

Fundamental trade-offs between privacy and utility in data sharing

Parastoo Sadeghi

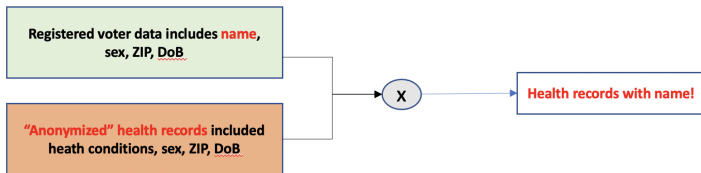


CSCIT 2021 - Croucher Summer Course in Information Theory
Hong Kong
August, 2021

- There is often a correlation between data that (most) people deem **useful** or **“sharable”** and data that (most) people deem **private** or **“protected”**.
- For example, (most) people consider their **highest education degree** sharable or non-private. Let us call this variable X .
- On the other hand, (most) people consider their **income** as private. Let us call this variable S .
- It is no secret that education and income are highly correlated. So if we reveal X we may inevitably reveal something about S .
- National census data may even be available on the joint distribution $P_{S,X}$ between income and education.

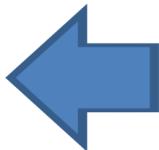
Background - 2

- Between 50% to 80% of people in the US are uniquely identified by their full DoB, ZIP code and sex.
- Simple “anonymisation” of datasets and publishing them or using them in training algorithms has a serious privacy risk.
- One of the first examples: (to prove a point) Latanya Sweeney (circa 1995) [1] managed to hack into “anonymised” health records of the Governor of Massachusetts by linking it with publicly available voter data.
- Even today, reconstruction attacks occur on the so called “anonymised” records.



This is a big challenge for national data curators. The following example is from a recent presentation by the US Census Bureau.

We now know that this publication can be reverse-engineered to reveal the confidential database.



	Count	Median	Mean
Total	7	30	38
# Female	4	30	33.5
# male	3	30	44
# black	4	51	48.5
# white	3	24	24
Married	4	51	54
Black F	3	36	36.7

This table can be expressed by 164 equations.
Solving those equations takes
0.2 seconds on a 2013 MacBook Pro. 29

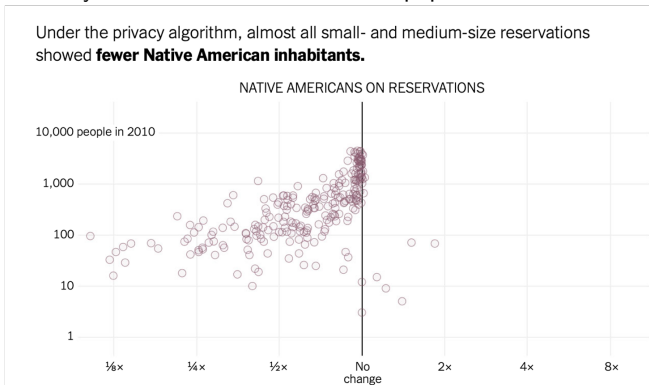
- US Census in 2020: “Every person matters for federal funding.”
- To preserve privacy: “Imaginary people will be added to some locations and real people will be removed from others.”
- “Minorities and rural areas are at most risk.”



<https://www.nytimes.com/interactive/2020/02/06/opinion/census-algorithm-privacy.html#commentsContainer>

Background - 5

- A very popular privacy preserving method (e.g., in differential privacy) is noise injection to the true data.
- Significant under-reporting would have occurred if (Laplace) noise were to be added to the true population count of native Americans in the 2010 US Census. Vertical middle line is the 2010 Census data (source of truth); Dots are noisy versions of different counties population.



<https://www.nytimes.com/interactive/2020/02/06/opinion/census-algorithm-privacy.html#commentsContainer>

- It should be intuitively understood that there is a **tradeoff** between privacy of individuals and utility (accuracy) of data.

Managing the Tradeoff



United States
Census
Bureau

U.S. Department of Commerce
Economic and Statistics Administration
U.S. CENSUS BUREAU
census.gov

Source of picture: <https://simson.net/ref/2019/2019-07-16%20Deploying%20Differential%20Privacy%20for%20the%202020%20Census.pdf>

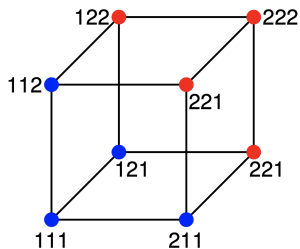
- A student has done research on the distribution of education, sex, and age versus postcode in a population of size $n = 100$.
- The question is whether this student should publicise their results **unperturbed**.
- It should be clear that even if the unperturbed data is published **"anonymously"**, it can lead to revealing or leaking information about **income** of individuals in such a small dataset.
- **Randomised or deterministic perturbation** of income data is required before publication to reduce **statistical disclosure**.
- Data perturbation will reduce (but not 100% eliminate) the **guessing or belief refining power of an adversary**.

- The question is how to **optimally** perturb data to protect privacy **AND** still provide accurate information.

ID	Postcode	Education
1	1000	PhD →?
2	1001	High School →?
3	1002	Bachelor →?
⋮	⋮	⋮
100	1000	Drop-out →?

Motivation - scenario 2

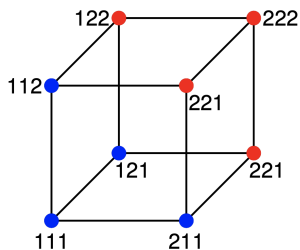
- Consider this toy example:



- Each dot is “a (toy) dataset”: the outcome of votes of three people (who chose between option 1 or 2).
- Each edge connects two datasets **differing only in one vote (neighbours)**.
- Colour of the dot represents the **true** majority outcome **Blue: majority 1** or **Red: majority 2**.

Motivation - scenario 2

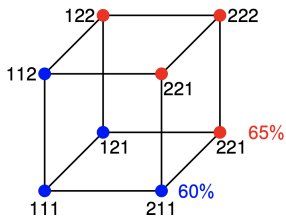
- The question is how should one release majority outcome while protecting the privacy of individuals (who voted what).



- Question: when trying to balance privacy of individuals with overall accuracy of results which dataset(s) pose the most challenge?

Motivation - scenario 2

- With what probability shall we report the true majority?
- Is 65% truthful response “good enough”? Is it “optimal”? What about 60%?
- What do these probabilities mean in terms of privacy and accuracy overall?



- Some privacy-preserving approaches are explicitly **statistical inference oriented**. They require knowledge of the distribution of the underlying sensitive and useful data $P_{S,X}$. One example is information-theoretic mutual information and its variations or generalisations.
- Some approaches aim to be **“agnostic”** to what the adversary may want to “guess” and aim for a **universal** “data-statistics-free” approach to minimise individual or group disclosure. One example is differential privacy and its variations. Another one is maximal leakage.
- They each have their own advantages and disadvantages.
- All approaches are underpinned by **information theory and statistics** and most of them can be studied and compared in a common framework.

- The aim of these 3 lectures is to provide a basic understanding of some of these privacy-preserving approaches, their relation, and formalisation of their fundamental privacy-utility tradeoffs.
- Due to time constraints, the main literature for the lectures is highly selected.
- A set of extra slides is provided at the end (after page ~ 115) for interested students or future reference.
- **Assumed knowledge for the main lectures:** UG knowledge of probability & statistics and a basic (trimester) course in information theory.

- **Model:** variables, general privacy and utility measures, inference cost model, optimisation framework (lecture 1).
- **Privacy funnel:** mutual information measure for both privacy and utility, basic optimisation setup, two greedy algorithms (lecture 1/2).
- **Local information privacy:** log-lift or information density privacy measure, its use in “local” privacy funnel, its use in privacy “watchdog” method (lecture 2).
- **Differential privacy:** basic definition, local differential privacy, their relation to the two information-theoretic privacy measures above, a graph-based (reasonable utility) optimal differential privacy scheme for binary functions (lectures 2/3).

- Privacy funnel and privacy watchdog with Ni Ding and Thierry Rakotoarivelo [2, 3].
- Differential privacy on graphs with Rafael D'Oliveira and Muriel Médard, [4].
- α -privacy watchdog with Ni Ding and Mohammad Amin Zarrabian, [5] (briefly discussed in the extra slides).
- Maximal leakage in index coding with Yucheng Liu, Ni Ding and Thierry Rakotoarivelo, [6] (not discussed).
- Maximal leakage in source/index coding with Yucheng Liu, Lawrence Ong, Phil Yeoh, Sarah Johnson, Joerg Kliewer, [7], [8] (not discussed).
- Differential privacy and low influence with Rafael D'Oliveira, Salman Salamatian, and Muriel Médard, [9] (not discussed).

Key references used for model section

- [10]: du Pin Calmon and Fawaz 2012
- [11]: Liao, Kosut, Sankar and du Pin Calmon, 2019
- [12]: Wang, Basciftci and Ishwar, 2017

- $s \in \mathcal{S}$: sensitive information to protect (e.g., race, political affiliation, vote, income, disease).
- $x \in \mathcal{X}$: useful information to share (e.g., shopping history, education, etc).
- Target application imposes the specific **statistical** data model:

$$P_{s,x}$$

- $y \in \mathcal{Y}$: released variable based on X .

- Markov model

$$S \rightarrow X \rightarrow Y$$

- The **mechanism** is specified by the conditional distribution $P_{Y|X}$.
- Y should provide **utility** about X while protecting **privacy** by limiting the information it reveals about S .
- There is a privacy-utility trade-off (PUT).

Numerical example - joint distribution $P_{S,X}$

$$P_{S,X}(s, x) = \begin{pmatrix} 0.0394 & 0.0306 & 0.0463 & 0.0463 & 0.0204 & 0.0317 & 0.0328 & 0.0317 & 0.0134 \\ 0.0438 & 0.0047 & 0.0466 & 0.0235 & 0.0442 & 0.0017 & 0.0366 & 0.0083 & 0.0022 \\ 0.0061 & 0.0135 & 0.0076 & 0.0387 & 0.0383 & 0.0410 & 0.0359 & 0.0341 & 0.0047 \\ 0.0441 & 0.0264 & 0.0469 & 0.0069 & 0.0464 & 0.0451 & 0.0190 & 0.0015 & 0.0398 \end{pmatrix}$$

- **Question:** How should we make sense of this distribution in the context of PUT?
- Is there anything such as a **safe** or **risky** outcome (s, x) ?

12

¹From now on, we do not explicitly specify what S or X physically or logically represent.

²Probability cells are rounded.

- The privacy of the mechanism is inversely quantified by a general privacy leakage measure $J(\mathcal{S}; \mathcal{Y})$:

$$J : \Delta_{|\mathcal{S}| \times |\mathcal{Y}|} \rightarrow \mathbb{R}_{\geq 0},$$

$\Delta_{|\mathcal{S}| \times |\mathcal{Y}|}$ is the set of all joint probability distributions over \mathcal{S} and \mathcal{Y} .

- The aim of privacy is to **minimise** $J(\mathcal{S}; \mathcal{Y})$.
- $J(\mathcal{S}; \mathcal{Y}) = 0$ when **perfect** privacy is achieved.

The privacy measure **need not be symmetric**. That is, in general

$$J(\mathcal{S}; \mathcal{Y}) \neq J(\mathcal{Y}; \mathcal{S}).$$

- Utility is highly application-dependent.
- The utility of the mechanism output Y about the useful information X is inversely proportional by a general distortion measure $D(P_{X,Y})$:

$$D : \Delta_{|\mathcal{X}| \times |\mathcal{Y}|} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0},$$

$\Delta_{|\mathcal{X}| \times |\mathcal{Y}|}$ is the set of all joint probability distributions over X and Y .

The aim of utility or accuracy is to **minimise** $D(P_{X,Y})$, which **may not be symmetric** in X and Y .

Distortion function examples

- Normally, we are concerned with **expected** distortion:

$$D(P_{X,Y}) = \mathbb{E}_{P_{X,Y}}[d(X, Y)],$$

for some hard distortion function

$$d : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}.$$

- This includes the **probability of error**

$$\mathbb{P}(Y \neq X),$$

by choosing $d(x, y) = \mathbb{1}_{[x \neq y]}$ being the **indicator function** for the condition $X \neq Y$.

- Choosing $d(x, y) = -\log P_{X|Y}(x|y)$ gives the conditional entropy $D(P_{X,Y}) = H(X|Y)$.

- The privacy-utility optimisation problems is

$$\begin{aligned}\epsilon^*(\rho) &= \inf_{P_{Y|X}} J(S; Y), \\ \text{s.t. } & D(P_{X,Y}) \leq \rho.\end{aligned}$$

where according to $S \rightarrow X \rightarrow Y$ we have

$$P_{Y|S}(y|s) = \sum_{x \in \mathcal{X}} P_{Y,X|S}(y, x|s) = \sum_{x \in \mathcal{X}} P_{X|S}(x|s) P_{Y|X}(y|x).$$

- The **privacy-centric** optimisation formulation is

$$\begin{aligned}\rho^*(\epsilon) &= \inf_{P_{Y|X}} D(P_{X,Y}), \\ \text{s.t. } & J(S; Y) \leq \epsilon.\end{aligned}$$

- A (machine) learning adversary outputs a function that captures the belief about S upon observing Y , denoted by $P_{\hat{S}|Y}$, according to

$$S \rightarrow Y \rightarrow \hat{S}.$$

- Denoting the loss by $c(s, y, P_{\hat{S}|Y})$ for each realisation $s \in \mathcal{S}$ and $y \in \mathcal{Y}$, the expected loss is

$$\mathbb{E}_{P_{S,Y}}[c(S, Y, P_{\hat{S}|Y})]$$

- Then a (machine) learning adversary outputs a belief of S to minimise an expected loss (also known as cost, risk or error).

$$P_{\hat{S}|Y}^* = \arg \min_{P_{\hat{S}|Y}} \mathbb{E}_{P_{S,Y}}[c(S, Y, P_{\hat{S}|Y})].$$

- By the same token, denoting the loss **before** observation of Y by $c(s, P_{\hat{S}})$ for each realisation $s \in \mathcal{S}$, the expected loss is

$$\mathbb{E}_{P_S}[c(S, P_{\hat{S}})].$$

- This leads to the optimal belief of S as

$$P_{\hat{S}}^* = \arg \min_{P_{\hat{S}}} \mathbb{E}_{P_S}[c(S, P_{\hat{S}})].$$

- The privacy risk incurred by an adversary's observation of Y is quantified as **the gain in the expected loss upon observing Y** :

$$\Delta C = \mathbb{E}_{P_S}[c(S, P_{\hat{S}})] - \mathbb{E}_{P_{S,Y}}[c(S, Y, P_{\hat{S}|Y})].$$

Log-loss and mutual information

- Log-loss with the observation of Y is defined as

$$c_{\log}(s, y, P_{\hat{S}|Y}) \triangleq -\log(P_{\hat{S}|Y}(s|y)).$$

- Let us expand the expected loss

$$\begin{aligned}\mathbb{E}_{P_{S,Y}}[-\log(P_{\hat{S}|Y}(s|y))] &= -\sum_{s,y} P_{S,Y}(s,y) \log(P_{\hat{S}|Y}(s|y)) \\ &= H(S|Y) \\ &\quad + \sum_y P_Y(y) \underbrace{D_{\text{KL}}(P_{S|Y}(S|Y=y) \| P_{\hat{S}|Y}(S|Y=y))}_{\text{minimised when } P_{\hat{S}|Y=y} = P_{S|Y=y}}.\end{aligned}$$

Therefore, the **optimal inference strategy** (resulting in zero KL divergence) is the true posterior $P_{\hat{S}|Y}^* = P_{S|Y}$.

