

# *l*-Diversity: Privacy Beyond *k*-Anonymity

Presenter: Scott Wolchok



# The Problem

- Want to release data for researchers
- Attackers repeatedly de-anonymize data (and make us look bad)
- Need better assurances of privacy



# Quasi-Identifiers

- Sets of attributes that can be linked with background knowledge to identify people
- Example: (gender, birthday, ZIP code) can link voter registration with medical data
- In general, surprisingly-unique attrs



# $k$ -Anonymity

- Previous state of the art
- Each tuple indistinguishable from  $k-1$  other tuples for all sets of quasi-identifiers

# $k$ -anonymity example

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

Figure 1. Inpatient Microdata

	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

Figure 2. 4-anonymous Inpatient Microdata



# Homogeneity Attacks

- If Alice knows that Bob is in a  $k$ -anonymous group where every node has the same sensitive value,  $k$ -anonymity doesn't help!
- Example: the cancer group



# Background Knowledge Attack

- If Alice knows that Umeko is Japanese, Japanese have low heart disease rates, and that her group has either a virus or heart disease, *k*-anonymity doesn't help either!



# Generalization

- This paper considers only generalization as a privacy-preserving measure
- Defn: Partition the values and only release which partition each value was in
- Defines a partial order and thus a lattice; bottom is unmodified, top is fully general



# Threat Model

- Adversary knows domains of attributes and how we anonymized the data
- May know some individuals in table, possibly including sensitive attrs
- May know distribution of attrs in population
- Nothing of form “if Alice has cancer, Bob has heart disease”



# Key Contributions

- Definition of Bayes-optimal Privacy
- Formal definition of  $l$ -diversity
- $l$ -diversity computable with  $k$ -anonymity algorithms



# Types of Disclosure

- Positive: attacker identifies sensitive value with high probability
- Negative: attacker rules out sensitive value with high probability



# Bayes Background

- Attacker's *prior belief* is  
 $P(\text{sensitive attr.} = \text{particular value})$
- Attacker's *posterior belief* is  
 $P(\text{sensitive attr.} = \text{particular value} \mid \text{released table})$
- The difference is related to the information gained from the release



# The Uninformative Principle

- “The published table should provide the adversary with little additional information beyond the background knowledge. In other words, there should not be a large difference between the prior and posterior beliefs.”



# Bayes-Optimal Privacy

- A privacy definition is *Bayes-optimal* if it follows the uninformative principle and bounds the change in the attacker's prior and posterior probabilities as a result of viewing the released table.
- $P(\text{Alice has cancer})$
- $P(\text{Alice has cancer} \mid \text{anonymized data})$



# Drawbacks

- Hard to use in practice
- Publisher has many unknowns:
  - Distribution of attributes in the population
  - Adversary's level of background knowledge
  - Classes of adversaries



# $l$ -Diversity

- Defn: a generalized attribute is  $l$ -diverse if it has at least  $l$  well-represented values for each sensitive attribute. A table is  $l$ -diverse if every attribute is  $l$ -diverse.
- Property: attacker needs  $l - 1$  pieces of background knowledge to deduce sensitive value

# l-Diversity Example

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	$\leq 40$	*	Heart Disease
4	1305*	$\leq 40$	*	Viral Infection
9	1305*	$\leq 40$	*	Cancer
10	1305*	$\leq 40$	*	Cancer
5	1485*	$> 40$	*	Cancer
6	1485*	$> 40$	*	Heart Disease
7	1485*	$> 40$	*	Viral Infection
8	1485*	$> 40$	*	Viral Infection
2	1306*	$\leq 40$	*	Heart Disease
3	1306*	$\leq 40$	*	Viral Infection
11	1306*	$\leq 40$	*	Cancer
12	1306*	$\leq 40$	*	Cancer



# Instantiating $l$ -diversity

- $l$ -diversity is only a principle defining a class of privacy definitions
- Two specific definitions here:
  - Entropy  $l$ -diversity
  - Recursive  $(c, l)$ -diversity



# Entropy $l$ -diversity

- Definition: entropy of dataset  $\geq \log(l)$
- Problem: splitting the table results in blocks with at most the same entropy!
- Can't use this definition if the table does not have entropy  $\log(l)$



# Recursive $(c, l)$ -diversity

- Key property: eliminate most informative sensitive value  $\Rightarrow (c, l-1)$ -diversity
- $c$  is a fudge factor for how bad the worst-case disclosure is
- Def:  $r_i < c(r_i + r_{i+1} + \dots + r_n)$
- $r_i$  is frequency of  $i$ th most frequent value
- $r_i + cr_i < \text{block size}$



# Extensions w/Disclosure

- Extend recursive  $(c, l)$ -diversity with sets of values we don't care about disclosing (like “Healthy”)
- Second extension: prevent negative disclosure by requiring values to show up in  $c$  percent of tuples



# Multiple Sensitive Attributes

- Can't calculate  $l$ -diversity separately for each sensitive attribute
- Must instead calculate for each sensitive attribute, **considering the others as identifiers**
- Requires large blocks, doesn't scale nicely in # sensitive attributes



# Implementing $l$ -diversity

- Algorithm: find a point in the generalization lattice that preserves privacy and is as useful as possible
- Useful property: if a generalization preserves privacy, all of its generalizations do too
- Can use efficient binary search-like algorithms (also for  $k$ -anonymity)



# Evaluation

- Modified *Incognito* by LeFevre et al.
- DBMS = DB2 (same as last paper)
- Adult dataset = 45k U.S. Census tuples
- Lands End = 4.6M POS tuples



# Homogeneity attacks

- 2 of 3 3-anonymous tables in Lands End vulnerable w/large homogeneous groups despite 147 Cost buckets
- 1 of 12 6-anonymous tables in Adult vulnerable w/Occupation
- 8 of 9 6-anonymous tables in Adult vulnerable w/Salary Class
- Non-vulnerable tables were 2+-diverse



# Performance

- Shows that  $l$ -diversity algorithms work and perform reasonably
- On the same order of perf as  $k$ -anonymity,  $< 1$  hour for Lands End



# Problems

- Bayes-optimal privacy is not resilient against attacker w/multiple anonymized views (e.g., automated data release)
- “Piece of background knowledge” may be more complex (disclose  $X \Rightarrow Y$ )



# Conclusion

- $k$ -anonymity is not strong enough
- $l$ -diversity provides a formal definition of privacy for released data
- Future work: guarantees about data utility