Achieving k-Anonmity* Privacy Protection Using Generalization and Suppression

Murat Kantarcioglu

Based on Sweeney 2002 paper

Releasing Private Data

- Problem: Publishing private data while, at the same time, protecting individual privacy
- Challenges:
  - How to quantify privacy protection
  - How to maximize the usefulness of published data
  - How to minimize the risk of disclosure
  - …

Sanitization

- Automated de-identification of private data with certain privacy guarantees
  - Opposed to “formal determination by statisticians” requirement of HIPAA
- Two major research directions
  1. Perturbation (e.g. random noise addition)
  2. Anonymization (e.g. k-anonymization)

Anonymization

- HIPAA revisited
  - Limited data set: no unique identifiers
- Safe enough?
  - Was not for the Governor of Massachusetts#
  - %87 of US citizens can possibly be uniquely identified using ZIP, sex and birth date #

 # L. Sweeney, "k-Anonymity: A Model for Protecting Privacy", International Journal on Uncertainty,

Anonymization

- Removing unique identifiers is not sufficient
- Quasi-identifier (QI)
  - Maximal set of attributes that could help identify individuals
  - Assumed to be publicly available (e.g., voter registration lists)

Anonymization

- As a process
  1. Remove all unique identifiers
  2. Identify QI-attributes, model adversary's background knowledge
  3. Enforce some privacy definition (e.g. k-anonymity)

k-Anonymity

- Each released record should be indistinguishable from at least (k-1) others on its QI attributes
- Alternatively: cardinality of any query result on released data should be at least k
- k-anonymity is (the first) one of many privacy definitions in this line of work
  - l-diversity, t-closeness, m-invariance, delta-presence...

Hardness

- Given some data set *R* and a QI *Q*, does *R* satisfy k-anonymity over *Q*?
  - Easy to tell in polynomial time, NP!
- Finding an *optimal* anonymization is not easy
  - NP-hard: reduction from k-dimensional perfect matching*
  - A polynomial solution implies P = NP
- Heuristic solutions
  - DataFly, Incognito, Mondrian, TDS, …

***A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In PODS'04.**

Tools

- Generalization
  - “Replacing (recoding) a value with a less specific but semantically consistent one”
- Suppression
  - “Not releasing any value at all”
- Advantages
  1. Reveals what was done to the data
  2. Truthful (no incorrect implications)
  3. Trade-off between anonymity and distortion
  4. Adjustable to the recipient’s needs (only one’s)

