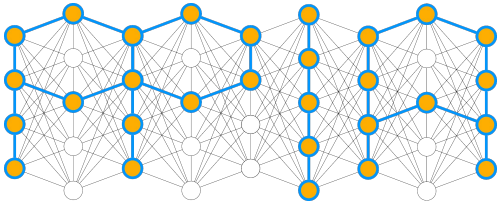


DIFFERENTIAL PRIVACY: PART I

PRIVACY PRESERVING INFORMATION ACCESS

PhD in Information Engineering
A.Y. 2025/2026

GUGLIELMO FAGGIOLI
Intelligent Interactive Information Access (IIIA) Hub
Department of Information Engineering
University of Padua

FUNDAMENTAL LAW OF INFORMATION RECOVERY

Overly accurate answers to too many questions will destroy privacy in a spectacular way.

Corollary: Data Cannot be Fully Anonymized and Remain Useful.

This result, which can be mathematically proven, applies to all techniques for privacy-preserving data analysis, and not just to differential privacy.

DATA CANNOT BE FULLY ANONYMIZED AND REMAIN USEFUL

The richer the data, the easier it is to carry out linkage attacks.

Conversely, fully anonymized data require complete removal of quasi-identifiers, making it impossible to extract useful information.

Differential privacy neutralizes linkage attacks: being differentially private is a property of the data access mechanism, and is unrelated to the presence or absence of information available to the malevolent user to carry that might enable linkage attacks.

RE-IDENTIFICATION IS NOT THE ONLY RISK

We already know that: k-anonymization should reduce the risk of re-identification, but it suffers from the background knowledge and skewness vulnerabilities.

QUERIES OVER LARGE SETS ARE NOT PROTECTIVE

We might assume that it is easy to apply techniques such as k-anonymity to large datasets and this might help protecting private information.

Nevertheless, even large datasets are vulnerable to **differencing attack**.

QUERIES OVER LARGE SETS ARE NOT PROTECTIVE

differencing attack: assume that we know the subject of interest, called X, is in a database which contains a (binary) sensitive attribute b.

By running two queries q_1 and q_2, defined as follows:

q_1 tells us how many subjects on the dataset have the attribute b = true

q_2 tells us how many subjects, not named X, have the attribute b = true.

the difference between the output for q_1 and q_2 tells us the value of b for the subject of interest (if it's 0, then b=true for the subject of interest).

A DIFFERENCING ATTACK

```
adult[“income”].sum()
```

> 6,039,000

```
adult[adult[“name”]!= “Steven”][“income”].sum()
```

> 6,020,000

Steven income: 19,000

QUERY AUDITING IS PROBLEMATIC

We might consider to *audit* queries, refusing to answer to queries which might disclose information, based on queries already executed (e.g., we might not answer to q_2 in the previous example).

This has 2 limitations:

- not disclosing something is still informative
- it is infeasible computationally

SUMMARY STATISTICS ARE NOT SAFE

The differencing attack highlights why summary statistics do not provide privacy.

Besides, **reconstruction attacks** done through **record linkage** allow to reconstruct secret information in linear time.

ORDINARY FACTS SHOULD NOT BE NEGLECTED

There is almost no fact that does not disclose information about the the subject.

“The subject was used to buy bread regularly for several years, then switched suddenly to buy it rarely.”

ORDINARY FACTS SHOULD NOT BE NEGLECTED

There is almost no fact that does not disclose information about the the subject.

“The subject was used to buy bread regularly for several years, then switched suddenly to buy it rarely.”

“The subject has been diagnosed with type 2 diabetes”

“JUST A FEW” POLICY

Sacrifice the privacy of “just a few”, to provide more informative data.

- Outliers are most likely those that will be in the “just a few” set: they are also likely those for which the privacy should be protected the most.
- We do not know what “just a few” is.

Differential privacy allows to achieve similar results, without the need for moral compromises.

THE PROBLEM

How can I protect my privacy completely? By not putting my record in the database!

Can we build an approach that allows a data analyst to learn about me as much information as if I did not contribute at all to the database?

My contribution/Non contribution corresponds to have two databases: one with me and one without.

AN EXAMPLE

Assume I can somehow be harmed by a certain trial - for example a trial about the correlation between the **consumption of coffee and being restless**.

Being restless is a risk factor: an insurance company might decide to increase the insurance premium to coffee drinkers if the correlation was proved.

I therefore might prefer avoid taking part into the trial to avoid giving evidence of such correlation.

Differential privacy aims at producing statistics which are the same independently of whether or not I was in the study.

DIFFERENTIAL PRIVACY IS NOT AN ALGORITHM

Differential privacy is a definition, not an algorithm.

For a given computational task T and a given value of ε there will be many algorithms falling into the differentially private framework for achieving T in an ε-differentially private manner

DIFFERENTIAL PRIVACY

Differential privacy will provide privacy by **process**; in particular it will introduce randomness.

COIN TOSS (RANDOMIZED RESPONSE)

Privacy by randomness algorithm.

Flip a coin:

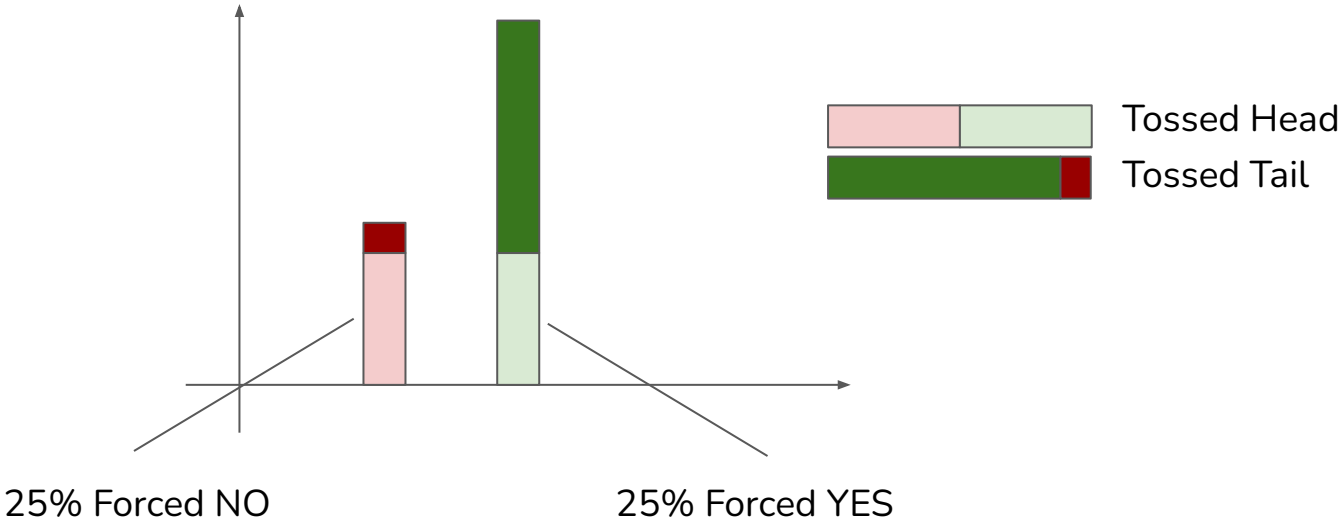
- if “tail”, respond truthfully
- if “head”, flip coin again:
  - if “tail”, respond “YES”
  - if “head”, respond “NO”

DENIABILITY OF THE OUTCOME

Technique developed by social scientists to collect information about embarrassing or illegal behaviours: the privacy is achieved thanks to the fact that the outcome is always deniable.

the probability of observing a “false NO” or a “false YES” is ¼ each. then, we know that among the “NO”s ¼ are yes and similarly, among “YES”s only ¾ are true.

