Checking for k-Anonymity Violation by Views

X. Sean Wang
The University of Vermont

Co-authors:
Chao Yao, Sushil Jajodia
George Mason University

THE UNIVERSITY OF VERMONT

Outline

- Motivation
- Problem definition
- Complexity of the general problem
- Polynomial cases
- Conservative checking methods
- Conclusion

Anonymity set and k-anonymity

- K-anonymity. Two versions:
  - Property k-anonymity: Given a set of k distinct *properties*, and a particular *person*. The adversary knows that the *person* has one of the k properties, but he does not know exactly which property the *person* has.
  - Person k-anonymity: Given a set of k distinct persons, and a particular *property*. The adversary knows that one of the k persons has the *property*, but she can't tell among the k persons who has that *property*.

  - *Implicit assumption: probability of a particular association (between person and property) is rather small.*

Prior work

- Samarati and Sweeney, PODS 1998; Meyerson and Williams PODS 2005, and others.
- Problem studied: Given a *private/base table* (one tuple per person), how to “generalize” or “obfuscate” values so that adversary can only tell that each published tuple “originates” from at least k tuples in the private table.
- Example (Person 2-anonymity):

SSN	Problem
111-11-1111	P11
111-11-1112	P21
111-11-1123	P31
111-11-1124	P32

SSN	Problem
111-11-111*	P11
111-11-111*	P21
111-11-112*	P31
111-11-112*	P32

Property 2-anonymity

- Not handled by prior work (although techniques do apply).
- Hybrid solution:

SSN	Problem
111-11-1111	P11
111-11-1112	P21
111-11-1123	P31
111-11-1124	P32

SSN	Problem
111-11-1111	P?1
111-11-1112	P?1
111-11-112*	P31
111-11-112*	P32

Publishing with views

Name	Job	Salary	Problem
George	Manager	70K	Cold
John	Manager	90K	Obesity
Bill	Lawyer	110K	HIV

Private table P1

$v_1 = \Pi_{Name,Job}(P_1)$

Name	Job
George	Manager
John	Manager
Bill	Lawyer

$v_2 = \Pi_{Job,Problem}(P_1)$

Job	Problem
Manager	Cold
Manager	Obesity
Lawyer	HIV

v1 and v2 together: *Violation of property 2-anonymity!*

A little more complicated example

Name	Job	Salary	Problem
George	Manager	70K	Cold
John	Manager	90K	Obesity
Bill	Lawyer	110K	HIV

Private table P1

$\Pi_{Name}\sigma_{Salary>80K}(P_1)$

Name
John
Bill

$\Pi_{Problem}\sigma_{80K<Salary<100K}(P_1)$

Problem
Obesity

Name
George
John

$\Pi_{Name}\sigma_{Salary<105K}(P_1)$

Functional Dependency: Name→Problem

Name	Problem	Charge
George	Cold	20K
John	Obesity	20K
John	Obesity	30K
Bill	HIV	30K

Private table P2

$\Pi_{Name,Charge}(P_2)$

Name	Charge
George	20K
John	20K
John	30K
Bill	30K

$\Pi_{Charge,Problem}(P_2)$

Charge	Problem
20K	Cold
20K	Obesity
30K	Obesity
30K	HIV

Prior work on views

- Miklau and Suciu 2004; Dalvi, Miklau, and Suciu 2005; Deutsch and Papakonstantinou 2005; Dalvi and Suciu 2005 VLDB (“to some extent”).
- Probability models
- *Not* at the tuple level

Outline

- Motivation
- ***Problem definition***
- Complexity of the general problem
- Polynomial cases
- Conservative checking methods
- Conclusion

Assumptions

- Provided to the public
  - View set v: a set of materialized views
  - View definitions (i.e., the queries)
- In addition, the “public” knows the constraints (FDs) on the private (base) table.
- Notation:

I^v is the set of all possible base/private table instances, each yielding (exactly) the given view set.

Assumptions (II)

- Two (fixed) attributes: ID and P (Property) on the base/private table
- The secret (to be protected) is the projection:

$$S(I) = \prod_{ID,P}(I)$$

- Looking for property k-anonymity

Definitions

- (Secret) association:
  - A binary tuple on *(ID, P)*
- An ***association cover*** A wrt a view set v is:
  - a set of associations,
  - all have the same ID value, and
  - for each I in I^v, $S(I) \cap A \neq \emptyset$.
- Intuition: If there exists A with $|A|<2$, then there is "information leak".
  - What if $|A|<k$?

Association cover example

Name	Job	Salary	Problem
George	Manager	70K	Cold
John	Manager	90K	Obesity
Bill	Lawyer	110K	HIV

Private table P1

$v_1 = \Pi_{Name,Job}(P_1)$

Name	Job
George	Manager
John	Manager
Bill	Lawyer

$v_2 = \Pi_{Job,Problem}(P_1)$

Job	Problem
Manager	Cold
Manager	Obesity
Lawyer	HIV

One association cover: {(Bill, HIV)}
Another: {(George, Cold), (George, Obesity)}

K-anonymity

Given a view set v and integer $k \geq 2$, we say v violates ***k-anonymity*** if there exists an association cover wrt v of size less than k.