- Similarly, without the observation of Y , the log-loss is

$$a_{\log}(s, P_{\hat{S}}) = -\log(P_{\hat{S}}(s)).$$

- And its expectation is

$$\begin{aligned}\mathbb{E}_{P_S}[-\log(P_{\hat{S}}(s))] &= -\sum_s P_S(s) \log(P_{\hat{S}}(s)) \\ &= H(S) + D_{\text{KL}}(P_S \| P_{\hat{S}}).\end{aligned}$$

Therefore, the **optimal inference strategy without Y** is the true prior $P_{\hat{S}}^* = P_S$.

Mutual information as a symmetric privacy measure

Therefore, the gain in the expected loss **upon** observing Y is simply

$$J_{\text{MI}}(S; Y) = I(S; Y) = H(S) - H(S|Y) = I(Y; S) = J_{\text{MI}}(Y; S).$$

Key references used for the privacy funnel section

- [13]: Makhdoumi, Salamatian, Fawaz and Médard, 2014 (see also its extended 2020 online version)
- [14]: Naftali, Pereira and Bialek, 2000
- [2]: Ding and Sadeghi, 2019

Problem formulation

- Both privacy and utility are measured through mutual information:

$$\min_{P_{Y|X}} J_{\text{MI}}(S; Y) = I(S; Y),$$

$$\text{s.t. } I(X; Y) \geq \theta.$$

- Equivalent (distortion-based) formulation: $D(P_{X;Y}) = H(X|Y)$ (noting that $H(X)$ is fixed)

$$\min_{P_{Y|X}} J_{\text{MI}}(S; Y) = I(S; Y),$$

$$\text{s.t. } H(X|Y) \leq \rho = H(X) - \theta.$$

Numerical example

$$P_{S,X}(s, x) = \begin{pmatrix} 0.0394 & 0.0306 & 0.0463 & 0.0463 & 0.0204 & 0.0317 & 0.0328 & 0.0317 & 0.0134 \\ 0.0438 & 0.0047 & 0.0466 & 0.0235 & 0.0442 & 0.0017 & 0.0366 & 0.0083 & 0.0022 \\ 0.0061 & 0.0135 & 0.0076 & 0.0387 & 0.0383 & 0.0410 & 0.0359 & 0.0341 & 0.0047 \\ 0.0441 & 0.0264 & 0.0469 & 0.0069 & 0.0464 & 0.0451 & 0.0190 & 0.0015 & 0.0398 \end{pmatrix}$$

- Originally, without any perturbation ($Y = X$)

$$I(S;X) = 0.2233; \quad I(X;Y) = H(X) = 3.11.$$

- Question is can we bring $I(S; Y)$ lower while not affecting $I(X; Y)$ “too much”?
- For this, we need to understand the nature of the optimisation problem.

Duality with information bottleneck

- Markov chain: $U \rightarrow X \rightarrow Y$.
- U is the underlying useful data (e.g., features of an image to be learned), X represents U in the physical world (a digital image) and Y is a compressed version of X .

$$\begin{aligned} \min_{P_{Y|X}} I(X; Y), \\ \text{s.t. } I(U; Y) \geq \mu. \end{aligned}$$

- The objective is to maximise the compression rate while maintaining a good level of information left between U and Y .
- This is the opposite of the privacy funnel optimisation and is a generalised rate-distortion problem with three variables connected via a Markov chain.

Challenge

- Both privacy funnel and information bottleneck problems are non-convex problems:
- The objective function say, $I(S; Y)$ in PF is a convex function of $P_{Y|X}$ (since $P_{Y|S}$ is a linear function of $P_{Y|X}$): *Good*
- However, the constraint region is not convex, e.g., $I(X; Y) \geq \theta$ is not a convex set in the PF method: *Bad*.
- We know that the optimum size of \mathcal{Y} is upper bounded as $|\mathcal{Y}| \leq |\mathcal{X}| + 1$ (see extended [13]).

Prevalent methods in the literature

- Iterative expectation-maximisation (EM) method (unlike the Blahut-Arimoto algorithm, converge to the global optimum is not guaranteed).
- Greedy/heuristic pairwise symbol **merging** algorithms.
- Iterative (general) merging based on the Lagrange method and a **submodularity-based** structure in the problem.
- Converting the privacy constraint to linear local (stronger) constraints based on **local differential privacy** or **local information privacy** \Rightarrow a convex problem. This will be discussed in lecture 2.

Greedy pairwise merging idea/example

- A lower bound θ on $I(X; Y)$ and $P_{S, X}$ are given.
- Start with initial useful symbols. Example:

$$\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$$

- While $I(X; Y) \geq \theta$, **iteratively merge** two symbols in \mathcal{X} that results in the **largest reduction** of $I(S; Y)$ (through exhaustive pairwise checks).
- Example of merged symbols x_1, x_8 at iteration 1:
 $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$, $W^1 = \{x_1, x_8\} \Rightarrow \tilde{\mathcal{X}}^1 = \{\overline{x_1 x_8}, x_2, x_3, x_4, x_5, x_6, x_7, x_9\}$
- Complexity of each iteration $\mathcal{O}(|\mathcal{X}|^2)$.

Numerical example

Let $\theta = 0.7H(X) \approx 2.17$

$P_{S,X}(s, x) =$

0.0394	0.0306	0.0463	0.0463	0.0204	0.0317	0.0328	0.0317	0.0134
0.0438	0.0047	0.0466	0.0235	0.0442	0.0017	0.0366	0.0083	0.0022
0.0061	0.0135	0.0076	0.0387	0.0383	0.0410	0.0359	0.0341	0.0047
0.0441	0.0264	0.0469	0.0069	0.0464	0.0451	0.0190	0.0015	0.0398

Merging means adding up corresponding columns.

$P_{S,\tilde{X}^1}(s, \tilde{x}^1) =$

0.0306	0.0463	0.0463	0.0204	0.0317	0.0328	0.0711	0.0134
0.0047	0.0466	0.0235	0.0442	0.0017	0.0366	0.0521	0.0022
0.0135	0.0076	0.0387	0.0383	0.0410	0.0359	0.0402	0.0047
0.0264	0.0469	0.0069	0.0464	0.0451	0.0190	0.0456	0.0398

This results in $\mathcal{Y} = \tilde{\mathcal{X}}^1$

$$I(S; \tilde{\mathbf{X}}^1) = 0.1637; \quad I(\mathbf{X}; \tilde{\mathbf{X}}^1) = 2.9127.$$

Numerical example

- Mechanism is very simple.
- Whenever, $x = x_1$ OR $x = x_8$, output the **OR** super symbol $\tilde{x} = \overline{x_1 x_8}$.
- E.g., the released data Y will say the customer bought either product 1 or product 8.

$$P_{Y|X}(y|x) = \begin{array}{c|cccccccc} \hline & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & \overline{x_1 x_8} & x_9 \\ \hline x_1 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ x_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ x_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ x_8 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ x_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

Numerical example

- Initial useful symbols

$$\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$$

- At iteration 1, optimal $W^1 = \{x_1, x_8\}$

$$\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\} \Rightarrow \tilde{\mathcal{X}}^1 = \{x_2, x_3, x_4, x_5, x_6, x_7, \overline{x_1 x_8}, x_9\}$$

$$I(S; \tilde{\mathcal{X}}^1) = 0.1637; \quad I(X; \tilde{\mathcal{X}}^1) = 2.9127.$$

- At iteration 2, optimal $W^2 = \{x_3, x_9\}$

$$\tilde{\mathcal{X}}^1 = \{x_2, x_3, x_4, x_5, x_6, x_7, \overline{x_1 x_8}, x_9\} \Rightarrow \tilde{\mathcal{X}}^2 = \{x_2, x_4, x_5, x_6, x_7, \overline{x_1 x_8}, \overline{x_3 x_9}\}$$

$$I(S; \tilde{\mathcal{X}}^2) = 0.1073; \quad I(X; \tilde{\mathcal{X}}^2) = 2.7500.$$

- At iteration 3 optimal $W^3 = \{x_4, x_6\}$

$$\tilde{\mathcal{X}}^2 = \{x_2, x_4, x_5, x_6, x_7, \overline{x_1 x_8}, \overline{x_3 x_9}\} \Rightarrow \tilde{\mathcal{X}}^3 = \{x_2, x_5, \overline{x_4 x_6}, x_7, \overline{x_1 x_8}, \overline{x_3 x_9}\},$$

$$I(S; \tilde{\mathcal{X}}^3) = 0.0516; \quad I(X; \tilde{\mathcal{X}}^3) = 2.4853.$$

- At iteration 4 optimal $W^4 = \{x_2, x_5\}$ (last iteration)

$$\tilde{\mathcal{X}}^3 = \{x_2, x_5, \overline{x_4 x_6}, x_7, \overline{x_1 x_8}, \overline{x_3 x_9}\} \Rightarrow \tilde{\mathcal{X}}^4 = \{\overline{x_2 x_5}, \overline{x_4 x_6}, x_7, \overline{x_1 x_8}, \overline{x_3 x_9}\},$$

$$I(S; \tilde{\mathcal{X}}^4) = 0.0286; \quad I(X; \tilde{\mathcal{X}}^4) = 2.2788.$$

Lagrange function formulation

$$L_{\text{PF}}(P_{Y|X}, \lambda) = I(S; Y) - \lambda I(X; Y).$$

For each λ the optimal solution is a boundary point of the PF problem.

$$\min_{P_{Y|X}} \{I(S; Y) - \lambda I(X; Y)\}.$$

However, this is as complex as the original problem.

Iterative merging-based Lagrange function optimisation

- We start with the original set \mathcal{X} .
- We want to find the best subset W of \mathcal{X} for merging (not necessarily pairwise merging where $|W| = 2$) as follows:

$$W^* = \arg \min \{I(S; \tilde{\mathcal{X}}) - \lambda I(X; \tilde{\mathcal{X}}) : W \subset \mathcal{X}\},$$

where $\tilde{\mathcal{X}} = (\mathcal{X} \setminus W) \cup \tilde{W}$ and \tilde{W} means **merge ALL** symbols in W .

- After finding W^* , update $\mathcal{X} \leftarrow (\mathcal{X} \setminus W^*) \cup \tilde{W}^*$ and start over.
- The question is how to find the best W , where exhaustive search is obviously exponential in $|\mathcal{X}|$ and out of the question.

Merging algorithm based on submodularity properties

Theorem

[2] In each iteration of finding the best merging solution is equivalently described:

$$\begin{aligned} & \arg \min \{ I(S; \tilde{\mathcal{X}}^{(k)}) - \lambda I(X; \tilde{\mathcal{X}}^{(k)}) : W \subset \tilde{\mathcal{X}}^{(k-1)} \} \\ & = \arg \min \{ (1 - \lambda)f(W) - g(W) : W \subset \tilde{\mathcal{X}}^{(k-1)} \} \end{aligned}$$

where f , g are submodular and non-increasing functions defined as

$$f(W) \triangleq \sum_{x \in W} p(x) \log \frac{p(x)}{\underbrace{\sum_{x \in W} p(x)}_{\text{merging symbols in } W}},$$

$$g(W) \triangleq \sum_{s \in S} \sum_{x \in W} p(s, x) \log \frac{p(s, x)}{\underbrace{\sum_{x \in W} p(s, x)}_{\text{merging symbols in } W}}.$$

Optimisation tool

$$\begin{aligned} & \arg \min \{ I(S; \tilde{\mathcal{X}}^{(k)}) - \lambda I(X; \tilde{\mathcal{X}}^{(k)}) : W \subset \tilde{\mathcal{X}}^{(k-1)} \} \\ & = \arg \min \{ (1 - \lambda)f(W) - g(W) : W \subset \tilde{\mathcal{X}}^{(k-1)} \} \end{aligned}$$

For $0 \leq \lambda \leq 1$, the optimisation problem is minimisation of the **difference of two submodular functions (MDSF)** $(1 - \lambda)f(W)$ and $g(W)$.

There are (suboptimal) polynomial-time toolboxes for greedily solving such problems in the ML literature.

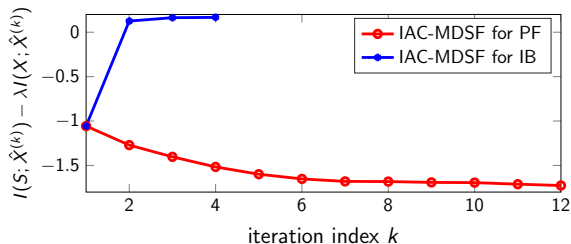
Convergence to global optimal is not guaranteed, but performance can be better than pairwise merging.

Numerical results

- Dataset: Asuncion et al, UCI ML repository.
- Sensitive Variables: {Age, Sex}.
- Useful Variables = {Sex, Cholesterol}.
-

$$\lambda = 0.8$$

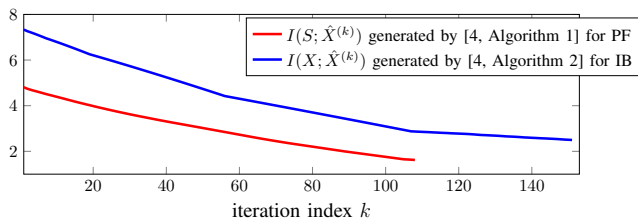
- Convergence performance using the submodularity-based merging.



