

# Data Science and Privacy

Fabrice Rossi

SAMM  
Université Paris 1 Panthéon Sorbonne

2018

# Privacy in a data-driven world

## Data collection

- ▶ is massive
- ▶ is here to stay (very probably)
- ▶ is invasive and potential dangerous
- ▶ is very useful and practical

## Data science

- ▶ is based on data
- ▶ provides better results with accurate data
- ▶ needs very personal data to provide personalized experiences

# Privacy in a data-driven world

## Data collection

- ▶ is massive
- ▶ is here to stay (very probably)
- ▶ is invasive and potential dangerous
- ▶ is very useful and practical

## Data science

- ▶ is based on data
- ▶ provides better results with accurate data
- ▶ needs very personal data to provide personalized experiences



## Trust is mandatory

- ▶ personal data are provided only to trusted collectors:
  - ▶ people will lie to collectors they do not trust
  - ▶ people will use protection techniques such as ad-blockers
- ▶ data science tolerates noisy data but not false ones!

## Trust is mandatory

- ▶ personal data are provided only to trusted collectors:
  - ▶ people will lie to collectors they do not trust
  - ▶ people will use protection techniques such as ad-blockers
- ▶ data science tolerates noisy data but not false ones!

## Collection and attack model

- ▶ a large number of individuals
- ▶ one or several *trusted* collectors
- ▶ external attackers who *cannot* access *directly* to the collected data
- ▶ but collectors share with the attackers some information about the collected data

## A limited model

- ▶ no rogue collector:
  - ▶ collectors are trusted
  - ▶ they operate as they declare to do
- ▶ perfect security:
  - ▶ data are secured in the collectors database systems
  - ▶ attackers cannot access the collected data

## Addressing the limitations

- ▶ out of scope of this course
- ▶ IT security
- ▶ legal enforcement

## The core challenge

How to publish information about the content of a database without compromising the privacy of the contributors?

## Practical examples

- ▶ data breach as a data release
- ▶ data leaks (e.g. misconfigured social networks)
- ▶ internal distribution, i.e. from collectors to data scientists (especially subcontractors)
- ▶ open data (public statistics)
- ▶ data reuse and data brokers

## GDPR

- ▶ General Data Protection Regulation (05/25/2018)
- ▶ Privacy by design and by default
  - ▶ data minimisation principle:  
*Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed*
  - ▶ anonymization: irreversible transformation that prevent any re-identification of the data
  - ▶ pseudonymization: re-identification is possible with additional data (that have to be kept separated)



Models

Full data release

Query answering

## Models

Full data release

Query answering

## Standard tabular data

- ▶ observations/instances/rows are elements of  $\mathcal{X}$
- ▶ with  $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_p$
- ▶  $\mathcal{X}_k$  is either  $\mathbb{R}$  (numerical data) or finite (categorical/nominal data)
- ▶ some variables are identifiers: they can be used to identify with certainty the associated person (e.g., social security number)

## Extensions

- ▶ relational data:
  - ▶ standard data
  - ▶ and in addition a graph of interaction between the instances
- ▶ multi-relational data: several graphs!

## Full data release

- ▶ a trusted collector wants to release her database at a *micro-level*:
  - ▶ the released database is comparable to the private one
  - ▶ it contains individual data (e.g. “rows” of the database)
- ▶ attackers gain access to this database and can do whatever they want with it

## Query answering

- ▶ a trusted collector wants to allow requests on her database:
  - ▶ sql like queries with only aggregate answers
  - ▶ no direct individual data results
- ▶ attackers can issue “arbitrary” queries (within some budget and other limitations)

## Identity disclosure (record linkage)

The attacker can link data in a published database to a specific person

## Attribute disclosure (attribute linkage)

The attacker can guess the value of a hidden attribute of a specific person

## Identity disclosure (record linkage)

The attacker can link data in a published database to a specific person

## Attribute disclosure (attribute linkage)

The attacker can guess the value of a hidden attribute of a specific person

## Some possible consequences

- ▶ anonymous publishing is impaired by identity disclosure
- ▶ potential dangerous hidden attributes include religious views, political views, sexual orientation, etc.
- ▶ publishing a database might allow an attacker to disclose information in another data source: the fact that collection of sensitive information is strongly regulated in some countries does not prevent its release through a breach of anonymity

## Naive solution

- ▶ just remove the identifier variables (or obfuscate them)
- ▶ (John, Doe, 36, Male, Roman Catholic, 50k) becomes (98b1aa7b4, 36, Male, Roman Catholic, 50k)
- ▶ pseudonymization if the obfuscated identifier can be mapped back to the original identifier

## Naive solution

- ▶ just remove the identifier variables (or obfuscate them)
- ▶ (John, Doe, 36, Male, Roman Catholic, 50k) becomes (98b1aa7b4, 36, Male, Roman Catholic, 50k)
- ▶ pseudonymization if the obfuscated identifier can be mapped back to the original identifier

## Unreliable scheme

- ▶ if the attacker knows (auxiliary information):
  - ▶ that John Doe is in the database
  - ▶ that he is Male and earns 50k a year
- ▶ then the attacker might guess John is 98b1aa7b4
- ▶ or more generally narrow down the possible records associated to John Doe



## Secondary identification

- ▶ identifiers are removed from an anonymized database by essence
- ▶ but some other variables can identify a person or at least a group of instances to which the person must belong
- ▶ *quasi-identifiers*

## Linkage attacks

- ▶ one of the main de-anonymization technique
- ▶ conditions:
  - ▶ auxiliary information
  - ▶ non anonymous data in the auxiliary information

# Well known de-anonymization cases

## Hospital discharge data (1997)

- ▶ in the USA, hospitals release anonymized discharge data:
  - ▶ include health related information (diagnoses, procedures, etc.)
  - ▶ and potential quasi-identifiers: date of birth, gender and ZIP code
- ▶ cross-referencing with publicly available voter lists:
  - ▶ identical quasi-identifiers!
  - ▶ on some experiments birth date + ZIP code identify exactly 69 % of the listed persons

