Privacy Preserving Data Mining

Presentation by Venkata Subbarao Chunduri and
Vincent Deuschle

Fachgebiet Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

http://www.dima.tu-berlin.de/

Agenda

- Introduction
  - Motivation
  - Examples
- Foundations
  - Randomization
  - K-Anonymity
  - Curse of Dimensionality
- Differential Privacy
- PPDM on Graphs and Social Networks
  - Methods
  - Summary
  - Open Problems
- Frameworks and Implementations
  - PPDM on Graphs with Map Reduce
- Summary
- References

Motivation

- Ability to store personal user data increases
- Data mining algorithms become increasingly sophisticated
- Avoiding identification of user from public data is often desired or legally required
- Even without concrete personal information users are often identifiable by inference
- Tradeoff in data science:
  - data privacy vs. data utility

https://www.internetsociety.org/sites/default/files/09_4-ndss2016-slides.pdf

Anonymization – Intuitive Approach

- For each record: Obfuscate all revealing variables
  - What are ‚revealing' variables? Are they alike in each situation/dataset?
- How much data utility is lost with this approach?

Firstname	Surname	Gender	Address	Zipcode	Birthdate
[redacted]	[redacted]	Male	[redacted]	02138	July 31, 1945

- Is this sufficient?

Anonymization – Historical Example

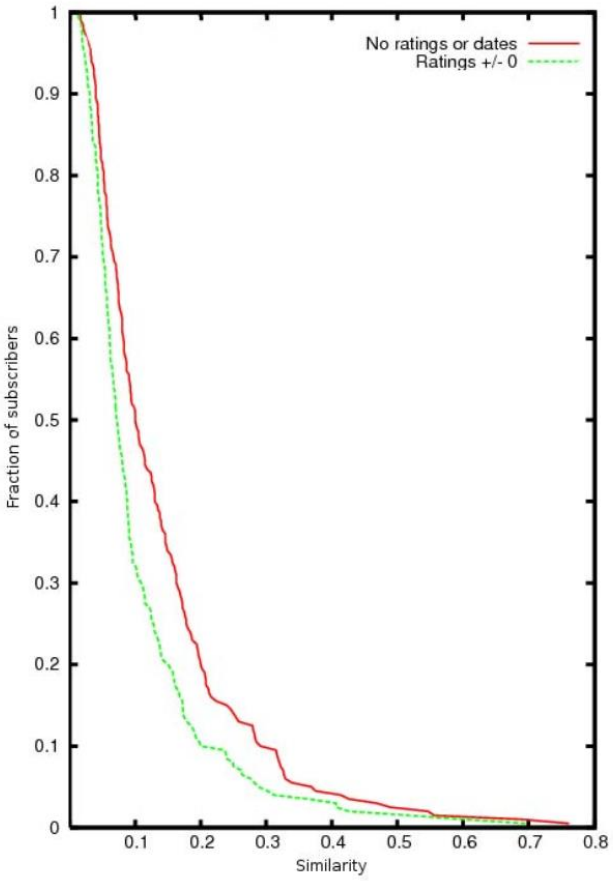
- For each record: Obfuscate all revealing variables
  - What are ‚revealing' variables? Are they alike in each situation/dataset?
- How much data utility is lost with this approach?

Firstname	Surname	Gender	Address	Zipcode	Birthdate
William	Weld	Male		02138	July 31, 1945

- Gov. William Weld has been identified by Prof. Latanya Sweeny in 1997 only by gender, zipcode and birthdate
  - Combination of gender, zipcode and date of birth is unique for 87% of the U.S. population
- Information had been included in (allegedly) anonymized medical records
- Gender, zipcode and birthdate reveal identity but might be relevant for health care analysis
  - Privacy/Utility tradeoff

Anonymization – Netflix Example

- A. Narayanan and V. Shmatikov de-anonymized Netflix userdata in 2008
- Set of userdata is large but high-dimensional
  - 500,000 user records with several thousand features
  - ➢ No two Netflix records are similar more than 50%
- Netflix data can be crossreferenced with other datasets
  - Netflix and IMDB data was joined to extend accessable user data
- Personal information (e.g., political opinions) might be infered from movie preferences



Narayanan, Shmatikov 2008

Foundations - Randomization

- 1. Approach: Add noise to (numerical) data
- Define private dataset as $X = \{x_1, x_2, \dots, x_n\}$
- Draw random noise as $Y = \{y_1, y_2, \dots, y_n\}$ from pdf $f_Y(y)$
  - Variance should be sufficiently large
- Add noise Y to data X to generate public set $Z = \{x_1 + y_1, x_2 + y_2, \dots, x_n + y_n\}$
- N instantiations of Z and distribution of Y are known
  - ➢ Original distribution of X can be approximated without knowing original records
- Distribution of X might be sufficient for several analytical tasks
- Pitfall: Only one-dimensional distributions of individual features can be approximated reliably
- Possible attack: Outliers might be distinguishable even with noise
- ➢ Noise compromises data utility in any case

