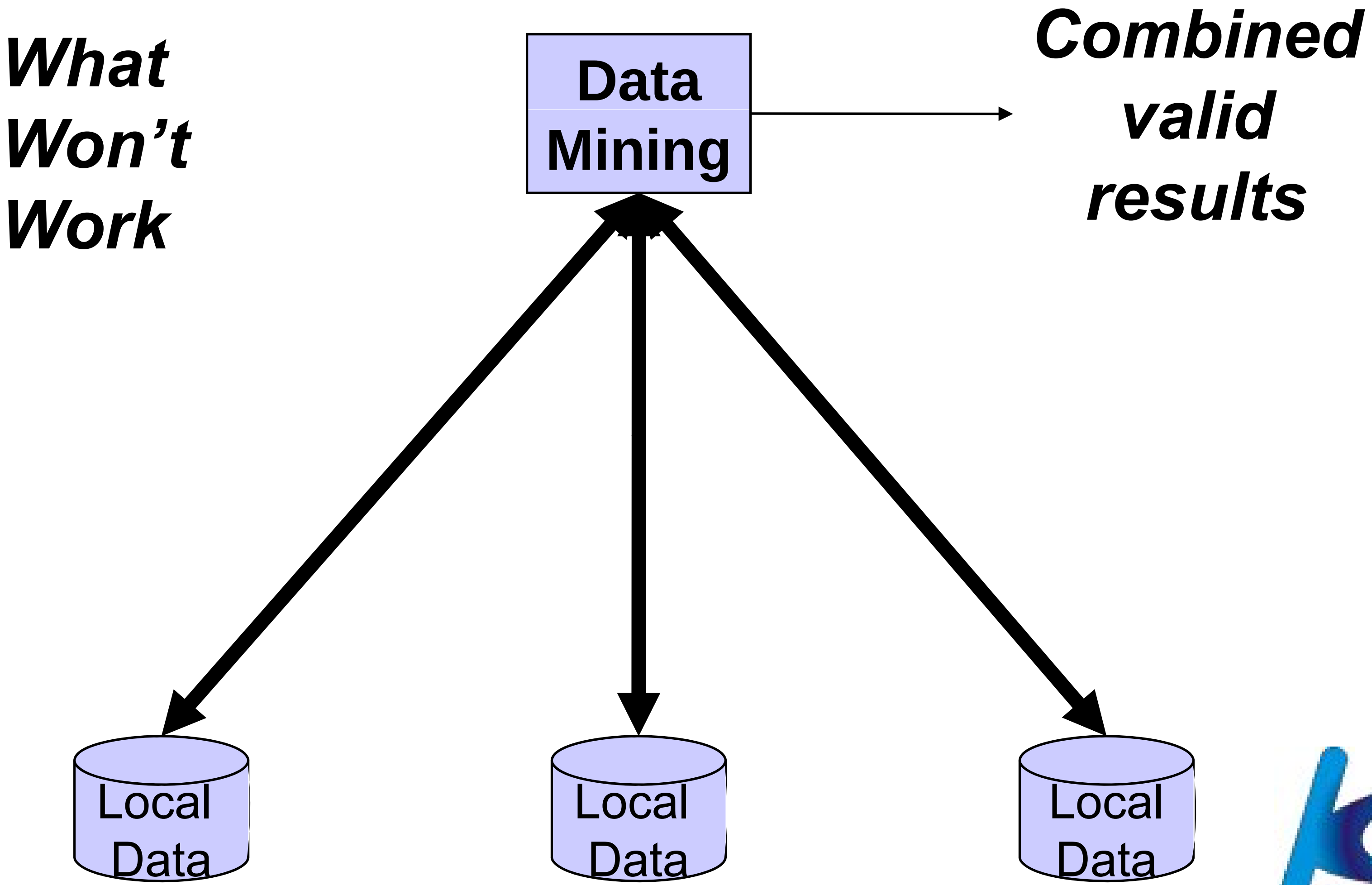
Distributed Privacy Preserving Data Mining

- C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y.Zhu. Tools for privacy preserving distributed data mining. SIGKDD Explor. Newsl., 4(2), 2002.
- M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02), 2002.
- B. Pinkas. Cryptographic techniques for privacy-preserving data mining. SIGKDD Explor. Newsl., 4(2), 2002.
- J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In Proceedings of ACM SIGKDD 2002.

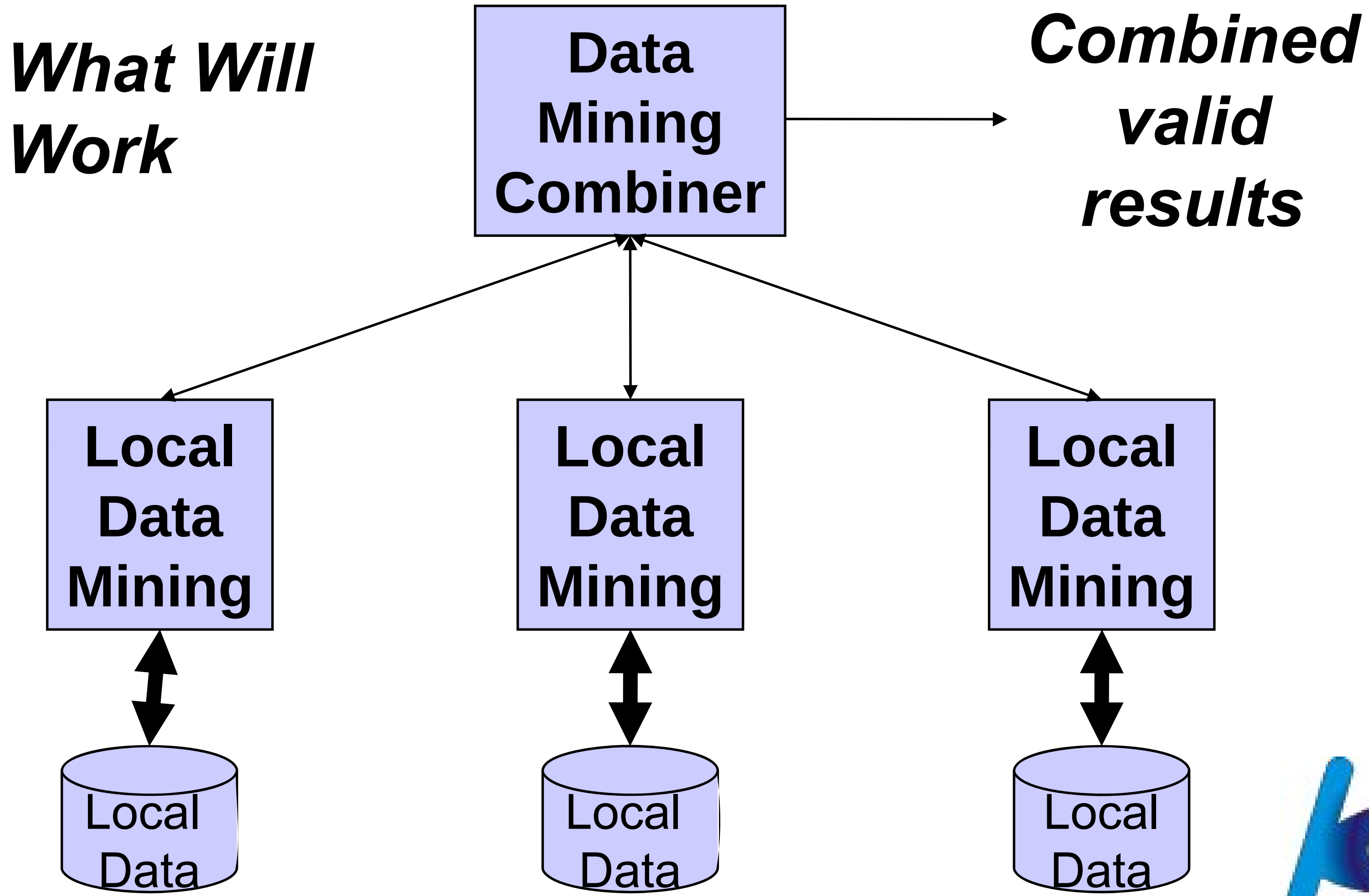
Distributed Data Mining: The "Standard" Method



Private Distributed Mining: What is it?