## DNA sequence identification (2004)

- ▶ DNA sequences can be shared for research (in the USA)
- ▶ they are associated to hospital visits, hence to discharge data
- ▶ trail matching algorithm

# Well known de-anonymization cases

## The AOL fiasco

- ▶ search data released in 2006, available a few days only:
  - ▶ 20 millions search keywords
  - ▶ 3-month period
  - ▶ 650 000 users
  - ▶ queries are associated to users
  - ▶ users are identified by unique numerical id
- ▶ de-anonymization by Barbaro and Zeller from the **NY times**
  - ▶ localization keywords (“landscapers in Lilburn, Ga”)
  - ▶ last name search
  - ▶ cross-reference with public data (e.g. phonebook listings)
- ▶ quasi-identifiers:
  - ▶ a single search query is seldom a quasi-identifier
  - ▶ identification become more and more precise with added queries

## The Netflix Prize

- ▶ ratings data released in 2006:
  - ▶ ~ 100 millions of ratings
  - ▶ ~ 480 thousands users
  - ▶ ~ 18 thousands movies
  - ▶ an observation: user ID (pseudonymous), movie ID (non anonymous), date of grade, grade
  - ▶ perturbations have been applied: rating deletions, rating insertions, rating date modifications
- ▶ de-anonymization by **Narayanan and Shmatikov** in 2007:
  - ▶ similar to AOL case: no quasi-identifier but a collection of discriminant variables (ratings with dates)
  - ▶ similarity based search
  - ▶ works well on *sparse* databases
  - ▶ IMDb as an example of auxiliary information source

# The need for data modification

## Anonymization is hard

- ▶ under a naive attack model (no auxiliary information), removing direct identifiers is sufficient
- ▶ **but** auxiliary information is always available (now more frequently than ever!)
- ▶ once non-anonymous data are available, quasi-identifiers enables one to propagate identities

## Modifications

- ▶ release a modified version of the database
- ▶ possible modifications:
  - ▶ noise
  - ▶ generalization (e.g. replace a complete 5 digits ZIP code by a truncated one)
  - ▶ etc.

## Trade-off

One cannot at the same time

- ▶ maximize the precision of the data
- ▶ and minimize the privacy risk

## Trade-off

One cannot at the same time

- ▶ maximize the precision of the data
- ▶ and minimize the privacy risk



## Trade-off

One cannot at the same time

- ▶ maximize the precision of the data
- ▶ and minimize the privacy risk



## Utility measures

- ▶ released databases must remain useful
- ▶ utility measures have been proposed to quantify this:
  - ▶ marginal distribution preservation
  - ▶ dependency preservation
  - ▶ machine learning oriented measures (e.g. AUC preservation)
  - ▶ etc.



# Utility versus Privacy

## Utility first

- ▶ utility preservation guarantees
- ▶ post hoc test of the privacy guarantees (e.g. the probability of re-identification under some threat model)
- ▶ quite common in official statistical institutes

## Privacy first

- ▶ privacy properties guarantees
- ▶ post hoc test of the utility guarantees
- ▶ main focus of the privacy research in computer science and mathematics

Models

Full data release

Query answering

## Threat model

- ▶ a trusted collector wants to release her database at a *micro-level*:
  - ▶ the released database is comparable to the private one
  - ▶ it contains individual data (e.g. “rows” of the database)
- ▶ attackers gain access to this database and can do whatever they want with it, including using auxiliary non-anonymous data

## Perturbation based solutions

- ▶ stochastic: additive noise, swapping and related methods
- ▶ partition based: approximation, recoding and generalization

R package: [sdcMicro](#)

## Rationale

- ▶ statistical point of view
- ▶ data released to enable researchers to conduct studies that involve human beings (sociology, medicine, etc.)
- ▶ researchers are implicitly trusted!
- ▶ utility first:
  - ▶ the methods try to preserve some important features (e.g., the covariance matrix)
  - ▶ privacy is checked afterward
- ▶ typically stochastic methods

## Simple additive noise

- ▶ rather than releasing  $X_k$  release  $X_k + \varepsilon_k$  where  $\varepsilon_k$  is a noise (e.g. Gaussian noise)
- ▶ properties:
  - ☺ straightforward
  - ☺ limited effects on univariate estimates (e.g. mean)
  - ☹ limited to numerical attributes
  - ☹ inconsistent multivariate estimates (e.g. covariance matrix)
  - ☹ low level of protection

## Correlated additive noise

- ▶ generate noise with a covariance matrix proportional to the one of the data
- ▶ solves the covariance estimation issue
- ▶ improves a bit the protection level
- ▶ variants preserve more elements

## Other tools

- ▶ random perturbation such as PRAM (Post Randomization Method) for nominal data
- ▶ data swapping: exchange values of attributes between instances (in a controlled way)

## Privacy first strategy

- ▶ identify a privacy threat
- ▶ build the perturbation as a protection against the threat
- ▶ identity disclosure:
  - ▶ threat: find a single record in the released database using part of its content (quasi-identifiers)
  - ▶ protection: make sure that no combination of quasi-identifiers can be used to select a single record
- ▶ typically deterministic methods from computer science

## Principle

- ▶ proposed by P. Samarati and L. Sweeney in 1998
- ▶ consider a database with  $P$  variables among which  $\mathbf{X}_1, \dots, \mathbf{X}_L$  form a quasi-identifier
- ▶ the database satisfies  $k$  anonymity for an integer  $k$  if for any value  $(\mathbf{x}_1, \dots, \mathbf{x}_L) \in \mathcal{X}_1 \times \dots \times \mathcal{X}_L$ , there are at least  $k$  instances in the database that begin with  $(\mathbf{x}_1, \dots, \mathbf{x}_L)$
- ▶ protection: if the attacker knows the quasi-identifier for a person, she cannot recover less than  $k$  compatible persons in the database



# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	M	02141	short breath
2	Black	1965	M	02142	chest pain
3	Black	1965	F	02131	hypertension
4	Black	1965	F	02132	hypertension
5	Black	1964	F	02131	obesity
6	Black	1964	F	02132	chest pain
7	White	1964	M	02131	chest pain
8	White	1964	M	02132	obesity
9	White	1964	M	02133	short breath
10	White	1967	M	02131	chest pain
11	White	1967	M	02132	chest pain

Original database

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	M	0214*	short breath
2	Black	1965	M	0214*	chest pain
3	Black	1965	F	0213*	hypertension
4	Black	1965	F	0213*	hypertension
5	Black	1964	F	0213*	obesity
6	Black	1964	F	0213*	chest pain
7	White	1964	M	0213*	chest pain
8	White	1964	M	0213*	obesity
9	White	1964	M	0213*	short breath
10	White	1967	M	0213*	chest pain
11	White	1967	M	0213*	chest pain

database with 2-anonymity with respect to the first 4 variables

## Identity protection

- ▶ is obvious
- ▶ but limited by the value of  $k$

## Attribute protection

- ▶ is not guaranteed (at all)
- ▶ without auxiliary information, the database releases the marginal distribution of private variables
- ▶ with auxiliary information, we have *conditional* distributions that might differ from the global one!

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	M	0214*	short breath
2	Black	1965	M	0214*	chest pain
3	Black	1965	F	0213*	hypertension
4	Black	1965	F	0213*	hypertension
5	Black	1964	F	0213*	obesity
6	Black	1964	F	0213*	chest pain
7	White	1964	M	0213*	chest pain
8	White	1964	M	0213*	obesity
9	White	1964	M	0213*	short breath
10	White	1967	M	0213*	chest pain
11	White	1967	M	0213*	chest pain

## Marginal distribution of Condition

chest pain	hypertension	obesity	short breath
0.4545	0.1818	0.1818	0.1818

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	M	0214*	short breath
2	Black	1965	M	0214*	chest pain
3	Black	1965	F	0213*	hypertension
4	Black	1965	F	0213*	hypertension
5	Black	1964	F	0213*	obesity
6	Black	1964	F	0213*	chest pain
7	White	1964	M	0213*	chest pain
8	White	1964	M	0213*	obesity
9	White	1964	M	0213*	short breath
10	White	1967	M	0213*	chest pain
11	White	1967	M	0213*	chest pain

Marginal distribution of Condition for (White, 1964, M, 02131)

chest pain	hypertension	obesity	short breath
0.3333	0.0000	0.3333	0.3333

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	M	0214*	short breath
2	Black	1965	M	0214*	chest pain
3	Black	1965	F	0213*	hypertension
4	Black	1965	F	0213*	hypertension
5	Black	1964	F	0213*	obesity
6	Black	1964	F	0213*	chest pain
7	White	1964	M	0213*	chest pain
8	White	1964	M	0213*	obesity
9	White	1964	M	0213*	short breath
10	White	1967	M	0213*	chest pain
11	White	1967	M	0213*	chest pain

Marginal distribution of Condition for (Black, 1965, F, 02131)

chest pain	hypertension	obesity	short breath
0.0000	1.0000	0.0000	0.0000

## Generalization

- ▶ proposed by P. Samarati and L. Sweeney in 1998
- ▶ based on the idea that data can be “generalized”, that is approximated, to hide identifying values:
  - ▶ full ZIP code: 5 digits (02141)
  - ▶ approximation: 4 first digits (0214\*)
  - ▶ progressive approximation
- ▶ data are not noisy but imprecise

## Domains

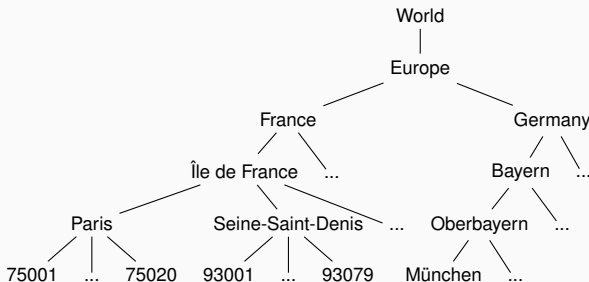
- ▶ domains are finite set of values
- ▶ domains are partially ordered (generality order)
- ▶ ground domain:  $\mathcal{X}_l$ , the most precise/complete description
- ▶ a maximal domain (for the partial order) contains only one value
- ▶ a domain is *more* general than another one if it has *fewer* values
- ▶ a domain has at most one *direct* more general domain
- ▶ example:
  - ▶ ground domain: age in years  $\mathcal{X}_l = \{0, 1, 2, \dots, 130\}$
  - ▶ direct generalization of  $\mathcal{X}_l$ : age rounded with 5 years precision  $\mathcal{X}_l^5 = \{0, 5, 10, \dots, 130\}$
  - ▶ direct generalization of  $\mathcal{X}_l^5$ : age rounded with 10 years precision  $\mathcal{X}_l^{10} = \{0, 10, 20, \dots, 130\}$
  - ▶ direct generalization of  $\mathcal{X}_l^{10}$ : age unreleased  $\mathcal{X}_l^{none} = \{unreleased\}$



# Hierarchical generalization

## Hierarchy of values

- ▶ values from one domain are mapped to values from its direct more general domain
- ▶ this creates a hierarchy of values from precise values to general ones



## Generalization

- ▶ generalization consists in replacing a value by its “generalized” version at an upper level of the corresponding hierarchy
- ▶ generalization is applied:
  - ▶ uniformly for each variable: all the values of a variable are generalized at the same level in the hierarchy
  - ▶ arbitrarily for different variables: two distinct variables can be generalized to different levels of their respective hierarchy
- ▶ the distance between a variable and its generalization is the number of levels in the hierarchy between the ground domain and the domain of the generalization (including this one)
- ▶ among all the generalizations that achieve k-anonymity, one prefers the database that is the closest to the original one