Numerical results

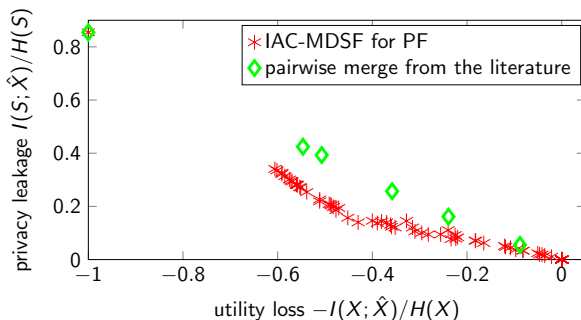
- Dataset: Asuncion et al, UCI ML repository.
- Sensitive Variables: {Age, Sex}.
- Useful Variables = {Sex, Cholesterol}.
-

$$\lambda = 0.8$$



Numerical results

- Dataset: Asuncion et al, UCI ML repository.
- Sensitive Variables: {Age, Sex}.
- Useful Variables = {Sex, Cholesterol}.
- Each point corresponds to a different $\lambda \in [0, 1] \Rightarrow$ PUT boundary point.
- Privacy-utility tradeoff performance using the submodularity-based merging is better.



Key references used for local information privacy, local privacy funnel and privacy watchdog

- [10]: du Pin Calmon and Fawaz 2012
- [15]: Hsu, Asoodeh and du Pin Calmon, 2019
- [3]: Sadeghi, Ding and Rakotoarivelo, 2020
- [16]: Lopuhaa-Zwakenberg, 2020

Motivation for local information privacy measure (log-lift)

- Let us revisit the mutual information between **sensitive variable S** and **released variable Y**

$$I(S; Y) = H(S) + H(Y) - H(S, Y) = H(S) - H(S|Y)$$

$$\begin{aligned} &= \sum_{s \in \mathcal{S}} \sum_{y \in \mathcal{Y}} P_{S, Y}(s, y) \log \frac{P_{S, Y}(s, y)}{P_S(s)P_Y(y)} \\ &= \sum_{s \in \mathcal{S}} \sum_{y \in \mathcal{Y}} P_{S, Y}(s, y) \log \frac{P_{S|Y}(s|y)}{P_S(s)} \\ &= \mathbb{E}_{P_{S, Y}} \left[\log \frac{P_{S|Y}}{P_S} \right]. \end{aligned}$$

- Each term $\frac{P_{S|Y}(s|y)}{P_S(s)}$ captures the adversary's **local gain** in terms of decrease or increase in the likelihood of the realisation $s \in \mathcal{S}$ **upon** the realisation $y \in \mathcal{Y}$.
- Mutual information $I(S; Y)$ is the average of this gain.

Motivation for local information privacy measure (log-lift)

- Instead of managing the average gain $I(S; Y)$, let us control **the worst-case** gain.

$$\begin{aligned} I(S; Y) &= H(S) - H(S|Y) \\ &= \sum_{s \in \mathcal{S}} \sum_{y \in \mathcal{Y}} P_{S,Y}(s, y) \log \frac{P_{S|Y}(s|y)}{P_S(s)} \\ &= \mathbb{E}_{P_{S,Y}} \left[\log \frac{P_{S|Y}}{P_S} \right]. \end{aligned}$$

- So in this lecture, we will mostly focus on the local information privacy (LIP) measure, how it can be used in privacy-utility tradeoff (PUT) optimisation, and how it relates to other measures such as **differential privacy**.

Local information privacy measure (log-lift)

- Define lift and log-lift (or the information density), respectively, as follows:

$$\ell(\mathbf{s}, \mathbf{y}) = \ell(\mathbf{y}, \mathbf{s}) \triangleq \frac{P_{S, Y}(\mathbf{s}, \mathbf{y})}{P_S(\mathbf{s})P_Y(\mathbf{y})} = \frac{P_{S|Y}(\mathbf{s}|\mathbf{y})}{P_S(\mathbf{s})} = \frac{P_{Y|S}(\mathbf{y}|\mathbf{s})}{P_Y(\mathbf{y})}, \quad \mathbf{s} \in \mathcal{S}, \mathbf{y} \in \mathcal{Y},$$

$$i(\mathbf{s}, \mathbf{y}) = i(\mathbf{y}, \mathbf{s}) \triangleq \log \ell(\mathbf{s}, \mathbf{y}) = \log \ell(\mathbf{y}, \mathbf{s}), \quad \mathbf{s} \in \mathcal{S}, \mathbf{y} \in \mathcal{Y},$$

- It is a **symmetric** measure $\ell(\mathbf{s}, \mathbf{y}) = \ell(\mathbf{y}, \mathbf{s})$.
- Recall, mutual information $I(S; Y)$ is the average of information density

$$I(S; Y) = \mathbb{E}_{P_{S, Y}}[i(S, Y)].$$

Example - joint distribution

Recall the joint distribution in our running example. Let us see with no perturbation (that is, when $X = Y$) what sorts of log-lift we get.

$$P_{S,X}(s, x) = \begin{pmatrix} 0.0394 & 0.0306 & 0.0463 & 0.0463 & 0.0204 & 0.0317 & 0.0328 & 0.0317 & 0.0134 \\ 0.0438 & 0.0047 & 0.0466 & 0.0235 & 0.0442 & 0.0017 & 0.0366 & 0.0083 & 0.0022 \\ 0.0061 & 0.0135 & 0.0076 & 0.0387 & 0.0383 & 0.0410 & 0.0359 & 0.0341 & 0.0047 \\ 0.0441 & 0.0264 & 0.0469 & 0.0069 & 0.0464 & 0.0451 & 0.0190 & 0.0015 & 0.0398 \end{pmatrix}$$

Example - log-lift

$$\log \ell(s, x) = \log \frac{P_{S|X}(s|x)}{P_S(s)} =$$

0.009	0.32	0.07	0.31	-0.76	-0.09	-0.10	0.35	-0.27
0.43	-1.21	0.40	-0.03	0.33	-2.68	0.33	-0.65	-1.74
-1.56	-0.20	-1.44	0.42	0.15	0.44	0.27	0.71	-1.03
0.18	0.24	0.14	-1.53	0.11	0.31	-0.59	-2.60	0.87

- Red values highlight the lowest negative and highest positive log-lifts.
- A very negative value $\log \frac{P_{S|X}(s|x)}{P_S(s)}$ means we can (almost) eliminate outcome s after observing x .
- A very large positive value $\log \frac{P_{S|X}(s|x)}{P_S(s)}$ means we can be (almost) certain s has occurred after observing x .

- Local information privacy is point-wise defined as

$$J_{\text{LIP}}^{\text{sym}}(S; Y) = \max_{s \in \mathcal{S}, y \in \mathcal{Y}} \left| \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \right|.$$

- In saying, we will sometimes refer to this as (maximum) absolute log-lift ((max-)abs-log-lift for short).
- Note that

$$J_{\text{LIP}}^{\text{sym}}(S; Y) \leq \epsilon \quad \Leftrightarrow \quad e^{-\epsilon} \leq \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \leq e^{\epsilon}.$$

Example - $J_{LIP}^{\text{sym}}(S; Y)$

$$\log \ell(s, x) = \log \frac{P_{S|Y}(s|y)}{P_S(s)} =$$

$$\begin{pmatrix} 0.009 & 0.32 & 0.07 & 0.31 & -0.76 & -0.09 & -0.10 & 0.35 & -0.27 \\ 0.43 & -1.21 & 0.40 & -0.03 & 0.33 & -2.68 & 0.33 & -0.65 & -1.74 \\ -1.56 & -0.20 & -1.44 & 0.42 & 0.15 & 0.44 & 0.27 & 0.71 & -1.03 \\ 0.18 & 0.24 & 0.14 & -1.53 & 0.11 & 0.31 & -0.59 & -2.60 & 0.87 \end{pmatrix}$$

$$-2.68 \leq \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \leq 0.87$$

$$\Rightarrow J_{LIP}^{\text{sym}}(S; Y) = \max_{s \in \mathcal{S}, y \in \mathcal{Y}} \left| \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \right| = 2.68.$$

Extension: asymmetric local information privacy

Upper LIP

$$J_{\text{LIP}}^{\text{U}}(S; Y) = \max_{s \in \mathcal{S}, y \in \mathcal{Y}} \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right).$$

It is meant to measure largest **gain** in inference (guessing capability).

Lower LIP

$$\begin{aligned} J_{\text{LIP}}^{\text{L}}(S; Y) &= \min_{s \in \mathcal{S}, y \in \mathcal{Y}} \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \\ &= - \max_{s \in \mathcal{S}, y \in \mathcal{Y}} \log \left(\frac{P_S(s)}{P_{S|Y}(s|y)} \right). \end{aligned}$$

This capture the largest **reduction** in the likelihood of s after observation of y .

Relation of Symmetric LIP to Upper and Lower LIP

$$J_{\text{LIP}}^{\text{sym}}(S; Y) = \max\{J_{\text{LIP}}^{\text{U}}(S; Y), |J_{\text{LIP}}^{\text{L}}(S; Y)|\}.$$

Note that for $\epsilon_u, \epsilon_l \geq 0$, we have

$$\begin{cases} J_{\text{LIP}}^{\text{U}}(S; Y) \leq \epsilon_u \\ J_{\text{LIP}}^{\text{L}}(S; Y) \geq -\epsilon_l \end{cases} \Leftrightarrow e^{-\epsilon_l} \leq \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \leq e^{\epsilon_u}$$

Upper local information privacy implies mutual information privacy

Lemma

$J_{LIP}^U(S; Y)$ implies $J_{MI}(S; Y)$. That is

$$J_{LIP}^U(S; Y) \leq \epsilon_u \Rightarrow J_{MI}(S; Y) \leq \epsilon_u.$$

Proof.

Simple: worst-case upper bound implies average upper bound.

$$J_{LIP}^U(S; Y) \leq \epsilon_u \Rightarrow i(s, y) = \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \leq \epsilon_u, \quad s \in \mathcal{S}, y \in \mathcal{Y}$$

$$\Rightarrow I(S; Y) = \sum_{s,y} P_{S,Y}(s, y) \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \leq \epsilon_u.$$



Note that we do not need a bound on the lower LIP in the above to deduce a bound on the mutual information privacy measure.

- Recall the (privacy-centric) privacy funnel problem

$$\begin{aligned} & \max_{P_{Y|X}} I(X; Y) \\ \text{s.t.} \quad & J_{\text{MI}}(S; Y) = I(S; Y) \leq \epsilon. \end{aligned}$$

- The idea of local privacy funnel is to maximise utility $I(X; Y)$ (or equivalently minimise distortion $H(X|Y)$) subject to local information privacy (LIP) constraints, which we now know imply mutual information privacy.

Optimisation

$$\max_{P_{X|Y}, P_Y} I(X; Y)$$

s.t. **normal probability axioms and consistencies**

$$P_Y(y) \geq 0, \quad \forall y \in \mathcal{Y}, \quad P_{X|Y}(x|y) \geq 0, \quad \forall x \in \mathcal{X}, y \in \mathcal{Y},$$

$$\sum_y P_{X|Y}(x|y)P_Y(y) = P_X(x), \quad \forall x \in \mathcal{X},$$

$$\sum_x P_{X|Y}(x|y) = 1, \quad \forall y \in \mathcal{Y},$$

AND also s.t. SYMMETRIC LIP conditions

$$e^{-\epsilon} P_S(s) \leq \underbrace{\sum_x P_{S|X}(s|x)P_{X|Y}(x|y)}_{P_{S|Y}(s|y), \quad S \rightarrow X \rightarrow Y} \leq e^{\epsilon} P_S(s), \quad \forall s \in \mathcal{S}, y \in \mathcal{Y}.$$

Optimisation solution structure

- Let Δ be the convex, closed, bounded, non-negative polytope in $\mathbb{R}^{|\mathcal{X}|}$ with elements denoted by $\mathbf{v} = (v_1, v_2, \dots, v_{|\mathcal{X}|}) \in \Delta$. That is,

$$\Delta = \left\{ \mathbf{v} \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|} : \sum_x v_x = 1, \quad e^{-\epsilon} P(s) \leq \sum_x P(s|x) v_x \leq e^{\epsilon} P(s), \forall s \in \mathcal{S} \right\}.$$

- Note coordinate \mathbf{v} represents $P_{X|Y=y}$ for a fixed y subject to $J_{\text{LIP}}(\mathcal{S}; Y) \leq \epsilon$ conditions.

Optimisation solution structure

- Let $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M\}$ be the **vertices** of the convex polytope Δ , where $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,|\mathcal{X}|})$ is the i -th vertex.
- For each $\mathbf{v}_i \in \mathcal{V}$, its entropy (representing $H(X|Y = y)$ for a fixed y) is:

$$H(X|Y = y) = H(\mathbf{v}_i) = - \sum_{x \in \mathcal{X}} v_{i,x} \log v_{i,x}.$$

Optimisation solution structure

- Note once the vertices \mathcal{V} of Δ are found, $H(\mathbf{v}_i)$ is fixed for all $i \in M$.
- Let $\mathbf{y}^* = (y_1^*, y_2^*, \dots, y_M^*)$ be the solution to the following linear programming problem:

$$\min_{\mathbf{y} \in \mathbb{R}^M} \sum_{i \in [M]} H(\mathbf{v}_i) y_i, \quad (\text{minimise } H(X|Y) \Rightarrow \text{maximise } I(X; Y))$$

$$\text{s.t. } y_i \geq 0, \quad \forall i \in [M],$$

$$\sum_{i \in [M]} v_{i,x} y_i = p_X(x), \quad \forall x \in \mathcal{X}, \quad (\text{consistency with probability } P_X).$$

Theorem

[16]:

- Then the mechanism that maximises $I(X; Y)$ subject to $J_{LIP}(S; Y) \leq \epsilon$ is given by:

$$\mathcal{Y}^* = \{i \in [M] : y_i^* > 0\}, \quad \text{mixing coeff of active vertices}$$

$$P_Y^*(Y = i) = y_i^*, \quad \forall i \in \mathcal{Y}^*,$$

$$P_{X|Y}^*(X = x|Y = i) = v_{i,x}, \quad \forall i \in \mathcal{Y}^* \quad \text{coord. of active vertices of } \Delta.$$

- Furthermore, $|\mathcal{Y}^*| \leq |\mathcal{X}|$.