DENIABILITY OF THE OUTCOME



DENIABILITY OF THE OUTCOME

How many “true YES” are there in the answers?

Called Y the proportion of “YES” observed and p is the proportion of “true YES”:

$$Y =$$

DENIABILITY OF THE OUTCOME

How many “true YES” are there in the answers?

Called Y the proportion of “YES” observed and p is the proportion of “true YES”:

$$Y = \boxed{3/4p}$$

Among the p true yes ½ answered yes freely, and ¼ answered yes thanks to the second coin toss

DENIABILITY OF THE OUTCOME

How many “true YES” are there in the answers?

Called Y the proportion of “YES” observed and p is the proportion of “true YES”:

$$Y = 3/4p + 1/4(1-p) = 1/4 + 2/4p$$

Among the 1-p true no ¼ answered answered yes thanks to the second coin toss

DENIABILITY OF THE OUTCOME

How many “true YES” are there in the answers?

Called Y the proportion of “YES” observed and p is the proportion of “true YES”:

$$Y = 3/4p + 1/4(1-p) = 1/4 + 2/4p$$

and thus:

$$p = 2Y - 1/2$$

PROBABILITY SIMPLEX

Probability Simplex: Given a discrete set B, the probability simplex over B, denoted Δ(B) is defined to be:

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|} \; x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{|B|} x_i = 1 \right\}$$

the probability simplex is nothing more than a vector of real numbers between 0 and 1 which add up to 1.

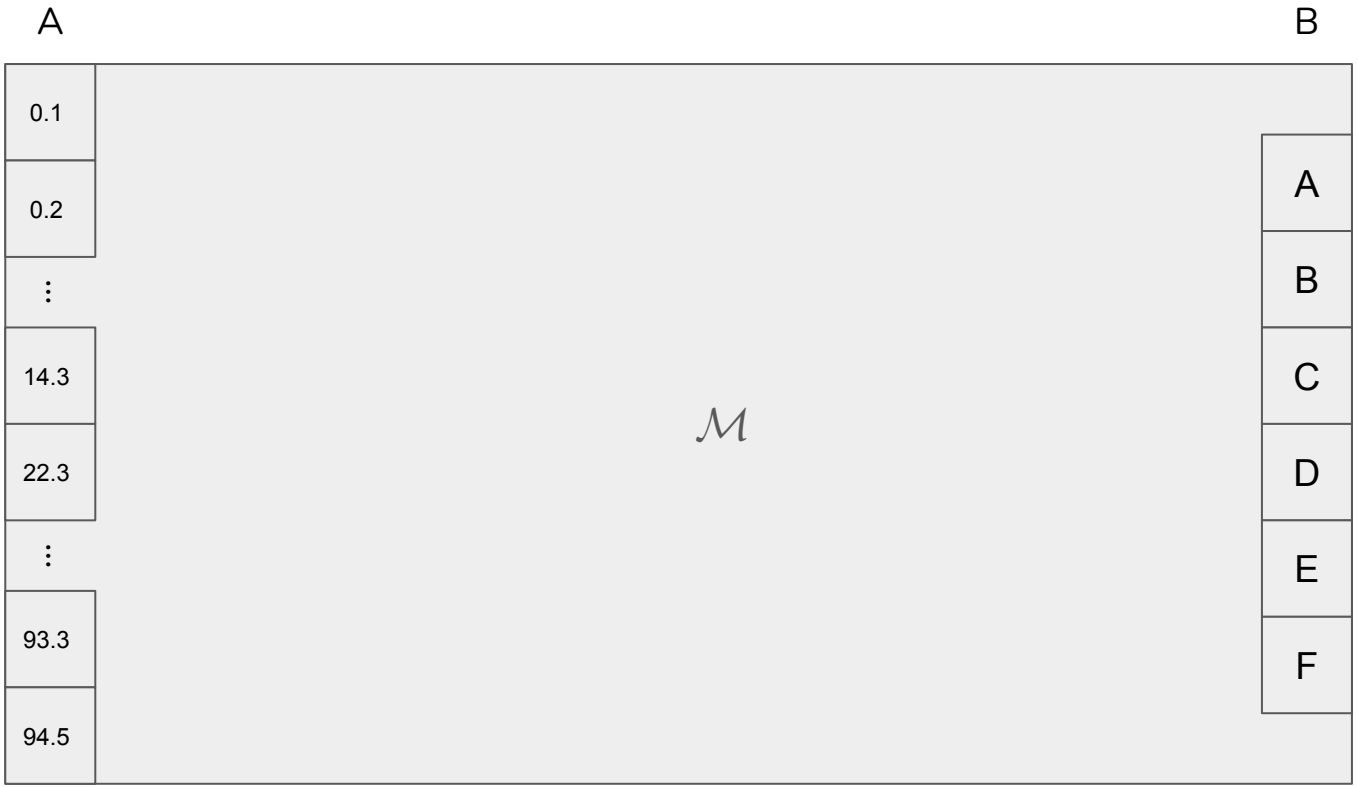
RANDOMIZED ALGORITHM

Randomized Algorithm: A randomized algorithm $\mathcal{M}$ with domain A and discrete range B is associated with a mapping $M : A \rightarrow \Delta(B)$.