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	M	02141	short breath
2	Black	1965	M	02142	chest pain
3	Black	1965	F	02131	hypertension
4	Black	1965	F	02132	hypertension
5	Black	1964	F	02131	obesity
6	Black	1964	F	02132	chest pain
7	White	1964	M	02131	chest pain
8	White	1964	M	02132	obesity
9	White	1964	M	02133	short breath
10	White	1967	M	02131	chest pain
11	White	1967	M	02132	chest pain

Original database

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	M	0214*	short breath
2	Black	1965	M	0214*	chest pain
3	Black	1965	F	0213*	hypertension
4	Black	1965	F	0213*	hypertension
5	Black	1964	F	0213*	obesity
6	Black	1964	F	0213*	chest pain
7	White	1964	M	0213*	chest pain
8	White	1964	M	0213*	obesity
9	White	1964	M	0213*	short breath
10	White	1967	M	0213*	chest pain
11	White	1967	M	0213*	chest pain

Generalization: (0,0,0,1,0)

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	*	02141	short breath
2	Black	1965	*	02142	chest pain
3	Black	1965	*	02131	hypertension
4	Black	1965	*	02132	hypertension
5	Black	1964	*	02131	obesity
6	Black	1964	*	02132	chest pain
7	White	1964	*	02131	chest pain
8	White	1964	*	02132	obesity
9	White	1964	*	02133	short breath
10	White	1967	*	02131	chest pain
11	White	1967	*	02132	chest pain

Generalization: (0,0,1,0,0)

# Example

	Ethnicity	Birth	Gender	ZIP	Condition
1	Black	1965	*	0214*	short breath
2	Black	1965	*	0214*	chest pain
3	Black	1965	*	0213*	hypertension
4	Black	1965	*	0213*	hypertension
5	Black	1964	*	0213*	obesity
6	Black	1964	*	0213*	chest pain
7	White	1964	*	0213*	chest pain
8	White	1964	*	0213*	obesity
9	White	1964	*	0213*	short breath
10	White	1967	*	0213*	chest pain
11	White	1967	*	0213*	chest pain

Generalization: (0,0,1,1,0)

## Outliers suppression

- ▶ rare values in a quasi-identifier are difficult to anonymize
- ▶ this can lead to over-generalization
- ▶ a simple solution consists in removing outliers (within specified limits)

## Multidimensional generalization

- ▶ multidimensional generalization function: use contexts to generalize an instance
- ▶ adaptive generalization level
- ▶ partition based:
  - ▶ e.g. **Mondrian**
  - ▶ cluster instances such that groups fulfill k-anonymity (or something else!)
  - ▶ replace values by statistics over the cluster (mean, span, etc.)

## Quality metrics

- ▶ minimal generalizations: databases that achieve  $k$ -anonymity with minimal distance on each variable
- ▶ multiple solutions in some situations
- ▶ ad hoc criteria can be used to choose one of the minimal solutions

## Complexity

- ▶ obtaining minimal generalization is NP-hard in general
- ▶ approximation algorithms do not have very good guarantees
- ▶ but heuristics give acceptable results ( $k$ -anonymity is guaranteed, minimality is not)



## Summary

- 😊 guarantees against identity disclosure
- 😊 flexible framework
- 😞 highly dependent to the chosen quasi-identifiers
- 😞 sub-optimal solutions (NP-hardness)
- 😞 no attribute protection

## l-diversity

- ▶ k-anonymity does not protect individual attributes
- ▶ l-diversity fixes this problem:
  - ▶ proposed in 2006 by Machanavajjhala et al.
  - ▶ general principle: a database is l-diverse if any group of instances identified by a quasi-identifier contains at least l “well represented” values for the sensitive attribute
- ▶ several instantiations:
  - ▶ minimal entropy
  - ▶ recursive diversity: bound on the ratio between the frequency of the most frequent value and the frequency of the less frequent values
  - ▶ variations around non-sensitive values (e.g. healthy) and sensitive-ones

## Limitations of I-diversity

- ▶ achievability: the original data could not satisfy I-diversity globally!
- ▶ semantic similarity:
  - ▶ I-diversity does not take into account links between the values of the variables
  - ▶ diversity among similar values is not sufficient to protect an attribute

## t-closeness

- ▶ proposed in 2007 by Li et al. (refined in 2010)
- ▶ core principle: ensuring conditional distributions (i.e. in group of instances) are similar to the marginal distribution
- ▶ instantiation via information theoretic measures (such as the KL-divergence) would only solve the achievability problem
- ▶ differences between distributions are measured via optimal transport (the earth mover distance)

## Protection against linkage attacks

- ▶ with respect to specific quasi-identifiers
- ▶ identity: k-anonymity
- ▶ attribute: closeness and related methods
- ▶ generalization/partition based (with help of suppression)
  - ▶ fast sub-optimal solutions
  - ▶ induce frequently a significative loss in data quality

## Independent anonymized releases

- ▶ several databases controlled by non coordinating collectors
- ▶ some common attributes
- ▶ each collector releases an anonymized database (with e.g. k-anonymity)
- ▶ some persons belong to more than one database

## Intersection attack

- ▶ analyzed by Ganta et al. in 2008
- ▶ consists simply in intersecting groups that match a quasi-identifier in different databases
- ▶ leverages the fact sensitive data is kept exact

## Quasi-identifiers

- ▶ must be specified before data release
- ▶ non obvious trade-off:
  - ▶ minimal set of attributes: low protection, but high quality data
  - ▶ large set of attributes: high protection, but might be impossible to reach without a massive loss in data quality
- ▶ skewed and long tail distributions:
  - ▶ typically power law distributed attributes
  - ▶ the vast majority of persons have the same value: intrinsically anonymous
  - ▶ **but** persons have very atypical values: must be aggressively modified  $\Rightarrow$  destroys marginal distributions

