Privacy Funnel Under Maximal Leakage Report

The privacy funnel problem is closely related to the information bottleneck one, since these two optimization problems have applications in machine learning, design of privacy algorithms, capacity problems (e.g., Mrs. Gerber’s Lemma), strong data processing inequalities, among others.

I will now start by introducing these two problems.

The goal of IB is to get a compact version of a specific variable that is maximally informative for inferring another variable. The compact version will be called T, and it is generated from X by applying a certain random function F in a way that is conditionally independent of Y given X, hence obtaining the following Markov chain Y 🡪 X 🡪 T.

Information bottleneck measures the ‘compactness’ of T using the mutual information I(X;T) and its informativeness by calculating I(Y;T). Hence to formulate this optimization problem, we can say that IB(R) = sup I(Y;T) . subject to I(X;T) <=R. with R being the measure of compactness.

In contrast, in the privacy funnel problem, a bottleneck variable T is sought to be maximally preserve "information" contained in X while revealing as little about Y as possible.

Therefore, PF(r) = inf I(Y;T) subject to I(X;T) >= r. where r is the parameter specifying the level of informativeness.

And finally I will introduce maximal leakage which is defined as the multiplicative increase, upon observing Y , of the probability of correctly guessing a randomized function of X, maximized over all such randomized functions.

Given two discrete random variables X and Y, we wanted to know how much information does Y leak about X? A quantitative answer to that question is crucial to assess the performance of privacy systems for which Y cannot be made independent of X, which is often the case in practice. Leakage for a specific function is considered to be the logarithm of the ratio of the probability of a correct guess when Y is observed, to the probability of a correct guess when it is not (i.e., a blind guess), which is identical to the Sibson mutual information of order infinity I∞(X; Y ).

I∞(X; Y ) = log .

Where Pc(U|V ) and Pc(U) characterize the optimal efficiency of guessing U with or without the observation V , respectively. Intuitively, I∞(U; V ) quantifies how useful the observation V is in estimating U: If it is small, then it means it is nearly as hard for an adversary observing V to guess U as it is without V . This observation motivates the use of I∞(U; V ) as a measure of privacy in lieu of I(Y ; T) in PF.

It is worth noting that I∞(U; V ) is not symmetric in general, i.e., I∞(U; V ) ≠ I∞(U; V ). Since observing T can only improve, we have Pc(Y |T) ≥ Pc(Y ); thus I∞(Y ; T) ≥ 0. However, I∞(Y ; T) = 0 does not necessarily imply independent of Y and T; instead, it means T is useless in estimating Y.

Maximal leakage, which we denote by L(X→Y ), is then defined as the maximum leakage over all such randomized functions. This maximization, which is formally over discrete random variables U for which the Markov chain U − X − Y holds.

I am going to list here some of the notions that I found interesting while reading papers related to this research.

The first notion is the notion of ‘relevant’ information since this has never been discussed especially in the Shannon theorem where the concentration has only been on how much information we are actually transmitting rather than determining what is ‘relevant’. Without an appropriate definition of relevance, the problem of extracting a relevant summary of data, a compressed description that captures only the relevant or significant information, is not well presented. The notion of relevance differs depending on the type of signal we are transmitting whether it is speech, images… A typical example is that of speech compression. One can consider lossless compression, but in any compression beyond the entropy of speech some components of the signal cannot be reconstructed. On the other hand, a transcript of the spoken words has much lower entropy (by orders of magnitude) than the acoustic waveform, which means that it is possible to compress (much) further without losing any information about the words and their meaning. The “rate distortion theory” describes the tradeoff between the rate, or signal representation size, and the average distortion of the reconstructed signal, which is the usual study of lossy source compression. The main problem with rate distortion theory is in the need to specify the distortion function first, which in turn determines the relevant features of the signal. Those features, however, are often not explicitly known and an arbitrary choice of the distortion function is in fact an arbitrary feature selection. This is the fundamental problem of feature selection in pattern recognition. Rate distortion theory does not provide a full answer to this problem since the choice of the distortion function, which determines the relevant features, is not part of the theory.

The second notion is the notion of evaluation of IB and PF. Since the optimization problem of IB and PF is hard to solve numerically due to the non linearity in the constraints, we then resolve to approximate them by their corresponding Lagrangian optimizations. LIB(β) = sup PT|X I(Y;T) − βI(X;T)

and LPF(β) = inf PT|X I(Y;T) − βI(X;T); where β ∈ R+ is the Lagrangian multiplier that controls the tradeoff between compression and informativeness in for IB and the privacy and informativeness in PF.