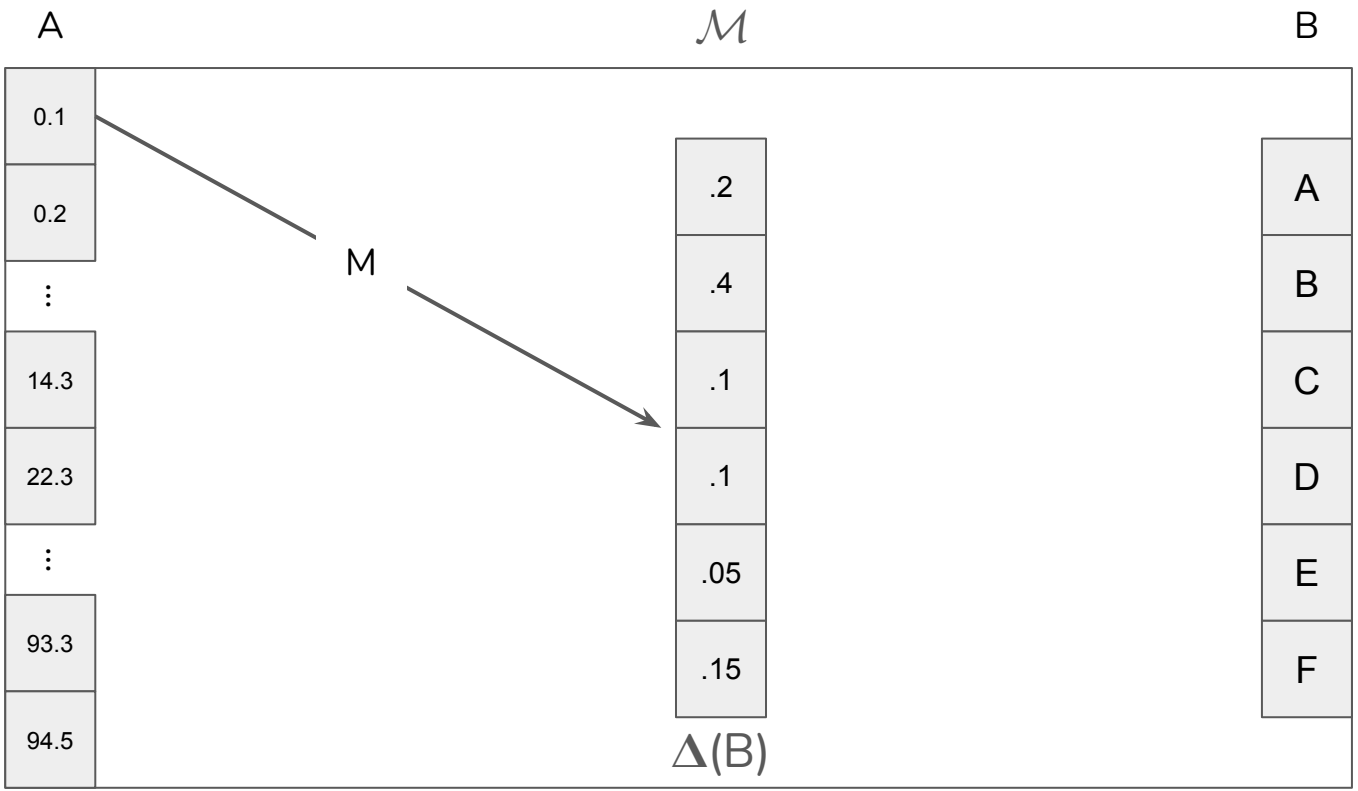
On input $a \in A$, the algorithm $\mathcal{M}$ outputs $\mathcal{M}(a) = b$ with probability $(M(a))_b$ for each $b \in B$.

The probability space is over the coin flips of the algorithm $\mathcal{M}$.