Foundations – K-Anonymization

- Reduce granularity of data by generalization or repression of informative features
- Define an integer k, so that each record cannot be distinguished from at least k other records in the dataset
- Example: Map birthdate
  - July 31, 1945 → July, 1945
- Computation of optimal k-anonymization is NP-hard
  - Heuristic methods are used in practice
- Applicable to table data and graph data
- Possible attack: As already shown, data might be joined with other datasets to reverse repression or generalization
- Pitfall: Exhaustive assessment of other available data that might be used for cross-referencing
- Data utility is necessarily compromised (to some extent)
- And yet: Frequently used in practice

Foundations – Curse of Dimensionality

- Many privacy preserving methods for data mining loose effectivity in high-dimensional space
  - Data becomes more sparse with increasing dimensions
  - Records become increasingly distinguishable
- K-anonymity becomes harder to achieve
  - Sparsity of data requires higher level of generalization or repression to find k similar records
- Cross-referencing of multiple databases may increase dimensionality even further than intended
- Repression of a substantial amount of features is often required
  - Balance between data utility and data privacy is usually hard to maintain

Differential Privacy

- Given two databases *D* and *D′* such that *D′* = *D U* {*X*} ,i.e. *D* and *D′* differ only by a single item, the probability distributions on the results of *D* and *D′* under differential privacy will be 'same'.

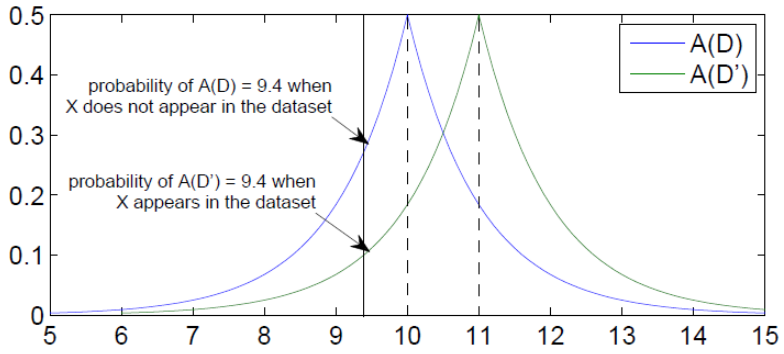


Image: Inria Nancy,2017

- Developed for statistics queries on private datasets.
- Adds noise to the output results.
- It assures one's participation in a dataset is not revealed.
- For example, on a dataset with 100 user records, 80 of with hold certain property P, queries for dataset size, and property P proportion noisy answers could be 102 and 81.5% respectively.
- Apple iOS 10 uses it for usage statistics anonymization. (https://www.youtube.com/watch?v=i5BGgM-E7mM)

Differential Privacy (continued…)

- Non-iterative Settings
  - Curator creates D’ dataset from D by adding noise and shuts down D given ϵ – privacy budget
  - ϵ – decides noise amount
  - Exposes D’ for unlimited queries
- Iterative Settings
  - Limited number of queries allowed on dataset
  - With ϵ -privacy budget, (q_i, ϵ_i) with $\sum_i \epsilon_i \leq \epsilon$
  - After every query privacy budget drops(more noisy answers)
  - If privacy budged reaches zero, no more queries allowed.

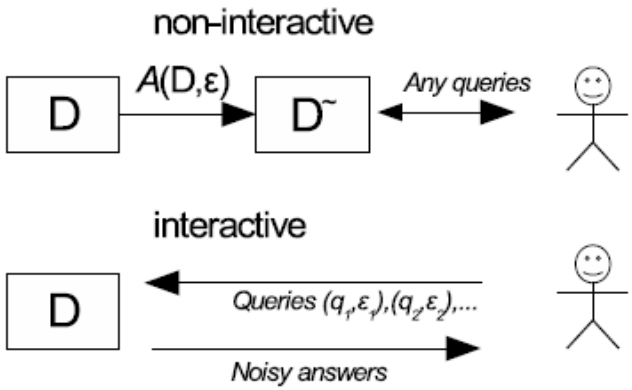


Image: Inria Nancy,2017

The lower the ϵ , higher the privacy.

Differential Privacy Criticism

- How much Noise?
  - **Global sensitivity:** Difference of results between adjacent datasets.
  - Higher global sensitivity results in more noisy results.
  - Sensitivity can be unbounded(average salary queries).
- Privacy vs Utility loss.
  - Differential privacy loses utility for the privacy it gains.
- Privacy budgeting.
  - Needs trying with different privacy budgets on data.
- More in-depth Criticism on Differential privacy(https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2326746)

Privacy Challenges with Graph Mining

- Any topological structure can be exploited
- Graph brings more dimensions like relationships, betweenness, degree, closeness centrality etc.
- Difficult to measure information loss and no standard quantifying measures.
- Impact of modifying edges (relationships) or nodes can spread across the graph.
  - Graph data is more dependent than tabular data

Privacy Breaches in Graphs

What can go wrong?