Private Distributed Mining: What is it?



Distributed Privacy Preserving Data Mining

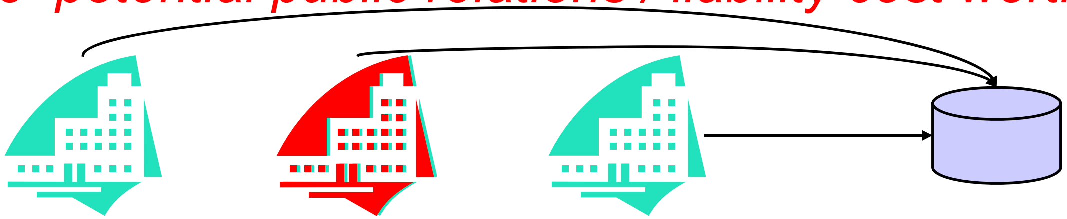
- This approach can be instantiated to association rules in two different ways corresponding to two different data partitions: vertically and horizontally partitioned data.

1. Each site *s* holds a portion *Is* of the whole vocabulary of items *I*, and thus each itemset is split between different sites. In such situation, the key element for computing the support of an itemset is the“secure” scalar product of vectors representing the subitemsets in the parties.

2. The transactions of *D* are partitioned in *n* databases *D1, . . . ,Dn*, each one owned by a different site involved in the computation. In such situation, the key elements for computing the support of itemsets are the “secure”union and “secure” sum operations.

Association Rule Mining: Horizontal Partitioning

- Distributed Association Rule Mining: Easy without sharing the individual data *[Cheung+'96]* *(Exchanging support counts is enough)*
- 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
    - *Is the potential public relations / liability cost worth it?*



Overview of the Method *(Kantarcioglu and Clifton '02)*

- 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

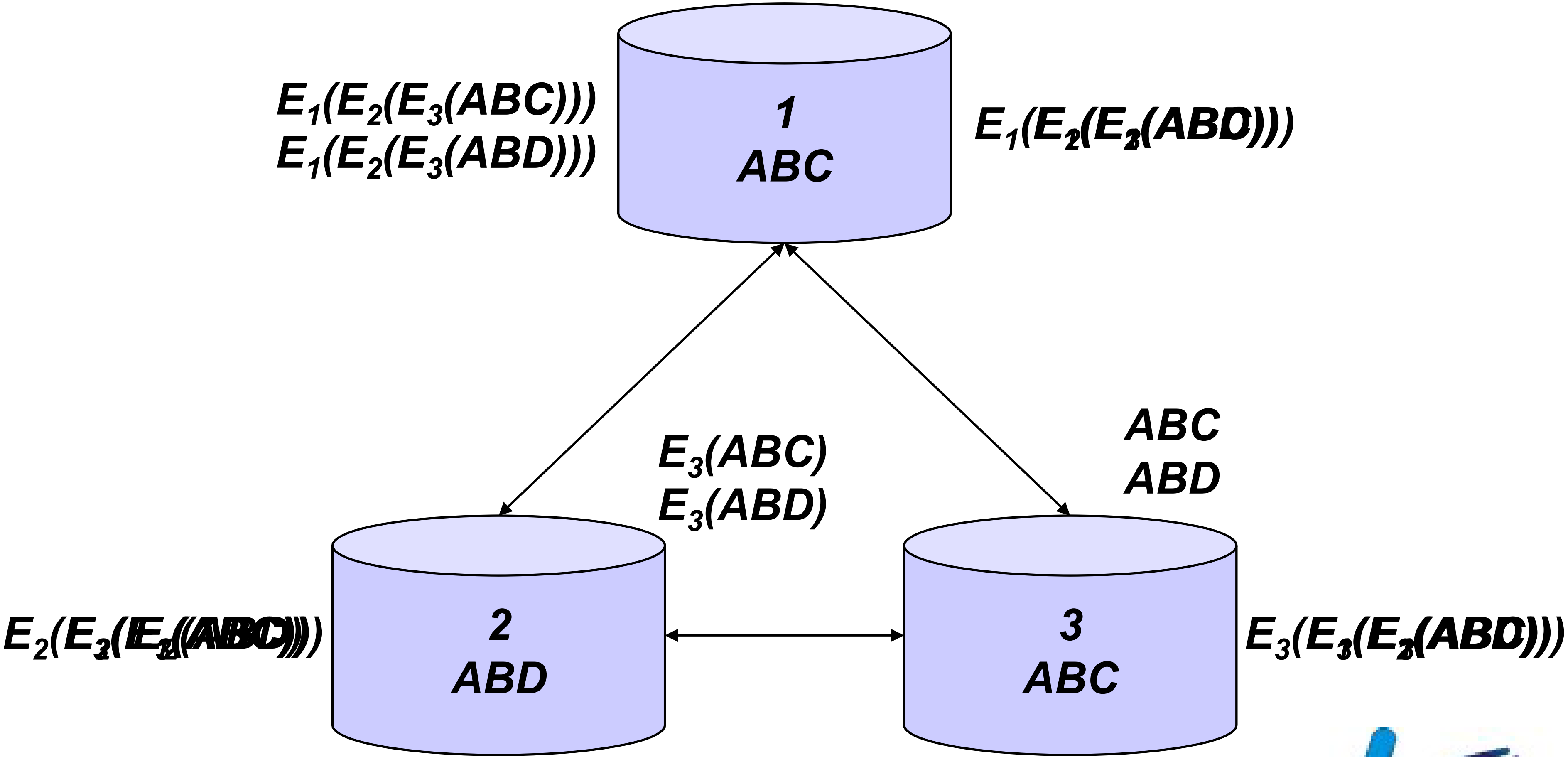
Securely Computing Candidates

- Key: Commutative Encryption ($E_a(E_b(x)) = E_b(E_a(x))$)
  - Compute local candidate set
  - Encrypt and send to next site
    - Continue until all sites have encrypted all rules
  - Eliminate duplicates
    - Commutative encryption ensures if rules the same, encrypted rules the same, regardless of order
  - Each site decrypts
    - After all sites have decrypted, rules left
- Care needed to avoid giving away information through ordering/etc.