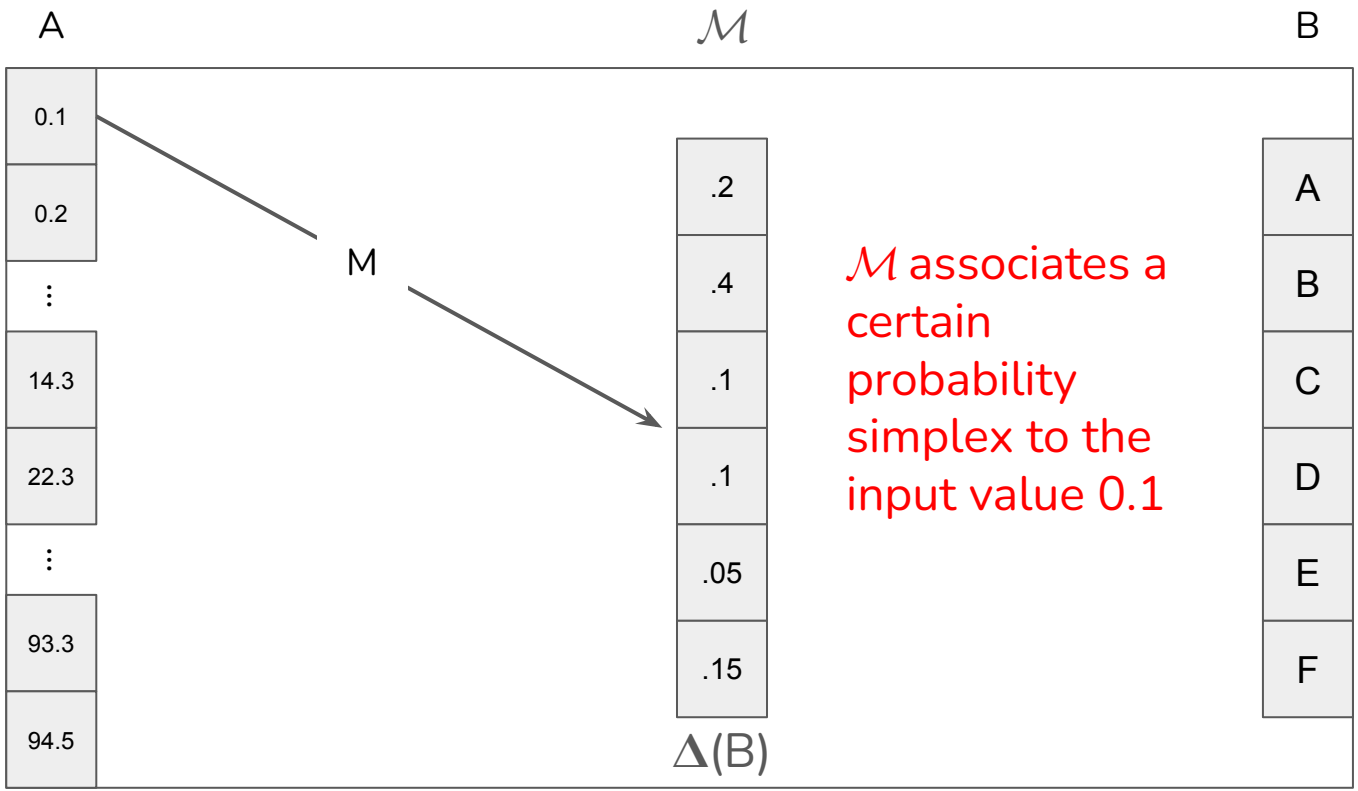
RANDOMIZED ALGORITHM



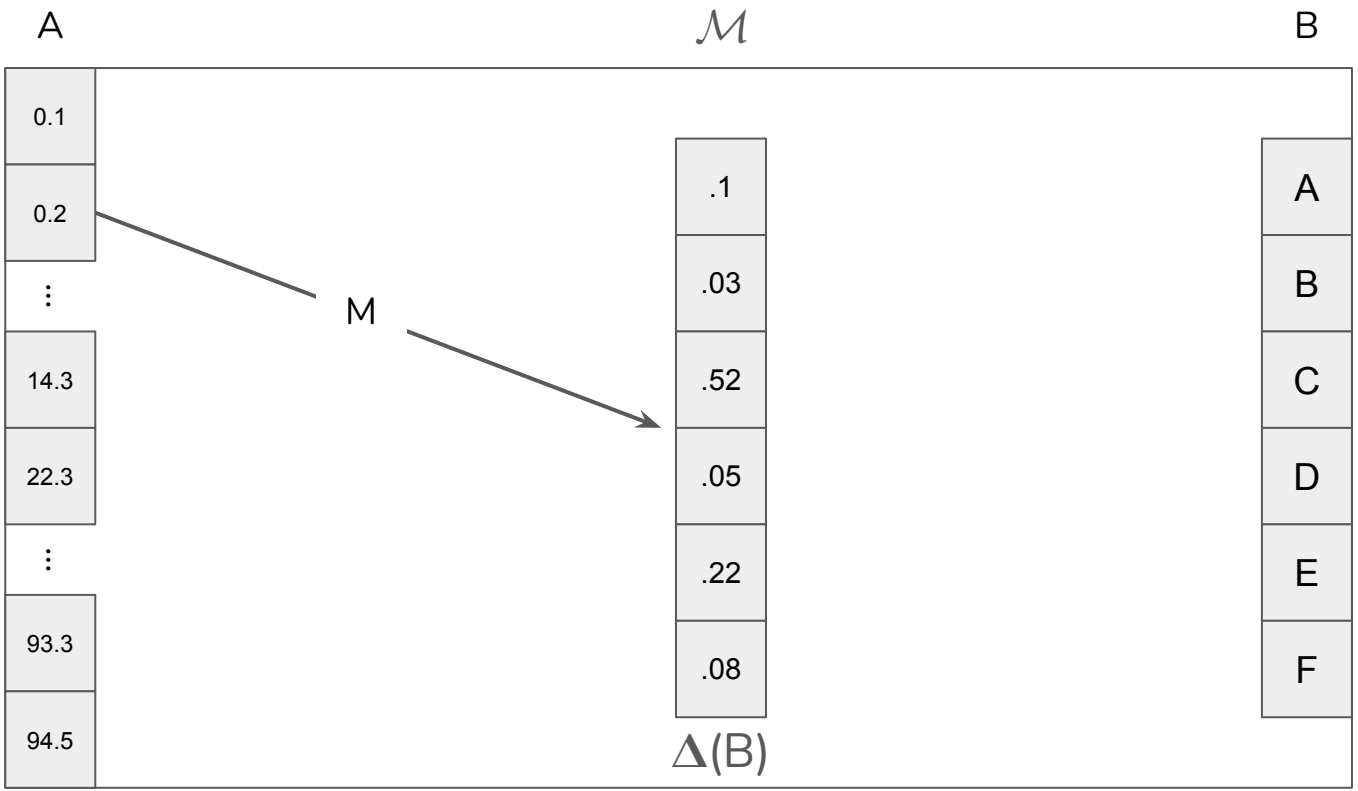
RANDOMIZED ALGORITHM



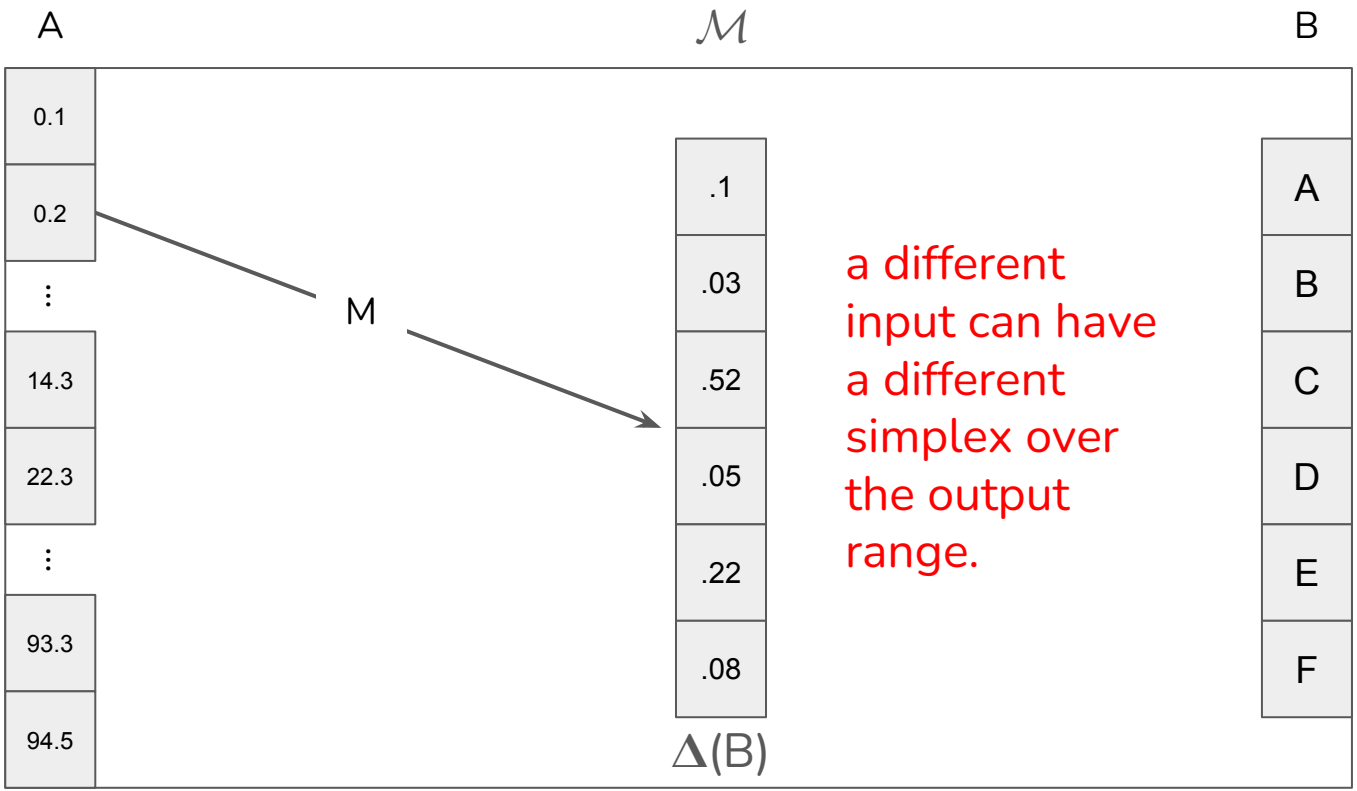
RANDOMIZED ALGORITHM



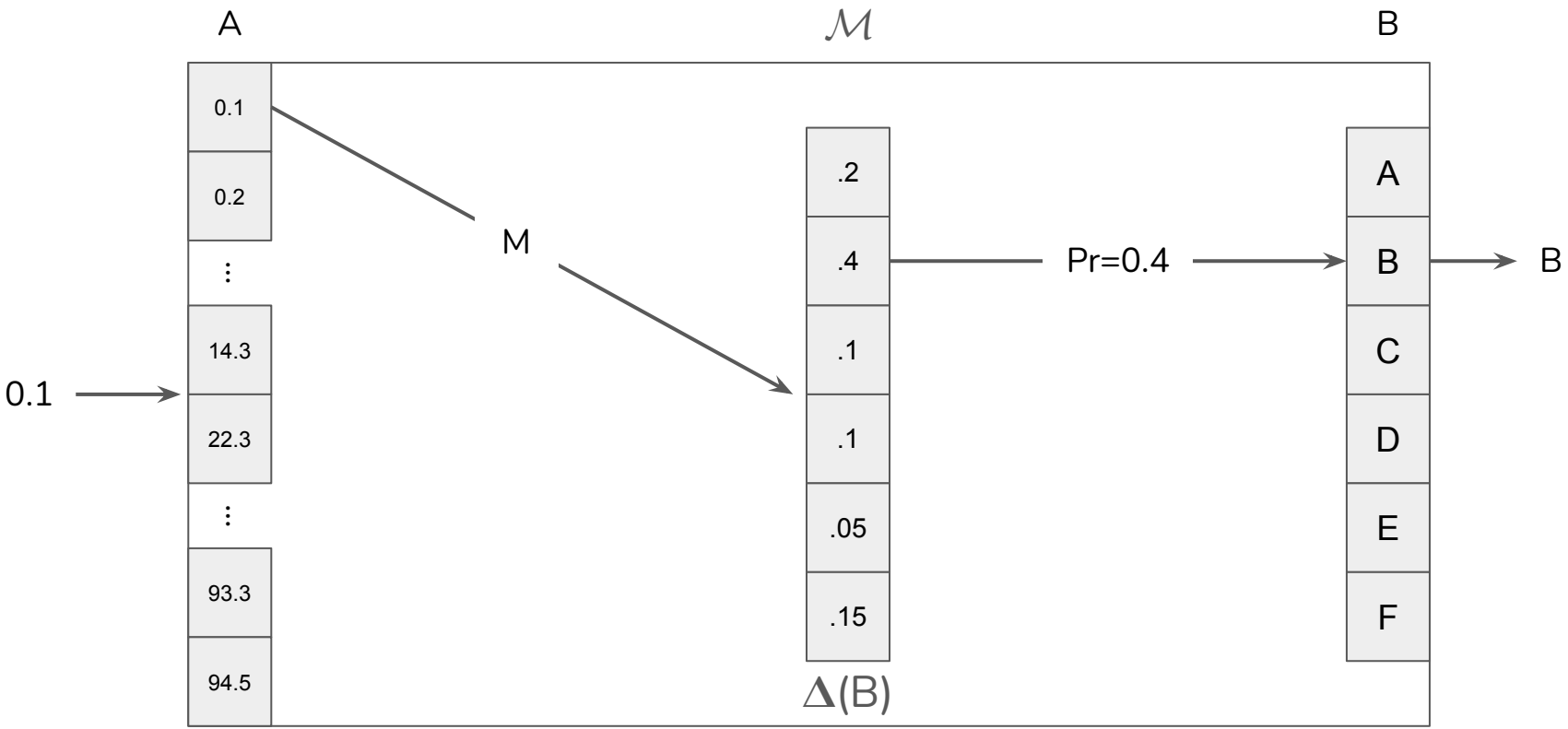
RANDOMIZED ALGORITHM



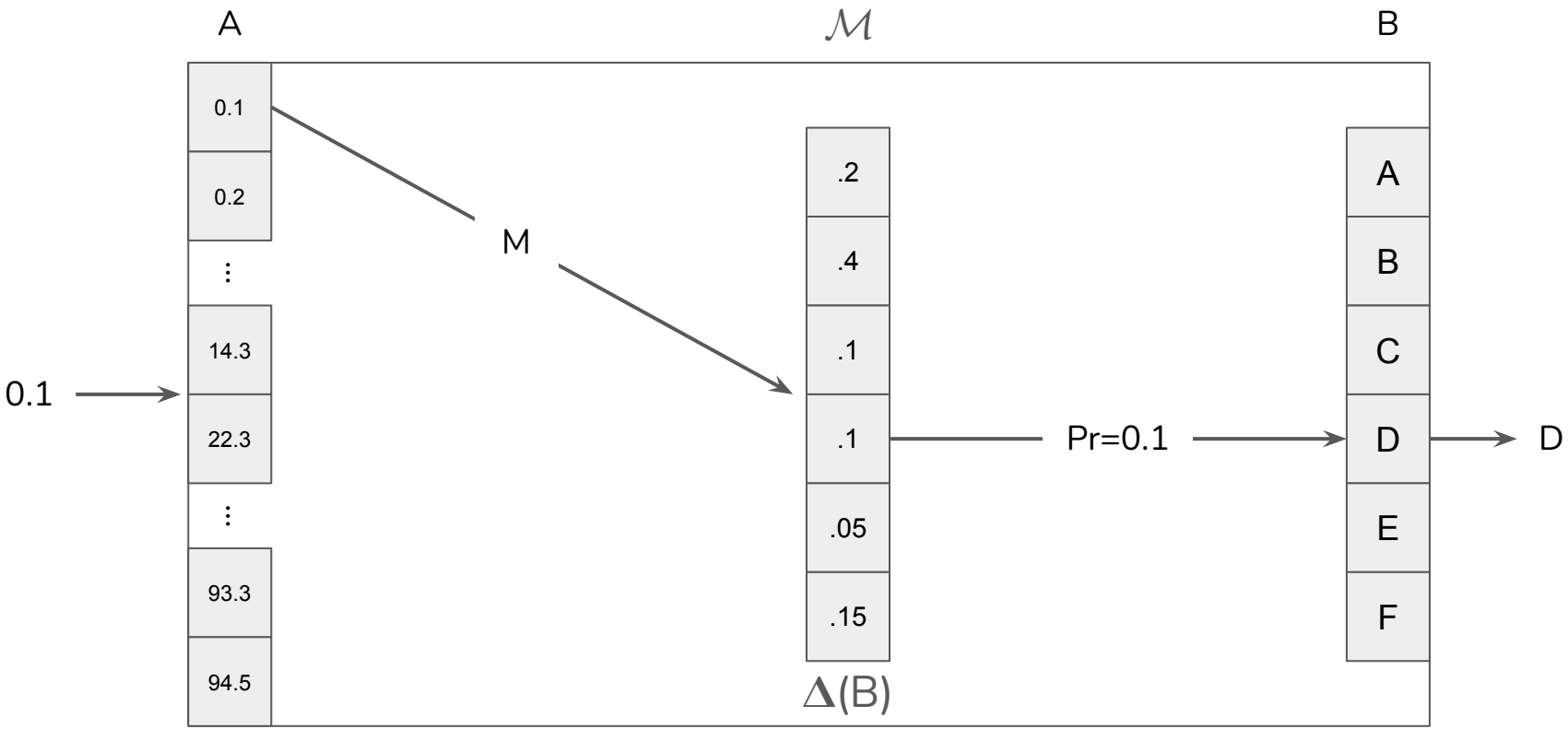
RANDOMIZED ALGORITHM



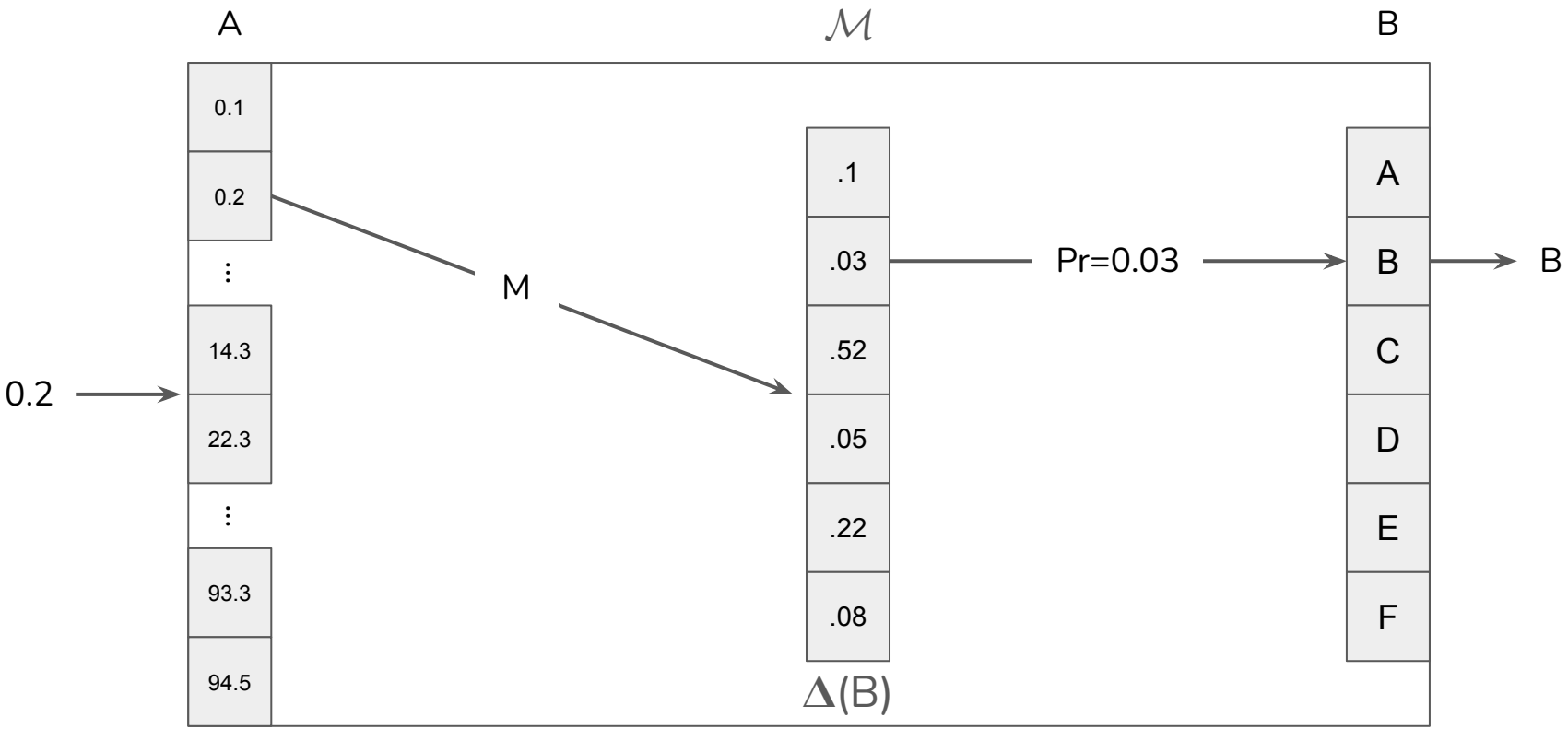
RANDOMIZED ALGORITHM



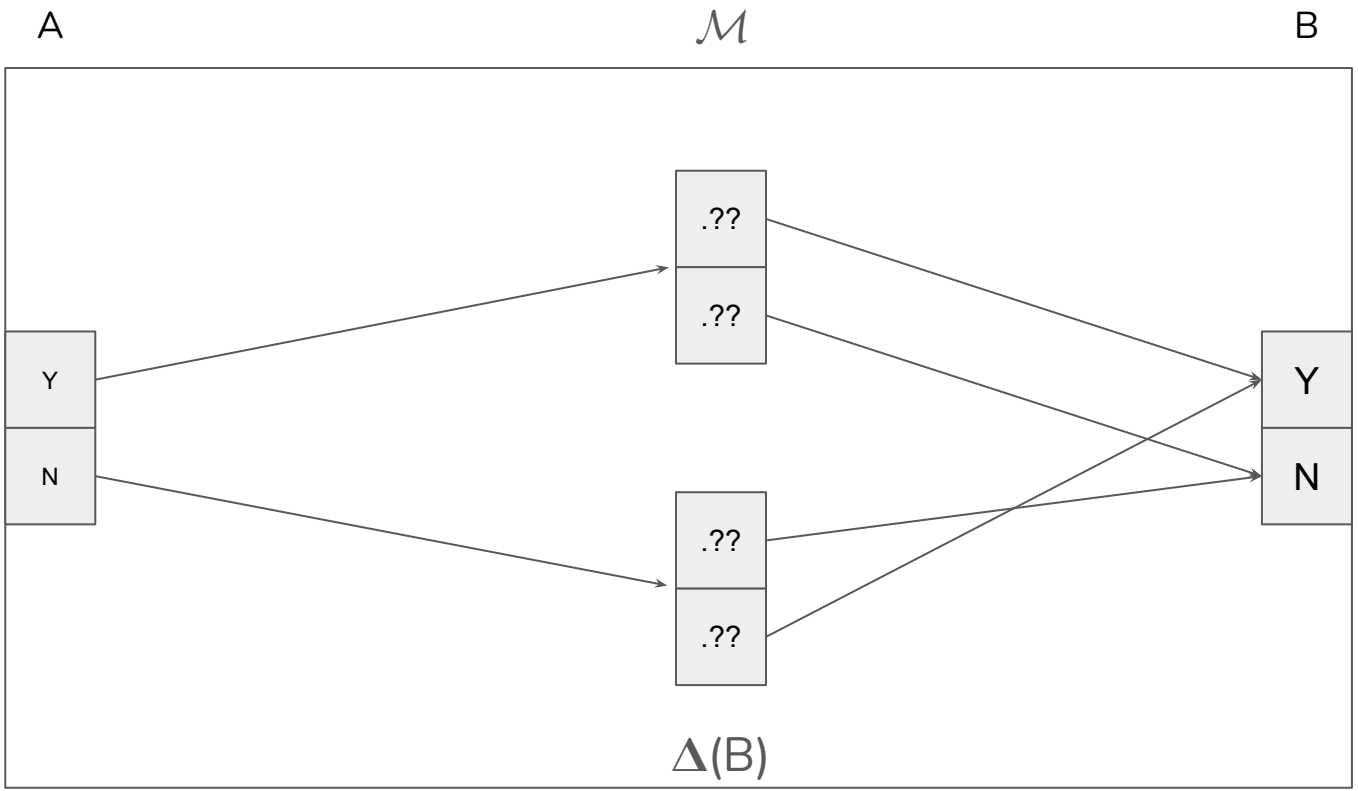
RANDOMIZED ALGORITHM



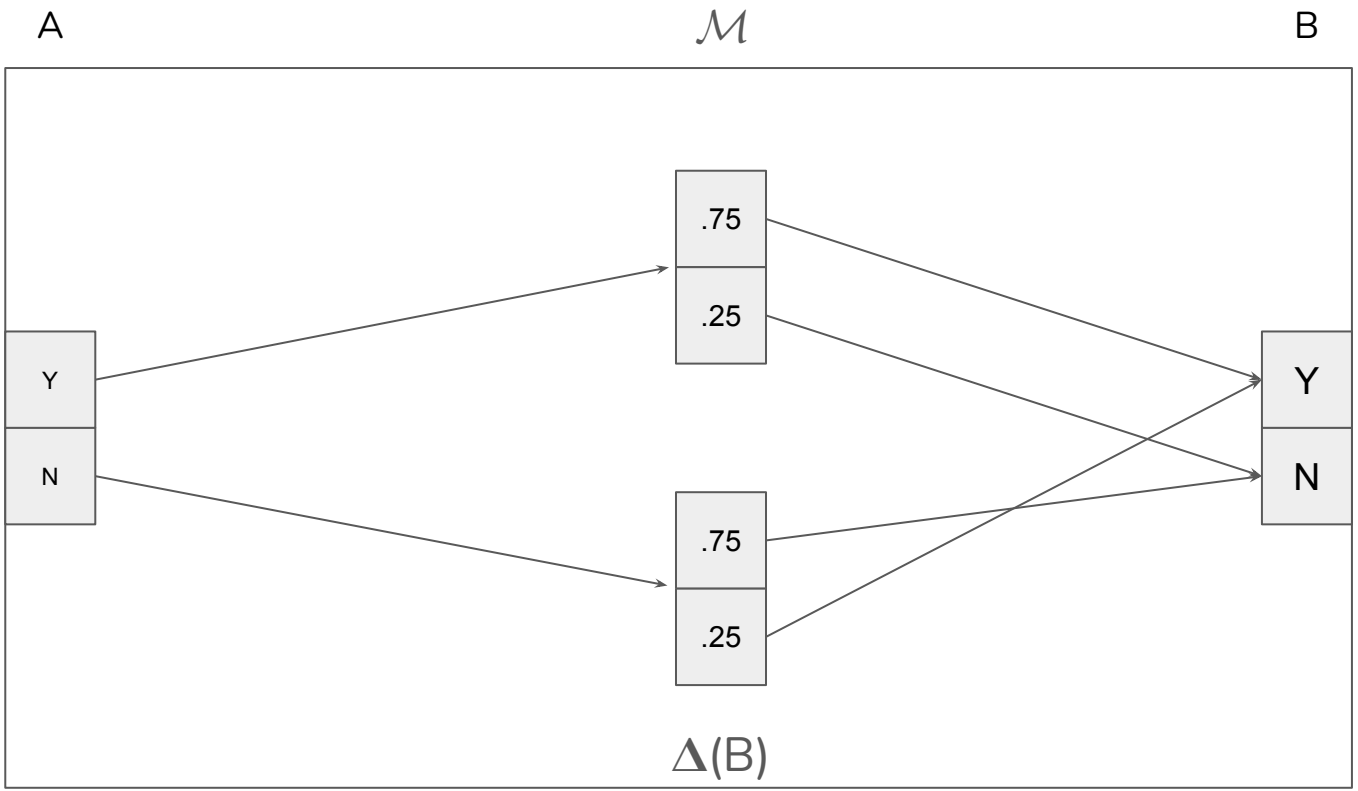
RANDOMIZED ALGORITHM



COIN TOSS



COIN TOSS



A NOTE ON THE OUTPUT

Even though we might use the randomized algorithm to output a single value, this is not very interesting: remember the coin-toss → **the response might be simply random**.

A much more relevant information is **the distribution of the outputs**: by looking at the (false) distribution of YES and NO in the coin toss example, we were able to compute the expected proportion of true YES.

A NOTE ON THE OUTPUT

Therefore, from now on, we will interpret $\mathcal{M}(x)$ **as a distribution over the domain of x**.