- Identity Disclosure
  - Revelation of individual identity associated with node.
- Link Disclosure
  - Revelation of sensitive relationships between users.
- Content Disclosure
  - Revelation of data associated with node or exchanged among nodes.

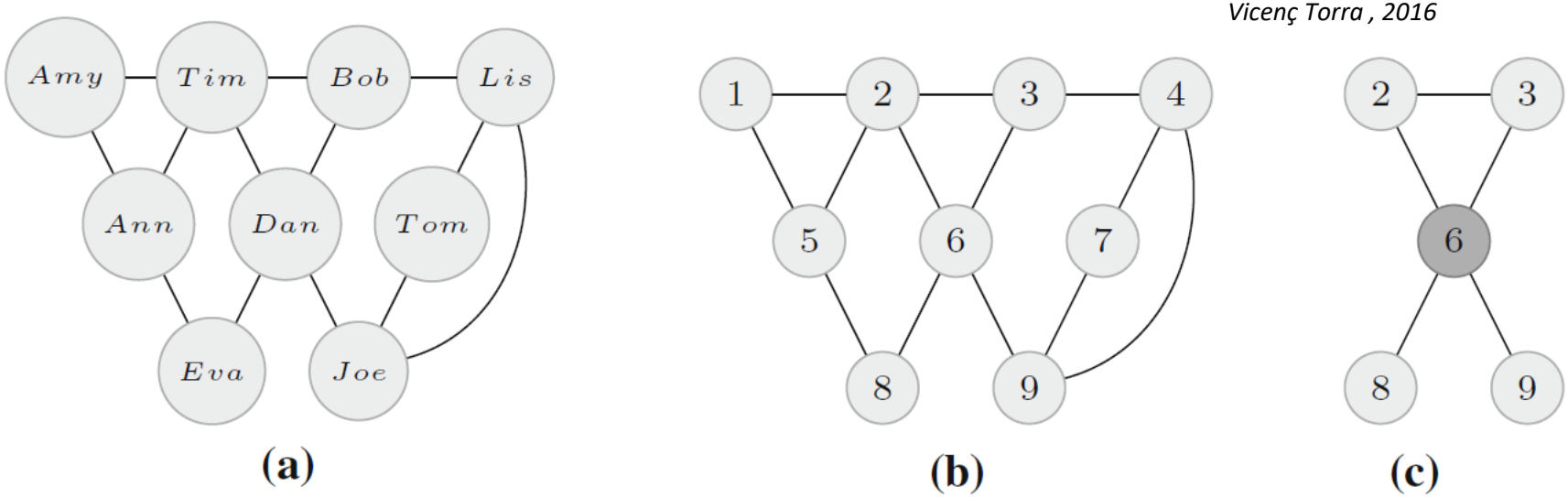
Types of Attacks:

- Active attacks
  - Create subnetwork before data is published.
- Passive attacks
  - Use knowledge about individual to re-identify.

Naïve Anonymization

Removes identity information from graph
- Known degree attacks

Image: *Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra , 2016*



a) Graph with identity information.
b) Graph without identity information.
c) Dan's 1-Neighborhood subgraph.

Graph Modification Techniques

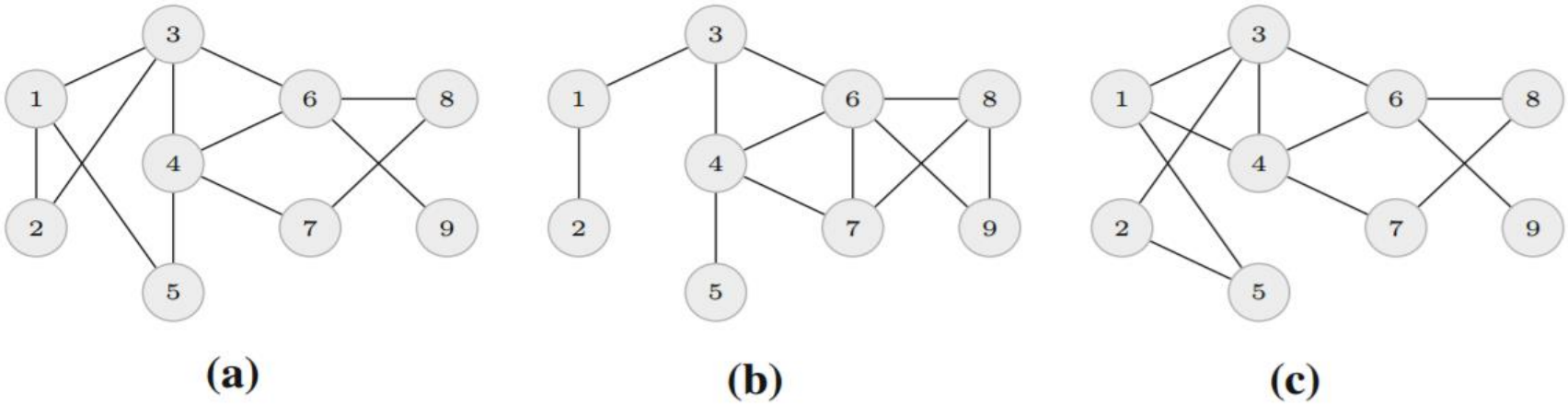
- Edge and vertex modification
  - Remove/add/rotate (or all) vertices and edges randomly (Randomization, random perturbation)
- Uncertain graphs
  - Add edges 'partially' with assigned probabilities
- Generalization or clustering-based approach
  - Group vertices and edges into super-vertices and super-edge groups