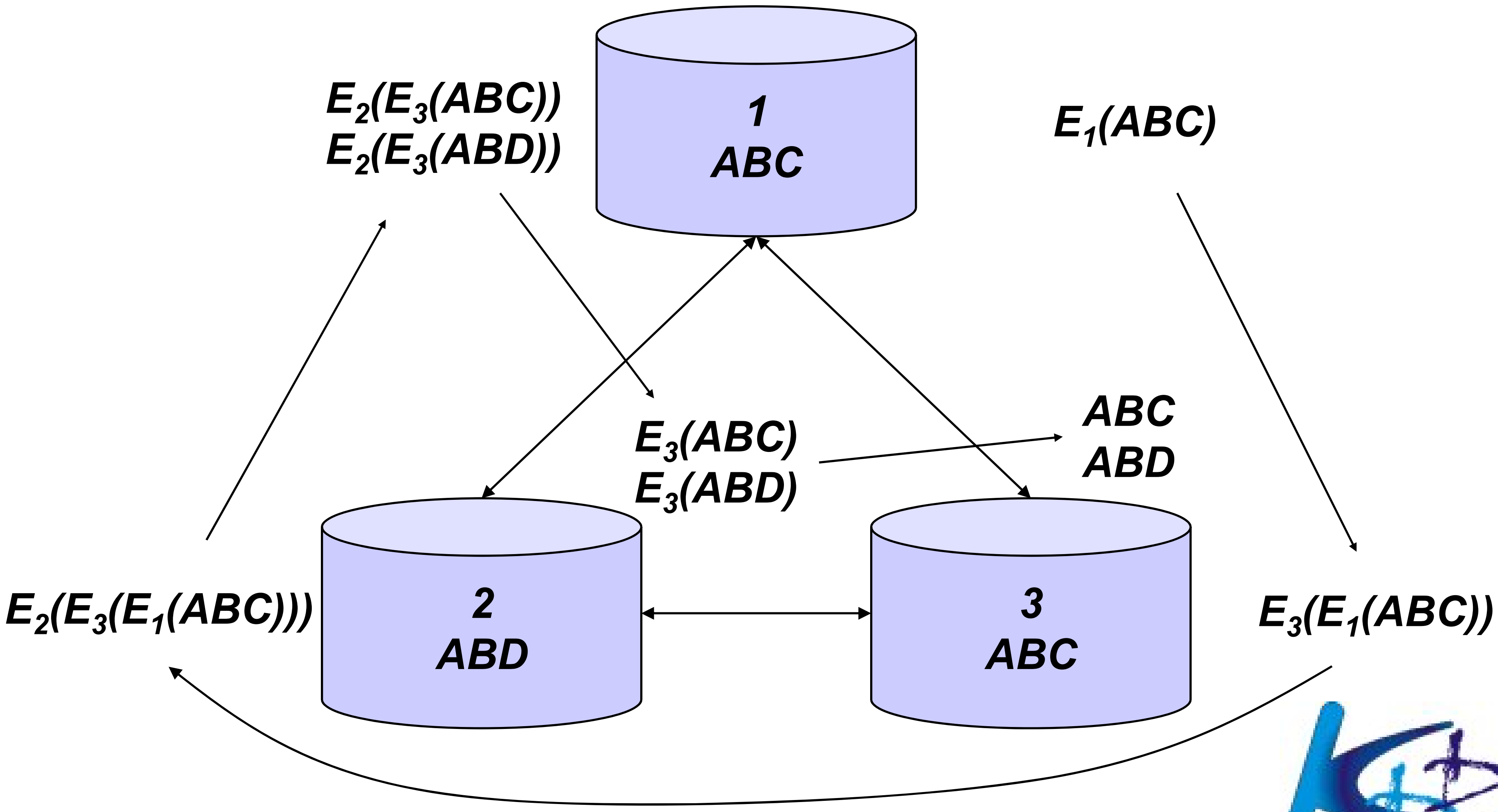
Redundancy maybe added in order to increase the security.

Not fully secure according to definitions of secure multi-party

Computing Candidate Sets



Computing Candidate Sets



Compute Which Candidates Are Globally Supported?

- Goal: To check whether

  X.sup $\geq s * \sum_{i=1}^{n} |DB_i|$ (1)

  $$\sum_{i=1}^{n} X.\sup_i \geq \sum_{i=1}^{n} s * |DB_i| \quad (2)$$

  $$\sum_{i=1}^{n} (X.\sup_i - s * |DB_i|) \geq 0 \quad (3)$$

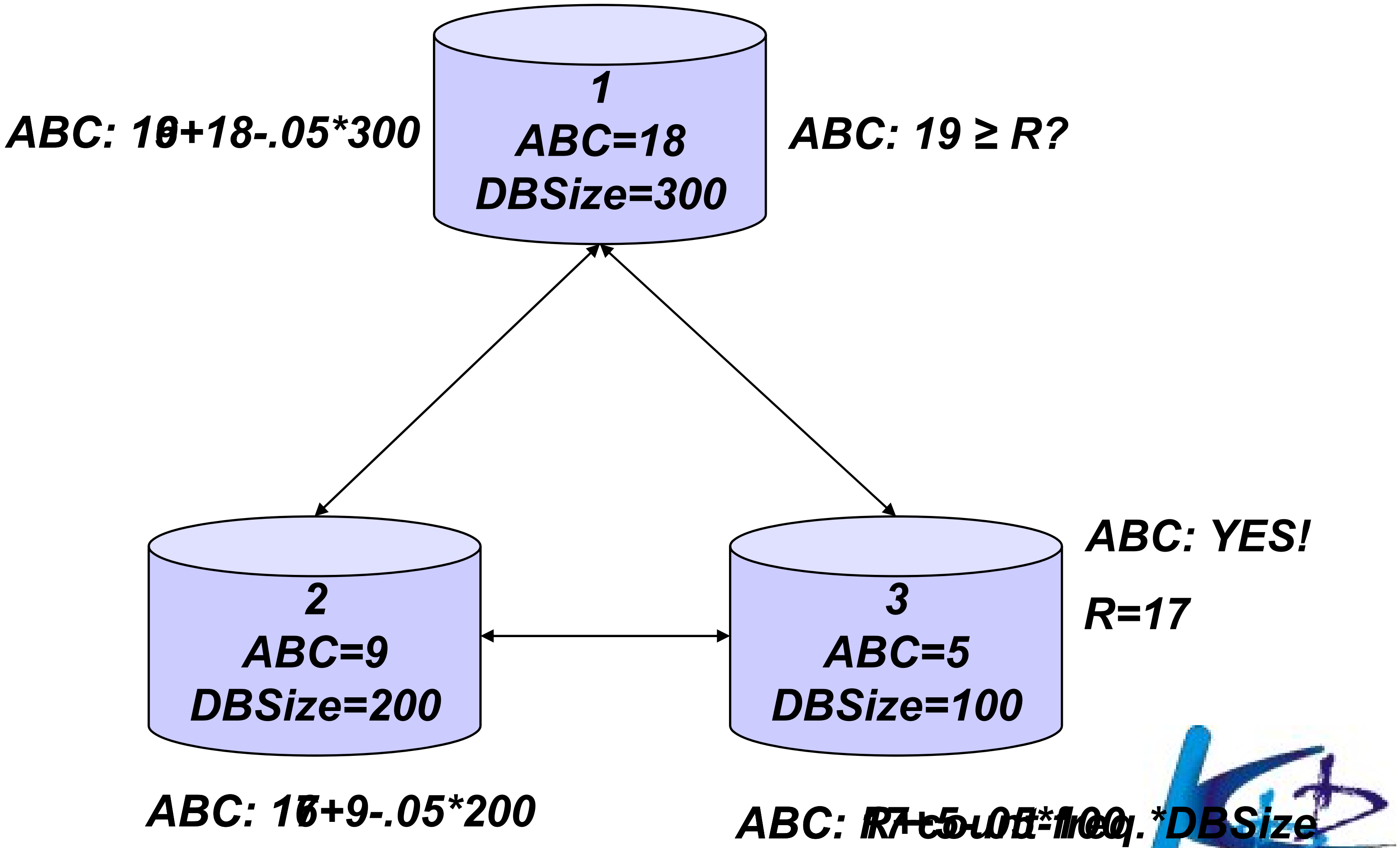
  Note that checking inequality (1) is equivalent to checking inequality (3)

Which Candidates Are Globally Supported? (Continued)