For example, the distribution of weights in our population, the distribution of subjects committing illegal behaviours and so on.

THE DATABASE HISTOGRAM

We will think of databases x as being collections of records from a universe $\mathcal{X}$.

Each database x can be further represented using a vector $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of type $i \in \mathcal{X}$.

This definition represents the **histogram of the database**: given the domain of all possible records, it describes the number of records (in a certain database) falling into each equivalence class.

DISTANCE BETWEEN DATABASES

The ℓ_1 norm of a database x is denoted $||x||_1$ and is defined as:

$$||x||_1 = \sum_{i=1}^{|\mathcal{X}|} |x_1|$$

ℓ_1 distance between two databases x and y is $||x - y||_1$

- $||x||_1$ is a measure of the size of a database x (i.e., the number of records it contains)
- $||x-y||_1$ is a measure of how many records differ between x and y.

DIFFERENTIAL PRIVACY

Differential Privacy:

intuitively guarantees that a randomized algorithm *behaves similarly* on similar input databases.

“behaves similarly” means that the output of the randomized algorithm will be similar for both databases.

NEIGHBOURING DATASETS

Differential privacy relies on the concept of **neighbouring databases**: databases that differ for only one record.

If the output is (almost) the same on two neighbouring databases, then **any record of the database might be fake: we would observe (almost) the same result!**

if $||x-y||_1 = 1$, then x and y are two neighbouring datasets.

DIFFERENTIAL PRIVACY

A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all S ⊆ Range($\mathcal{M}$) and for all x, y ∈ $\mathbb{N}^{|\mathcal{X}|}$ such that $||x - y||_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(y) \in S] + \delta$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If δ = 0, we say that M is ε-differentially private.

DIFFERENTIAL PRIVACY

considering δ = 0, since y and x are neighbouring datasets (which is a symmetrical property) it is also true that:

$$\Pr[\mathcal{M}(y) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x) \in S]$$

therefore:

$$e^{-\varepsilon} \Pr[\mathcal{M}(y) \in S] \leq \Pr[\mathcal{M}(x) \in S]$$