All the above methods transform data with different types of graph modification, then release data for unconstrained analysis

On the other hand, 'privacy-aware methods' such as differential privacy does not release data but only output of the computation.

Edge and Vertex Modification

Random Perturbation:



a) Original graph
b) Random edge deletions and additions
c) Random edge switch
- Any observable problems with random switch?

Image: *Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra , 2016*

Advantages:
- Most simple approach
- Lowest complexity thus scales well for Big Data

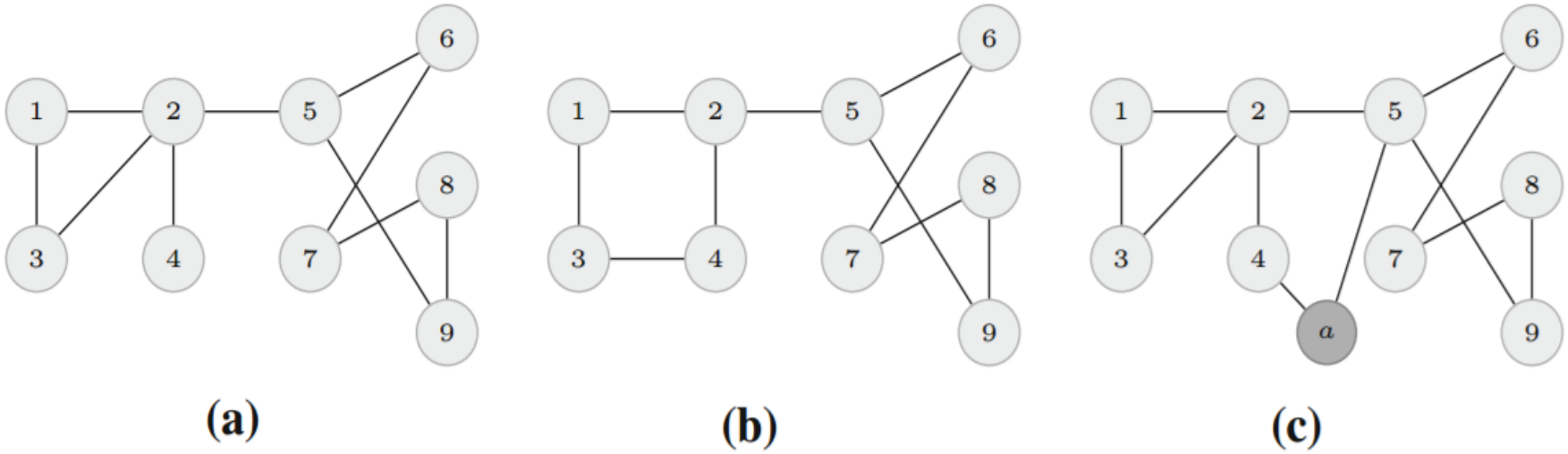
Disadvantage:
- No privacy guarantees

Edge and Vertex Modification

Constrained perturbation:

k-degree anonymity

Image: *Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra , 2016*



- Probability of re-identification not greater than 1/k
- Other variations- k-neighborhood, k-candidate, k-automorphic, etc.

