Privacy Preserving Frequent Itemset Mining

Stanley R. M. Oliveira[1,2]
oliveira@cs.ualberta.ca
[1]Embrapa Information Technology
Andre Tosello, 209, PO Box 6041
13083-886, Campinas, SP, Brasil

Osmar R. Zaïane[2]
zaiane@cs.ualberta.ca
[2]Database Systems Laboratory
Computing Science Department
University of Alberta, Canada

Workshop on Privacy, Security, and Data Mining
ICDM - Maebashi City, Japan December 9, 2002

Outline

- Motivation
- Basics Concepts
- The Framework for Privacy Preservation
- The Sanitizing Algorithms
- Experimental Results
- Related Work
- Conclusions and Future Research

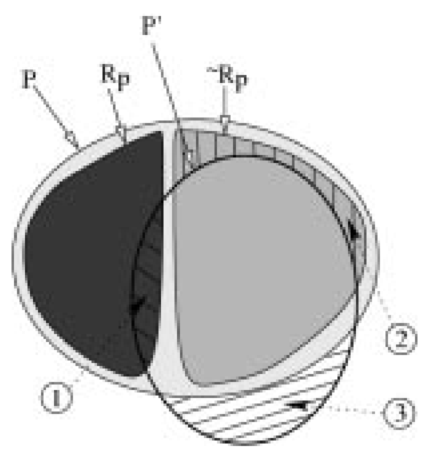
Motivation

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions

- Privacy issues in data mining have emerged globally;
- Broad application of frequent itemsets;
- The traditional solution “all or nothing” has been too rigid;
- The need for techniques to enforce privacy concerns when mining.

Privacy Preservation Problem

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions



Visual representation of restrictive and non-restrictive patterns and the patterns effectively discovered after transaction sanitization.

ψ allows a trade-off between problems (1) and (2)

Restrictive Patterns and Sensitive Transactions

- Motivation
- **Basic Concepts**
- Framework
- Algorithms
- Experiments
- Related Work
- Conclusions

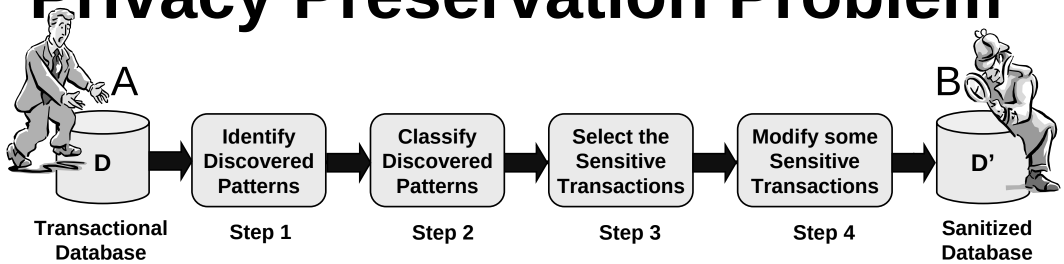
- Definition 1: Let *D* be a transactional database, *P* be a set of all frequent patterns that can be mined from *D*, and $Rules_H$ be a set of decision support rules that need to be hidden according to some security policies. **A set of patterns, denoted by R_P, is said to be restrictive if $R_P \subset$ P and if and only if R_P would derive the set $Rules_H$.** $\neg R_P$ is the set of non-restrictive patterns such that $\neg R_P \cup R_P = P$.

Restrictive Patterns and Sensitive Transactions

- Motivation
- **Basic Concepts**
- Framework
- Algorithms
- Experiments
- Related Work
- Conclusions

- Definition 2: Let *T* be a set of all transactions in a transactional database *D* and R_P be a set of restrictive patterns mined from *D*. **A set of transactions is said to be sensitive, as denoted by S_T, if $S_T \subset T$ and iff all restrictive patterns can be mined from S_T and only from S_T.**