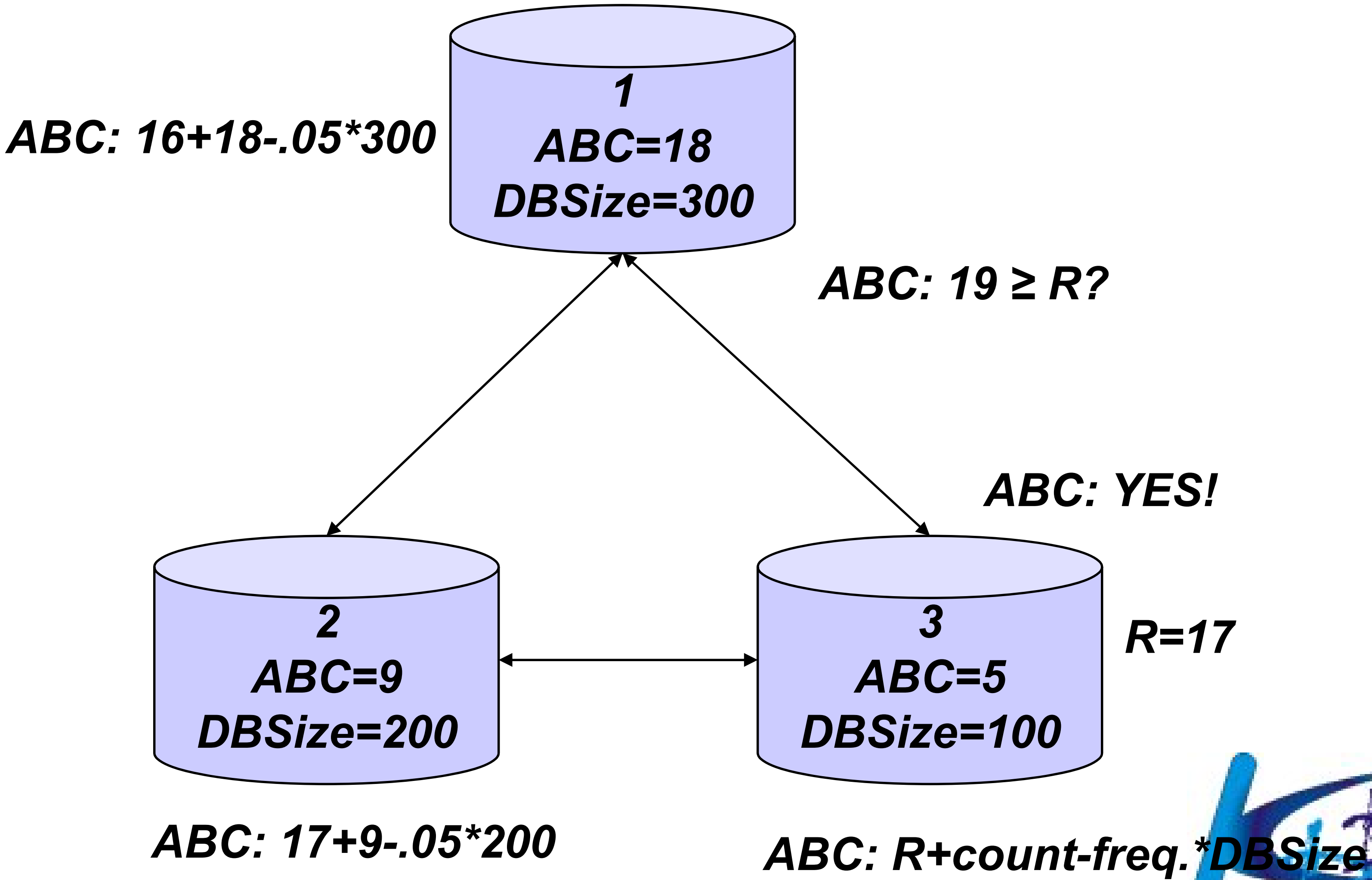
- Now securely compute Sum ≥ 0:
  - $Site_0$ generates random R

    Sends $R + count_0 - frequency * dbsize_0$ to $site_1$
  - $Site_k$ adds $count_k - frequency * dbsize_k$, sends to $site_{k+1}$
- Final result: Is sum at $site_n - R \geq 0$?
  - Use Secure Two-Party Computation
- This protocol is secure in the semi-honest model

Computing Frequent:
Is *ABC* ≥ 5%?



Computing Frequent: Is $ABC \geq 5\%$?



Computing Confidence

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

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

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

Association Rules in Vertically Partitioned Data

- 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*

JSV	Brain Tumor	Diabetic	*JSV*	*5210*	*Li/Ion*	*Piezo*

Vertically Partitioned Data *(Vaidya and Clifton ’02)*

- Learn globally valid association rules
- Prevent disclosure of individual relationships
  - Join key revealed
  - Universe of attribute values revealed
- Many real-world examples
  - Ford / Firestone
  - FBI / IRS
  - Medical records

Basic idea

- Find out if itemset $\{A_1, B_1\}$ is frequent (i.e., If support of $\{A_1, B_1\} \geq k$)

A

Key	A_1
k_1	1
k_2	0
k_3	0
k_4	1
k_5	1

B

Key	B_1
k_1	0
k_2	1
k_3	0
k_4	1
k_5	1

- Support of itemset is defined as number of transactions in which all attributes of the itemset are present
- For binary data, support $=|A_i \wedge B_i|$
- Boolean AND can be replaced by normal (arithmetic) multiplication.

Basic idea

- Thus, $Support = \sum_{i=1}^{n} A_i \times B_i$
- This is the scalar (dot) product of two vectors
- To find out if an arbitrary (shared) itemset is frequent, create a vector on each side consisting of the component multiplication of all attribute vectors on that side (contained in the itemset)
- 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 = $\prod B_2 B_3$
  - Securely compute the dot product of X and Y

The algorithm

$L_1 = \{\text{large 1-itemsets}\}$
for (k=2; $L_{k-1} \neq \phi$; k++) do begin
    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 $= \cup_k L_k$

Protocol

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

$$\left\langle x_1 + a_{1,1} * R_1 + a_{1,2} * R_2 + \cdots + a_{1,n/2} * R_{n/2} \right\rangle$$
$$\left\langle x_2 + a_{2,1} * R_1 + a_{2,2} * R_2 + \cdots + a_{2,n/2} * R_{n/2} \right\rangle$$
$$\vdots$$
$$\left\langle x_n + a_{n,1} * R_1 + a_{n,2} * R_2 + \cdots + a_{n,n/2} * R_{n/2} \right\rangle$$

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

Protocol (cont.)

- 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.

$$S =$$

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

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

Protocol (cont)

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

$$S =$$

$$\boxed{x_1 * y_1 + x_2 * y_2 + \cdots + x_n * y_n}$$

$$+\left(a_{1,1} * y_1 * R_1 + a_{1,2} * y_1 * R_2 + \cdots + a_{1,n/2} * y_1 * R_{n/2}\right)$$

$$+\left(a_{2,1} * y_2 * R_1 + a_{2,2} * y_2 * R_2 + \cdots + a_{2,n/2} * y_2 * R_{n/2}\right)$$

$$\vdots$$

$$+\left(a_{n,1} * y_n * R_1 + a_{n,2} * y_n * R_2 + \cdots + a_{n,n/2} * y_n * R_{n/2}\right)$$

Grouping components vertically and factoring out R_i

Protocol (complete)

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

- *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 Analysis

- A sends to B
  - n values (which are linear equations in 3n/2 unknowns – the n x-values and n/2 R-values)
  - The final result (which reveals another linear equation in the n/2 R-values) (Note – this can be avoided by allowing A to only report if scalar product exceeds threshold)
- B sends to A
  - The sum, S (which is one linear equation in the n y-values)
  - n/2 values (which are linear equations in n unknowns – the n y-values)

Security Analysis

- Security based on the premise of revealing less equations than the number of unknowns – possible solutions infinite!
- Security of both is symmetrical
- Just from the protocol, nothing can be found out
- Everything is revealed *only* when about half the values are revealed

Knowledge Hiding

Privacy issue and knowledge discovery

- Security and privacy threats from data mining and similar applications

- Possible solutions to prevent data mining of significant knowledge:
  - Releasing only subsets of the source database
  - Augmenting the database
  - Disclosing an aggregated but not individual value

Knowledge Hiding

- What is disclosed?
  - the data (modified somehow)
- What is hidden?
  - some "sensitive" knowledge (i.e. secret rules/patterns)
- How?
  - usually by means of data **sanitization**
    - the data which we are going to disclose is modified in such a way that the sensitive knowledge can non longer be inferred,
    - while the original database is modified as less as possible.