Remaining drawbacks

- The worst-case complexity of finding all vertices of convex $\epsilon - J_{\text{LIP}}$ region Δ is **exponential** with $|\mathcal{X}|$.
- The final optimal solution is the solution to a linear program and one cannot draw much insight from it.

Idea

- Takes a targeted approach to first categorise symbols $x \in \mathcal{X}$ based on their worst-case local information (LIP) value

$$\max_{s \in \mathcal{S}} |\log \ell(s, x)| = \max_{s \in \mathcal{S}} \left| \log \frac{P(s|x)}{P(s)} \right|.$$

- And then sanitise those elements in \mathcal{X} whose $\max_{s \in \mathcal{S}} |\log \ell(s, x)|$ is larger than a desired ϵ **threshold**.
- The privacy watchdog operation is easy to understand and also **easy to implement** with excellent privacy guarantees as we show next.

Example - joint distribution

$P_{S,X}(s, x) =$

0.0394	0.0306	0.0463	0.0463	0.0204	0.0317	0.0328	0.0317	0.0134
0.0438	0.0047	0.0466	0.0235	0.0442	0.0017	0.0366	0.0083	0.0022
0.0061	0.0135	0.0076	0.0387	0.0383	0.0410	0.0359	0.0341	0.0047
0.0441	0.0264	0.0469	0.0069	0.0464	0.0451	0.0190	0.0015	0.0398

Example - log-lift

$\log \ell(s, x) =$

0.009	0.32	0.07	0.31	-0.76	-0.09	-0.10	0.35	-0.27
0.43	-1.21	0.40	-0.03	0.33	-2.68	0.33	-0.65	-1.74
-1.56	-0.20	-1.44	0.42	0.15	0.44	0.27	0.71	-1.03
0.18	0.24	0.14	-1.53	0.11	0.31	-0.59	-2.60	0.87

where red values highlight the lowest negative and highest positive log-lifts.

Example - absolute log-lift

- As an example, let us set $\epsilon = 1$ as our symmetric local information privacy target (for abs-log-lift).

$$|\log \ell(s, x)| = |i(s, x)| =$$

0.0093	0.3298	0.0710	0.3169	0.7614	0.0983	0.1024	0.3598	0.2723
0.4383	1.2163	0.4017	0.0391	0.3369	2.6855	0.3308	0.6598	1.7406
1.5647	0.2054	1.4484	0.4227	0.1537	0.4449	0.2731	0.7189	1.0352
0.1809	0.2419	0.1418	1.5348	0.1178	0.3127	0.5935	2.6078	0.8747

where red values highlight (s, x) pairs whose abs-log-lift $> \epsilon = 1$.

$$\mathcal{X}_\epsilon = \{x \in \mathcal{X} : |i(s, x)| \leq \epsilon, \quad \forall s \in \mathcal{S}\} \quad \text{low-risk symbols}$$

$$\mathcal{X}_{\epsilon^c} = \mathcal{X} \setminus \mathcal{X}_\epsilon \quad \text{high risk symbols.}$$

The general mechanism structure

$$P(y|x) = \begin{cases} 1_{\{x=y\}} & x, y \in \mathcal{X}_\epsilon, & \text{(no sanitisation required)} \\ R(y|x) & x, y \in \mathcal{X}_\epsilon^c, & \text{(sanitisation required)} \\ 0 & \text{otherwise.} \end{cases}$$

where

$$\sum_{y \in \mathcal{X}_\epsilon^c} R(y|x) = 1, \quad x \in \mathcal{X}_\epsilon^c$$

Question: What is the optimal $R^*(y|x)$?

We will focus on bringing the abs-log-lift down to lowest possible value and see what happens to utility: **privacy-centric approach**.

The general mechanism properties

$$i(s, y) \triangleq \log \left(\frac{P(s, y)}{P(s)P(y)} \right) = \log \left(\frac{P(y|s)}{P(y)} \right) = \log \left(\frac{P(s|y)}{P(s)} \right)$$

Recall $S \rightarrow X \rightarrow Y$:

$$i(s, y) = \log \left(\frac{P(y|s)}{P(y)} \right) = \log \left(\frac{\sum_{x \in \mathcal{X}_\epsilon^c} P(y|x)P(x|s)}{\sum_{x \in \mathcal{X}_\epsilon^c} P(y|x)P(x)} \right)$$

An X -invariant mechanism structure

An X -invariant mechanism $R_{Y|X}(y|x) = R(y)$ for all $x, y \in \mathcal{X}_\epsilon^c$.

$$\begin{aligned} i(s, y) &= \log \left(\frac{P(y|s)}{P(y)} \right) = \log \left(\frac{\sum_{x \in \mathcal{X}_\epsilon^c} P(y|x)P(x|s)}{\sum_{x \in \mathcal{X}_\epsilon^c} P(y|x)P(x)} \right) \\ &= \log \left(\frac{\sum_{x \in \mathcal{X}_\epsilon^c} R(y)P(x|s)}{\sum_{x \in \mathcal{X}_\epsilon^c} R(y)P(x)} \right) \\ &= \log \frac{\sum_{x \in \mathcal{X}_{\epsilon^c}} P(x|s)}{\sum_{x \in \mathcal{X}_{\epsilon^c}} P(x)}. \end{aligned}$$

Optimal privacy watchdog mechanism

Merging **high-risk symbols** is an example of X -invariant mechanism

$\mathcal{X}_\epsilon = \{x \in \mathcal{X} : |i(s, x)| \leq \epsilon, \forall s \in \mathcal{S}\}$ low-risk symbols released w/o perturbation

$\mathcal{X}_{\epsilon^c} = \mathcal{X} \setminus \mathcal{X}_\epsilon$ high risk symbols are all merged into y^* .

$$p(y|x) = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ \vdots & & & & \vdots & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{pmatrix} \begin{matrix} y^* \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}$$

$$P_{Y|X}(y|x) = R(y) = \mathbf{1}, \quad y = y^*, x \in \mathcal{X}_{\epsilon^c}$$

We say that the randomisation mechanism $p(y|x)$ attains $(\epsilon', \mathcal{X}_\epsilon^c)$ -log-lift if

$$|i(s, y)| = \left| \log \frac{p(y|s)}{p(y)} \right| \leq \epsilon', \quad \forall y \in \mathcal{X}_\epsilon^c, s \in \mathcal{S}.$$

The question is what is the best ϵ' for a given \mathcal{X}_ϵ^c .

Theorem

[3]:

For a given value of ϵ (subsequently determining \mathcal{X}_ϵ and \mathcal{X}_{ϵ^c}), the *minimum* value of ϵ' such that $(\epsilon', \mathcal{X}_\epsilon^c)$ -log-lift is attainable is given by

$$\epsilon^c \doteq \epsilon(\mathcal{X}_\epsilon^c) = \max_{s \in \mathcal{S}} \left| \log \frac{\sum_{x \in \mathcal{X}_{\epsilon^c}} p(x|s)}{\sum_{x \in \mathcal{X}_{\epsilon^c}} p(x)} \right|,$$

which is achieved by *any valid X-invariant mechanism* $R_{Y|X}(y|x) = R(y)$, for all $x, y \in \mathcal{X}_{\epsilon^c}$, $\sum_{y \in \mathcal{X}_{\epsilon^c}} R(y) = 1$.

Optimal privacy watchdog mechanism

Example

$$P_{S,X}(s, x) = \begin{pmatrix} & & \text{merged } \mathcal{X}_\epsilon^c \\ 0.0204 & 0.0328 & 0.2392 \\ 0.0442 & 0.0366 & 0.1308 \\ 0.0383 & 0.0359 & 0.1457 \\ 0.0464 & 0.0190 & 0.2108 \end{pmatrix}$$

The resulting abs -log-lift is much smaller now:

$$|i(s, x)| = |\log \ell(s, x)| = \begin{pmatrix} 0.7614 & 0.10248 & 0.1188 \\ 0.3369 & 0.3308 & \mathbf{0.1618} \\ 0.1537 & 0.2731 & 0.0920 \\ 0.1178 & 0.5935 & 0.0496 \end{pmatrix}$$

A utility boosting method

- Crudely applying a merging solution to all \mathcal{X}_ϵ^c may hurt utility a lot.
- One solution:
- Move elements from $x' \in \mathcal{X}_\epsilon^c$ back to \mathcal{X}_ϵ as long as probability of violating $\ell(s, x) > \epsilon$ remains below a suitably chosen small slack threshold δ .

Example

$$|\log \ell(s, x)| = |i(s, x)| =$$

0.0093	0.3298	0.0710	0.3169	0.7614	0.0983	0.1024	0.3598	0.2723
0.4383	1.2163	0.4017	0.0391	0.3369	2.6855	0.3308	0.6598	1.7406
1.5647	0.2054	1.4484	0.4227	0.1537	0.4449	0.2731	0.7189	1.0352
0.1809	0.2419	0.1418	1.5348	0.1178	0.3127	0.5935	2.6078	0.8747

For **red elements**, the corresponding probability of pair (s, x) occurring is **really small**. They correspond to a large reduction in likelihood of s ($i(s, x) \ll 0$). Indeed, these correspond to small negative log-lifts (with large abs value):

$$P_{S,X}(s, x) =$$

0.0394	0.0306	0.0463	0.0463	0.0204	0.0317	0.0328	0.0317	0.0134
0.0438	0.0047	0.0466	0.0235	0.0442	0.0017	0.0366	0.0083	0.0022
0.0061	0.0135	0.0076	0.0387	0.0383	0.0410	0.0359	0.0341	0.0047
0.0441	0.0264	0.0469	0.0069	0.0464	0.0451	0.0190	0.0015	0.0398

Example

$$|\log \ell(s, x)| = |i(s, x)| =$$

0.0093	0.3298	0.0710	0.3169	0.7614	0.0983	0.1024	0.3598	0.2723
0.4383	1.2163	0.4017	0.0391	0.3369	2.6855	0.3308	0.6598	1.7406
1.5647	0.2054	1.4484	0.4227	0.1537	0.4449	0.2731	0.7189	1.0352
0.1809	0.2419	0.1418	1.5348	0.1178	0.3127	0.5935	2.6078	0.8747

$$P_{S,X}(s, x) =$$

0.0394	0.0306	0.0463	0.0463	0.0204	0.0317	0.0328	0.0317	0.0134
0.0438	0.0047	0.0466	0.0235	0.0442	0.0017	0.0366	0.0083	0.0022
0.0061	0.0135	0.0076	0.0387	0.0383	0.0410	0.0359	0.0341	0.0047
0.0441	0.0264	0.0469	0.0069	0.0464	0.0451	0.0190	0.0015	0.0398

- Every column that has a **blue element** now belongs back to \mathcal{X}_ϵ : **low risk symbols**.
- Overall, the probability of $\ell(s, x) > \epsilon$ is $\delta = 0.0015 + 0.0017 + 0.0047 < 0.01$

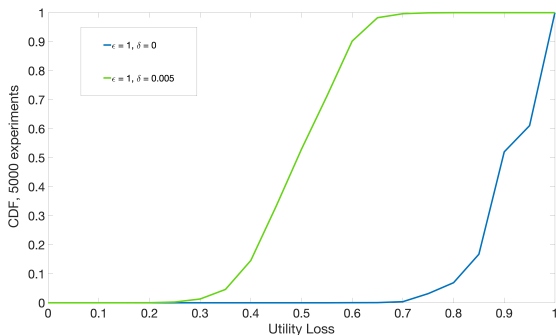
Numerical results

The simulation is the result from 5000 randomly generated $P_{S,X}$. It shows the distribution of the utility loss defined as

$$1 - \frac{I(X; Y)}{H(X)}.$$

The more to the left a curve is, the better.

It is clear that much **lower utility loss** is achievable with a small relaxation of absolute log-lift $\ell(s, x)$ going above ϵ with probability $\delta = 0.005$.



References for differential privacy section and its relation to information-theoretic measures

- [17]: Dwork, McSherry, Nissim and Smith, 2006
- [18]: Kasiviswanathan, Lee, Nissim, Raskhodnikova and Smith, 2011
- [10]: du Pin Calmon and Fawaz 2012
- [4]: D'Oliveira, Sadeghi, Médard, 2021 (and references therein)

Differential privacy

A randomised mechanism $M : \mathcal{X}^n \rightarrow \mathcal{Y}$ is said to satisfy (ϵ, δ) -differential privacy or (ϵ, δ) -DP for short, if for all **neighboring** $x, x' \in \mathcal{X}^n$ **differing on a single element** and all events $E \subset \mathcal{Y}$, we have

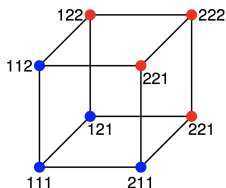
$$P[M(x) \in E] \leq e^\epsilon P[M(x') \in E] + \delta.$$

Pure differential privacy

Pure differential privacy occurs when $\delta = 0$. In pure DP, the condition for privacy can be written as:

$$e^{-\epsilon} P[M(x') \in E] \leq P[M(x) \in E] \leq e^\epsilon P[M(x') \in E], \quad \forall x \sim x', E \subset \mathcal{Y}.$$

Example



A subset of pure ϵ -DP constraints + normal probability constraints:

For any neighbouring x, x' :

$$e^{-\epsilon} P[M(x') = R] \leq P[M(x) = R] \leq e^{\epsilon} P[M(x') = R],$$

$$e^{-\epsilon} P[M(x') = B] \leq P[M(x) = B] \leq e^{\epsilon} P[M(x') = B],$$

$$P[M(x) = R] + P[M(x) = B] = 1,$$

$$P[M(x') = R] + P[M(x') = B] = 1,$$

Roughly speaking, probability of same response for neighbouring datasets (with same or different true colors) must be **more or less** the same.