## A limited model

- ▶ quasi-identifiers are *public* (non sensitive) data that can be used to identify a person
- ▶ but the attacker might know *private* (sensitive) data also!
- ▶ **Netflix Prize:**
  - ▶ private information: movie ratings with dates
  - ▶ typical skewed distribution: rare movies, compulsive watchers, etc.
  - ▶ re-identification *from private data* is very easy: e.g. 99 % of users are unique given 8 movie ratings and approximate rating dates!
  - ▶ private data obtained from IMDb, but other sources could be used (e.g. blog posts, direct interaction, etc.)
  - ▶ perturbations of the ratings would reduce strongly the interest of the database

## Relational data

- ▶ data + graph
- ▶ new disclosure risk: link disclosure
- ▶ much more complex anonymization problem:
  - ▶ added value of relational data: the graph structure!
  - ▶ new identification source: the graph structure!
  - ▶ typical example:
    - ▶ degree based identification
    - ▶ degree anonymity
    - ▶ obtained by inserting links, deleting links and swapping links
    - ▶ but the degree follows generally a skewed distribution!
- ▶ generalization at the graph level:
  - ▶ cluster of nodes
  - ▶ cluster of edges



## Correlated neighborhood

- ▶ measure the resemblance between two nodes as the agreement between their connection:
  - ▶  $A$ : adjacency matrix ( $A_{ij} = 1 \Leftrightarrow i$  and  $j$  are connected)
  - ▶  $s(i, j) = \frac{1}{N} \sum_k A_{ik} A_{jk} - \frac{1}{N^2} (\sum_k A_{ik}) (\sum_k A_{jk})$
- ▶ characteristic vector of a node:
  - ▶ vector of agreements,  $(s(i, 1), \dots, s(i, N))$
  - ▶ very robust to limited random modification of the graph
- ▶ re-identification via characteristic vectors
  - ▶ ordering sensitive and theoretically NP-hard
  - ▶ efficient heuristics for sparse graphs
  - ▶ very efficient re-identification scheme, even against protected graphs

## Solutions...

- ▶ a collection of data release methods
- ▶ utility oriented (noise)
- ▶ privacy oriented (generalization)

## Solutions...

- ▶ a collection of data release methods
- ▶ utility oriented (noise)
- ▶ privacy oriented (generalization)

## with strong limitations!

- ▶ quasi-identifiers are naive, anything interesting can be used to re-identify persons
- ▶ k-anonymity (and related constraints) is essentially impossible to apply in high dimension
- ▶ the lack of guaranteed composition properties creates dangerous future opportunities for attackers
- ▶ full data release is inherently dangerous

Models

Full data release

Query answering

## Threat model

- ▶ a trusted collector wants to allow requests on her database:
  - ▶ sql like queries with only aggregate answers
  - ▶ no direct individual data results
- ▶ attackers can issue “arbitrary” queries (within some budget and other limitations)

## Links to full data release

- ▶ queries can use quasi-identifiers (QI) to select groups exactly as in full data release
- ▶ aggregate answers can be used to infer attributes via differentiating attacks (comparing the results of two queries):
  - ▶ how many persons in the database have aids?
  - ▶ how many persons expected those with QI  $x$  have aids?

## Query auditing

- ▶ verify that a query cannot leak information, taking into account previous ones
- ▶ but refusing to answer can leak valuable information
- ▶ and rich query language can lead to undecidable problems

## Perturbated data

- ▶ execute queries on perturbated but unreleased versions of the database
- ▶ mostly identical to full data release with perturbation!

## Sampling

- ▶ compute the query on a sample of the database
- ▶ different samples for different queries

## Noisy answers

- ▶ compute the exact answer on the original database
- ▶ return a noisy version of the answer
- ▶ close to sampling in some situations

## Rationale

- ▶ provide strong privacy guarantees (mathematically proven!)
- ▶ protection against identity disclosure in a strong sense: the attacker cannot guess whether a person belongs to a database or not
- ▶ very broad threat model: the attacker can use whatever auxiliary information she wants

## Informal definition

A query mechanism is **differentially private** if its results do not change significantly when applied to two databases which differ only by the inclusion of one person



## Background

- ▶  $\mathcal{D}$ : a database
- ▶  $\mathcal{X}^N$ , the set of all databases of size at most  $N$
- ▶  $d(\mathcal{D}_1, \mathcal{D}_2)$ : distance between databases, the number of distinct instances
- ▶ randomized algorithm: an algorithm with random outputs

## Definition (Dwork, Nissim, McSherry and Smith, 2006)

A randomized algorithm  $\mathcal{M}$  is  $(\epsilon, \delta)$ -differentially private if for any possible solution set  $\mathcal{S}$ , and any pair of databases  $\mathcal{D}_1$  and  $\mathcal{D}_2$  with  $d(\mathcal{D}_1, \mathcal{D}_2) \leq 1$  we have

$$\mathbb{P}(\mathcal{M}(\mathcal{D}_1) \in \mathcal{S}) \leq \delta + \exp(\epsilon)\mathbb{P}(\mathcal{M}(\mathcal{D}_2) \in \mathcal{S}).$$

When  $\delta = 0$ ,  $\mathcal{M}$  is  $\epsilon$ -differentially private.