Privacy Preservation Problem

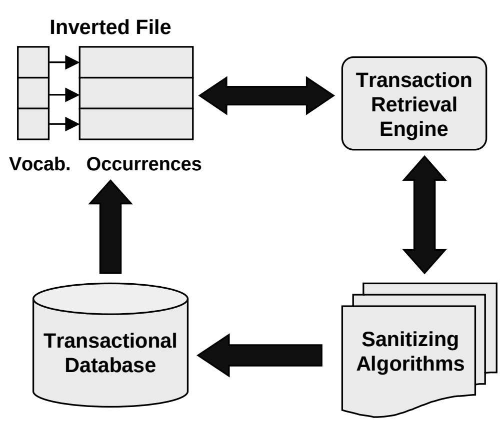


The sanitization process

Problem Definition: If *D* is the source database of transactions and *P* is a set of relevant patterns that could be mined from *D*, the goal is to transform *D* into a database *D'* so that the most frequent patterns in *P* can still be mined from *D'* while others will be hidden.

The goal: Hide restrictive patterns while minimizing the impact on the sanitized database.

Privacy Preservation Framework



Privacy Preservation Framework

The Inverted File Index

Docs	Items/Terms
T1	A B C D
T2	A B C
T3	A B D
T4	A C D
T5	A B C
T6	B D

Item	Freq		Transaction IDs
A	5	→	T1, T2, T3, T4, T5
B	5	→	T1, T2, T3, T5, T6
C	4	→	T1, T2, T4, T5
D	4	→	T1, T3, T4, T6

Vocabulary — Transaction IDs

An example of transactions modeled by documents and the corresponding inverted file.

Conflicting Transactions

Docs	Items/Terms
T1	A B C D
T2	A B C
T3	A B D
T4	A C D
T5	A B C
T6	B D

Sample Transactional Database

Example: R_P = {ABD, ACD}

S_T = {T1, T3, T4}

ABD = {T1, T3}
ACD = {T1, T4}

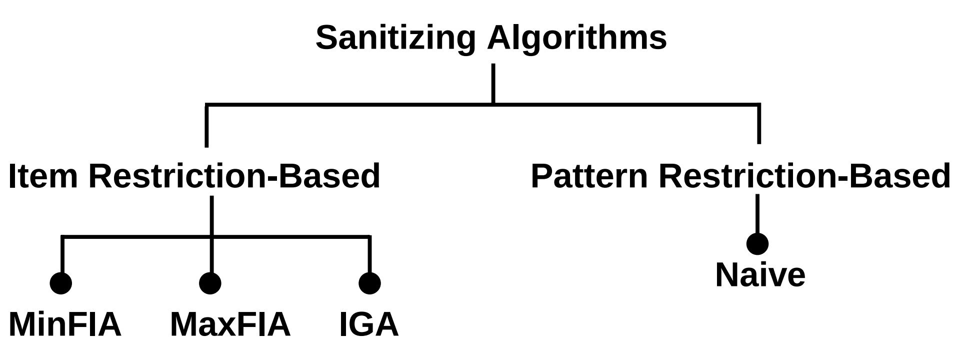
Degree (T1) = 2
Degree (T3) = 1
Degree (T4) = 1

Sanitizing Algorithms: Major Steps

1. Identify sensitive transactions for each restrictive patterns;
2. For each restrictive pattern, identify a candidate item that should be eliminated (victim item);
3. Based on the disclosure threshold ψ, compute the number of sensitive transactions to be sanitized;
4. Based on the number found in 3, remove the victim items from the sensitive transactions.

A Taxonomy of Sanitizing Algorithms

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions



A taxonomy of sanitizing algorithms

The Naïve Algorithm

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions

Naive_Algorithm
Input: D, R_P, ψ
Output: D'

Step 1. For each restrictive pattern $rp_i \in R_P$ do
  1. $T[rp_i] \leftarrow$ Find_Sensitive_Transactions(rp_i, D);

Step 2. For each restrictive pattern $rp_i \in R_P$ do
  1. $Victims(rp_i) \leftarrow \forall\ item_k$ such that $item_k \in rp_i$

Step 3. For each restrictive pattern $rp_i \in R_P$ do
  1. $NumTrans(rp_i) \leftarrow |T[rp_i]| \times (1 - \psi)$ // $|T[rp_i]|$: number of sensitive transac. for rp_i