- Google, for sharing historical traffic statistics [19](Erlingsson et al. '14).
- Apple's private learning of users' preferences [20](Apple DP team '17).
- Microsoft for telemetry in Windows [21](Microsoft, Ding et al. '17).
- The 2020 United States Census [22].
- Federated learning [23](Edited by Kairouz and McMahan, 2021);

Definition

For a dataset X^n consisting of n elements, a randomised mechanism $M : \mathcal{X}^n \rightarrow \mathcal{Y}$ is said to satisfy ϵ -differential privacy or ϵ -LDP for short, **if for all** $x, x' \in \mathcal{X}^n$ and all events $E \subset \mathcal{Y}$, we have

$$e^{-\epsilon} P[M(x') \in E] \leq P[M(x) \in E] \leq e^{\epsilon} P[M(x') \in E].$$

Local DP is a **much stronger** notion than standard DP because it has to be satisfied for **all datasets**, not just the neighbouring ones.

Relation between differential privacy and information-theoretic privacy

- Differential privacy is intentionally designed to be free of statistical models (prior distribution) for the underlying data.
- It does not explicitly distinguish “sensitive data” (to protect) from “useful data” (to share).
- An effort in DP is made to separate unavoidable statistical inference (using statistical side information) from individual or group disclosure.
- It may therefore appear that differential privacy and information-theoretic privacy measures cannot be compared directly.
- **This is not the case.** However, we do require a common interpretation or notion of what constitutes as sensitive data, S and what constitute as useful data, X .
- At least two approaches have been used in the literature.
- We briefly discuss one of these approaches.

First relational approach

- Variable $S = (S_1, S_2, \dots, S_n)$ is used to represent the dataset variable which is considered as private, where each discrete entry $S_i \in S$ belongs to an individual.
- Variable X is used to represent the output of the query function on the dataset $q : S^n \rightarrow \mathcal{X}$.
- Variable Y is used to represent the mechanism output from input X , which is probabilistically mapped from X into Y according to $P_{Y|X}$, with the goal to hide individual entries S_i from the adversary.

Example

Let us assume $\mathcal{S} = \{0, 1\}^n$ represents the truthful response from n individuals about whether or not they have a certain disease D . The counting query asks $X = q(\mathcal{S}) = \sum_{i=1}^n \mathbb{1}_D(S_i)$, where

$$\mathbb{1}_D(z) = \begin{cases} 1, & \text{If } z \text{ has property } D, \\ 0, & \text{otherwise.} \end{cases}$$

We say $s \sim s'$ if they differ in a single binary value, leading to the corresponding query values $|x - x'| = 1$.

DP with sensitive variable S

The mechanism is said to be ϵ -differentially private if

$$e^{-\epsilon} P_{Y|S}(y|s') \leq P_{Y|S}(y|s) \leq e^{\epsilon} P_{Y|S}(y|s'), \quad s, s' \in \mathcal{S} : s \sim s', \forall y \in \mathcal{Y}$$

where $s \sim s'$ denotes any two datasets that are neighbors differing in a single sensitive coordinate (Hamming distance of 1).

LDP with sensitive variable S

The mechanism is said to be ϵ -**locally** differentially private if

$$e^{-\epsilon} P_{Y|S}(y|s') \leq P_{Y|S}(y|s) \leq e^{\epsilon} P_{Y|S}(y|s'), \quad \forall s, s' \in \mathcal{S}, \forall y \in \mathcal{Y}$$

This removes the requirement on s, s' being neighbors, making it a **stronger** notion of differential privacy.

Upper and lower local information privacy imply local differential privacy

Lemma

$J_{LIP}^U(S; Y)$ and $J_{LIP}^L(S; Y)$ **together** imply local differential privacy. That is

$$J_{LIP}^U(S; Y) \leq \epsilon_u \quad \& \quad J_{LIP}^L(S; Y) \geq -\epsilon_l \quad \Rightarrow \quad (\epsilon_u + \epsilon_l) - LDP.$$

Proof.

First, note from that

$$\begin{aligned} J_{LIP}^L(S; Y) &= \min_{s \in \mathcal{S}, y \in \mathcal{Y}} \log \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right) \geq -\epsilon_l \\ &\Rightarrow \max_{s \in \mathcal{S}, y \in \mathcal{Y}} \log \left(\frac{P_S(s)}{P_{S|Y}(s|y)} \right) \leq \epsilon_l. \end{aligned}$$



Upper and lower local information privacy imply local differential privacy

Proof.

Recalling the definition of local differential privacy in the context of Markov chain $S \rightarrow X \rightarrow Y$:

$$\begin{aligned} & \sup_{y \in \mathcal{Y}, s, s'} \frac{P_{Y|S}(y|s)}{P_{Y|S}(y|s')} \\ &= \sup_{y \in \mathcal{Y}, s, s'} \frac{P_{S|Y}(s|y)P_Y(y)}{P_S(s)} \frac{P_S(s')}{P_{S|Y}(s'|y)P_Y(y)} \\ &\leq e^{\epsilon_U} e^{\epsilon_I} = e^{\epsilon_U + \epsilon_I}. \end{aligned}$$



Lemma

Local differential privacy implies differential privacy. That is (ϵ, δ) -LDP \Rightarrow (ϵ, δ) -DP.

Follows immediately from more strict definition of local differential privacy.

Lemma

*Pure local differential privacy implies upper and lower local information privacy.
That is*

$$\epsilon - \text{LDP} \Rightarrow J_{LIP}^U(S; Y) \leq \epsilon, \quad J_{LIP}^L(S; Y) \geq -\epsilon.$$

Pure local differential privacy implies upper and lower LIP

Proof.

Recall the definition of $(\epsilon, 0)$ -LDP in context of information theory measures:

$$e^{-\epsilon} P_{Y|S}(y|s') \leq P_{Y|S}(y|s) \leq e^{\epsilon} P_{Y|S}(y|s'), \quad \forall y \in \mathcal{Y}, s, s' \in \mathcal{S}$$

$$\begin{aligned} \frac{P_{S|Y}(s|y)}{P_S(s)} &= \frac{P_{Y|S}(y|s)}{P_Y(y)} = \frac{P_{Y|S}(y|s)}{\sum_{s' \in \mathcal{S}} P_{Y|S}(y|s') P_S(s')} \\ &\leq \frac{P_{Y|S}(y|s)}{\sum_{s' \in \mathcal{S}} P_{Y|S}(y|s') e^{-\epsilon} P_S(s')} \leq e^{\epsilon} \end{aligned}$$

Similarly,

$$\begin{aligned} \frac{P_{S|Y}(s'|y)}{P_S(s')} &= \frac{P_{Y|S}(y|s')}{P_Y(y)} = \frac{P_{Y|S}(y|s')}{\sum_{s \in \mathcal{S}} P_{Y|S}(y|s) P_S(s)} \\ &\geq \frac{P_{Y|S}(y|s')}{\sum_{s \in \mathcal{S}} P_{Y|S}(y|s') e^{\epsilon} P_S(s)} \geq e^{-\epsilon} \end{aligned}$$

□

Lemma

For every $\epsilon \geq 0$ and $\delta \geq 0$, there exists a (S, Y) vector such that the mechanism $P_{Y|S}$ satisfies $(\epsilon, 0)$ -DP but, $I(S; Y) \geq \delta$.

The following counting query counterexample is from [10].

Differential privacy does NOT imply mutual information privacy

Proof

Let $\mathcal{S} = \{0, 1\}^n$ and the counting query asks $X = q(\mathcal{S}) = \sum_{i=1}^n \mathbb{1}_D(S_i)$. Let $n \bmod k = 0$ and P_S be such that

$$P_X(x) = \begin{cases} \frac{1}{1+\frac{n}{k}}, & \text{if } x \bmod k = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Differential privacy does NOT imply mutual information privacy

Proof

Let the DP mechanism be the standard Laplace mechanism:

$$Y = X + N, \quad N \sim \text{Lap}(1/\epsilon),$$

which is guaranteed to satisfy $J_{\text{DP}}(X; Y) \leq \epsilon$.

The probability of correctly guessing X from Y according to Laplace noise distribution is then:

$$P(X = Y | X = k) = \int_{-k/2}^{k/2} \frac{\epsilon}{2} \exp(-|x|\epsilon) dx = 1 - \exp\left(-\frac{k\epsilon}{2}\right).$$

$$\begin{aligned} I(X; Y) &\geq I(X, E; Y) - 1 \geq I(X; Y|E) - 1 \geq P(E = 0)I(X; Y|E = 0) - 1 \\ &= \left(1 - \exp\left(-\frac{k\epsilon}{2}\right)\right) \log\left(1 + \frac{n}{k}\right) - 1, \end{aligned}$$

where k and n can be chosen appropriately to make $I(X; Y)$ arbitrarily large.

Graph-based optimisation of DP, setting [4]

- Family of datasets \mathcal{D} .
- A symmetric relationship in \mathcal{D} where $d \sim d'$ are said to be neighbors.³
- An output space \mathcal{V} . We consider binary $\mathcal{V} = \{1, 2\}$.
- True function $f : \mathcal{D} \rightarrow \mathcal{V}$.
- Random function $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{V}$ called random mechanism.

Definition: Differential Privacy for binary values

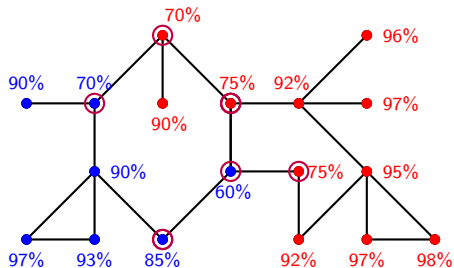
\mathcal{M} is (ϵ, δ) -differentially private if, for any $d \sim d'$ and $v \in \mathcal{V}$,

$$\Pr[\mathcal{M}(d) = v] \leq e^\epsilon \Pr[\mathcal{M}(d') = v] + \delta$$

Goal: Approximate the true function f by an (ϵ, δ) -DP mechanism \mathcal{M} .

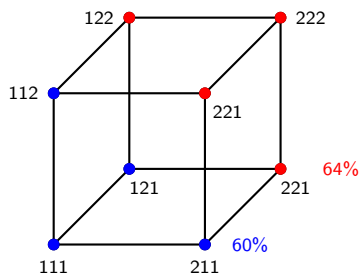
³For the rest of this lecture, we use the notation d, d' to represent datasets.

Main contributions



- We introduce a graph-theoretical framework for differential privacy, where:
 - Vertices represent datasets.
 - Edges connect neighbouring datasets.
 - Colours represent the output of true function.
 - A mechanism is a randomised colouring.
- We characterise the optimal mechanism in terms of its values at the boundary.
- We present a closed form for the optimal mechanism when the values at the boundary satisfy a homogeneity condition.

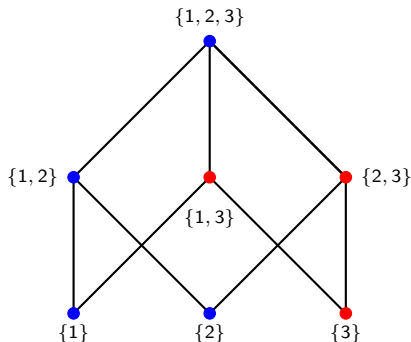
Differential privacy as a randomised graph colouring



- A dataset with three voters $\mathcal{D} = \{1, 2\}^3$.
- The outputs are the candidates $\mathcal{V} = \{1, 2\}$.
- The true function f is the majority function.
- Vote is private.
- A mechanism \mathcal{M} is a random coloring.

• DP implies that neighboring datasets behave almost the same under \mathcal{M} .
 $(\epsilon, \delta) = (0.5, 0)$: if $\Pr[\mathcal{M}(211) = 1] = 0.6$, then $\Pr[\mathcal{M}(221) = 1] \geq 0.36$ and $\Pr[\mathcal{M}(221) = 2] \leq 0.64$.

Differential privacy as a randomised graph colouring



- A dataset of who voted $\mathcal{D} = \mathcal{P}(\{1, 2, 3\}) - \{\emptyset\}$.
 - The outputs are the candidates $\mathcal{V} = \{1, 2\}$.
 - The true function f is the majority function.
 - 1 and 2 vote blue, 3 votes red. Ties are resolved in favour of red.
 - Identity of voter is private.
 - A mechanism \mathcal{M} is a random coloring.
- DP implies that neighboring datasets behave almost the same under \mathcal{M} .

How well does the mechanism approximate the true function?

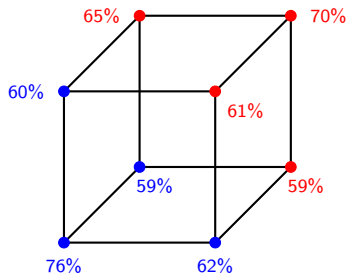
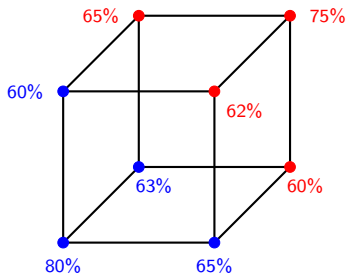
- Let \mathfrak{M} be the set of all mechanisms.
- Utility function $U : \mathfrak{M} \rightarrow \mathbb{R}$.
- $U[\mathcal{M}] > U[\mathcal{M}']$ means \mathcal{M} is better than \mathcal{M}' .

Definition: reasonable utility function

The utility U is reasonable if, for every $d \in \mathcal{D}$,
 $\Pr[\mathcal{M}(d) = f(d)] \geq \Pr[\mathcal{M}'(d) = f(d)]$ implies $U[\mathcal{M}] \geq U[\mathcal{M}']$.

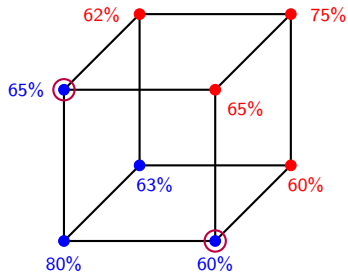
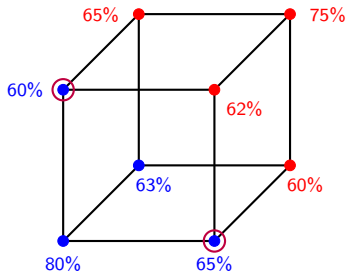
- If $\Pr[\mathcal{M}(d) = f(d)] \geq \Pr[\mathcal{M}'(d) = f(d)]$ for every $d \in \mathcal{D}$, we say that the mechanism \mathcal{M} dominates \mathcal{M}' .
- We say a mechanism is optimal if it dominates every mechanism it is comparable to.