Knowledge Hiding: Association Rules

- This approach can be instantiated to association rules as follows:
  - *D* source database;
  - *R* a set of association rules that can be mined from *D;*
  - R_h a subset of *R* which must be hidden.

  - Problem: how to transform *D* into *D'* (the database we are going to disclose) in such a way that R/R_h can be mined from *D'.*

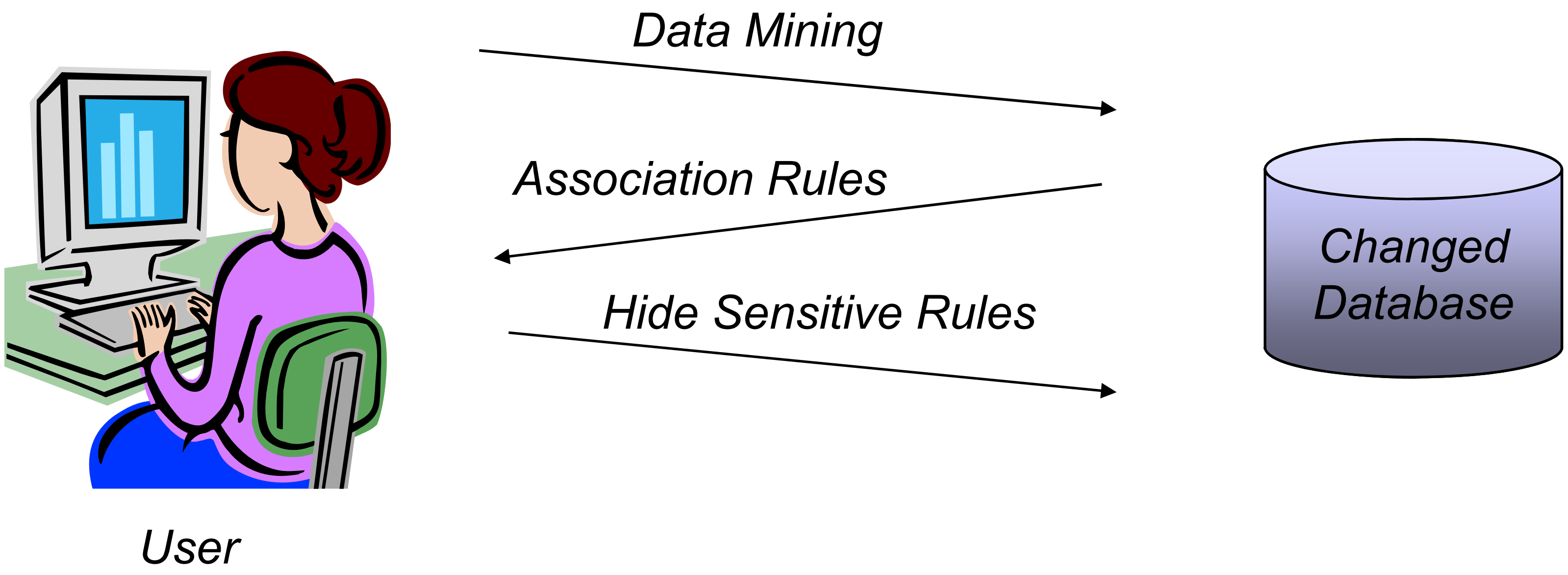
Knowledge Hiding

- E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. *Hiding association rules by using confidence and support*. In Proceedings of the 4th International Workshop on Information Hiding, 2001.
- Y. Saygin, V. S. Verykios, and C. Clifton. *Using unknowns to prevent discovery of association rules*. SIGMOD Rec., 30(4), 2001.
- S. R. M. Oliveira and O. R. Zaiane. *Protecting sensitive knowledge by data sanitization*. In Third IEEE International Conference on Data Mining (ICDM'03), 2003.
- O. Abul, M. Atzori, F. Bonchi, F. Giannotti: *Hiding Sequences.* ICDE Workshops 2007

Hiding association rules by using confidence and support

E. Dasseni, V. S. Verykios,
A. K. Elmagarmid, and E. Bertino

Scenario


Data Mining
Association Rules
Hide Sensitive Rules
Changed Database
User

Association Rule Discovery

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items.

A set of items $X \subset I$ is called an itemset.

Let D be a set of transactions, where each transaction T is an itemset such that $T \subseteq I$.

A transaction T contains an itemset X, if $X \subseteq T$.

Association Rule Discovery

An association rule is an implication of the form:

$$X \Rightarrow Y \text{ where } X \subset I, Y \subset I, \text{ and } X \bigcap Y = 0.$$

$$\text{confidence} = \frac{|X \bigcup Y|}{|X|}, \text{ and support} = \frac{|X \bigcup Y|}{N}$$

Example

Example Database

TID	Items
T1	ABCD
T2	ABC
T3	ACD

Frequent Itemsets	Support
AB	2
AC	3
AD	2
BC	2
CD	2
ABC	2
ACD	2

Optimal Sanitization is NP-hard

- Let D be the source database. Let R be a set of "significant" association rules that are mined from D. Let r be a "sensitive" rule in R. Transform D into D' so that all rules in R can still be mined from D' but r

- Optimal sanitization is NP-Hard
- Reduction from the NP-Hard problem of Hitting Set

Hiding Methods

- Reduce the support of frequent itemsets containing sensitive rules
  - Cyclic Method
  - Greedy Method
  - Isolated items and safe transactions
- Reduce the confidence or support of rules

Hiding Association Rules by using Confidence and Support

- **Assumptions**
  - hide a rule by decreasing either its confidence or its support
  - decrease either the support or the confidence one unit at a time (we modify the value of one transaction at a time)
  - hide one rule at a time
  - consider only set of disjoint rules: rules supported by large itemsets that do not have any common item

Hiding a rule X→Y by using Confidence and Support

- **Conf(X→Y) = Supp(XY) / Supp(X)**
- **Strategies:**
  - Decreasing confidence of rule
    - Increasing the support of X in transactions not supporting Y
    - Decreasing the support of Y in transactions supporting both X and Y
  - Decreasing support of rule
    - Decreasing the support of the corresponding large itemset (XY)

Strategies: basic idea

- Transactions viewed as lists
- One element for each item in DB

TID	Items
T1	ABC
T2	A

TID	A	B	C
T1	1	1	1
T2	1	0	0

- Decreasing support of S = turning to 0 one item in one transaction supporting S
- Increasing support of S = turning to 1 one item in one transaction partially supporting S