Step 4. $D' \leftarrow D$
  For each restrictive pattern $rp_i \in R_P$ do
    1. Sort_Transactions($T[rp_i]$);   //in ascending order of degree of conflict
    2. $TransToSanitize \leftarrow$ Select first $NumTrans(rp_i)$ transactions from $T[rp_i]$
    3. in D' foreach transaction $t \in TransToSanitize$ do
      3.1. $t \leftarrow [t - Victims(rp_i)]$

End

The Minimum Frequency Item Algorithm (MinFIA)

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions

Minimum_Frequency_Item_Algorithm
Input: D, R_P, ψ
Output: D'

Step 1. For each restrictive pattern $rp_i \in R_P$ do
  1. $T[rp_i] \leftarrow$ Find_Sensitive_Transactions(rp_i, D);

Step 2. For each restrictive pattern $rp_i \in R_P$ do
  1. $Victim(rp_i) \leftarrow item_v$ such that $item_v \in rp_i$ and $\forall\ item_k \in rp_i$
    support($item_k$, D) $\geq$ support($item_v$, D)

Step 3. For each restrictive pattern $rp_i \in R_P$ do
  1. $NumTrans(rp_i) \leftarrow |T[rp_i]| \times (1 - \psi)$ // $|T[rp_i]|$: number of sensitive transac. for rp_i

Step 4. $D' \leftarrow D$
  For each restrictive pattern $rp_i \in R_P$ do
    1. Sort_Transactions($T[rp_i]$);   //in ascending order of degree of conflict
    2. $TransToSanitize \leftarrow$ Select first $NumTrans(rp_i)$ transactions from $T[rp_i]$
    3. in D' foreach transaction $t \in TransToSanitize$ do
      3.1. $t \leftarrow [t - Victim(rp_i)]$

End

The Maximum Frequency Item Algorithm (MaxFIA)

Maximum_Frequency_Item_Algorithm
Input: D, R_P, ψ
Output: D'

Step 1. For each restrictive pattern $rp_i \in R_P$ do
   1. $T[rp_i] \leftarrow$ Find_Sensitive_Transactions(rp_i, D);

Step 2. For each restrictive pattern $rp_i \in R_P$ do
   1. $Victim(rp_i) \leftarrow item_v$ such that $item_v \in rp_i$ and $\forall item_k \in rp_i$ support($item_k$, D) $\leq$ support($item_v$, D)

Step 3. For each restrictive pattern $rp_i \in R_P$ do
   1. $NumTrans(rp_i) \leftarrow |T[rp_i]| \times (1 - \psi)$ // $|T[rp_i]|$: number of sensitive transac. for rp_i

Step 4. $D' \leftarrow D$
   For each restrictive pattern $rp_i \in R_P$ do
      1. Sort_Transactions($T[rp_i]$);   //in ascending order of degree of conflict
      2. $TransToSanitize \leftarrow$ Select first $NumTrans(rp_i)$ transactions from $T[rp_i]$
      3. in D' foreach transaction $t \in TransToSanitize$ do
         3.1. $t \leftarrow [t - Victim(rp_i)]$

End

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions

The Item Grouping Algorithm (IGA)

Item_Grouping_Algorithm
Input: D, R_P, ψ **Output:** D'

Step 1. For each restrictive pattern $rp_i \in R_P$ do
   1. $T[rp_i] \leftarrow$ Find_Sensitive_Transactions(rp_i, D);

Step 2.
   1. Group restrictive patterns in a set of groups GP such that $\forall$ $G \in GP$, $\forall$ rp_i, $rp_j \in G$, rp_i and rp_j share the same itemset I. Give the class label α to G such that $\alpha \in I$ and $\forall$ $\beta \in$ I, support(α, D) $\leq$ support(β, D).
   2. Order the groups in GP by size in terms of number of restrictive patterns in the group.
   3. Compare groups pairwise G_i and G_j starting with the largest.
   For all $rp_k \in G_i \cap G_j$ do
      3.1. if size(G_i) $\neq$ size(G_j) then remove rp_k from smallest(G_i, G_j)
      3.2. else remove rp_k from group with class label α such that support(α, D) $\leq$ support(β, D) and α, β are class labels of either G_i or G_j
   4. For each restrictive pattern $rp_i \in R_P$ do
      4.1. Victim(rp_i) $\leftarrow \alpha$ such that α is the class label of G and $rp_i \in$ G