## Intuitive interpretation

- ▶ symmetric definition:

$$\exp(-\epsilon) (\mathbb{P}(\mathcal{M}(\mathcal{D}_2) \in \mathcal{S}) - \delta) \leq \mathbb{P}(\mathcal{M}(\mathcal{D}_1) \in \mathcal{S}) \leq \delta + \exp(\epsilon) \mathbb{P}(\mathcal{M}(\mathcal{D}_2) \in \mathcal{S}).$$

- ▶  $\mathbb{P}(\mathcal{M}(\mathcal{D}_1) \in \mathcal{S}) \simeq \mathbb{P}(\mathcal{M}(\mathcal{D}_2) \in \mathcal{S})$
- ▶ an attacker cannot decide based on  $\mathcal{M}(\mathcal{D}_?)$  whether the database is  $\mathcal{D}_1$  or  $\mathcal{D}_2$
- ▶ protects  $x$  who is in  $\mathcal{D}_1$  and not in  $\mathcal{D}_2$  (or vice versa)
- ▶ notice that in practice,  $\epsilon$  should be small, so  $\exp(\epsilon) \simeq 1 + \epsilon$

## Important property

- ▶  $f$  a probability distribution depending on the result of  $\mathcal{M}$
- ▶  $u$  a function from the support of  $f$  to  $\mathbb{R}$
- ▶ if  $\mathcal{M}$  is  $\epsilon$ -differentially private

$$\exp(-\epsilon)\mathbb{E}_{A\sim f(\mathcal{M}(\mathcal{D}_2))}u(A) \leq \mathbb{E}_{A\sim f(\mathcal{M}(\mathcal{D}_1))}u(A) \leq \exp(\epsilon)\mathbb{E}_{A\sim f(\mathcal{M}(\mathcal{D}_2))}u(A)$$

## Interpretation: utilitarian point of view

- ▶  $A$ : state-of-the-world
- ▶  $u$ : utility function for a given person
- ▶  $f(\mathcal{M}(\mathcal{D}))$ : probability distribution on the states of the world after releasing the result of  $\mathcal{M}$
- ▶  $\epsilon$ -df: no significant effect of a data release on the average utility

## Exact answers?

- ▶ arbitrary queries, e.g.  $\mathcal{M}$  = “how many persons expected those with QI  $x$  are hiv positive?”
- ▶ an *exact* answer cannot be  $\epsilon$ -df in a useful way:
  - ▶ exact answers are deterministic:  $\mathbb{P}(\mathcal{M}(\mathcal{D}_i) = r_i) = 1$
  - ▶ if  $x$  is hiv positive, with  $x \in \mathcal{D}_1$  and  $x \notin \mathcal{D}_2$ ,  $\mathbb{P}(\mathcal{M}(\mathcal{D}_1) = r_1) = 1$  and  $\mathbb{P}(\mathcal{M}(\mathcal{D}_2) = r_1) = 0$
  - ▶  $\mathbb{P}(\mathcal{M}(\mathcal{D}_1) \in S) \leq \exp(-\epsilon) \mathbb{P}(\mathcal{M}(\mathcal{D}_2) \in S)$  is impossible!

## Distortion is mandatory

- ▶ we must give approximate answers
- ▶ randomized ones are appropriate (unpredictable)

## Embarrassing question

- ▶ objective: obtain an accurate estimate of the proportion of persons engaging in “insert here an embarrassing activity”
- ▶ question: “did you engage in ... last week?”
- ▶ answering algorithm:
  1. flip a coin
  2. if Tail, then respond truthfully
  3. if Head, flip another coin:
    - 3.1 if Tail, answer Yes
    - 3.2 if Head, answer No
- ▶ provides plausible deniability 😊

## Estimating the frequency

- ▶  $p$ : true frequency of the activity (that shall not be named)
- ▶  $\mathbb{P}(\text{answer}=\text{true}) = \mathbb{P}(\text{answer}=\text{true}|\text{Tail as first result})\frac{1}{2} + \mathbb{P}(\text{answer}=\text{true}|\text{Head first result})\frac{1}{2}$
- ▶  $\mathbb{P}(\text{answer}=\text{true}) = \frac{p}{2} + \frac{1}{4}$
- ▶ thus  $p = 2\mathbb{P}(\text{answer}=\text{true}) - \frac{1}{2}$

# Randomized response

## Estimating the frequency

- ▶  $p$ : true frequency of the activity (that shall not be named)
- ▶  $\mathbb{P}(\text{answer}=\text{true}) = \mathbb{P}(\text{answer}=\text{true}|\text{Tail as first result})\frac{1}{2} + \mathbb{P}(\text{answer}=\text{true}|\text{Head first result})\frac{1}{2}$
- ▶  $\mathbb{P}(\text{answer}=\text{true}) = \frac{p}{2} + \frac{1}{4}$
- ▶ thus  $p = 2\mathbb{P}(\text{answer}=\text{true}) - \frac{1}{2}$

## Differential privacy like analysis

- ▶  $\mathbb{P}(\text{answer}=\text{true}|\text{doing it} = \text{true}) = \frac{3}{4}$
- ▶  $\mathbb{P}(\text{answer}=\text{true}|\text{doing it} = \text{false}) = \frac{1}{4}$
- ▶ ratio: 3 (also for answer=false), so we are in a way  $\ln 3$ -differentially private

## Definition (Sensitivity)

Let  $f$  be a function from  $\mathcal{X}^N$  to  $\mathbb{R}^k$ . The sensitivity of  $f$  is

$$\Delta f = \max_{d(\mathcal{D}_1, \mathcal{D}_2) \leq 1} \|f(\mathcal{D}_1) - f(\mathcal{D}_2)\|_1.$$

## Interpretation

- ▶ the sensitivity of  $f$  is the maximum value by which the output of  $f$  can change by removing someone from the database
- ▶ e.g. if  $f$  is “how many persons in the database do this and that”, then  $\Delta f = 1$



## Definition (Laplace distribution)

The centered Laplace distribution with scale  $b$  is a continuous distribution on  $\mathbb{R}$  with density  $f(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$ . Notation:

$$Y \sim \text{Lap}(b)$$

## Definition (Laplace mechanism)

Let  $f$  be a function from  $\mathcal{X}^N$  to  $\mathbb{R}^k$ . The Laplace mechanism  $\mathcal{M}_{f,\epsilon}$  is defined from  $\mathcal{X}^N$  to  $\mathbb{R}^k$  as the random algorithm that answers  $\mathcal{M}_{f,\epsilon}(\mathcal{D}) = f(\mathcal{D}) + (Z_1, \dots, Z_k)^T$ , where the  $Z_j$  are independent Laplace distributed random variables with scale  $\frac{\Delta f}{\epsilon}$ .