MIN_SUPP = 1/5=20%

MIN_CONF = 80%

TID	Items
T1	ABC
T2	ABC
T3	A C
T4	A
T5	B

AR	Conf
AB→C	100%
BC→A	100%

Example:hiding AB→C by increasing support of AB

- Turn to 1 the item B in transaction T4

TID	Items
T1	ABC
T2	ABC
T3	A C
T4	A
T5	B

TID	Items
T1	ABC
T2	ABC
T3	A C
T4	AB
T5	B

AR	Conf
AB→C	*66%*
BC→A	100%

Example: hiding AB→C by decreasing support of C

- Turn to 0 the item C in transaction T1

TID	Items
T1	ABC
T2	ABC
T3	A C
T4	A
T5	B

TID	Items
T1	AB
T2	ABC
T3	A C
T4	A
T5	B

AR	Conf
AB→C	*50%*
BC→A	100%

Hiding Sequences

O. Abul, M. Atzori, F. Bonchi, F. Giannotti
ISTI-CNR - Pisa, Italy

Knowledge Hiding: Sequential Patterns

- **Definitions**
  - Let S be a simple sequence[§] defined over an alphabet Σ, i.e. $S \in \Sigma^*$, and D be a database of simple sequences.
  - $S \in \Sigma^*$ is a subsequence of $T \in \Sigma^*$, denoted $S \sqsubseteq T$, iff S can be obtained by deleting some elements (not necessarily contiguous) from T
  - Support of sequence of S on D is defined as

$$sup_{\mathcal{D}}(S) = |\{T \in \mathcal{D} \,|\, S \sqsubseteq T\}|$$

§ *This is not a restriction but preferred for the sake of simplicity. Later it will be generalized so each element of S is a subset of Σ.*

The Sequence Hiding Problem

Problem 1 (The Sequence Hiding Problem)
Let $\mathcal{S}_h = \{S_1, \ldots, S_n\}$ *with* $S_i \in \Sigma^*, \forall i \in \{1, \ldots, n\}$, *be the set of sensitive sequences that must be hidden from* $\mathcal{D}$. *Given a disclosure threshold* ψ, *the Sequence Hiding Problem requires to transform* $\mathcal{D}$ *in a database* $\mathcal{D}'$ *such that:*

1. $\forall S_i \in \mathcal{S}_h, sup_{\mathcal{D}'}(S_i) \leq \psi$;
2. $\sum_{S \in \Sigma^* \setminus \mathcal{S}_h} |sup_{\mathcal{D}}(S) - sup_{\mathcal{D}'}(S)|$ *is minimized.*

Note that a special case occurs when ψ*=0, where every instance needs to be hidden.*

Matching set

- **Matching set** allows to identify all instances of sensitive patterns in a sequence

Definition 1 (Matching Set) *Given two sequences* $S \in \mathcal{S}_h$ *and* $T \in \mathcal{D}$, *we define the matching set of* S *in* T, *denoted* $\mathcal{M}_S^T$, *as the set of all sets with size* $|S|$ *of indices for which* $S \sqsubseteq T$. *For instance, let* $S = \langle a, b, c \rangle$ *and* $T = \langle a, a, b, c, c, b, a, e \rangle$, *in this case we got* $\mathcal{M}_S^T = \{(1,3,4), (1,3,5), (2,3,4), (2,3,5)\}$. *Moreover, given a sequence* $T \in \mathcal{D}$ *we define* $\mathcal{M}_{\mathcal{S}_h}^T = \bigcup_{S \in \mathcal{S}_h} \mathcal{M}_S^T$.

Sequence Sanitization

- Sanitization operator **Marking** replaces certain positions with a special symbol $\Delta \notin \Sigma$

Problem 2 (Sequence Sanitization) **Given:** *A sequence T and a set of patterns $\mathcal{S}_h$ to be hidden.* **Objective:** *Find a set of position indices of T such that, replacing the symbols in the positions with Δ results in $\mathcal{M}^T_{\mathcal{S}_h} = \emptyset$.*

Theorem 1 *Optimal Sequence Sanitization Problem is NP-Hard.*

Note that Problem2 is at sequence level while Problem1 was at database level

A Sanitization Algorithm

- A 2-stage greedy algorithm
  - First stage: Select a subset of *D* for sanitization
  - Second stage: For each sequence chosen to be sanitized (the output from the first stage), select marking positions
- The heuristic
  - Recalling the objective is introducing minimum number of Δs,
    - For the first stage: Sort the sequences in ascending order of matching set size, and select top $|D|- \psi$ for sanitization
    - For the second stage: Choose the marking position that is involved in most matches

A Sanitization Algorithm

- Illustrating the heuristic

Example 1 *Consider again the case* $S = \langle a, b, c \rangle$ *and* $T = \langle a, a, b, c, c, b, a, e \rangle$*. In this situation marking the symbol* e *(*$T[8]$*) does not affect the matching set while marking the symbol* b *in* $T[3]$ *position will cause* $\mathcal{M}_S^T = \emptyset$*. Note that the latter marking removes all the matching which is equivalent of hiding all sensitive pattern instances and thus provides sanitization. Also note that marking* $T[1]$ *reduces the number of matches without providing sanitization, while marking* $T[1]$ *and* $T[2]$ *together provides sanitization.*

Experimental Evaluation

- Two datasets:
  - SYNTHETIC: 300 discretized trajectories of synthetic car movements generated in our lab.
    - $|\Sigma|=100$ (a grid of 10x10), and average sequence length=20.1 (after repetitions removed)
  - TRUCKS: 273 discretized trajectories of real truck movement data [Frentzos et al. 2005]
    - $|\Sigma|=100$ (a grid of 10x10), and average sequence length=6.8 (after repetitions removed)

Experimental Evaluation

- 4 algorithms are experimented to get informed about the contribution of ***global*** (at the first stage) and ***local*** (at the second stage) heuristics over random selections:
  - *HH*: The proposed heuristics at both level
  - *HR*: The heuristic in second stage while random subset selection in the first stage
  - *RH, RR*: defined accordingly

Utility Measures

- Three different distortion measures, *M1, M2* and *M3*

- M1 (Data distortion): total number of marking symbols in $\mathcal{D}'$.
- M2 (Frequent Pattern Distortion):

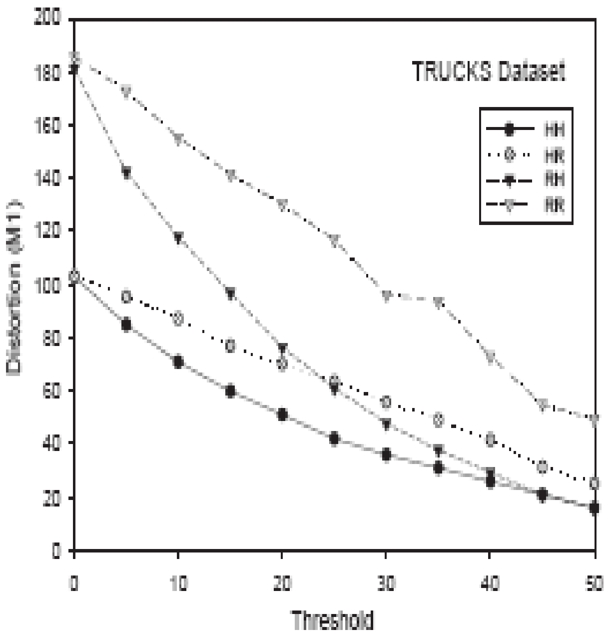
$$\frac{|\mathcal{F}(\mathcal{D},\sigma)| - |\mathcal{F}(\mathcal{D}',\sigma)|}{|\mathcal{F}(\mathcal{D},\sigma)|}$$

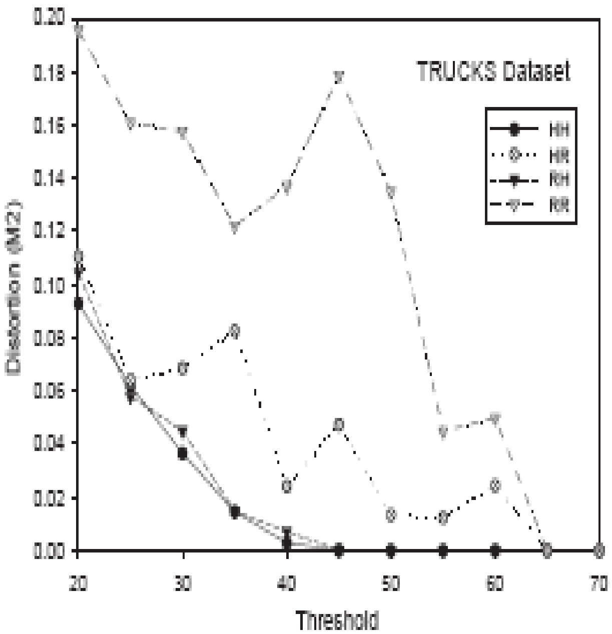
- M3 (Frequent Pattern Support Distortion):