thus:

$$e^{-\varepsilon} \Pr[\mathcal{M}(y) \in S] \leq \Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(y) \in S]$$

WHICH EPSILON SHOULD WE USE?

if ε is small...

$$e^0 \Pr[\mathcal{M}(y) \in S] \leq \Pr[\mathcal{M}(x) \in S] \leq e^0 \Pr[\mathcal{M}(y) \in S]$$

$$\Pr[\mathcal{M}(y) \in S] \leq \Pr[\mathcal{M}(x) \in S] \leq \Pr[\mathcal{M}(y) \in S]$$

$$\Pr[\mathcal{M}(y) \in S] = \Pr[\mathcal{M}(x) \in S]$$

2 different (neighbouring) datasets, same output.
maximum privacy: **the output does not depend on the data (i.e., its noise)**.

WHICH EPSILON SHOULD WE USE?

Giving a precise rule-of-thumb is very hard.

If ε is very small, even if a mechanism is not $(\varepsilon, 0)$-differentially private, then it might be $(2\varepsilon, 0)$-differentially private, **which might be still acceptable**.

WHICH EPSILON SHOULD WE USE?

Failure to be (15, 0)-differentially private tells that there exist neighboring databases and an output *o* for which the ratio of probabilities of observing *o* conditioned on the database being, respectively, *x* or *y*, is large.