Uncertain Graph

- Partial additions with some probability
- Partial probabilities add noise required for privacy

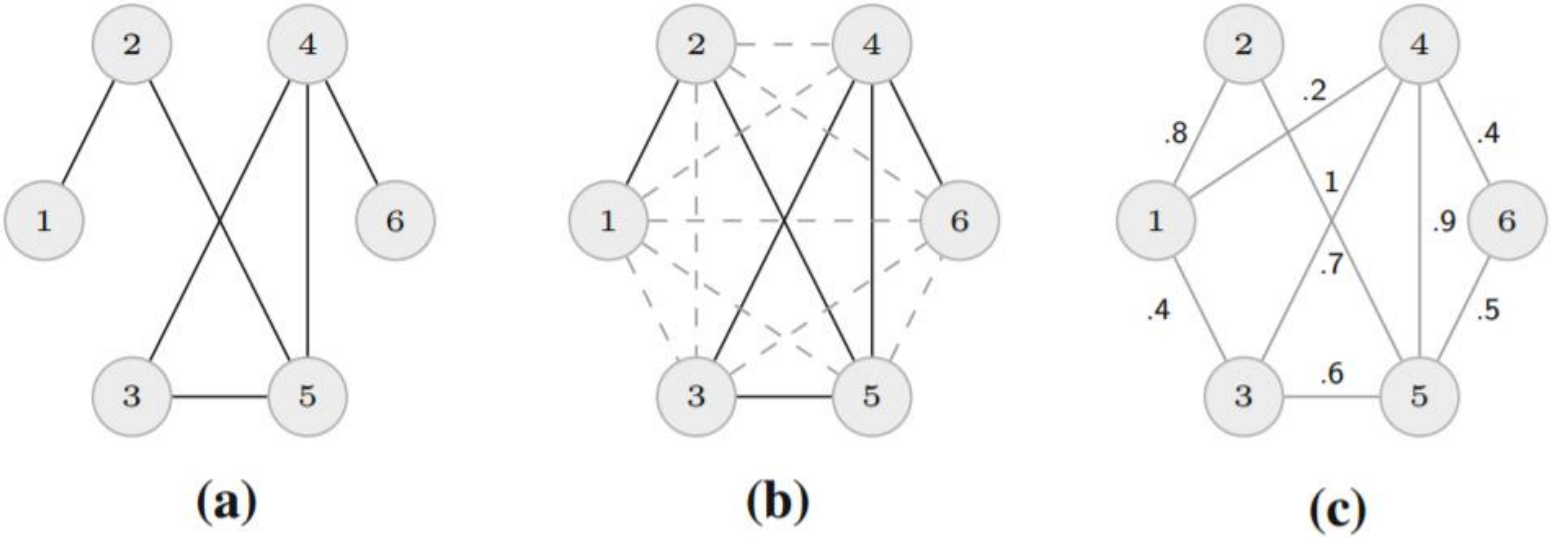


Image: *Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra , 2016*

Generalization Approaches

- Also called clustering-based approach
- Partitions vertices and edges into super-vertices and super-edges

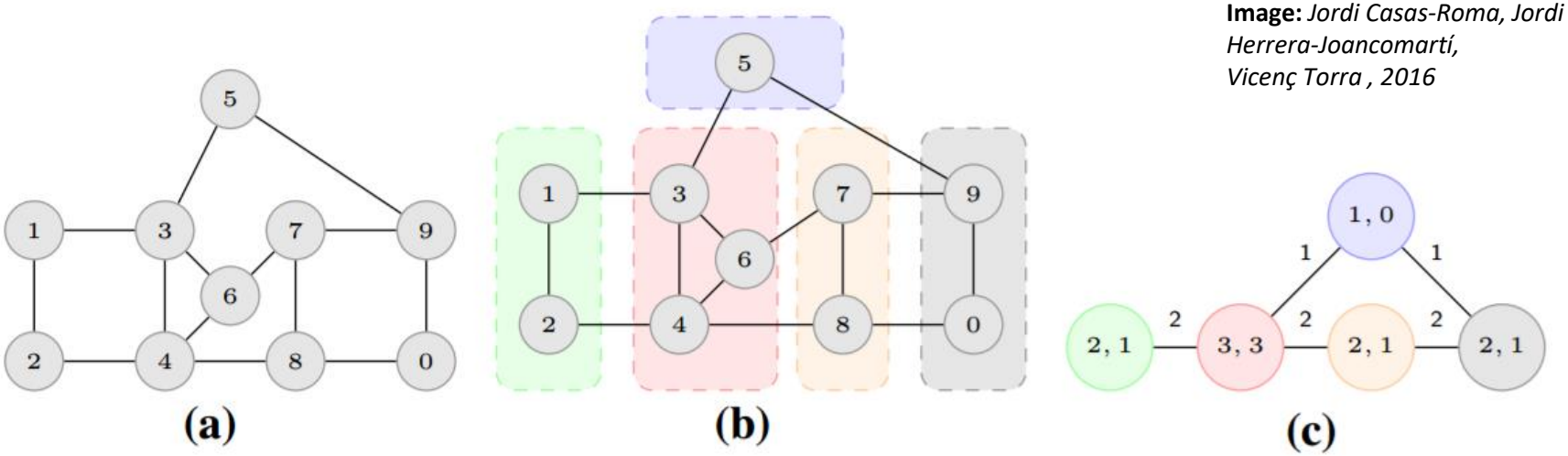


Image: *Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra , 2016*

- Individual details can be hidden properly but graph could shrink more
- Useful for macro-properties (aggregations)
- Increased privacy with decreased utility