Example: mechanism domination



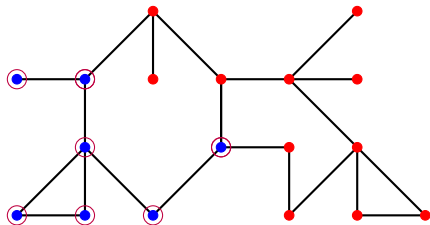
- The mechanism on the left dominates the one on the right.

Example: mechanism domination



- Each mechanism performs better on at least one dataset compared to the other mechanism.
- Neither mechanism dominates the other.

Optimal mechanisms and boundaries

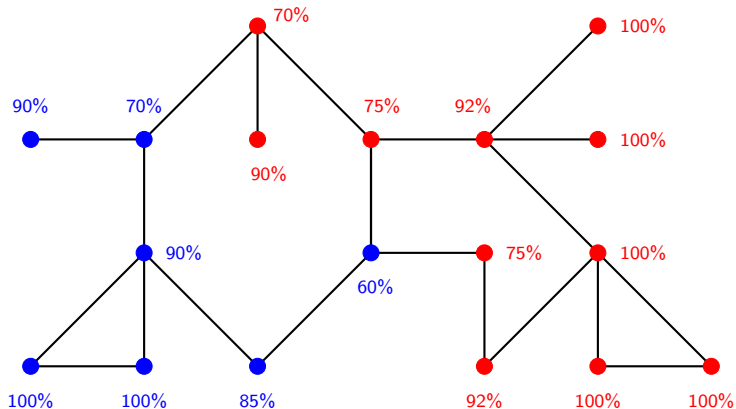


- **Blue set:** $B = \{d \in \mathcal{D} : f(d) = 1\}$.
- **Interior:**
 $B^\circ = \{d \in B : d \sim d' \Rightarrow d' \in B\}$.
- **Boundary:** $\partial B = B - B^\circ$.
- Analogous concepts for Red.

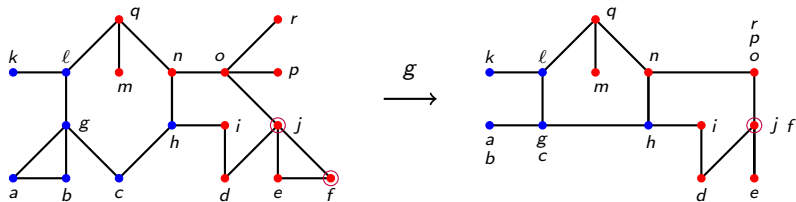
Theorem: optimal mechanism

Let $m_d \in [0, 1]$ be a fixed value for every $d \in \partial B$. Then, there exists at most one optimal (ϵ, δ) -DP mechanism \mathcal{M} such that $\Pr[\mathcal{M}(d) = 1] = m_d$, for every $d \in \partial B$.

Example: $(\epsilon, \delta) = (\log(2), 0.1)$



Morphisms



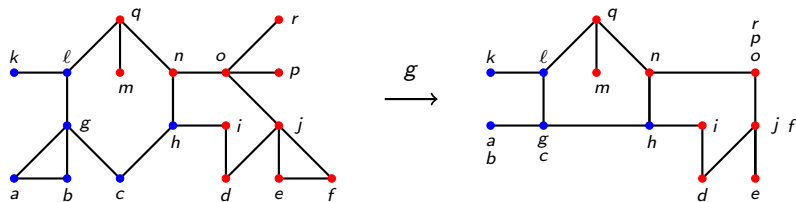
Definition: Morphisms

A morphism is a function $g : \mathcal{D}_1 \rightarrow \mathcal{D}_2$ such that $d \stackrel{1}{\sim} d'$ implies in either $g(d) \stackrel{2}{\sim} g(d')$ or $g(d) = g(d')$.

Example: on the left, j, f are neighbours. On the right, they collapse to the same dataset $g(d) = g(d')$.

Example: on the left, o, j are neighbours. On the right, they remain neighbours $g(d) \stackrel{2}{\sim} g(d')$.

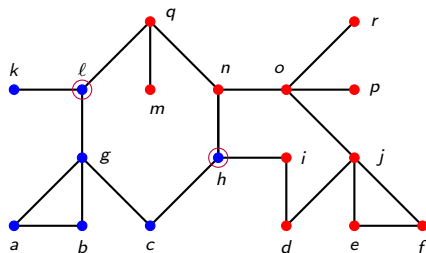
Morphisms



Theorem: DP via pullbacks

Let \mathcal{M}_2 be an (ϵ, δ) -DP mechanism on \mathcal{D}_2 . Then, $\mathcal{M}_1 = \mathcal{M}_2 \circ g$ is (ϵ, δ) -DP on \mathcal{D}_1 .

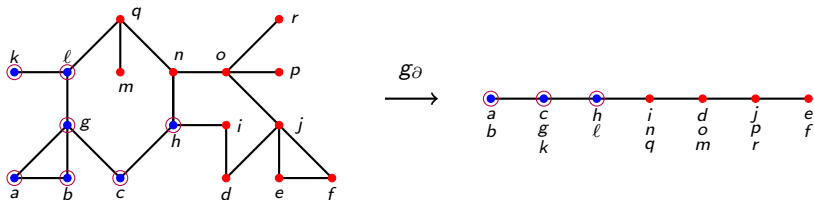
Boundary homogeneity



Definition: boundary homogeneous mechanisms

A mechanism \mathcal{M} is boundary homogeneous if the probabilities at the boundary are the same.

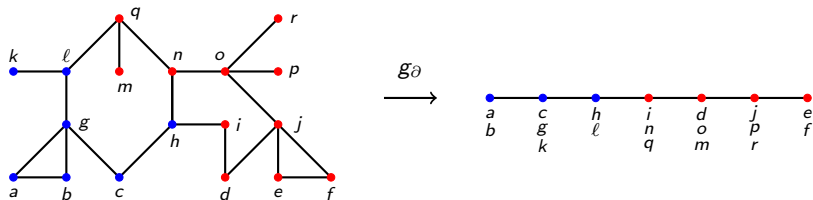
The boundary morphism



Definition: boundary morphism

The boundary morphism g_{∂} maps \mathcal{D} to a line which preserves the distance from any vertex to the boundary.

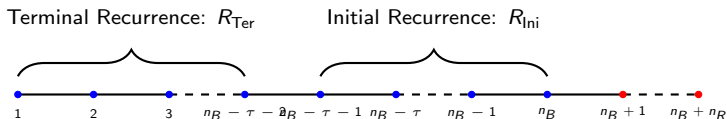
Optimal mechanisms



Theorem: optimal boundary homogeneous mechanisms

Let \mathcal{M}_{∂} be the optimal (ϵ, δ) -DP mechanism on the boundary graph of \mathcal{D} . Then, the pullback $\mathcal{M} = \mathcal{M}_{\partial} \circ g_{\partial}$ is the optimal boundary homogeneous (ϵ, δ) -DP mechanism on \mathcal{D} .

Optimal mechanism on the (n_B, n_R) -line



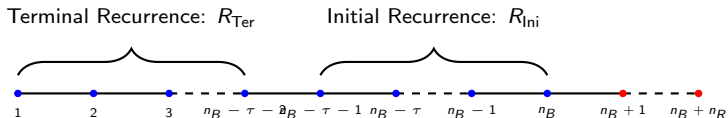
Definition: initial recurrence

$$R_{Ini}(n, i) = 1 - e^{\epsilon i} (1 - R_n) - \frac{\delta(e^{i\epsilon} - 1)}{e^{\epsilon} - 1}$$

Definition: terminal recurrence

$$R_{Ter}(n, i) = \frac{R_n}{e^{\epsilon i}} - \frac{\delta(e^{\epsilon i} - 1)}{e^{\epsilon i}(e^{\epsilon} - 1)}$$

Optimal mechanism on the (n_B, n_R) -line



Theorem: optimal mechanism on a line

The unique optimal (ϵ, δ) -DP mechanism on the (n_B, n_R) -line with $\Pr[\mathcal{M}(n_B) = 2] = R_{n_B}$ is such that

$$R_{n_B - i} = \begin{cases} R_{Ini}(n_B, i) & \text{if } i \leq \tau + 1, \\ R_{Ter}(n_B - \tau - 1, i - \tau - 1) & \text{if } \tau + 1 < i, \end{cases}$$

for every $i \in [1, n_B - 1]$ and where τ is defined as

$$\tau = \left\lceil \frac{1}{\epsilon} \log \left(\frac{e^\epsilon + 2\delta - 1}{(1 - R_{n_B})(e^{3\epsilon} - e^\epsilon) + \delta(e^{2\epsilon} + e^\epsilon)} \right) \right\rceil,$$

Definition: initial recurrence

A mechanism \mathcal{M} is balanced if $\Pr[\mathcal{M}(d) = 1] = \Pr[\mathcal{M}(d') = 2]$ for every $d \in \partial B$ and $d' \in \partial R$.

Corollary: optimal balanced mechanism

The optimal balanced (ϵ, δ) -DP mechanism is such that, for every $d \in B^o$,

$$\Pr[\mathcal{M}(d) = 2] = \frac{e^\epsilon - 1 - \delta(e^{\epsilon(\text{dist}(d, \partial B)+1)} + e^{\epsilon \text{dist}(d, \partial B)} - 2)}{e^{\epsilon \text{dist}(d, \partial B)}(e^\epsilon + 1)(e^\epsilon - 1)}.$$

- Extension to non-homogenous boundary conditions (how to propagate the probabilities down from different boundary values) - almost done.
- Extension to non-binary functions (more colours) - in train.
- Contextualising other mechanisms such as exponential and Laplace on the dataset graph and comparisons - in train.

Extra slides on model section and privacy measures

Model - variables with noisy observation variable W [12]

- $s \in \mathcal{S}$: sensitive information to protect.
- $x \in \mathcal{X}$: useful information to share.
- $w \in \mathcal{W}$: directly observable data, which may be a noisy version of variables \mathcal{S} and/or \mathcal{X} .
- Target application imposes the specific **statistical** data model:

$$P_{S,X}$$

- The observation constraint is:

$$P_{W|S,X}$$

- $y \in \mathcal{Y}$: released variable based on W .

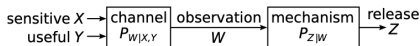


Figure from [12].

- Markov model

$$(S, X) \rightarrow W \rightarrow Y$$

- The **mechanism** is specified by the conditional distribution $P_{Y|W}$.
- Y should provide **utility** about X while protecting **privacy** by limiting the information it reveals about S .

Privacy-utility trade-off (PUT)

- We say a particular privacy-utility pair (ϵ, ρ) is **achievable** if there exists a mechanism $P_{Y|W}$ such that we **simultaneously** have:

$$J(S; Y) \leq \epsilon,$$

and

$$D(P_{X,Y}) \leq \rho.$$

- The set of all achievable privacy-utility tradeoffs forms the **achievable region** of privacy-utility tradeoff.

- The optimal region **boundary** is the solution to the optimisation problem:

$$\begin{aligned}\epsilon^*(\rho) &= \inf_{P_{Y|W}} J(S; Y), \\ \text{s.t. } & D(P_{X,Y}) \leq \rho.\end{aligned}$$

- We can alternatively write the **privacy-centric** optimal boundary:

$$\begin{aligned}\rho^*(\epsilon) &= \inf_{P_{Y|W}} D(P_{X,Y}), \\ \text{s.t. } & J(S; Y) \leq \epsilon.\end{aligned}$$

$$\begin{aligned} P_{Y|X}(y|x) &= \sum_{w \in \mathcal{W}, s \in \mathcal{S}} P_{Y,w,s|X}(y, w, s|x) \\ &= \sum_{w \in \mathcal{W}, s \in \mathcal{S}} P_{S|X}(s|x) P_{W|S,X}(w|s, x) P_{Y|W}(y|w), \end{aligned}$$

$$\begin{aligned} P_{Y|S}(y|s) &= \sum_{w \in \mathcal{W}, x \in \mathcal{X}} P_{Y,w,x|S}(y, w, x|s) \\ &= \sum_{w \in \mathcal{W}, x \in \mathcal{X}} P_{X|S}(x|s) P_{W|S,X}(w|s, x) P_{Y|W}(y|w). \end{aligned}$$

- $P_{S,X}$ is general, but

$$W = (S, X),$$

capturing the situation when the mechanism has **direct noiseless access** to both sensitive and useful information.

- For this case, the privacy-utility optimisation problems reduce to

$$\begin{aligned} \epsilon_{\text{FD}}^*(\rho) &= \inf_{P_{Y|S,X}} J(S; Y), \\ \text{s.t. } D(P_{X,Y}) &\leq \rho. \end{aligned}$$

- And for the privacy-centric optimisation:

$$\begin{aligned} \rho_{\text{FD}}^*(\epsilon) &= \inf_{P_{Y|S,X}} D(P_{X,Y}), \\ \text{s.t. } J(S; Y) &\leq \epsilon. \end{aligned}$$

Special output perturbation observation model

- $P_{S,X}$ is general, but

$$W = X,$$

where mechanism has **direct noiseless access** to useful information **only**.

- That is, we have the additional Markov chain constraint

$$S \rightarrow X \rightarrow Y.$$

- The privacy-utility optimisation problems reduce to

$$\epsilon_{\text{OP}}^*(\rho) = \inf_{P_{Y|X}} J(S; Y),$$

$$\text{s.t. } D(P_{X,Y}) \leq \rho.$$

where

$$P_{Y|S}(y|s) = \sum_{x \in \mathcal{X}} P_{Y,X|S}(y, x|s) = \sum_{x \in \mathcal{X}} P_{X|S}(x|s) P_{Y|X}(y|x).$$

- It is clear that full data performs better than output perturbation in terms of privacy measure because of access to more data directly

$$\epsilon_{\text{FD}}^*(\rho) \leq \epsilon_{\text{OP}}^*(\rho).$$

α -loss with Y

Recall the Markov chain $S \rightarrow Y \rightarrow \hat{S}$, where \hat{S} is an estimator of S from Y . For any $\alpha \in [1, \infty]$, the α -loss of the strategy $P_{\hat{S}|Y}$ is defined as

$$c_\alpha(s, y, P_{\hat{S}|Y}) = \begin{cases} \frac{\alpha}{\alpha-1} (1 - P_{\hat{S}|Y}(s|y))^{1-\frac{1}{\alpha}}, & \alpha \in (1, \infty), \\ -\log P_{\hat{S}|Y}(s|y), & \alpha = 1, \\ 1 - P_{\hat{S}|Y}(s|y), & \alpha = \infty. \end{cases}$$

For large α , larger probabilities $P_{\hat{S}|Y}(s|y)$ give lower losses.