Step 3. For each restrictive pattern $rp_i \in R_P$ do
   1. $NumTrans(rp_i) \leftarrow |T[rp_i]| \times (1 - \psi)$ // $|T[rp_i]|$ is the number of sensitive transac. for rp_i

Step 4. $D' \leftarrow D$
   For each restrictive pattern $rp_i \in R_P$ do
      1. Sort_Transactions($T[rp_i]$);   //in descending order of degree of conflict
      2. $TransToSanitize \leftarrow$ Select first $NumTrans(rp_i)$ transactions from $T[rp_i]$
      3. in D' foreach transaction $t \in TransToSanitize$ do
         3.1. $t \leftarrow [t - Victim(rp_i)]$

End

The Item Grouping Algorithm (IGA)

Docs	Items/Terms
T1	A B C D
T2	A B C
T3	A B D
T4	A C D
T5	A B C
T6	B D

Sample Transactional Database

Ex.: R_P = {ABD, ACD}

S_T = {T1, T3, T4}

ABD = {T1, T3}
ACD = {T1, T4}

1. Group restrictive patterns

G_1= {ABD} Class Label = {D}
G_2= {ACD} Class Label = {C}
G_3= {ABD, ACD} Class Label = {A,D}

2. Order the groups by size

G_3= {ABD, ACD} Class Label = {A,D}
G_1= {ABD} Class Label = {D}
G_2= {ACD} Class Label = {C}

3. Compare the groups pairwise

G_3= {ABD, ACD} Class Label = {D}

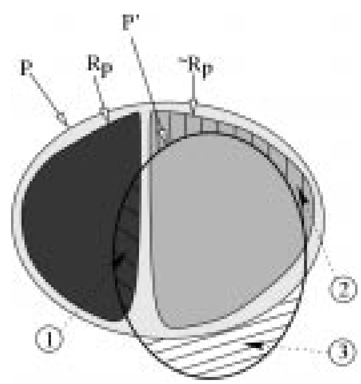
Support(D)<=Support(A)

Experimental Results

- PC AMD Athlon 1900/1600, with 1.2 GB of RAM
- Dataset: 100K transactions, 500 different items
- Minimum size per transaction: 40 items
- Restricted patterns: 10 patterns (support: 20% to 40%)
- Restrictive patterns ranging from 2 to 5 items
- 22,479 patterns became restricted (out of 1,866,693)
- Time required to build the inverted file: 4.05 sec.
- Time for retrieving all sensitive transactions: 1.02 sec.

Experimental Results

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions



Measuring three possible problems

1. Hiding Failure (HF) $HF = \frac{\# R_P(D')}{\# R_P(D)}$

2. Misses Cost (MC)

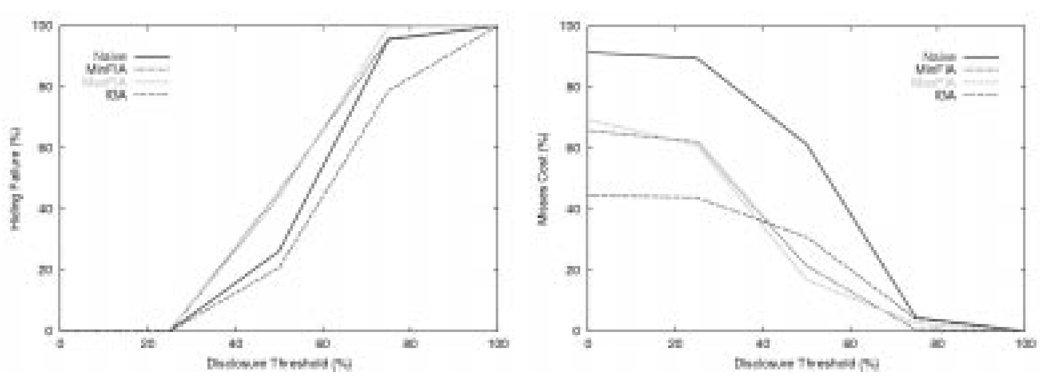
$$MC = \frac{\# \neg R_P(D) - \# \neg R_P(D')}{\# \neg R_P(D)}$$

