K-Anonymity

REU Summer 2007

Advisors: Ryan Williams and Manuel Blum

How do you publicly release a database without compromising individual privacy?

The Wrong Approach:

- Just leave out any *unique* identifiers like name and SSN and hope that this works.
- The triple (DOB, gender, zip code) suffices to uniquely identify at least 87% of US citizens in publicly available databases (Sweeney).
- Moral: Any real privacy guarantee must be proved and established mathematically.

Definitions

- *Database* – a table with n rows (records) and m columns (attributes)
- *Alphabet of a Database* (Σ) – the range of values that individual cells in the database can take.
- Note that the alphabet of the k-anonymized database is $\Sigma \cup \{*\}$

How do you publicly release a database without compromising individual privacy?

- Models: K-Anonymity (Sweeney), Output Perturbation
- K-Anonymity: attributes are suppressed or generalized until each row is identical with at least k-1 other rows. At this point the database is said to be k-anonymous.
- K-Anonymity thus prevents definite database linkages. At worst, the data released narrows down an individual entry to a group of k individuals.
- Unlike Output Perturbation models, K-Anonymity guarantees that the data released is accurate.

Methods for Achieving K-Anonymity

- Suppression – can replace individual attributes with a *
- Generalization – replace individual attributes with a broader category Example: (Age: 26 => Age: [20-30])
- We will be looking at K-Anonymity with suppression

Examples

The following database:

first	last	age	race
Harry	Stone	34	Afr-Am
John	Reyser	36	Cauc
Beatrice	Stone	34	Afr-Am
John	Delgado	22	Hisp

Can be 2-Anonymized with suppression as follows:

first	last	age	race
*	Stone	34	Afr-Am
John	*	*	*
*	Stone	34	Afr-Am
John	*	*	*

Note: Rows 1 and 3 are identical and Rows 2 and 4 are identical

Minimum Cost K-Anonymity

- Obviously, we can guarantee k-anonymity by replacing every cell with a *, but this renders the database useless.
- The cost of K-Anonymous solution to a database is the number of *'s introduced.
- A minimum cost k-anonymity solution suppresses the fewest number of cells necessary to guarantee k-anonymity.

Results

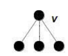
- Minimum Cost 3-Anonymity is NP-Hard for $|\Sigma| = O(n)$ (Meyerson, Williams 2004)
- Minimum Cost 3-Anonymity is NP-Hard for $|\Sigma| = 3$ (Aggarwal et al. 2005)
- Minimum Cost 3-Anonymity is NP-Hard for $|\Sigma| = 2$ (Dondi et al. July 2007)
- We independently proved the same thing this summer.

Theorem: Minimum Cost 3-Anonymity is NP-Hard even with $|\Sigma| = 2$

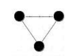
- Lemma 1: There is a polynomial time reduction from the Edge Partition into Triangles and 4-stars problem to binary 3-Anonymity
- Lemma 2: Edge Partition into Triangles and 4-stars is NP-Complete

Triangles and 4-Stars

- A *4-Star* is a simple graph with three edges, all three of which are incident to a common vertex ***v***. ***v*** is called the center of the *4-Star*. The other vertices are called the leaves of the *4-Star*.



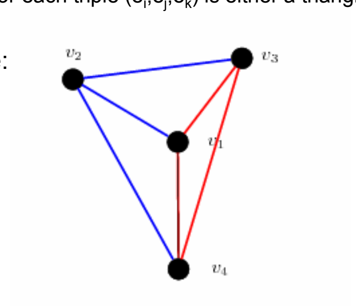
- A *triangle* is the complete graph with three vertices.



Edge Partition into Triangles And 4-Stars

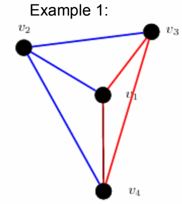
Given a graph G=(E,V) partition the set E into triples (e_i,e_j,e_k) such that for each triple (e_i,e_j,e_k) is either a triangle or a 4-Star.

Example:



Lemma 1: Edge Partition into Triangles and 4-Stars $\leq_p$ Minimum Cost binary 3-Anonymity

Example 1:



	v_1	v_2	v_3	v_4
$\{v_1, v_2\}$	1	1	0	0
$\{v_2, v_3\}$	0	1	1	0
$\{v_2, v_4\}$	0	1	0	1
$\{v_1, v_3\}$	1	0	1	0
$\{v_1, v_4\}$	1	0	0	1
$\{v_3, v_4\}$	0	0	1	1

	v_1	v_2	v_3	v_4
$\{v_1, v_2\}$	*	1	*	*
$\{v_2, v_3\}$	*	1	*	*
$\{v_2, v_4\}$	*	1	*	*
$\{v_1, v_3\}$	*	0	*	*
$\{v_1, v_4\}$	*	0	*	*
$\{v_3, v_4\}$	*	0	*	*

Claim: Database can be 3-Anonymized using exactly 3 *'s per column ⇔ G can be edge partitioned into triangles and 4-Stars.

Lemma 1: Edge Partition into Triangles and 4-Stars $\leq_p$ Minimum Cost binary 3-Anonymity

Example 2:

	v_1	v_2	v_3	v_4
$\{v_1, v_2\}$	1	1	0	0
$\{v_2, v_3\}$	0	1	1	0
$\{v_3, v_4\}$	0	0	1	1

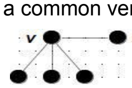
	v_1	v_2	v_3	v_4
$\{v_1, v_2\}$	*	*	*	*
$\{v_2, v_3\}$	*	*	*	*
$\{v_3, v_4\}$	*	*	*	*

Lemma 2: Exactly One In Three SAT $\leq_p$ Edge Partition into Triangles And 4-Stars

- Exactly One In Three Sat: Given a formula ϕ whose clauses each contain 3 variables, is there an assignment such that each clause contains exactly one true variable?
- Exactly One In Three SAT is known to be NP-Complete.
- Given a formula ϕ we construct a triangle free graph G_ϕ such that $E(G_\phi)$ can be partitioned into 4-Stars $\Leftrightarrow$ ϕ is satisfiable.
- G_ϕ is constructed from clause gadgets and variable gadgets.