α -loss without Y

By the same token, the α -loss of the strategy $P_{\hat{S}}$ in the absence of Y is

$$c_{\alpha}(s, P_{\hat{S}}) = \begin{cases} \frac{\alpha}{\alpha-1}(1 - P_{\hat{S}}(s))^{1-\frac{1}{\alpha}}, & \alpha \in (1, \alpha), \\ -\log P_{\hat{S}}(s), & \alpha = 1, \\ 1 - P_{\hat{S}}(s), & \alpha = \infty. \end{cases}$$

Minimum expected α -loss with Y

Minimum α -loss with Y

For $\alpha \in [1, \infty]$, the minimal expected α -loss **with** observation Y is

$$\min_{P_{\hat{S}|Y}} \mathbb{E}_{P_{S,Y}} [c_{\alpha}(S, Y, P_{\hat{S}|Y})] = \begin{cases} \frac{\alpha}{\alpha-1} \left(1 - \sum_{y \in \mathcal{Y}} P_Y(y) \|P_{S|Y=y}\|_{\alpha}\right), & \alpha \in (1, \infty), \\ H(S|Y), & \alpha = 1, \\ 1 - \sum_{y \in \mathcal{Y}} P_Y(y) \max_{s \in \mathcal{S}} P_{S|Y}(s|y), & \alpha = \infty, \end{cases}$$

Optimal estimation strategy with Y is the normalised α -norm of distribution

$$P_{\hat{S}|Y}^*(s|y) = \frac{P_{S|Y}(s|y)^{\alpha}}{\sum_{s \in \mathcal{S}} P_{S|Y}(s|y)^{\alpha}}, \quad s \in \mathcal{S}, y \in \mathcal{Y}.$$

Minimum expected α -loss with Y

Proof.

$$\min_{P_{\hat{S}|Y}} \mathbb{E}_{P_{S,Y}} [c_{\alpha}(S, Y, P_{\hat{S}|Y})] = \min_{P_{\hat{S}|Y}} \left[\frac{\alpha}{\alpha - 1} \left(1 - \sum_{s,y} P_{S,Y}(s, y) P_{\hat{S}|Y}(s|y)^{1 - \frac{1}{\alpha}} \right) \right]$$

$$\min_{P_{\hat{S}|Y}} \mathbb{E}_{P_{S,Y}} [c_{\alpha}(S, Y, P_{\hat{S}|Y})] = \frac{\alpha}{\alpha - 1} \left(1 - \max_{P_{\hat{S}|Y}} \left[\sum_{s,y} P_{S,Y}(s, y) P_{\hat{S}|Y}(s|y)^{1 - \frac{1}{\alpha}} \right] \right)$$

$$\min_{P_{\hat{S}|Y}} \mathbb{E}_{P_{S,Y}} [c_{\alpha}(S, Y, P_{\hat{S}|Y})] = \frac{\alpha}{\alpha - 1} \left(1 - \max_{P_{\hat{S}|Y}} \left[\sum_y P(y) \sum_s P(s|y) P_{\hat{S}|Y}(s|y)^{1 - \frac{1}{\alpha}} \right] \right)$$

□

Proof.

$$\begin{aligned} & \max_{P_{\hat{S}|Y}} \left[\sum P(s|y) P_{\hat{S}|Y}(s|y)^{1-\frac{1}{\alpha}} \right] \\ \text{s.t.} \quad & \sum_s P_{\hat{S}|Y}(s|y) = 1, \quad \forall y \in \mathcal{Y} \\ & P_{\hat{S}|Y}(s|y) \geq 0 \quad \forall s \in \mathcal{S}, y \in \mathcal{Y} \end{aligned}$$

$$\Rightarrow P_{\hat{S}|Y}^*(s|y) = \frac{P_{S|Y}(s|y)^\alpha}{\sum_{s \in \mathcal{S}} P_{S|Y}(s|y)^\alpha}, \quad s \in \mathcal{S}, y \in \mathcal{Y}$$

$$\max_{P_{\hat{S}|Y}} \left[\sum P(s|y) P_{\hat{S}|Y}(s|y)^{1-\frac{1}{\alpha}} \right] = \|P_{S|Y=y}\|_\alpha = \left(\sum_{s \in \mathcal{S}} P_{S|Y}(s|y)^\alpha \right)^{\frac{1}{\alpha}}.$$

□

Minimum expected α -loss without Y

Minimum α -loss without Y

For $\alpha \in [1, \infty]$, the minimal expected α -loss **without** any observation Y is

$$\min_{P_{\hat{S}}} \mathbb{E}_{P_S} [c_{\alpha}(S, P_{\hat{S}})] = \begin{cases} \frac{\alpha}{\alpha-1} \left(1 - (\sum_s P_S(s)^{\alpha})^{\frac{1}{\alpha}} \right) = \frac{\alpha}{\alpha-1} (1 - \|P_S\|_{\alpha}), & \alpha \in (1, \infty) \\ H(S), & \alpha = 1, \\ 1 - \max_{s \in \mathcal{S}} P_S(s) & \alpha = \infty. \end{cases}$$

Optimal estimation strategy without Y is the normalised α -norm of distribution

$$P_{\hat{S}}^*(s) = \frac{P_S(s)^{\alpha}}{\sum_{s \in \mathcal{S}} P_S(s)^{\alpha}}, \quad s \in \mathcal{S}.$$

Definition

The α -leakage between S and Y is (the scaled logarithm of) the multiplicative increase in the maximal expected α -reward gained by the adversary:

$$J_{\alpha L}(S; Y) = \lim_{\alpha' \rightarrow \alpha} \frac{\alpha'}{\alpha' - 1} \log \left(\frac{\max_{P_{\hat{S}|Y}} \mathbb{E}_{P_{S,Y}} [P_{\hat{S}|Y}(S|Y)^{1-\frac{1}{\alpha'}}]}{\max_{P_{\hat{S}}} \mathbb{E}_{P_S} [P_{\hat{S}}(S)^{1-\frac{1}{\alpha'}}]} \right).$$

Theorem

α -leakage is equal to Arimoto mutual information of order α . That is,

$$J_{\alpha L}(S; Y) = I_{\alpha}^A(S; Y).$$

Definition (Maximal α -leakage)

Maximal α -leakage between variable X and released variable Y is defined as

$$J_{\text{MaxL}}(X; Y) = \sup_{S \rightarrow X \rightarrow Y} J_{\alpha\text{L}}(S; Y),$$

which measures adversary's capability to infer **any (possibly random) function** of X from Y .

Theorem

Maximal α -leakage is given by

$$J_{M\alpha L}(X; Y) = \begin{cases} \sup_{P_{\tilde{X}} \ll P_X} I_{\alpha}^A(\tilde{X}; Y) = \sup_{P_{\tilde{X}} \ll P_X} I_{\alpha}^S(\tilde{X}; Y), & \alpha \in (1, \infty], \\ I(X; Y) & \alpha = 1, \end{cases}$$

where the supremum is over all distributions $P_{\tilde{X}}$ whose support is a subset of (or equal to) that of P_X .

Special case for $\alpha = \infty$: maximal leakage [24]

Definition (Maximal leakage)

Maximal leakage between variable X and released variable Y is the maximal α -leakage for $\alpha = \infty$

$$J_{ML}(X; Y) = J_{M\alpha L}(X; Y)|_{\alpha=\infty} = \sup_{S \rightarrow X \rightarrow Y} I_{\infty}^A(S; Y),$$

Theorem

Maximal leakage is given by

$$J_{ML}(X; Y) = I_{\infty}^S(X; Y) = \log \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} P_{Y|X}(y|x).$$

- Tunable α -lift is inspired by the two notions of local information privacy (lift) and tunable α -leakage.
- Score a tunable version of the lift, $\ell(s, y)$ for $\alpha \in (1, \infty)$ with the likelihood of s , $P_S(s)$ as follows:

$$\begin{aligned} \ell_\alpha(y) &\triangleq \left(\sum_{s \in \mathcal{S}} P_S(s) \ell(s, y)^\alpha \right)^{\frac{1}{\alpha}} = \left(\sum_{s \in \mathcal{S}} P_S(s) \left(\frac{P_{S|Y}(s|y)}{P_S(s)} \right)^\alpha \right)^{\frac{1}{\alpha}} \\ &= \frac{\alpha - 1}{\alpha} \exp(D_\alpha(P_{S|Y=y} \| P_S)), \quad y \in \mathcal{Y}. \end{aligned}$$

This is the expected α -norm of lift, also rewritten as

$$\ell_\alpha(y) = \left(\sum_{s \in \mathcal{S}} P_S(s) \left(\frac{P_{Y|S}(y|s)}{P_Y(y)} \right)^\alpha \right)^{\frac{1}{\alpha}} = \frac{1}{P_Y(y)} \left(\sum_{s \in \mathcal{S}} P_S(s) P_{Y|S}(y|s)^\alpha \right)^{\frac{1}{\alpha}}.$$

The Sibson mutual information of order α between S and Y is the expected α -lift over marginal distribution of the released information Y .

$$\begin{aligned}
 I_{\alpha}^S(S; Y) &= \frac{\alpha}{\alpha - 1} \log \sum_{y \in \mathcal{Y}} \left(\sum_{s \in \mathcal{S}} P_S(s) P_{Y|S}(y|s)^\alpha \right)^{\frac{1}{\alpha}} \\
 &= \frac{\alpha}{\alpha - 1} \log \sum_{y \in \mathcal{Y}} \frac{P_Y(y)}{P_Y(y)} \left(\sum_{s \in \mathcal{S}} P_S(s) P_{Y|S}(y|s)^\alpha \right)^{\frac{1}{\alpha}} \\
 &= \frac{\alpha}{\alpha - 1} \log \sum_{y \in \mathcal{Y}} P_Y(y) l_{\alpha}(y) = \frac{\alpha}{\alpha - 1} \log \mathbb{E}_{P_Y}[l_{\alpha}(Y)].
 \end{aligned}$$

Definition

The (worst-case) α -lift measure between \mathcal{S} and \mathcal{Y} is maximum of logarithm of $\ell_\alpha(y)$ over $y \in \mathcal{Y}$. That is,

$$J_{\alpha\text{Lift}}(\mathcal{S}; \mathcal{Y}) = \max_{y \in \mathcal{Y}} \log(\ell_\alpha(y)) = \max_{y \in \mathcal{Y}} \log \left(\left(\sum_{s \in \mathcal{S}} P_S(s) \ell(s, y)^\alpha \right)^{\frac{1}{\alpha}} \right),$$

where

$$\ell(s, y) = \frac{P(s|y)}{P(s)},$$

is the lift.

Privatisation Scheme: replace worst-case $i(s, x)$ by $\log l_\alpha(x)$ in watchdog method [15].

For $\epsilon > 0$: $\mathcal{X}_\epsilon = \{x \in \mathcal{X} : \log l_\alpha(x) \leq \epsilon\}$, $\mathcal{X}_\epsilon^c = \mathcal{X} \setminus \mathcal{X}_\epsilon$.

$$P(y|x) = \begin{cases} 1 & x = y \in \mathcal{X}_\epsilon, \\ 0 & x, y \in \mathcal{X}_\epsilon, x \neq y, \\ r(y|x) & x, y \in \mathcal{X}_\epsilon^c, \end{cases} \quad (1)$$

By randomisation $P(y|x)$, the resulting α -lift of $y \in \mathcal{Y}$ is:

Resulted α -lift

$$l_\alpha(y) = \begin{cases} l_\alpha(x) & x, y \in \mathcal{X}_\epsilon : x = y, \\ \left(\sum_{s \in \mathcal{S}} P(s) \left(\frac{P(y|s)}{P(y)} \right)^\alpha \right)^{1/\alpha} & y \in \mathcal{X}_\epsilon^c. \end{cases} \quad (2)$$

for $y \in \mathcal{X}_\epsilon^c$: $P(y) = \sum_{x \in \mathcal{X}_\epsilon^c} r(y|x)P(x)$, $P(y|s) = \sum_{x \in \mathcal{X}_\epsilon^c} r(y|x)P(x|s)$.

Optimal $r^*(y|x)$

For watchdog randomisation, it suffices to determine the optimal $r(y|x)$. It was shown in [5] that the optimal $r^*(y|x)$ minimizing the worst-case $\ell_\alpha(y)$ in the high-risk symbols $\forall x, y \in \mathcal{X}_\epsilon^c$ is X -invariant.

Optimal $r^*(y|x)$

For all $\alpha \in (1, \infty)$ and $\epsilon > 0$,

$$r^*(y|x) = R(y) \in \arg \min_{r(y|x)} \max_{y \in \mathcal{X}_\epsilon^c} \ell_\alpha(y), \quad \forall x, y \in \mathcal{X}_\epsilon^c \quad (3)$$

Example - Uniform Mechanism: $R(y) = \frac{1}{|\mathcal{X}_\epsilon^c|}, \forall y \in \mathcal{X}_\epsilon^c$.

Example - Merging Mechanism: $R(y^*) = 1$: for supersymbol $y^* \in \mathcal{X}_\epsilon^c$.

This generalises the existing results in [3].

Extra slides on relations and properties of different privacy measures

Lemma

$J_{LIP}^U(S; Y)$ implies $J_{\alpha L}(S; Y)$. That is

$$J_{LIP}^U(S; Y) \leq \epsilon_u \quad \Rightarrow \quad J_{\alpha L}(S; Y) \leq \frac{\alpha}{\alpha - 1} \epsilon_u.$$

Proof.

First, note:

$$\begin{aligned}
 J_{\text{LIP}}^U(S; Y) &\leq \epsilon_u \\
 \Rightarrow i(s, y) &= \log \left(\frac{P_{S|Y}(S = s | Y = y)}{P_S(S = s)} \right) \leq \epsilon_u, \quad \forall s \in \mathcal{S}, y \in \mathcal{Y} \\
 \Rightarrow P_{S|Y}(S = s | Y = y) &\leq e^{\epsilon_u} P_S(S = s), \quad \forall s \in \mathcal{S}, y \in \mathcal{Y} \\
 \Rightarrow \|P_{S|Y=y}\|_\alpha &\leq e^{\epsilon_u} \|P_S\|_\alpha, \quad \forall y \in \mathcal{Y}.
 \end{aligned}$$

Recall that $J_{\alpha\text{L}}(S; Y) = I_\alpha^A(S; Y)$. Therefore, for $\alpha \in (1, \alpha)$:

$$I_\alpha^A(S; Y) = \frac{\alpha}{\alpha - 1} \log \left(\frac{\sum_{y \in \mathcal{Y}} P_Y(y) \|P_{S|Y=y}\|_\alpha}{\|P_S\|_\alpha} \right) \leq \frac{\alpha}{\alpha - 1} \epsilon_u.$$



Upper local information privacy implies α -leakage

Proof.

