Privacy for Computations

Vicenç Torra

February 2024

Umeå University, Sweden

V. Torra (2022) A guide to data privacy, Springer (Chapter 5)

Outline

1. Computation-driven approaches
   - Integral privacy

Introduction

Integral privacy

Privacy models

Privacy models: for computations

- Privacy for re-identification (to data) + computation
- k-Anonymity (to data) + computation
- Differential privacy directly to the computation

We proposed

- Integral privacy

Privacy models

Integral privacy: for a computation or algorithm f

- $f(X)$ is private if there are different ways to reach $f(X)$, i.e., different databases X which are different enough.

Integral privacy

Integral privacy: for computation f.

Some preliminaries ...

- P the population, f be a function or algorithm that given a data set $S \subseteq P$ computes an output $f(S)$ that belongs to another domain $\mathcal{G}$.
- Given G in $\mathcal{G}$, previous knowledge S^* with $S^* \subset P$, the set of possible generators of G is:

$$Gen(G, S^*) = \{S' | S^* \subseteq S' \subseteq P, f(S') = G\}.$$

We use $Gen^*(G, S^*) = \{S' \setminus S^* | S^* \subseteq S' \subseteq P, f(S') = G\}$
(when no information is known on S^*, we use $S^* = \emptyset$

Integral privacy

Integral privacy: for function f, definition:

- P data, $f : S \to \mathcal{G}$, S^* background knowledge, $Gen(G, S^*)$ databases that generate G and are consistent with background knowledge S^*.
  Then, integral privacy is satisfied when $Gen(G, S^*)$ is large and diverse.

Integral privacy

Integral privacy: for function f, definition:

- P data, $f : S \rightarrow \mathcal{G}$, S^* background knowledge, $Gen(G, S^*)$ databases that generate G and are consistent with background knowledge S^*.
  Then, integral privacy is satisfied when $Gen(G, S^*)$ is large=at least k databases and diverse:

$$\cap_{g \in Gen^*(G,S^*)} g = \emptyset.$$

Requirements: why? / what?

- Empty intersection to avoid all generators sharing a record (e.g., avoiding membership inference attacks)
- $Gen(G, S^*)$ large. large = k-flavor.

Integral privacy vs Differential privacy

Integral privacy, and differential privacy

- Differential privacy, smooth function
  $f(D) \sim f(D \oplus x)$ where $D \oplus x$ means to add the record x to D
- Integral privacy, recurrent function
  If $f^{-1}(G)$ is the set of all (real) databases that can generate the output G, we require $f^{-1}(G)$ to be a large and diverse set for G.

Integral privacy vs Differential privacy

Integral privacy, and differential privacy

- Differential privacy, smooth function
  $f(D) \sim f(D \oplus x)$ where $D \oplus x$ means to add the record x to D
- Integral privacy, recurrent function
  If $f^{-1}(G)$ is the set of all (real) databases that can generate the output G, we require $f^{-1}(G)$ to be a large and diverse set for G.

- Simple integrally private function:
  f an algorithm that is 1 if the number of records in D is even, and 0 if the number of records in D is odd.
  That is, $f(D) = 1$ if and only if $|D|$ is even.

Integral privacy vs Differential privacy

Pros and cons:

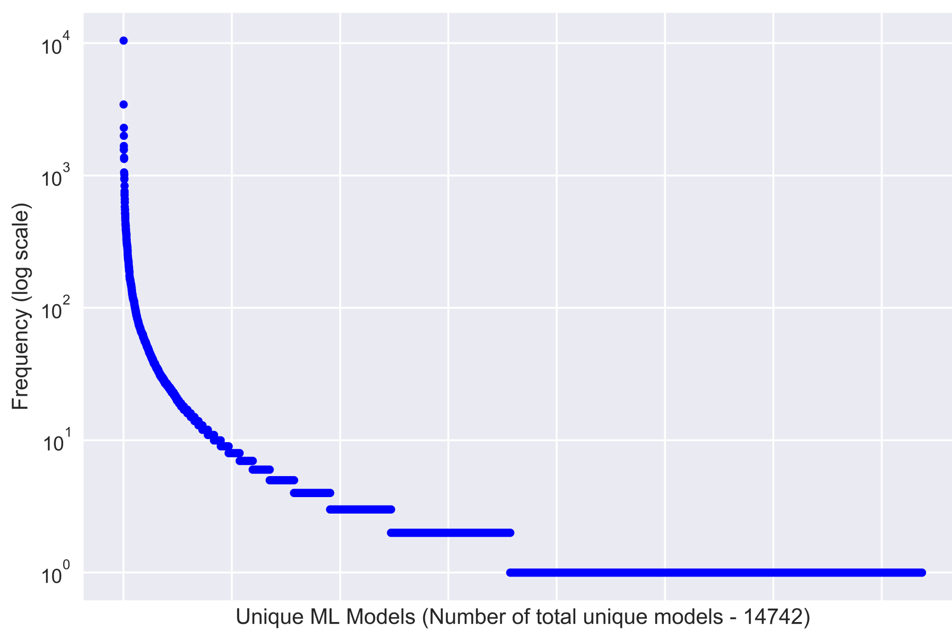
- Cons:
  - If S^* is all population P but a record. Not "strong" guarantees.
- Pros:
  - Integral privacy, and plausible deniability
    - IP satisfies plausible deniability if for any record r in P such that $r \notin S^*$, there is a set/database $\sigma \in Gen^*(G, S^*)$ such that $r \notin \sigma$.
  - Our definition satisfies plausible deniability