3. Artifactual Patterns (AP)

$$AP = \frac{|P'| - |P \cap P'|}{|P'|}$$

Experimental Results

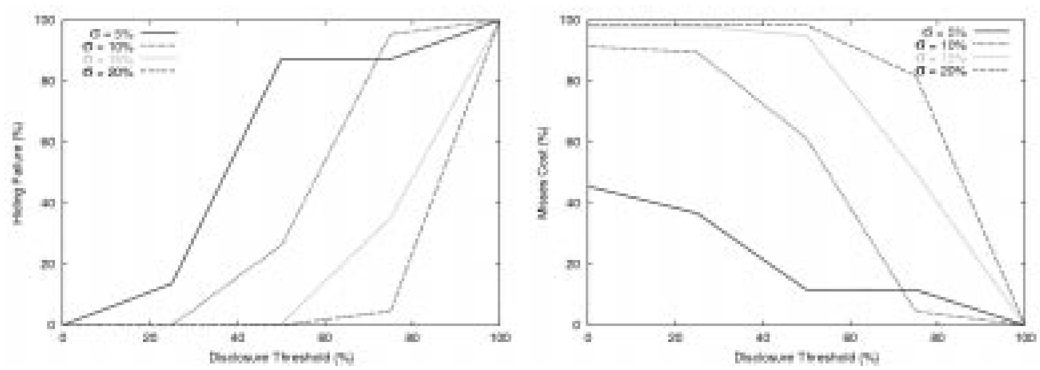
Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions



Effect of ψ on the hiding failure and the misses cost

Experimental Results

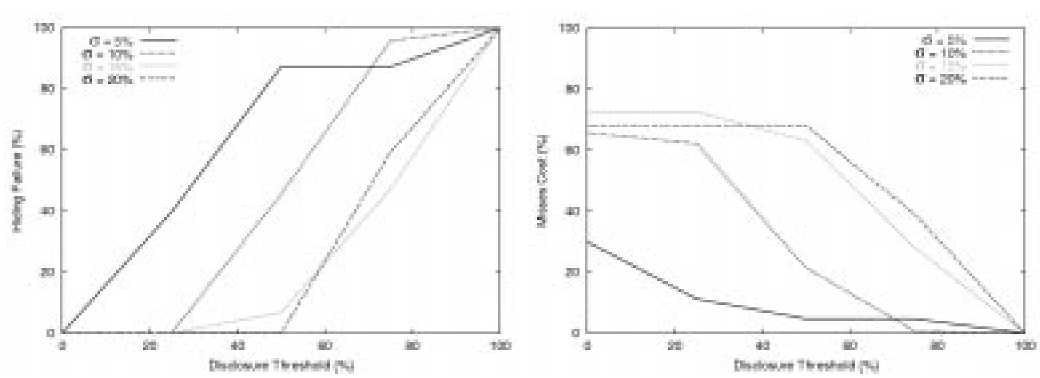
- Motivation
- Basic Concepts
- Framework
- Algorithms
- **Experiments**
- Related Work
- Conclusions



Effect of support threshold σ on privacy preservation (Naïve)