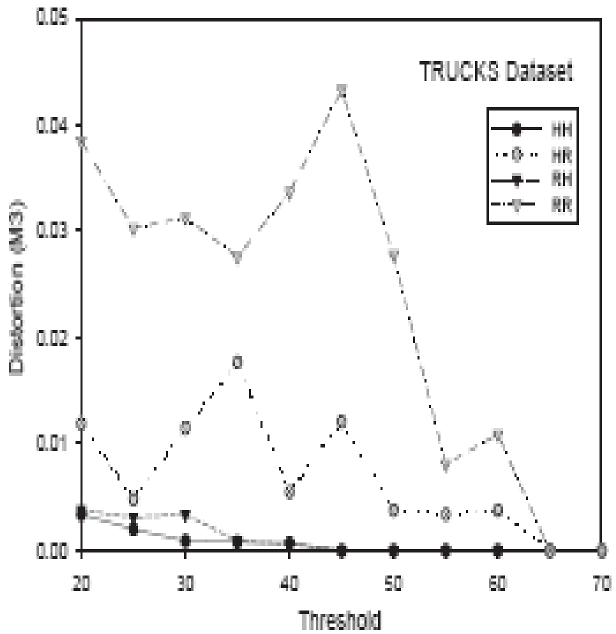
$$\frac{1}{|\mathcal{F}(\mathcal{D}',\sigma)|} \sum_{S \in \mathcal{F}(\mathcal{D}',\sigma)} \frac{sup_{\mathcal{D}}(S) - sup_{\mathcal{D}'}(S)}{sup_{\mathcal{D}}(S)}$$

Experimental Evaluation

- Results (effect of heuristics, TRUCKS)



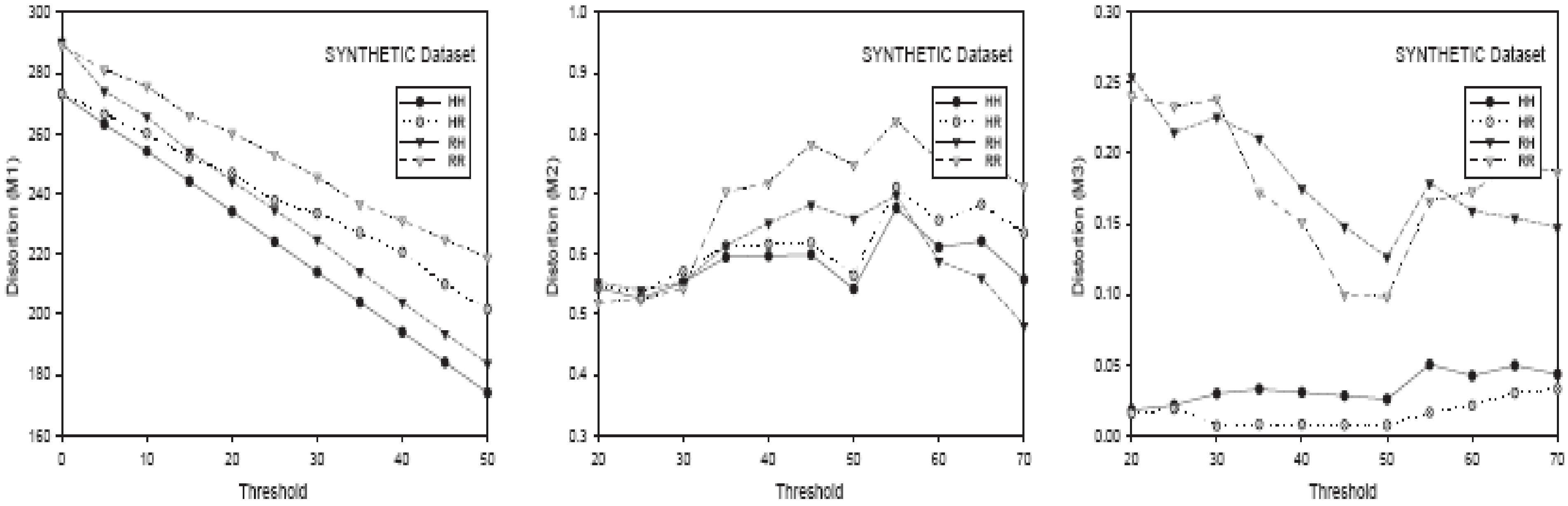




The heuristics causes relatively smaller distortions at all thresholds

Experimental Evaluation

- Results (effect of heuristics, SYNTHETIC)



The heuristics causes relatively smaller distortions at all thresholds

Privacy Preserving Outsourcing of Data Mining

Secure Outsourcing of Data Mining

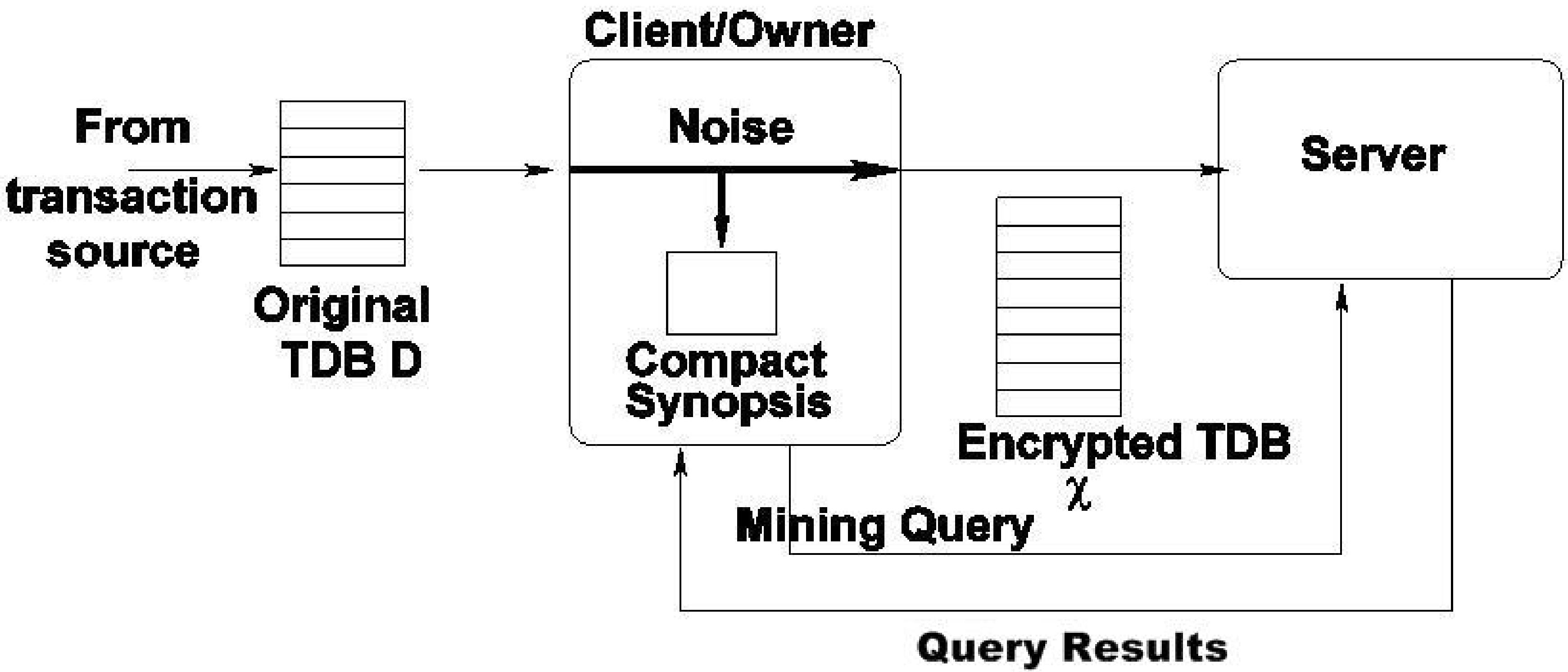
- Organizations could do not posses
  - **in-house expertise** for doing data mining
  - **computing infrastructure** adequate
- **Solution**: Outsourcing of data mining to a service provider
  - specific human resources
  - technological resources
- The server has access to data of the owner
- Data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