Integral privacy

Finding. Recurrent models appear also in machine learning

- **Recurrent models?** Large set of generators
- **Generators?** DB generator of m_1 if $f(DB) = m_1$

Decision trees with Iris dataset. Models/freq.

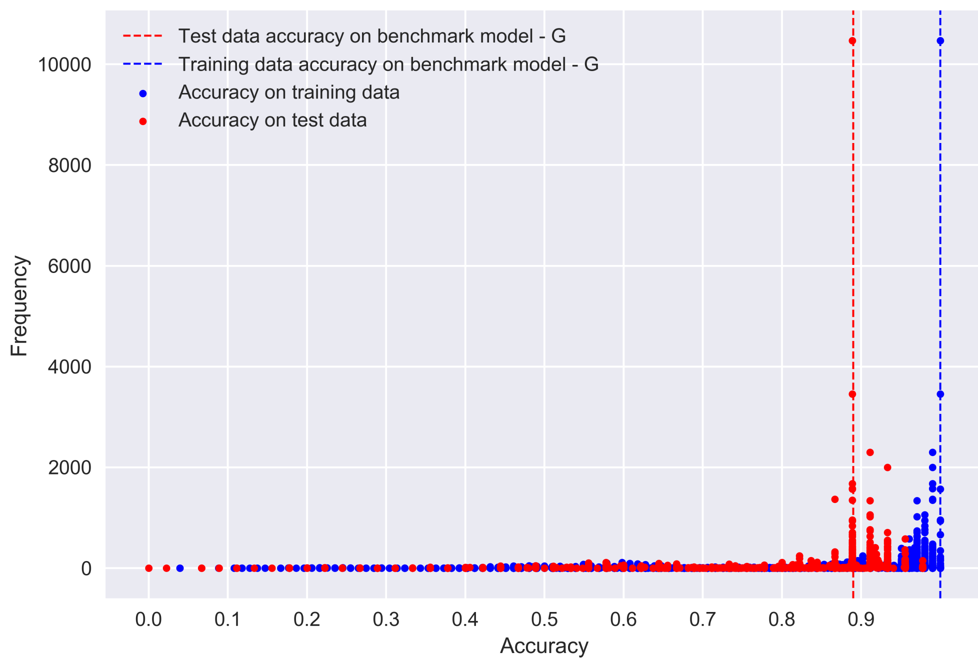


Integral privacy

Finding N. 1. Recurrent models appear also in machine learning

Finding N. 2. Recurrent models may have good accuracy

- accuracy + frequency. DT with Iris. Acc./freq.



Integrally private means

Integral privacy

How to implement IP mean (numerical database)

- Round numbers in the database
  - All number multiples of r
- Sample the database and build subsets
- Compute means of subsets
- Take a frequent mean such that satisfies the privacy constraints
  E.g., at least k generators with empty intersection

Integral privacy

How to implement IP mean (numerical database)

- k is a privacy requirement, and relates to distortion
  - larger k, larger distortion
- Larger r in rounding, larger distortion
- Amount of distortion also depends on the query
  - See mean vs. maximum / minimum
    (to produce the same $maximum$ we will need larger rounding)

Integrally private ML models

Integral privacy

How to implement ML models

- Sampling the database (DT)
  - Create databases from the original database
  - Create models m for each database db ($db =$ generator of m)
  - Compare models and generators
- Partition the database (SVM, DL)
  - Create a database from each part
  - Create models
  - If the models are the same, by construction they satisfy the privacy constraints (or models similar enough)

Integrally private clustering

κ-centroid c-means

Formalization of the problem

Informal description:

- Database X
- Macro-clusters: c
- Micro-clusters: κ

So, $c \times \kappa$ disjoint groups or parts

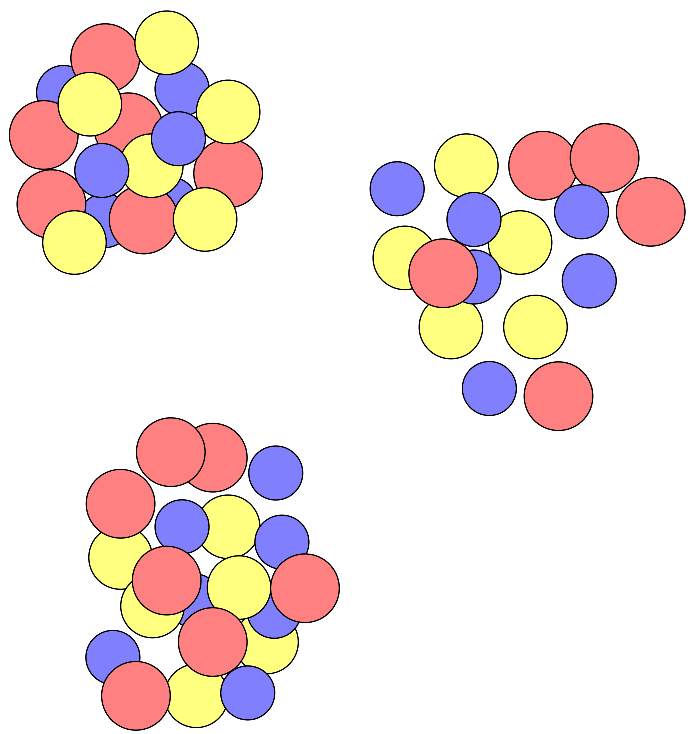
- Macro-clusters are distinct and distant
- Micro-clusters of a macro-cluster are similar and overlapping in the data space

Formalization of the problem

Data and parameters:

- Database X
- Macro-clusters: c
- Micro-clusters: κ

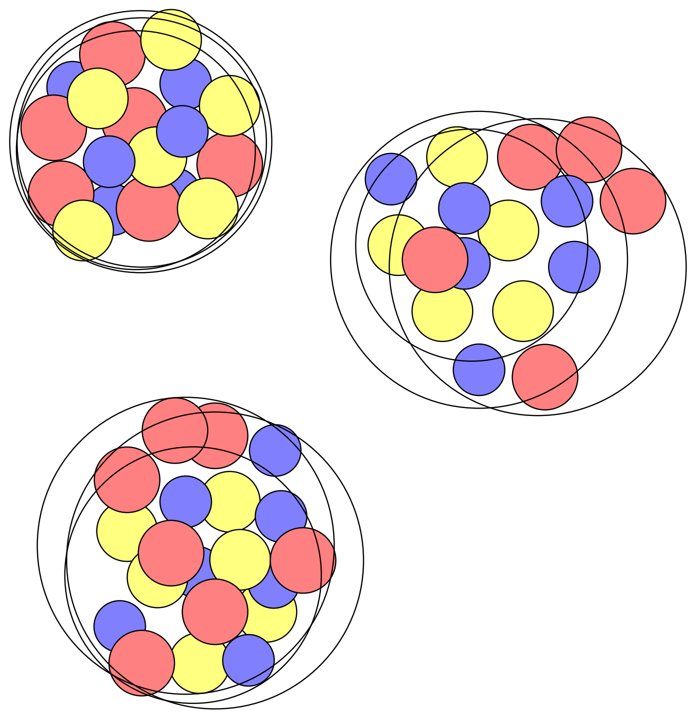
Data points and clusters:



Formalization of the problem

Notation

- centroids: v_{jk} for $j = 1, \ldots, c$ and $k = 1, \ldots, \kappa$ be the centroid of kth micro-centroid of the jth macro-cluster.
- assignment: $\mu_{jk}(x)$ represent the membership of x to the kth micro-centroid of the jth macro-cluster. We assume $\mu_{jk} \in \{0, 1\}$.



Formalization of the problem

Parameters: X,
A (difference on number of records), δ (distance for centroids)

$$\min J(\mu, v) = \sum_{j=1}^{c} \sum_{k=1}^{\kappa} \sum_{x \in X} \mu_{jk}(x) ||x - v_{jk}||^2$$

subject to $\sum_{j=1}^{c} \sum_{k=1}^{\kappa} \mu_{jk}(x) = 1$ for all $x \in X$

$|\sum_{x \in X} \mu_{jk_1}(x) - \sum_{x \in X} \mu_{jk_2}(x)| \leq A$
for all $j \in \{1, \ldots, c\}$, $k_1 \neq k_2 \in \{1, \ldots, \kappa\}$