If *o* is particularly unlikely, then x and y looks particularly “strange” or artifact, and thus the malevolent data scientist wouldn’t gain much information.

as a rule of thumb, most common values for ε are between 0.1 and 5 - typical values are ln(2) and ln(3).

WHICH EPSILON SHOULD WE USE?

Authors	Value(s) of ε	Application
McSherry-Mahajan [32]	0.1—10	Network Trace Analysis
Chaudhuri-Monteleoni [7]	0.1	Logistic Regression
Machanavajjhala et al. [30]	< 7	Census Data
Korolova et al. [24]	$\ln 2, \ln 5, \ln 10$	Click Counts
Bhaskar et al. [5]	1.4	Frequent Items
Machanvajjhala et al. [31]	0.5—3	Recommendation System
Bonomi et al. [6]	0.01—10	Record Linkage
Li et al. [28]	0.1—1	Frequent Items
Ny-Pappas [38]	$\ln 3$	Kalman Filtering
Chaudhuri et al. [8]	0.1—2	Principal Component Analysis
Narayan-Haeberlen. [34]	0.69	Distributed Database Joins
Chen et al. [10]	$1.0 - 5.0$	Queries over Distributed Clients
Acs-Castelluccia [2]	1	Smart Electric Meters
Uhler et al. [41]	0.1—0.4	Genome Data
Xiao et al. [43]	0.05—1	Histograms
Li-Miklau. [27]	0.1—2.5	Linear Queries
Chen et al. [9]	0.5—1.5	Trajectory Data
Cormode et al. [12]	0.1—1	Location Data
Chaudhuri et al. [40]	0.01—0.5	Empirical Risk Minimization

Table 1: Values of ε in the literature

J. Hsu *et al*., "Differential Privacy: An Economic Method for Choosing Epsilon," *2014 IEEE 27th Computer Security Foundations Symposium*, 2014, pp. 398-410, doi: 10.1109/CSF.2014.35.

DIFFERENTIAL PRIVACY: δ

a typical value for δ is far less than the inverse of any polynomial in the size of the database.

If $\delta = 1/||x||_1$ permits to “preserve the privacy”, by publishing a small number of records (“Just a few” policy).

DIFFERENTIAL PRIVACY: δ

δ = 0 grants that, for every run of the randomized algorithm $\mathcal{M}$, the output is (almost) equally likely to be observed on every neighboring database, simultaneously.

(ε, δ)-differential privacy says that for every pair of neighboring databases x, y, it is unlikely (with probability δ) that, the observed value M(x) will be much more or much less likely to be generated when the database is x than when the database is y.

EPSILON AND DELTA

epsilon: privacy budget. the lower, the higher the noise and thus data protection.

delta: probability of failing in protecting the privacy. It represents a form of relaxation of DP, where we assume that, for some rare inputs, the privacy bounds are larger than expected.

PRIVACY LOSS

A very important measure is the **privacy loss** incurred if the output *o* is observed and which can be computed as:

$$\mathcal{L}^{o}_{\mathcal{M}(x)||\mathcal{M}(y)} = \ln\left(\frac{\Pr[\mathcal{M}(x) = o]}{\Pr[\mathcal{M}(y) = o]}\right)$$

An important lemma tells us that:

$$\left|\mathcal{L}^{o}_{\mathcal{M}(x)||\mathcal{M}(y)}\right| \leq \varepsilon, \text{ with probability } 1 - \delta$$

PRIVACY LOSS

We can now revisit our coin toss algorithm. Assume to have two databases, with a single record, b and b' which are neighbouring.

if they are neighbouring, they must differ for a single record: then assuming, without loss of generality, b=Y, b'=N.

we know that Pr[M(b) = Y] = ¾, while P[M(b')=Y] =¼. thus:

$$\varepsilon \geq \left| \log \left(\frac{3/4}{1/4} \right) \right| = \log 3$$

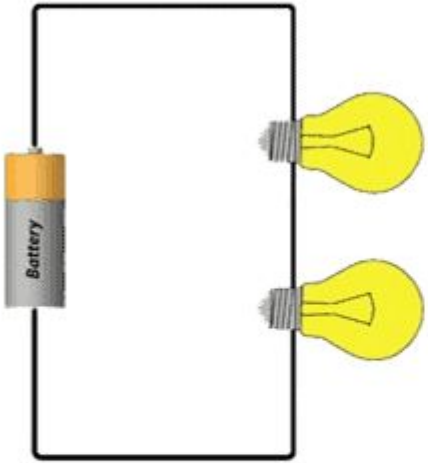
The coin toss algorithm is (log 3, 0)-differentially private.

COMPOSITION

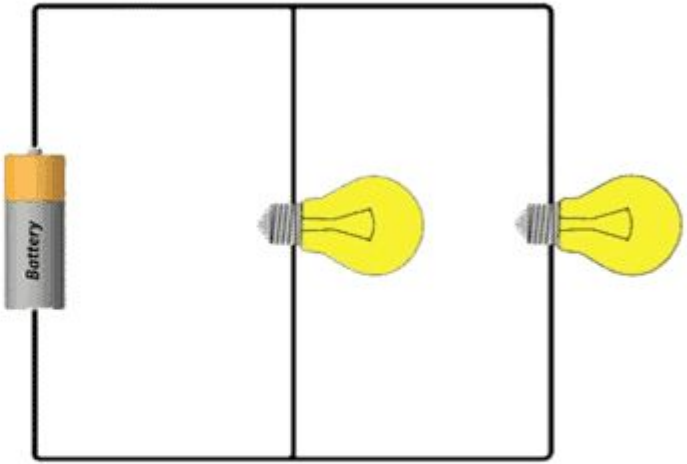
The combination of two differentially private mechanisms is again a differentially private mechanism.

You can use the simplest differential privacy mechanisms as building blocks for more complex DP solutions.

COMPOSITION



If (independent) mechanisms are placed in series, the privacy guarantee (upper bound) is the **sum of the privacy guarantees**.



If (independent) mechanisms are placed in parallel - on different partitions of the dataset - the **privacy guarantee is the maximum of ε**.

GROUP DIFFERENTIAL PRIVACY

What if in our data we have a family of k, rather than a person? Would the privacy of the family remain protected?

GROUP DIFFERENTIAL PRIVACY: THEOREM

Any (ε, 0)-differentially private mechanism $\mathcal{M}$ is (kε, 0)-differentially private for groups of size k. That is, for all $||x - y||_1 \leq k$ and all $S \subseteq \text{Range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(x) \in S] \leq e^{k\varepsilon} \Pr[\mathcal{M}(y) \in S]$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$.

POST-PROCESSING

Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ be a randomized algorithm that is (ε, δ)-differentially private. Let $f : R \rightarrow R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R'$ is (ε, δ)-differentially private.

A function applied to a randomized algorithm, returns again a randomized algorithm: it is impossible to go back to the original input.

POST-PROCESSING

Differential Privacy is immune to post processing: it is not possible to compute a function of the output of a differentially private algorithm to make it less differentially private.

PROPERTIES OF THE DIFFERENTIAL PRIVACY

- Automatic neutralization of linkage attacks (aggregate statistics)
- Quantification of the privacy loss
- Composition
- Group Privacy
- Closure Under Post-Processing