DGH / VGH

- ZIP attribute

$Z_3=\{*****\}$

$\uparrow$

$Z_2=\{021**\}$

$\uparrow$

$Z_1=\{0213*,0214*\}$

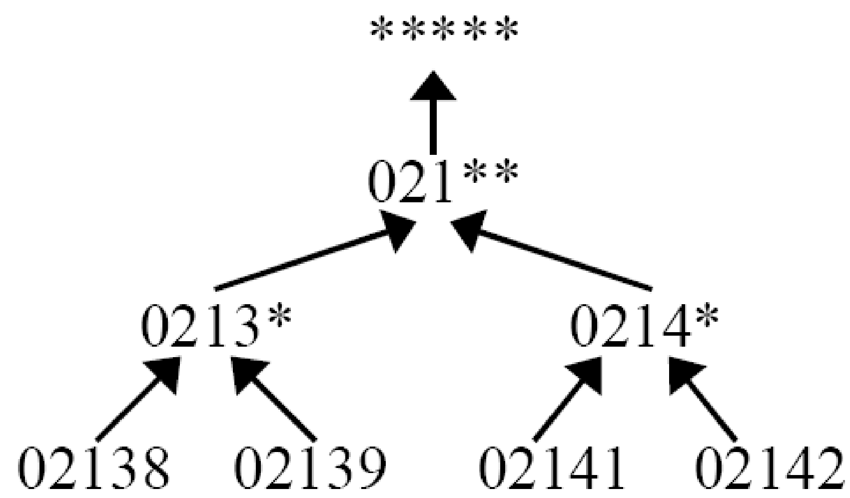
$\uparrow$

$Z_0=\{02138, 02139, 02141, 02142\}$

DGH_{Z0}

Suppression value

Ground domain



VGH_{Z0}

Example

- QI = {Race, ZIP}
- k = 2

- k-anonymous relation should have at least 2 tuples with the same values on

$$Dom(Race_i) \times Dom(ZIP_j)$$

  where $Race_i$ and ZIP_j are chosen from corresponding DGHs

Example

Race E_0	ZIP Z_0
Black	02138
Black	02139
Black	02141
Black	02142
White	02138
White	02139
White	02141
White	02142

PT

Race E_1	ZIP Z_0
Person	02138
Person	02139
Person	02141
Person	02142
Person	02138
Person	02139
Person	02141
Person	02142

$GT_{[1,0]}$

Race E_1	ZIP Z_1
Person	0213*
Person	0213*
Person	0214*
Person	0214*
Person	0213*
Person	0213*
Person	0214*
Person	0214*

$GT_{[1,1]}$

Race E_0	ZIP Z_2
Black	021**
Black	021**
Black	021**
Black	021**
White	021**
White	021**
White	021**
White	021**

$GT_{[0,2]}$

Race E_0	ZIP Z_1
Black	0213*
Black	0213*
Black	0214*
Black	0214*
White	0213*
White	0213*
White	0214*
White	0214*

$GT_{[0,1]}$

k-Minimal Generalization

- Given |R| $\geq$ k, there is always a trivial solution
  - Generalize all attributes to VGH root
  - Not very useful if there exists another k-anonymization with higher granularity (more specific) values
- k-minimal generalization
  - Satisfies k-anonymity
  - None of its specializations satisfies k-anonymity
  - E.g., [0,2] is not minimal, since [0,1] is k-anonymous
  - E.g., [1,0] is minimal, since [0,0] is not k-anonymous

Precision Metric, *Prec(.)*

- Multiple k-minimal generalizations may exist
  - E.g., [1,0] and [0,1] from the example
- Precision metric indicates the generalization with minimal information loss, maximal usefulness
  - Informally, since *Prec* is not based on entropy

- Problem: how to define usefulness

Precision Metric, *Prec(.)*

- Precision: average height of generalized values, normalized by VGH depth per attribute per record
- N_A: number of attributes
- $|PT|$: data set size
- $|DGH_{Ai}|$: depth of the VGH for attribute A_i

$$Prec(\mathrm{RT}) = 1 - \frac{\sum_{i=1}^{N_A} \sum_{j=1}^{N} \frac{h}{|\mathrm{DGH}_{Ai}|}}{|\mathrm{PT}| \bullet |N_A|}$$

Precision Metric, *Prec(.)*

- Notice that precision depends on DGH/VGH
- Different DGHs result in different precision measurements for the same table
- Structure of DGHs might determine the generalization of choice
- DGHs should be semantically meaningful
  - I.e., created by domain experts

k-Minimal Distortion

- Most precise release that adheres to k-anonymity
- Precision measured by *Prec(.)*
- Any k-minimal distortion is a k-minimal generalization

- In the example, only [0,1] is a k-minimal distortion
  - [0,0] is not k-anonymous
  - [1,0] and others are less precise

MinGen Algorithm

- Steps:
  - Generate all generalizations of the private table
  - Discard those that violate k-anonymity
  - Find all generalizations with the highest precision
  - Return one based on some preference criteria
- Unrealistic
  - Even with attribute level generalization/suppression, there are too many candidates

MinGen Algorithm

- Attribute level – global recoding

$$\prod_{i=1}^{n} \left(\left| \mathrm{DGH}_i \right| + 1 \right)$$

- Cell (tuple) level – local recoding

$$\prod_{i=1}^{n} \left(\left| \mathrm{DGH}_{Ai} \right| + 1 \right)^{\left| \mathrm{PT} \right|}$$

MinGen Algorithm

Input: Private Table **PT**; quasi-identifier QI = (A_1, …, A_n), k constraint; domain generalization hierarchies DGH_{A_i}, where i=1,…,n, and *preferred()* specifications.