Experimental Results

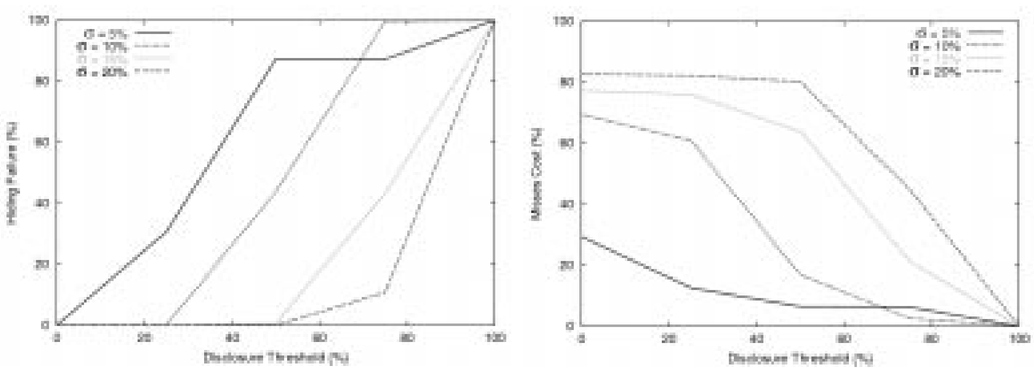
- Motivation
- Basic Concepts
- Framework
- Algorithms
- **Experiments**
- Related Work
- Conclusions



Effect of support threshold σ on privacy preservation (MinFIA)

Experimental Results

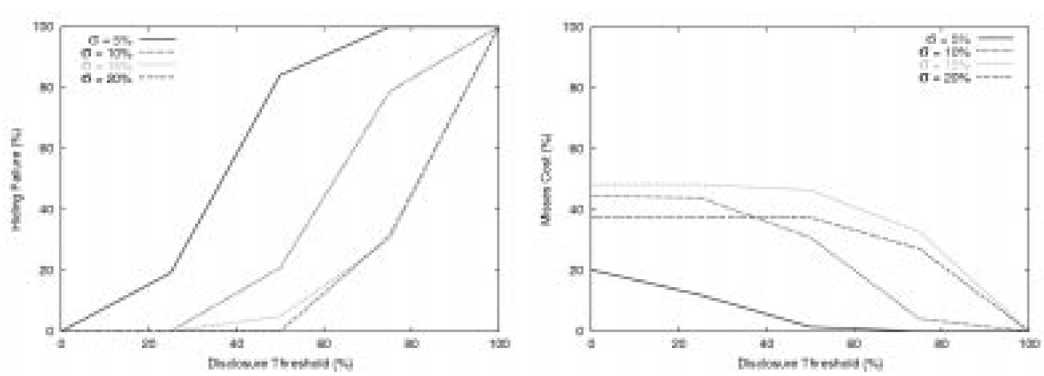
- Motivation
- Basic Concepts
- Framework
- Algorithms
- **Experiments**
- Related Work
- Conclusions



Effect of support threshold σ on privacy preservation (MaxFIA)

Experimental Results

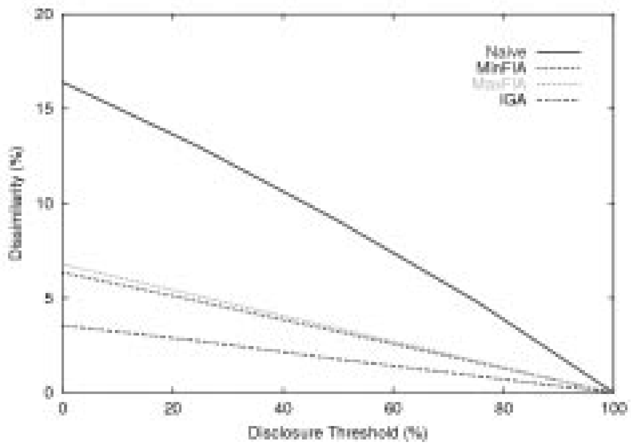
- Motivation
- Basic Concepts
- Framework
- Algorithms
- **Experiments**
- Related Work
- Conclusions



Effect of support threshold σ on privacy preservation (IGA)