## Theorem

*The Laplace mechanism is  $\epsilon$ -differentially private.*

Comparing densities of the outputs  $\mathcal{M}_{l,f,\epsilon}(\mathcal{D}_1)$  and  $\mathcal{M}_{l,f,\epsilon}(\mathcal{D}_2)$

$$\begin{aligned}\frac{p_{\mathcal{D}_1}(t)}{p_{\mathcal{D}_2}(t)} &= \prod_{i=1}^k \frac{\exp\left(-\epsilon \frac{|f(\mathcal{D}_1)_i - t_i|}{\Delta f}\right)}{\exp\left(-\epsilon \frac{|f(\mathcal{D}_2)_i - t_i|}{\Delta f}\right)} \\ &= \prod_{i=1}^k \exp\left(\epsilon \frac{|f(\mathcal{D}_2)_i - t_i| - |f(\mathcal{D}_1)_i - t_i|}{\Delta f}\right) \\ &\leq \prod_{i=1}^k \exp\left(\epsilon \frac{|f(\mathcal{D}_2)_i - f(\mathcal{D}_1)_i|}{\Delta f}\right) \\ &= \exp\left(\epsilon \frac{\|f(\mathcal{D}_2)_i - f(\mathcal{D}_1)_i\|_1}{\Delta f}\right) \\ &\leq \exp(\epsilon)\end{aligned}$$

# Noise of the Laplace Mechanism

## Theorem

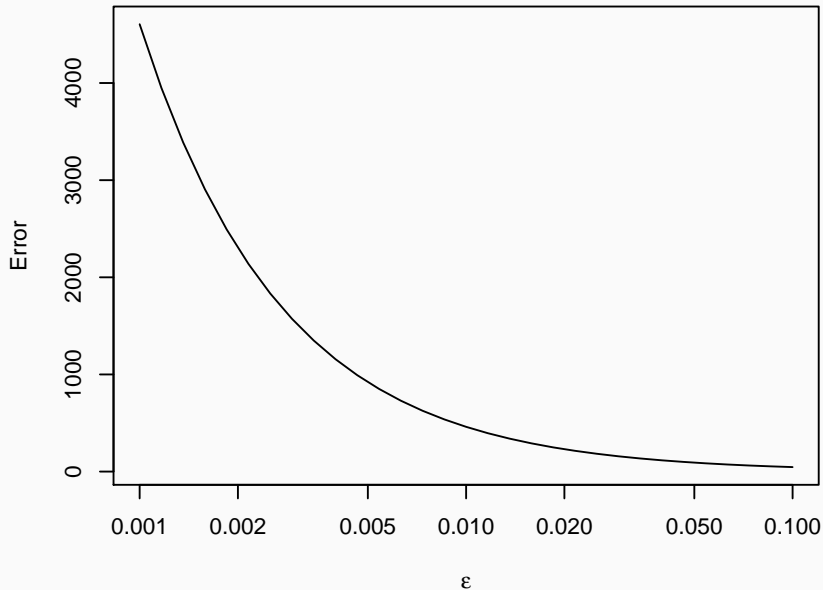
If  $f$  is from  $\mathcal{X}^N$  to  $\mathbb{R}^k$ , then

$$\mathbb{P} \left( \|f(\mathcal{D}) - \mathcal{M}_{l,f,\epsilon}(\mathcal{D})\|_{\infty} \geq \frac{\Delta f}{\epsilon} \ln \left( \frac{k}{\delta} \right) \right) \leq \delta.$$

## Example

- ▶ medical database
- ▶  $f$ : counting query of the form “how many persons have medical condition  $z$ ?” ( $k = 1$ )
- ▶  $\Delta f = 1$  (true in general for counting queries!)
- ▶ bound with 1% confidence, i.e.  $\delta = 0.01$
- ▶ in at least 99% of the queries, the count is at most  $\frac{\log 100}{\epsilon}$  away from the true count

# Example



## Discussion

- ▶  $\epsilon = 0.01$ 
  - ▶ guarantees that probabilities with or without any person are within 1% one from another
  - ▶ induces a noise of at most 460 in 99 % of the cases
  - ▶ the size of the database has not effect on those values (for counting queries!)
- ▶ this is:
  - ▶ enormous for small size data and small size answers
  - ▶ well within margins for large scale data
- ▶ Differential privacy is big data oriented

# Repeating queries

## An obvious attack

- ▶ just repeatedly ask the same query and average the results!
- ▶ queries can be carefully crafted to avoid being obviously identical!

## Protection is impossible

- ▶ theoretical results show that if one allows arbitrary complex queries, either the answers are very inaccurate or the underlying database can be recovered using less than a linear number of queries (with respect to the size of the database)
- ▶ in practice one must limit the number of queries that can be answered
- ▶ access control is mandatory!

## Principle

- ▶ allow to each user a total privacy budget
- ▶ each query to a  $\epsilon$ -dp mechanism reduces the budget by  $\epsilon$
- ▶ when the budget is exhausted, the user cannot issue any more request to the database

## Consequences

- ▶ access control is mandatory!
- ▶ a very important issue is to reduce the noise in the results for a fixed value of  $\epsilon$ : better use of the budget!
- ▶ a possible solution when the budget is exhausted is to throw away the data

# Composing queries

## Theorem

let  $\mathcal{M}_i$  be  $\epsilon_i$ -dp for  $i \in \{1, \dots, k\}$ . Then

$$\mathcal{M}(\mathcal{D}) = (\mathcal{M}_1(\mathcal{D}), \dots, \mathcal{M}_k(\mathcal{D}))$$

is  $\sum_{i=1}^k \epsilon_i$ -dp.