Outline

- Motivation
- Problem definition
- ***Complexity of the general problem***
- Polynomial cases
- Conservative checking methods
- Conclusion

Computationally hard

- With FD present, it is $\sum_2^p$-complete to test if a view set violates k-anonymity
- Data complexity
  - Complexity is in terms of the number of tuples

Outline

- Motivation
- Problem definition
- Complexity of the general problem
- ***Polynomial cases***
- Conservative checking methods
- Conclusion

Polynomial case

- No FDs
- Selection and projection queries
  - Conjunctive selection conditions

Basic definitions

- *Tuple cover* for a view set v:
  - A set of tuples T such that for each I in I^v, $I \cap T \neq \emptyset$
- A_{min}: the set of all minimal association covers
- T_{min}: the set of all minimal tuple covers

Basic mechanism

- Given a view set v,

$$A_{min} \subseteq \{\Pi_{ID,P}(T) | T \text{ in } T_{min}\}$$

- Why useful?
  - if $|A|<k$ for A in A_{min}, then $|\Pi_{ID,P}(T)| < k$ for a T in T_{min}
  - if $|\Pi_{ID,P}(T)| < k$ for T in T_{min}, then $|A|<k$ for an A in A_{min}.
    - $\Pi_{ID,P}(T)$ is an association cover by definition
    - minimality of A_{min}

Basic mechanism

- *A projection fact* (PF) is a tuple in a view (q_i, r_i) in the view set v
- *Tuple Set* for a PF p in a view (q_i, r_i) in v is the set of all the tuples t in I^v such that $q_i(t) = p$.
- *u(p):* the tuple set for PF p

Basic mechanism

$$T_{min} \subseteq \{u(p) \mid p \text{ is a } PF\}$$

One more… we are there

- Given a tuple p in a view (q_i, r_i) (of a view set v with n views)
- *u(p)* can be computed as the intersection of the following n sets
  - All the tuples t that returns p with q_i
  - All the tuples that returns a FP in the first remaining view, and tuples that do not satisfy the selection condition of that first remaining view,
  - …
  - All the tuples that returns a FP in the last remaining view, and tuples that do not satisfy the selection condition of that last remaining view.

Going back to an example

$v_1 = \Pi_{Name,Job}(P_1)$

Name	**Job**
George	Manager
John	Manager
Bill	Lawyer

$v_2 = \Pi_{Job,Problem}(P_1)$

Job	**Problem**
Manager	Cold
Manager	Obesity
Lawyer	HIV

Let *p*=(George, Manager), then *u(p)* consists of all the tuples that project to *p*, and project to a tuple in v_2 *(note there is no selection condition).*

Therefore, *u(p)* ={ (George, Manager, Cold),

(George, Manager, Obesity) }

Going back to example 2

$\Pi_{Name}\sigma_{Salary>80K}(P_1)$

Name
John
Bill

Name
George
John

$\Pi_{Name}\sigma_{Salary<105K}(P_1)$

$\Pi_{Problem}\sigma_{80K<Salary<100K}(P_1)$

Problem
Obesity

Let *p*=(Obesity) in the right view, then *u(p)* consists of all the tuples that satisfy:

- 80K < salary < 105K,
- name=John or name=George (due to middle view; note selection condition must be satisfied).
- name = John or name=Bill (due to the left view)

Hence: u(p)={(John, s, Obesity)} where 80K<s<100K.

The algorithm

- Represent tuple sets for each projection fact as a formula (from selection condition, or it's complement)
- Perform all the intersections as indicated earlier
- Count the number of possible tuples in each intersection.
- Complexity: basically $|v|^n$, where n is the number of views and $|v|$ is the number of tuples in each view (data complexity).

With FDs

- Some special cases based on observations on FDs
- Consider two views in the view set
  - If an FD does not contain attributes from *both* views, then we can safely ignore this FD.
  - If the two view do not have common attributes and there is a single FD $ID \rightarrow P$, then checking is easy.
  - If the single FD is not ID→P, checking is NP-complete.

Outline

- Motivation
- Problem definition
- Complexity of the general problem
- Polynomial cases
- ***Conservative checking methods***
- Conclusion

Conservative algorithms

- Let $S_a(I) = \sigma_{ID=a}(\Pi_{ID,P}(I))$
- b_1 and b_2 are symmetric for a: Given (a, b_1) and (a, b_2) in $ID^D \times P^D$, if exactly one of the two is in $S_a(I)$, where I is in I^v, then there is I' in I^v such that $S_a(I')$ differs from $S_a(I)$ only in having the other association (among (a, b_1) and (a, b_2)).

K-anonymity

- Given a view set v and a value a in ID^D, v does not violate k-anonymity for a, if there exists I in I^v, such that the following condition is satisfied: For each association (a, b) in $S(I)$, there exists a set of $k-1$ distinct values b_i such that b_i is symmetric to b for a and (a, b_i) is not in $S(I)$.

Going back to example

$v_1 = \Pi_{Name,Job}(P_1)$

Name	**Job**
George	Manager
John	Manager
Bill	Lawyer

$v_2 = \Pi_{Job,Problem}(P_1)$

Job	**Problem**
Manager	Cold
Manager	Obesity
Lawyer	HIV

Given ID value John, Cold and Obesity are symmetric. Then for John, 2-anonymity is NOT violated.

Conclusion & future work

- Introduced k-anonymity violation for views
- Showed computational hardness of the problem
- Gave a polynomial algorithm for a no-FD case
- Provided a general approach for conservative algorithms
- Future work
  - Value obfuscation with views?
  - Experiments?
  - Duplicate preserving projection?
  - More complex views?