Summarization of Techniques

Technique	Graph type	Disclosure	Background	Method	Characteristics	References
Random perturbation	Simple, undirected	Identity	Vertex degree	Randomization	Edge modification	Hay et al. (2007)
	Simple, undirected	Link	Structural properties	Spectrum preserving	Edge modification	Ying and Wu (2008)
	Simple, undirected	Identity and link	Structural properties	Random sparsification	Edge deletion	Bonchi et al. (2011, 2014)
Constrained perturbation	Simple, undirected	Identity	Vertex degree	k-degree anonymity	Edge modification	Liu and Terzi (2008); Lu et al. (2012); Casas-Roma et al. (2013, 2016)
	Simple, undirected	Identity	Vertex degree	k-degree anonymity	Vertex and edge addition	Chester et al. (2013a)
	Simple, undirected	Identity	Coreness and vertex degree	(k, δ)-core anonymity	Vertex and edge addition	Assam et al. (2014)
	Simple, undirected	Identity	Neighbourhood	k-neighbourhood	Edge modification	Zhou and Pei (2011); Tripathy and Panda (2010)
	Simple, undirected	Identity	Structure properties	k-automorphism	Edge modification	Zou et al. (2009)
	Bipartite	Link	Sensitive edges	(k,ℓ)-grouping	Edge clustering	Cormode et al. (2010)
	Edge-labelled	Identity	Edge attributes	k-anonymity	Linear programming	Das et al. (2010)

Image: *Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra , 2016*

Summarization of Techniques(continued..)

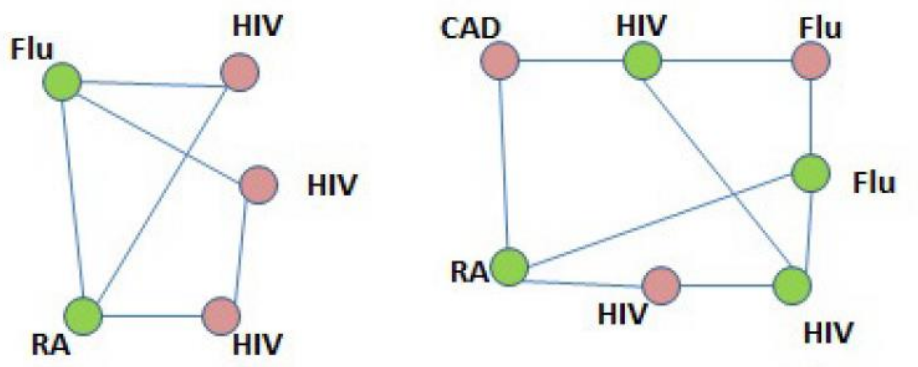
Technique	Graph type	Disclosure	Background	Method	Characteristics	References
Uncertain graphs	Simple, undirected	Identity	Vertex properties	(k, ε)-obfuscation	Partially edge modification	Boldi et al. (2012)
	Simple, undirected	Identity	Vertex degree	Adjacency matrix obfuscation	Partially edge switch	Nguyen et al. (2015)
Generalization	Vertex-labelled	Identity and attribute	Vertex properties	k-anonymity	Vertex and edge clustering	Campan and Truta (2008, 2009)
	Vertex-labelled	Identity and attribute	Sensitive attributes	p-sensitive k-anonymity	Vertex and edge clustering	Ford et al. (2009)

Open Problems

- Anonymization of time-varying graphs/multi-layer graph
- Anonymization of streaming data
- Computability of anonymization techniques to scale to Big Data
- Decentralized anonymization
- Linkability of data from differently anonymized data from different sources for Big Data analysis
- Preserving accuracy in linked datasets

Big Graph Privacy - Overview

- Privacy framework to prevent attribute disclosure in large graphs
- Scenario: Attacker has vertex knowledge (victim node degrees) and pursues victim‘s attribute values
- Concept: Values of sensitive attributes should sattisfy l-diversity for nodes with same degree
- Full graph structure is maintained and published
- Map Reduce is utilized to ensure scalability



(a) *Vulnerable graph* (b) *Satisfying 2-diversity*

Zakerzadeh et al.: 2015

Big Graph Privacy - Notations

- Graph is formally defined as $G = (V, E, S, f)$
- A graph G with degree set D is called an l-diversified graph $\leftrightarrow$ $\forall d \in D$, the set of nodes eq^d with degree d satisfies the l-diversity privacy