Experimental Results

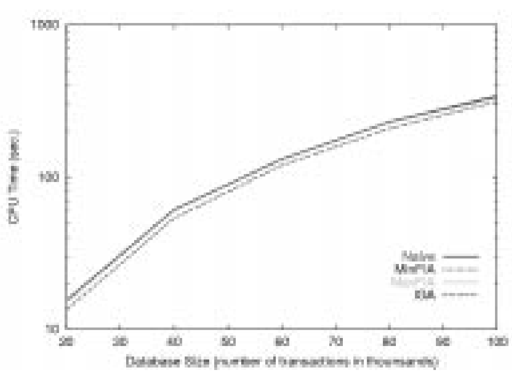
- Motivation
- Basic Concepts
- Framework
- Algorithms
- **Experiments**
- Related Work
- Conclusions



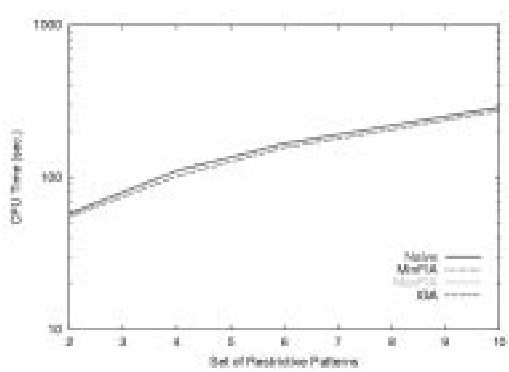
The difference in size between D and D'

Experimental Results

- Motivation
- Basic Concepts
- Framework
- Algorithms
- **Experiments**
- Related Work
- Conclusions



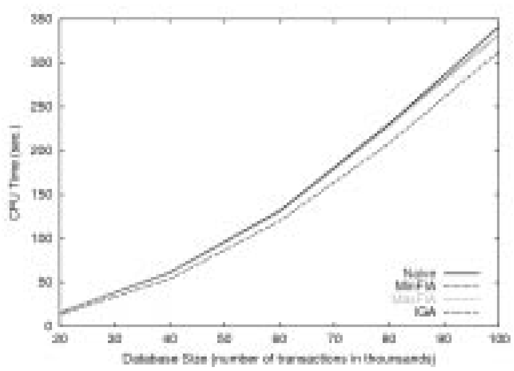
CPU time wrt database size



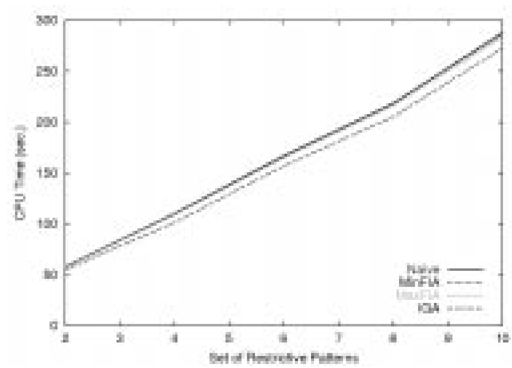
CPU time wrt number of restrictive patterns

Experimental Results

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions



CPU time wrt database size



CPU time wrt number of restrictive patterns

Related Work

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions

1. M. Atallah, E. Bertino, A. K. Elmagarmid, M. Ibrahim, and V. Verykios. **Disclosure Limitation of Sensitive Rules**. In *IEEE Knowledge and Data Engineering Workshop*, Chicago, Illinois, USA, November 1999, pp.45-52.

2. E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. **Hiding Association Rules by Using Confidence and Support.** In the *4th Information Hiding Workshop (IHW)*, Pittsburg, PA, USA, April 2001, pp.369-383.

3. Y. Saygin, V.S. Verykios and C. Clifton. **Using Unknowns to Prevent Discovery of Association Rules**. *SIGMOD Record* 30(4), December 2001, pp.45-54.

Motivation
Basic Concepts
Framework
Algorithms
Experiments
Related Work
Conclusions

Conclusions and Future Work

- Main contributions of this paper:
  - The design and implementation of the framework;
  - A taxonomy of sanitizing algorithms;
  - Performance measures for mining frequent patterns.
- Future Work:
  - Investigating optimizing the “negative” impact of the sanitization process;
  - Adjusting the sanitizing algorithms for association rule mining;
  - Studying the impact of data sanitization in distributed environment;
  - Integrating this framework with RBAC.

Questions?