## Discussion

- ▶ differential privacy is one of the only framework that guarantees composition
- ▶ explains the issue with repeated queries:
  - ▶ applying  $k$ -times a  $\epsilon$ -dp mechanism corresponds to query once a  $k\epsilon$ -dp mechanism
  - ▶ from  $\epsilon = 0.01$  with probabilities with 1% we move to
    - ▶  $k = 10$ : probabilities within 10 %
    - ▶  $k = 50$ : probabilities within 65 %!
- ▶ budget drain...



## Theorem

let  $\mathcal{M}_i$  be  $\epsilon_i$ -dp for  $i \in \{1, \dots, k\}$ . Let  $C_1, \dots, C_k$  be arbitrary disjoint subsets of a database  $\mathcal{D}$ . Then

$$\mathcal{M}(\mathcal{D}) = (\mathcal{M}_1(\mathcal{D} \cap C_1), \dots, \mathcal{M}_k(\mathcal{D} \cap C_k))$$

is  $\max_{i \in \{1, \dots, k\}} \epsilon_i$ -dp.

## Application

- ▶ parallel composition enables non naive extension of the Laplace framework
- ▶ particularly useful for related queries
- ▶ efficiently limits the budget spending

# Histogram queries

## Setting

- ▶ assume given a partition of  $\mathcal{D}$  into  $k$  subsets
- ▶ ask for the number of instances in each subset

## Naive solution

- ▶ apply the Laplace mechanism to  $k$  queries, one per subset
- ▶ if each query is answered with  $\epsilon$ -dp, then the composed query is  $k\epsilon$ -dp

## Histogram analysis

- ▶ consider the  $k$  dimensional query that answers the  $k$  counts at once
- ▶ its sensitivity is 1 as the subsets are disjoint
- ▶ thus using  $k$  independent Laplace noise leads to a  $\epsilon$ -dp mechanism!

## Setting

- ▶ compute the empirical distribution of some property
- ▶ report the most common value (and the number of times it occurs)

## Histogram case

- ▶ when the values of the property are mutually exclusive
- ▶ straightforward application of the histogram query
- ▶ the most common value is computed by the analyst after receiving the histogram

## More general setting

- ▶ in some situations, the values are not exclusive, e.g. in case of repeated measurements over the same persons
- ▶ then the histogram case does not apply: the sensitivity is proportional to the number of values!

## Report noisy max mechanism

- ▶ compute internally all the counts needed
- ▶ add independent Laplace noise with scale  $\frac{1}{\epsilon}$  to each count
- ▶ report the winning value based on the noisy counts (and the winning count)

Report noisy max is  $\epsilon$ -differentially private.

## Generalization

- ▶ selecting the “best” something according to some external utility measure
- ▶ applies in particular when the mapping between instances and utility is very sensitive

## Setting

- ▶ a set of possible answers  $\mathcal{R}$
- ▶ a utility measure  $u$  from  $\mathcal{X}^N \times \mathcal{R}$  to  $\mathbb{R}$
- ▶ ideal answer:  $\arg \max_{r \in \mathcal{R}} u(\mathcal{D}, r)$

## Sensitivity

- ▶ the sensitivity of  $u$  is given by

$$\Delta u = \max_{r \in \mathcal{R}} \max_{d(\mathcal{D}_1, \mathcal{D}_2) \leq 1} |u(\mathcal{D}_1, r) - u(\mathcal{D}_2, r)|$$

- ▶ notice this is *not* a sensitivity with respect to  $r$ !

## Exponential mechanism

- ▶ output  $r$  with probability proportional to  $\exp\left(\frac{\epsilon u(\mathcal{D}, r)}{\Delta u}\right)$
- ▶ somewhat related to the softmax principle

The Exponential mechanism is  $\epsilon$ -differentially private.

## Mechanisms

- ▶ many other mechanisms have been designed
- ▶ the main idea is to exploit the structure of the query to reduce the budget consumption
- ▶ a particular attention has been given to answering to a set of queries rather than to a single one
- ▶ limited by a recent result from Ullman: if we do not restrict the range of queries or accept exponential running time, the Laplace mechanism is essentially optimal

## Practical implementations

- ▶ **PINQ** and related models
- ▶ tools to analyze automatically release mechanisms

## Differentially private data science

- ▶ very active field of research
- ▶ request based point of view: contradictory with the data science day to day practice
- ▶ dp version of machine learning algorithms:
  - ▶ decision trees
  - ▶ general stochastic gradient descent
  - ▶ k-means and other unsupervised models

## Synthetic data release

- ▶ an old solution: build a statistical model of the data and release a sample generated by the model
- ▶ ongoing work on relating this approach to differential privacy



## Differential privacy

- 😊 strong theoretical guarantees
- 😊 very active field with constant progress
- 😞 very complex
- 😞 negative results

## Future

- ▶ privacy guarantees are here to stay
- ▶ more and more large scale adoption (official statistical institutes, Google, Apple, etc.)
- ▶ regulation will probably impose some minimal guarantees in the future

# Take home message

- ▶ k-anonymity and related deterministic methods are slowly being abandoned
- ▶ full data release is impossible without introducing privacy risks
- ▶ privacy breaches propagate and cannot be undone
- ▶ the “look first” approach of data science is fundamentally in contradiction with the request oriented approach of secure systems
- ▶ differential privacy and related concepts are slowly becoming the main solution for privacy preservation

- ▶ Captain Obvious image:

<https://imgur.com/gallery/PazzF>



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-sa/4.0/>

Last git commit: 2019-01-17

By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)

Git hash: 21a7fea6a63ad78a37f024769a5abf0bebccd946