The Problem

PROBLEM: Given a plain database D, construct an encrypted database D* such that:

- all encrypted transactions in D* and items contained in it are secure
- given any mining query the server can compute the encrypted result
- encrypted mining and analysis results are secure
- the owner can decrypt the results and so, reconstruct the exact result
- the space and time incurred by the owner in the process has to be minimum

Framework Architecture



Secure Outsourcing of Data Mining

- W. K. Wong, D. W. Cheung, E. Hung, B. Kao, N. Mamoulis. *Security in Outsourcing of Association Rule Mining*. VLDB 2007.

- L. Qiu, Y. Li, and X. Wu. *Protecting business intelligence and customer privacy while outsourcing data mining tasks*. Know. and Inf. Sys., 17(1):99-120, 2008.

Security in Outsourcing of Association Rule Mining

W.K. Wong, D.W. Cheung, E. Hung, B. Kao, N. Mamoulis.

Background knowledge

- Where does the attacker get knowledge? (Assumption)
  - In many cases, the statistics of the global industry is public (background knowledge)
- Background Knowledge (with two parameters)
  - alpha: knows alpha% of large itemsets in original database
  - beta: the support in the knowledge is in the range
    - real support * $(1 \pm \text{beta})$

Framework

- Generation of mappings
  - One-to-n mappings
  - Item Extend
- Transformation of transactions
  - Mapping f(x)
  - Subsets of unique mapping set
  - Fake items
- Recovering association rules
  - Reverse mappings and filtering

Generation of mappings

- One-to-n vs one-to-one?
  - Intuitively, one-to-n should be more secure
- Unfortunate Scenario:
  - one-to-n + item mapping = one-to-one + item mapping
- Solution :
  - Add a random component to transaction transformation
  - It will make one-to-n always better (more secure) than one-to-one

One-to-n Transformation

one-to-one mapping

- a -> { 1 }, b -> { 2 }, …
- t = { a, b } → t′ = { 1, 2 }

one-to-n mapping

- a -> { 1, 3 }, b -> { 2, 3 }, …
- t = { a, b } → t′ = { 1, 2, 3 }

Randomly generated

one-to-n transformation

- a -> { 1, 3 }, b -> { 2, 3 }, …
- t = { a, b } → t′ = { 1, 2, 3, 4, 6 }

Transaction transformation

- $M: 2^I \to 2^B$, based on a one-to-n itemset mapping m
- N: transaction transformation
  - Maps from 2^I to 2^{BUF}
- $t' = N(t) = M(t) \cup E$
  - E is a random subset of B U F; F is a set of items not in B
- $N^{-1}(t') = \{x \mid m(x) \text{ in } t'\}$

Example transformation

$N(t) = M(t) \cup E$

T =

- {a}
- {b}
- {c}
- {a, b}
- {a, c}
- {b, c}
- {a, b, c}

m:
a -> {1, 4, 5}
b -> {2}
c -> {3, 5}

m':
a -> {1}
b -> {2}
c -> {3}

T' =

- {1, 4, 5}
- {2, 1, 3}
- {3, 5, 4}
- {1, 2, 4, 5}
- {1, 3, 4, 5}
- {2, 3, 5, 1}
- {1, 2, 3, 4, 5}

Protecting business intelligence and customer privacy while outsourcing data mining tasks

L. Qiu, Y. Li, and X. Wu

What we want to protect?

- When outsourcing mining tasks to protect the following three elements which may expose BI and customer privacy:
  - the source data which contain all transactions and items;
  - the mining requests which are itemsets of interests;
  - the mining results which are frequent itemsets and association rules.

Framework

- **Goal**: how to outsource the association rule data mining tasks, at the same time, protect BI and customer privacy
- A **Bloom filter** based approach is proposed
- Bloom filter is a simple, space-efficient, randomized data structure for representing a set of objects so as to support membership queries

Definition 3.1 Given an n-element set $S = \{s_1, \ldots, s_n\}$ and k hash functions $h_1, \ldots, h_k$ of range m, the Bloom filter of S, denoted as $B(S)$, is a binary vector of length m that is constructed by the following steps: (i) every bit is initially set to 0; (ii) every element $s \in S$ is hashed into the bit vector through the k hash functions, and the corresponding bits $h_i(s)$ are set to 1.[4] A Bloom filter function, denoted as $B(\cdot)$, is a mapping from a set (not necessarily n-element set) to its Bloom filter.

Process

- Source data are converted to **Bloom filter** representation and handed over to a third party together with mining algorithms
- The first party sends its mining requests to the third party
- Mining requests are actually candidates of frequent itemsets which are also represented by Bloom filters
- Lastly, the third party runs the mining algorithms with source data and mining requests, and comes out the mining results which are
  - frequent itemsets or association rules represented by Bloom filters
- The third party would not be able to distill down private information from Bloom filters.

Problem Definition

Problem 2 ***Our research problem: privacy preserving frequent itemsets mining.*** *Given* (i) *a collection of Bloom filters* $\{B(T_1), \ldots, B(T_N)\}$ *for transaction database* $\mathcal{D}$ *over* $\mathcal{I}$, (ii) *a set of Bloom filters* $\{B(I_1), \ldots, B(I_d)\}$ *for items in* $\mathcal{I}$, *and* (iii) *a threshold* $\tau \in [0, 1]$, *find all Bloom filters* $B(FS)$ *of itemsets* $FS \in 2^{\mathcal{I}}$ *such that* $\mathrm{freq}(FS) \geq \tau$.

- A framework of this method is based on an algorithm that computes the frequent patterns from Bloom Filters
- Thi algorithm has 3 steps
  - counting phase
  - pruning phase
  - candidates generating phase

Algorithm

Algorithm 1 Mining frequent itemsets from Bloom filters

$C_1 = \{B(I_1), \ldots, B(I_d)\}$ // $B(I_i)$ is the Bloom filter of item I_i
for ($\ell = 1$; $C_\ell \neq \varnothing$; ℓ++) **do**
  for each $B(S) \in C_\ell$ and each transaction filter $B(T_i)$ **do**
    if $S \subseteq_B T_i$ **then** Bsupport(S)++ // $S \subseteq_B T_i$ iff $B(S) \wedge B(T_i) = B(S)$
  end for
  for each $B(S) \in C_\ell$ **do**
    if $Bsupport(S) < N \cdot \tau'$ **then** delete $B(S)$ from C_ℓ // τ' is the revised threshold in data mining
  end for
  $F_\ell = C_\ell$ // F_ℓ is the collection of Bloom filters of all "frequent" itemsets with length ℓ
  $C_{\ell+1} = can\_gen(F_\ell)$ // generate filters of candidate itemsets for the next round
end for
Answer = $\bigcup_\ell F_\ell$ // all filters of frequent itemsets