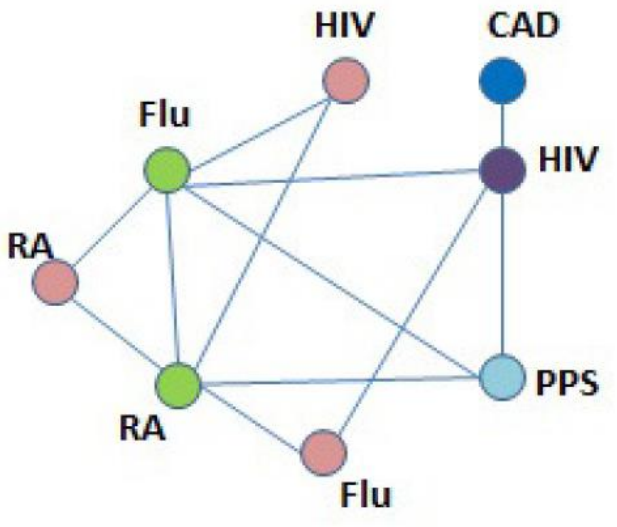
Table 1: List of notations

notation	explanation
v_i	i^{th} $vertex$
$f(v_i)$	function returning the sensitive value of v_i
$deg(v_i)$	degree of vertex v_i
eq^d	set of nodes with degree d (an equivalence class)
N_i	set of immediate neighbors of v_i
S^X	set of sensitive values of set of nodes X
$\|.\|$	size of a set

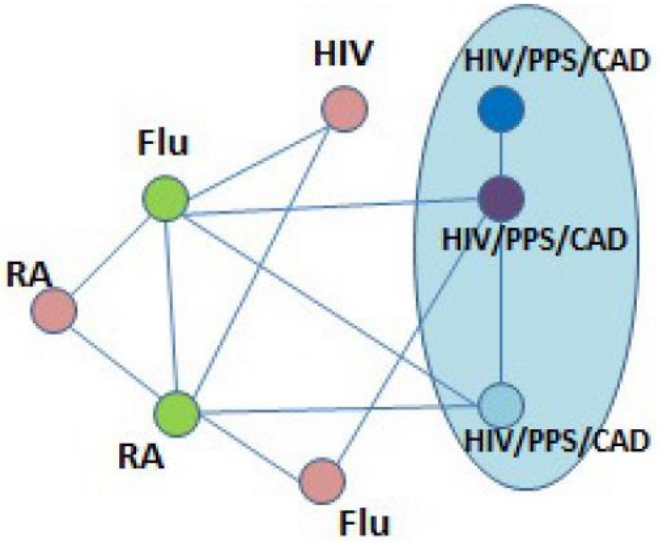
Zakerzadeh et al.: 2015

- An equivalence class eq^d is l-diverse $\leftrightarrow \forall v_i \in eq^d$ and multisets of sensitive values (S^{eq^d}, g) where $S^{eq^d} = \{f(v_i) | v_i \in eq^d\}$ and $g(\cdot)$ returns the frequency of each sensitive value in eq^d, $\forall x \in S^{eq^d}$ the inequality $\frac{g(x, eq^d)}{\Sigma_{y \in S^{eq^d}} g(y, eq^d)} \leq \frac{1}{l}$ holds

Big Graph Privacy - Example



(a) A 2-diversity violating graph



(b) 2-divertised version of (a)

Zakerzadeh et al.: 2015

Big Graph Privacy - Algorithm

- Filter all equivalence classes for nodes that violate l-diversity condition
- Cluster all violating nodes so that each cluster fulfills l-diversity condition
  - Hierarchical clustering (bottom-up) is used
  - Entropy of sensitive values is merge criterion
- Sensitive values are shared amongst all cluster members

Algorithm 1 Big Graph Anonymization Steps

1: $AnonymizationScheme(G)$
2: //G is a simple graph of form (V,E,S,f)
3: $\mathbb{EQ}$= assign nodes with degree d to equivalence class eq^d and form equivalence classes set
4: **foreach** (eq in $\mathbb{EQ}$)
5: **if** (eq does not satisfy the *ℓ-diversity condition)* append nodes $v_i \in eq$ to the violating nodes set VN
6: $\mathbb{C}$=cluster nodes $v_j \in VN$ such that each cluster $c \in \mathbb{C}$ satisfies the *ℓ-diversity condition*
7: define function $f\prime : V \rightarrow S \times \mathbb{N}$ such that
8: **foreach** (v_i in V)
9: **if** (v_i not in VN)
10: $f\prime(v_i) = (f(v_i), 1)$
11: **else**
12: $f\prime(v_i)$ = the multiset of sensitive values of nodes in cluster $c \in \mathbb{C} | v_i \in c$
13: publish $G\prime(V, E, S, f\prime)$