Output: MGT, a minimal distortion of PT[QI] with respect to k chosen according to the preference specifications

Assumes: |PT |≥ k

Method:

1. **if PT**[QI] satisfies k-anonymity requirement with respect to k **then do**
   1.1. MGT ← { **PT** } // **PT** is the solution
2. **else do**
   2.1. *allgen* ← {T_i : T_i is a generalization of **PT** over QI}
   2.2. *protected* ← {T_i : $T_i \in$ *allgen* ∧ T_i satisfies k-anonymity of k}
   2.3. MGT ← {T_i : $T_i \in$ *protected* ∧ there does not exist $T_z \in$ *protected* such that *Prec*(T_z) > *Prec*(T_i) }
   2.4. MGT ← **preferred**(MGT) // select the preferred solution
3. **return** MGT

DataFly Algorithm

- Steps:
  - Create equivalences over the Cartesian product of QI attributes
  - Heuristically select an attribute to generalize
  - Continue until < k records remain (suppression)

- Too much distortion due to attribute level generalization and greedy choices
- k-anonymity is guaranteed

DataFly Algorithm

Input: Private Table **PT**; quasi-identifier QI = (A_1, …, A_n),
k constraint; hierarchies DGH_{Ai}, where i=1,…,n.
Output: MGT, a generalization of PT[QI] with respect to k
Assumes: |PT |≥ k
Method:

1. freq ← a frequency list contains distinct sequences of values of PT[QI], along with the number of occurrences of each sequence.
2. **while there exists** sequences in freq occurring less than k times that account for more than k tuples **do**
   - 2.1. **let** A_j be attribute in freq having the most number of distinct values
   - 2.2. freq ← generalize the values of A_j in freq
3. freq ← suppress sequences in freq occurring less than k times.
4. freq ← enforce k requirement on suppressed tuples in freq.
5. **Return** MGT ← construct table from freq

μ -Argus Algorithm

- Steps:
  - Generalize until each QI attribute appears k times
  - Check k-anonymity over 2/3-combinations
  - Keeps generalizing according to data holder's choices
  - Suppress any remaining *outliers*

- k-anonymity is not guaranteed
- Faster than DataFly

μ -Argus Algorithm

Input: Private Table **PT**; quasi-identifier QI = (A_1, …, A_n), disjoint subsets of QI known as *Identifying*, *More*, and *Most* where QI = *Identifying* $\cup$ *More* $\cup$ *Most*, *k* constraint; domain generalization hierarchies DGH_{Ai}, where i=1,…,n.

Output: MT containing a generalization of PT[QI]

Assumes: |PT |≥ *k*

Method:

1. freq ← a frequency list containing distinct sequences of values of PT[QI], along with the number of occurrences of each sequence.
2. Generalize each $A_i \in$ QI in freq until its assigned values satisfy *k*.
3. Test 2- and 3- combinations of *Identifying*, *More* and *Most* and **let** *outliers* store those cell combinations not having *k* occurrences.
4. Data holder decides whether to generalize an $A_j \in$ QI based on *outliers* and if so, identifies the A_j to generalize. freq contains the generalized result.
5. **Repeat** steps 3 and 4 until the data holder no longer elects to generalize.
6. Automatically suppress a value having a combination in *outliers*, where precedence is given to the value occurring in the most number of combinations of *outliers*.

What's Next?

- l-Diversity: homogenous distribution of sensitive attribute values within anonymized data

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Japanese Umeko has viral infection

Neighbor Bob has cancer

UTD Anonymization Library

- Contains 5 different methods of anonymization
- Soon to come:
  - Support for 2 other anonymity definitions
  - Integration with Weka
  - Perturbation methods