$||v_{jk_1} - v_{jk_2}||^2 \leq \delta$
for all $j \in \{1, \ldots, c\}$, $k_1 \neq k_2 \in \{1, \ldots, \kappa\}$

$\mu_{jk}(x) \in \{0, 1\}$
for all $j \in \{1, \ldots, c\}$, $k \in \{1, \ldots, \kappa\}$, and $x \in X$

Experiments

Experiments

Implementation:

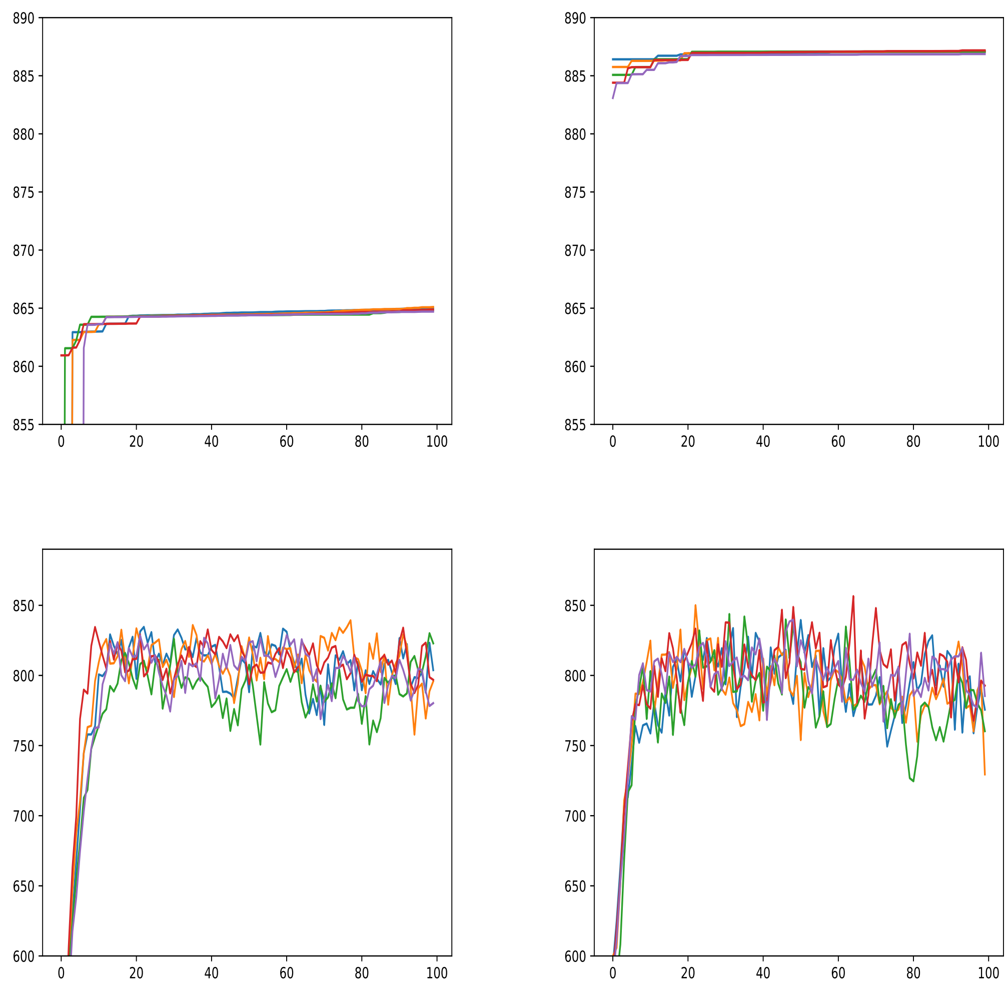
- (Clustering $+$) Genetic algorithms
- MDAV to produce k-size clusters
  so all clusters have the same number of records,
  better partition of macro-clusters into micro-clusters
  (better approximation of δ)

Parameters: $\delta = 0.0005$, $A = 5$
(5 runs, 100 epochs; $c = 2, \kappa = 3$ also $c = 4, \kappa = 10$)

Dataset: Concrete and CASC

Experiments

Example: Concrete, $c = 2, \kappa = 3$
(best top,mean bottom; random (left) and MDAV (right)



Discussion

Discussion

Discussion:

- Solution satisfies integral privacy constraints (κ parts with empty intersection); but,
- the optimization with $\kappa \neq 1$ and the full dataset X, and a reduced problem (say X_k) with one of the subsets, may lead to different results; but,
  - separated enough clusters will produce same results for X and X_k,
  - clustering algorithms lead to local optimal,
- So, maybe good enough?

Discussion

Discussion:

- Database changes. We want models that do not change.

Thank you