The case for $\alpha = 1$ was explicitly proved. The case for $\alpha = \infty$ can explicitly be proved as follows, noting that

$$\begin{aligned} P_{S|Y}(S = s|Y = y) &\leq e^{\epsilon_u} P_S(S = s), \quad \forall s \in \mathcal{S}, y \in \mathcal{Y} \\ \Rightarrow \max_{s \in \mathcal{S}} P_{S|Y}(S = s|Y = y) &\leq e^{\epsilon_u} P_S(S = s), \quad \forall y \in \mathcal{Y}. \end{aligned}$$

Therefore,

$$\begin{aligned} I_{\infty}^A(S; Y) &= \log \left(\frac{\sum_{y \in \mathcal{Y}} P_Y(y) \max_{s \in \mathcal{S}} P_{S|Y}(s|y)}{\max_{s \in \mathcal{S}} P_S(s)} \right) \\ &\leq \log \left(\frac{\sum_{y \in \mathcal{Y}} P_Y(y) e^{\epsilon_u} P_S(S = s)}{\max_{s \in \mathcal{S}} P_S(s)} \right) \\ &= \log \left(\frac{e^{\epsilon_u} P_S(s) \sum_{y \in \mathcal{Y}} P_Y(y)}{\max_{s \in \mathcal{S}} P_S(s)} \right) \leq \epsilon_u. \end{aligned}$$



Lemma

$J_{LIP}^U(S; Y)$ implies $J_{\alpha Lift}(S; Y)$. That is

$$J_{LIP}^U(S; Y) \leq \epsilon_u \Rightarrow J_{\alpha Lift}(S; Y) \leq \epsilon_u.$$

Proof.

First note,

$$J_{\text{LIP}}^U(S; Y) \leq \epsilon_u$$

$$\Rightarrow i(s, y) = \log \left(\frac{P_{S|Y}(S = s | Y = y)}{P_S(S = s)} \right) = \log \ell(s, y) \leq \epsilon_u, \quad \forall s \in \mathcal{S}, y \in \mathcal{Y}$$

$$\Rightarrow \ell(s, y) \leq e^{\epsilon_u}, \quad \forall s \in \mathcal{S}, y \in \mathcal{Y}$$

$$\begin{aligned} \Rightarrow J_{\alpha\text{Lift}}(S; Y) &= \max_{y \in \mathcal{Y}} \log \left(\left(\sum_{s \in \mathcal{S}} P_S(s) \ell(s, y)^\alpha \right)^{\frac{1}{\alpha}} \right) \\ &\leq \max_{y \in \mathcal{Y}} \log \left(\left(\sum_{s \in \mathcal{S}} P_S(s) e^{\alpha \epsilon_u} \right)^{\frac{1}{\alpha}} \right) \\ &= \epsilon_u \end{aligned}$$



Lemma

Maximal α -leakage implies α -leakage. That is

$$J_{M\alpha L}(X; Y) \leq \epsilon \quad \Rightarrow \quad J_{\alpha L}(X; Y) \leq \epsilon.$$

Follows immediately from the definition of maximal α -leakage which is the supremum of α -leakage over all Markov chains $S \rightarrow X \rightarrow Y$.

Maximal α -leakage implies maximal α' -leakage for $\alpha \leq \alpha'$

Lemma

Maximal α' -leakage implies maximal α -leakage. In particular, for $1 \leq \alpha \leq \alpha' \leq \infty$, we have

$$\begin{aligned} J_{M\alpha L}(X; Y) \leq \epsilon &\Rightarrow I(X; Y) \leq \epsilon, \\ J_{M\alpha' L}(X; Y) \leq \epsilon &\Rightarrow J_{M\alpha L}(X; Y) \leq \epsilon, \\ (\alpha' = \infty) \quad J_{ML}(X; Y) \leq \epsilon &\Rightarrow J_{M\alpha' L}(X; Y) \leq \epsilon. \end{aligned}$$

Follows from the monotonicity property of the Sibson mutual information in order α .

In the remaining slides we discuss basic desired properties of privacy measures.
Adapted mainly from [12].

Independence

Any meaningful privacy measure must be non-negative $J(S; Y) \geq 0$ with $J(S; Y) = 0$ if and only if S and Y are independent.

Definition

A privacy measure $J(S; Y)$ satisfies the post-processing inequality if and only if for any $S \rightarrow Y_1 \rightarrow Y_2$ that form a Markov chain, we have $J(S; Y_1) \geq J(S; Y_2)$.

- This property is a fundamental, axiomatic requirement for any reasonable privacy measure.
- It establishes that a privacy measure cannot deteriorate by independent post-processing of the released data.

Definition

A privacy measure $J(S_1; Y)$ satisfies the linkage inequality if and only if for any $S_2 \rightarrow S_1 \rightarrow Y$ that form a Markov chain, we have that $J(S_1; Y) \geq J(S_2; Y)$.

Having this property in the privacy measure is important because:

- If the release was generated from only the primary sensitive data S_1 , then the privacy-leakage for the secondary sensitive data S_2 is bounded by the privacy-leakage for the primary sensitive data S_1 .
- It provides stronger privacy guarantees, especially when there may be unforeseen secondary sensitive data correlated with the primary sensitive data originally considered in the release.

The relationship between linkage and post-processing inequalities

Lemma

For symmetric privacy measures where $J(S; Y) = J(Y; S)$, post-processing and linkage inequalities are equivalent.

Proof.

Assume post-processing inequality holds in $S \rightarrow Y_1 \rightarrow Y_2 \Leftrightarrow Y_2 \rightarrow Y_1 \rightarrow S$.

$$J(S; Y_1) \geq J(S; Y_2) \Rightarrow J(Y_1; S) \geq J(Y_2; S)$$

where the last inequality is the linkage inequality if rename the variables in Markov chain $Y_2 \rightarrow Y_1 \rightarrow S$ to $S_2 \rightarrow S_1 \rightarrow Y$ to conclude:

$$J(S_1; Y) \geq J(S_2; Y).$$

The other direction to establish the equivalence can be similarly proved. □

Lemma

The differential privacy measure $J_{DP}(S; Y)$ does not satisfy the linkage inequality.

Pure Differential Privacy Measure

$$J_{DP}(S; Y) = \sup_{s, s': s \sim s', E \subset \mathcal{Y}} \left| \log \left(\frac{P(Y \in E | s)}{P(Y \in E | s')} \right) \right|.$$

proof

The counterexample is from [12].

- Consider two databases S_1, S_2 , each with two binary entries $S_1, S_2 \in \{0, 1\}^2$.
- Let $Y \in \{0, 1\}$.
- Let database $S_1 = (s_1^1, s_2^1)$ be a deterministic and correlated entry function of database $S_2 = (s_1^2, s_2^2)$.
- Let database entries of S_1 be identically correlated and determined as $s_1^1 = s_2^1 = s_1^2 \vee s_2^2$.
- E.g., S_1 represents disease (e.g., covid-19, delta variant) in *any* member in a family of two, whereas S_2 contains the original disease indicator per family member.

Differential privacy holds post-processing but not linkage inequality

proof

- The release mechanism for $Y \in \{0, 1\}$ is based on access to S_1 only:

$$P_{Y|S_1}(Y = 1|S_1 = (s_1^1, s_2^1)) = \begin{cases} q, & (s_1^1, s_2^1) = (0, 0), \\ s, & (s_1^1, s_2^1) = (1, 1), \\ r, & \text{otherwise,} \end{cases}$$

where $0 < q < r < s < 1$.

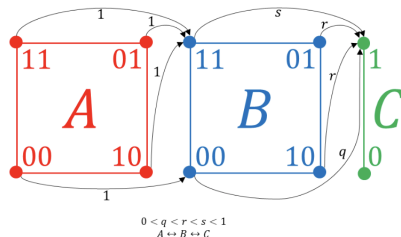


Figure from [12].

proof

- Now consider the $J_{\text{DP}}(S_1; Y)$ which is given as

$$J_{\text{DP}}(S_1; Y) = \max \left\{ \log \frac{s}{r}, \log \frac{r}{q}, \log \frac{\bar{r}}{\bar{s}}, \log \frac{\bar{q}}{\bar{r}} \right\}.$$

- Due to the construction of the mechanism $0 < q < r < s < 1$, we have









$$\max \left\{ \frac{s}{r}, \frac{r}{q} \right\} < \frac{s}{q} \qquad \max \left\{ \frac{\bar{r}}{\bar{s}}, \frac{\bar{q}}{\bar{r}} \right\} < \frac{\bar{q}}{\bar{s}}.$$










- Therefore,










$$J_{\text{DP}}(S_2; Y) = \max \left\{ \log \frac{s}{q}, \log \frac{\bar{q}}{\bar{s}} \right\} > J_{\text{DP}}(S_1; Y),$$

which shows violation of linkage inequality.

- Connections between conditional worst-case mutual information and differential privacy: [25]: Cuff and Yu, 2016
- Connections between mutual information, identifiability and differential privacy: [26]: Wang, Ying and Zhang, 2016
- Quantitative information flow, Bayesian inference, ϵ -leakage, axomatic reasoning about privacy measures, [27, 28, 29]: Smith, 2009; Alvim et al, 2014, Alvim et al, 2016

- 
- L. Sweeney, “Only you, your doctor, and many others may know,” *J. Technology Science*, 2015.
- 
- N. Ding and P. Sadeghi, “A submodularity-based clustering algorithm for the information bottleneck and privacy funnel,” in *ITW*, 2019.
- 
- P. Sadeghi, N. Ding, and T. Rakotoarivelo, “On properties and optimization of information-theoretic privacy watchdog,” in *IEEE Inf. Theory Workshop (ITW)*, Apr. 2020, pp. 1–5.
- 
- R. G. L. D’Oliveira, S., M. Medard, and P. Sadeghi, “Differential privacy for binary functions via randomized graph colorings,” in *Proc. Int. Symp. on Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 473–478.
- 
- N. Ding, M. A. Zarrabian, and P. Sadeghi, “ α -information-theoretic privacy watchdog and optimal privatization scheme,” in *Proc. Int. Symp. on Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 2584–2589.
- 
- Y. Liu, N. Ding, P. Sadeghi, and T. Rakotoarivelo, “Privacy-utility tradeoff in a guessing framework inspired by index coding,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, LA, CA, Jun. 2020, pp. 926–931.
- 
- Y. Liu, O. Lawrence, S. J. Johnson, J. Kliewer, P. Sadeghi, and P. L. Yeoh, “Information leakage in zero-error source coding: A graph-theoretic perspective,” in *Proc. Int. Symp. on Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 2590–2595.
- 
- Y. Liu, O. Lawrence, P. L. Yeoh, P. Sadeghi, J. Kliewer, and S. J. Johnson, “Information leakage in index coding,” in *Proc. IEEE Information Theory Workshop (ITW)*, Kanazawa, Japan, Oct. 2021.

-  R. G. L. D'Oliveira, S. Salamatian, , M. Medard, and P. Sadeghi, "Low influence, utility, and independence in differential privacy: A curious case of $\binom{3}{2}$," *IEEE J. on Selected Areas in Inf. Theory*, vol. 2, no. 1, pp. 240–252, Mar. 2021.
-  F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *Allerton Conf.*, Monticello, IL, 2012, pp. 1401–1408.
-  J. Liao, O. Kosut, L. Sankar, and F. du Pin Calmon, "Tunable measures for information leakage and applications to privacy-utility tradeoffs," *IEEE Trans. on Inform. Theory*, vol. 65, no. 12, pp. 8043–8066, 2019.
-  Y. Wang, Y. O. Basciftci, and P. Ishwar. (2017) Privacy-utility tradeoffs under constrained data release mechanisms. [Online]. Available: arXiv:1710.09295v1
-  A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," in *ITW*, 2014.
-  N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.
-  H. Hsu, S. Asodeh, and F. P. Calmon, "Information-theoretic privacy watchdogs," in *ISIT*, Paris, France, 2019, pp. 552–556.
-  M. Lopuhaa-Zwakenberg, "The privacy funnel from the viewpoint of local differential privacy," 2020. [Online]. Available: <https://arxiv.org/abs/2002.01501v1>
-  C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

- 
- S. Kasiviswanathan, H. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.
- 
- Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 1054–1067.
- 
- D. P. Team, *Learning with Privacy at Scale*, 2017 (last accessed May 2021). [Online]. Available: <https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html>
- 
- B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," *arXiv preprint arXiv:1712.01524*, 2017.
- 
- Disclosure Avoidance and the 2020 Census*, 2020 (last accessed May 2020). [Online]. Available: https://www.census.gov/about/policies/privacy/statistical_safeguards/disclosure-avoidance-2020-census.html
- 
- E. by: Peter Kairouz and H. B. McMahan, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1, pp. –, 2021. [Online]. Available: <http://dx.doi.org/10.1561/22000000083>
- 
- I. Issa, A. B. Wagner, and S. Kamath, "An operational approach to information leakage," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1625–1657, Mar. 2020.
- 
- P. Cuff and L. Yu, "Differential privacy as a mutual information constraint," in *CSS*, 2016, pp. 43–54.
- 
- W. Wang, L. Ying, and J. Zhang, "On the relation between identifiability, differential privacy, and mutual-information privacy," *IEEE Trans. Inf. Theory*, pp. 5018–5029, 2016.



G. Smith, "On the foundations of quantitative information flow," in *Foundations of Software Science and Computational Structures*, L. de Alfaro, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 288–302.



M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *Proceedings of the 2012 IEEE 25th Computer Security Foundations Symposium*, ser. CSF '12. USA: IEEE Computer Society, 2012, p. 265–279. [Online]. Available: <https://doi.org/10.1109/CSF.2012.26>



M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, , C. Palamidess, and G. Smith, "Axioms for information leakage," in *IEEE Computer Security Foundations Symposium*, 2016, pp. 77–92.