Zakerzadeh et al.: 2015

Big Graph Privacy – MapReduce: Pruning

- Checks if equivalence class satisfies l-diversity condition

Algorithm 2 Neighborhood Discovery Job

```
1: Mapper(k,v)
2:   // v can be either of form (v_i,v_j) or (v_i,f(v_i))
3:   if (input chunk belongs to the relationship file)
4:     emit(v_i,v_j)
5:     emit(v_j,v_i)
6:   else
7:     emit(v_i,f(v_i))
```

Algorithm 3 Neighborhood Discovery Job

```
1: Reducer(k,V)
2:   // k is a vertex and V contains all neighbours and the sensitive
       value f(k)
3:   emit(k/f(k),V)
```

Algorithm 4 Filtering Job

```
1: Mapper(k,v)
2:   //v is of form (v_i/f(v_i), N_i)
3:   emit(|N_i|,<v_i/f(v_i), N_i>)
```

Algorithm 5 Filtering Job

```
1: Reducer(k,V)
2:   //V is a list of (v_i/f(v_i), N_i) where deg(v_i) = d
3:   violation=false
4:   foreach (v_i/f(v_i), N_i) ∈ V
5:     if (freq(f(v_i),S^d) / |S^d| > 1/ℓ)
6:       violation=true
7:       break
8:   if (violation)
9:     foreach (v_i/f(v_i), N_i) ∈ V
10:      emit(v_i/f(v_i), N_i)
```

Zakerzadeh et al.: 2015

Big Graph Privacy – MapReduce: Clustering

- Groups privacy violating verticies and shares sensitive attributes amongst all cluster members

Algorithm 6 Clustering Job

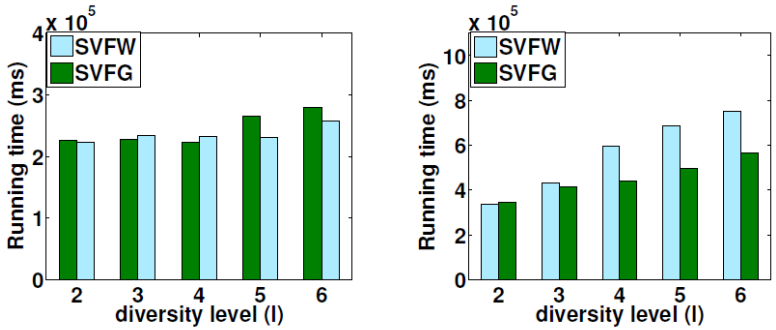
```
1: Mapper(k,v)
2:   // v is of form (v_i/f(v_i), N_i)
3:   append each pair (v_i/f(v_i), N_i) to buffer
4:   if (no more pair)
5:     ℂ = cluster(buffer) //either SVFW or SVFG clustering
6:   for (c in ℂ)
7:     for (v_j in c)
8:       emit(v_j, multiset(c))
```

- SVFW = *‚Sensitive Value Frequency aWare‘* Clustering
- *SVFG = ‚Sensitive Value Frequency aGnostic‘* Clustering

Big Graph Privacy - Benchmarks

- Authors tested their algorithm with Hadoop 1.0.4
- Two testsets:
  - LastFM co-Group Graph: Up to 177000 nodes (users) with over 10 million edges (friendship relations)
  - US Patent Citation Graph: Over 2.9 million nodes (patents) with around 16.5 million edges
- Run on ACENet cluster:
  - 32 nodes with 16 cores, 64 GB RAM and Gigabit Ethernet connection

Figure 9: Running time vs. ℓ



(a) *Patent data set* (b) *LastFM data set*

Zakerzadeh et al.: 2015

Summary

- Most of the tabular anonymization concepts apply to graphs in a modified way.
- Differential privacy guarantees one's participation not revealed.
- Most of current graph anonymization methods do not scale very well to Big Data.
- Frameworks like Map Reduce can increase the scalability of anonymization methods
- Unlimited Privacy and unlimited utility cannot be achieved together.
- Relevance of privacy and utility depends on:
  - Type of data (medical records, geo-locational data, etc.)
  - Environment (legal requirements, prevailing public opinion, etc.)
  - Perspective (user vs. analyst)
  - Resources (how much privacy can analysts guarantee reliably)
  - Application (how important is the retrieval of information in given data)
  - …

References

- A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In Security and Privacy, 2008.
- C. Aggarwal and P. Yu. A General Survey of Privacy-Preserving Data Mining Models and Algorithms. In Privacy-preserving data mining, 2008.
- H. Zakerzadeh, C. Aggarwal. Big Graph Privacy. EDBT/ICDT Workshops, 2015.
- Jordi Casas-Roma, Jordi Herrera-Joancomartí and Vicenç Torra. A survey of graph-modification techniques for privacy-preserving on networks, 2016.
- Kun Liu, Kamalika Das and Tyrone Grandison.Privacy-Preserving Data Analysis on Graphs and Social Networks, 2008.
- Inria Nancy. Social Graph Anonymization, 2017.