Clause Gadget

- A *5-Star* is a simple graph with 4 edges all incident with a common vertex v (the center).

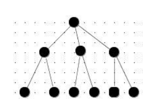


In our usage, v and p are considered *private*, while the other vertices are considered *shared*

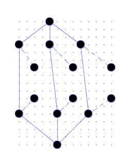
Note: In any 4-Star edge partition of a graph G which contains the clause gadget, v must be the center of exactly one 4-Star since v is the only vertex adjacent to p and has deg(v) = 4. Hence, the 4-Star must use exactly two of the shared edges.

Variable Gadget

- Let d$\in$**N** be given, a *3-Binary Tree* of depth d is a complete tree of depth d where the root has three children and all other nodes have two children.

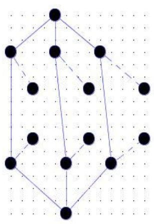


- Let d$\in$**N** be given, G_d is the graph formed by taking two 3-Binary trees of depth d, deleting 3 leaf nodes from each and adding 3 edges between the parents of the deleted leaf nodes so that each parent node still has degree 3.



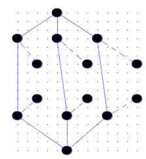
Lemma 2: Exactly One In Three SAT $\leq_p$ Edge Partition into Triangles And 4-Stars

- G_d is a gadget corresponding to each variable, the leaf vertices are consider *shared*, while all other vertices are considered *private*

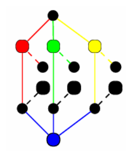


Lemma 2: Exactly One In Three SAT $\leq_p$ Edge Partition into Triangles And 4-Stars

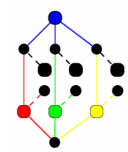
- Motivation: In any 4-Star edge partition P of a graph G which contains G_d, if any of the *shared* vertices on the top (bottom) 3-Binary Tree are the leafs of a 4-Star in P then all of the *shared* vertices on top are leaves of a 4-Star in P and all of the shared vertices on the bottom (top) are the center of a 4-Star in P. Accordingly, we can say that G_d is true (false) partitioned.



True Partition



False Partition



Lemma 2: Exactly One In Three Sat $\leq_p$ Edge Partition into Triangles And 4-Stars

Proof Motivation:

Given a formula ϕ with variables $x_1,\ldots,x_n$ and clauses $c_1,\ldots,c_n$, we can build a graph G using clause and variable gadgets such that any partition of G into 4-Stars corresponds to a satisfying assignment of ϕ and vice versa.

Is Minimum Cost 2-Anonymity NP-Hard?

- Without loss of generality, a 2-Anonymization partitions the rows into doubles and triples. Larger groups of rows could be split into smaller subgroups.
- Intuition 1: Minimum Weight Matching is easy and triples can only increase the number of stars per row.
- Problem: In some cases it is actually beneficial to use groups of three. Example:

10000000000…
00000000000…
10000000000…
11111111111…
01111111111…
11111111111…

****000000000…**
****000000000…**
****000000000…**
****111111111…**
****111111111…**
****111111111…**

Theorem: 2-Anonymity is in P

- We can reduce a 2-Anonymity instance to the Simplex Matching Problem
- Anshelevich and Karagiozova just showed that there is a polynomial time algorithm to solve Simplex Matching (STOC, 2007)

Simplex Matching

Given a hypergraph H with hyperedges of size 2 and 3, and a cost function C(e) such that:

1. $(u,v,w) \in E(H) \rightarrow (u,v),(v,w),(u,w) \in E(H)$
2. $C(u,v) + C(u,w) + C(v,w) \leq 2\, C(u,v,w)$

Find the minimum cost node partition into hyperedges

2-Anonymity $\leq_p$ Simplex Matching

- Given a database D, build a hypergraph H with a node v_i for each row r_i.
- Let $C_{i,j,}$ denote the number of *'s needed to anonymize the rows r_i, r_j. Similarly, define $C_{i,j,k}$.
- For every pair of rows (r_i,r_j) add a hyperedge $e_{i,j}$ with cost $C(e_{i,j})=C_{i,j}$
- For every triple (r_i,r_j,r_k) add a hyperedge $e_{i,j,k}$ with $C(e_{i,j,k})=C_{i,j,k}$

Do the Simplex Conditions Apply?

- $(u,v,w) \in E(H) \rightarrow (u,v),(v,w),(u,w) \in E(H)$
  Because E(H) contains every pair.
- Note that adding an extra row to a double can only increase the number of *'s per row.

$$\frac{1}{3}C_{i,j,k} \geq \frac{1}{2}C_{i,j}, \frac{1}{2}C_{j,k}, \frac{1}{2}C_{i,k}$$

Therefore,

$$2C_{i,j,k} \geq C_{i,j} + C_{j,k} + C_{i,k}$$

2-Anonymity $\leq_p$ Simplex Matching

- Recall that the optimal 2-Anonymity solution partitions the rows into groups of size 2 and 3. Larger groups can be split into smaller groups of size 2 and 3.
- Therefore, the optimal 2-Anonymity solution corresponds to the minimum cost partition of V(H) into hyperedges.
- Because the Simplex Conditions apply we can find the minimum cost partition of V(H) into hyperedges in polynomial time.