Text, letter

Description automatically generatedText

Description automatically generatedThe third notion is the expansion of IB and PF formulas to include different degrees of Arimoto’s mutual information. For example, if we have a pair of random variables (X,Y) PXY over finite sets X and Y and α, γ ∈ [1, ∞], we define IB(α,γ) and PF(α,γ) as

With Ia being Arimoto’s mutual information of order α > 1 that is defined as Iα(U; V ) = Hα(U) − Hα(U|V ), with Hα(U) and Hα(U|V ) being the renyi entropy of order a and the arimoto’s conditional entropy of order a respectively having the following formulas:

A picture containing graphical user interface

Description automatically generated 

If we take a special case of this generalization, and we define IB (∞,1) that characterizes the best error probability in recovering Y among all R-bit summaries of X, and PF(∞,1) that characterizes the smallest probability of revealing private feature Y among all representations of X preserving at least r bits information of X. These two cases have the following formula

A picture containing schematic

Description automatically generated 

They were able to find bounds of these two formulations as shown below:

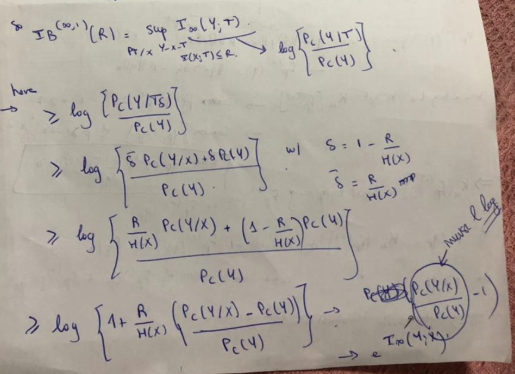


And

A picture containing text, clock, gauge

Description automatically generated

I recreated the proof for the first boundary and projected the findings of the second one given that they are really similar to prove.

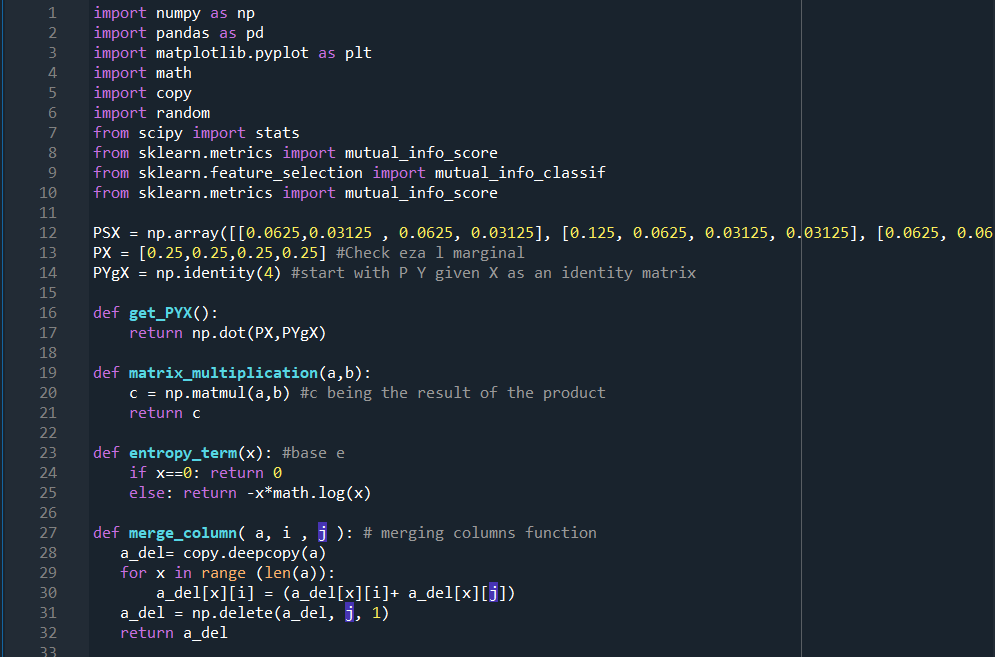


A piece of paper with writing on it

Description automatically generated with medium confidence

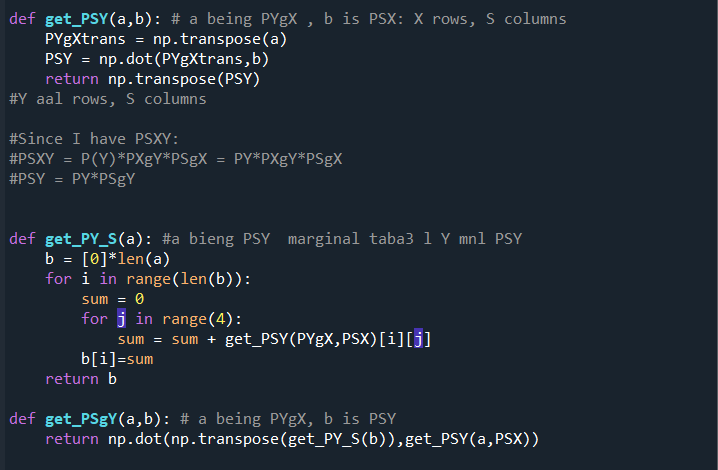
The fourth notion is the robustness of the definition of maximal leakage. In the definition of L(X →Y ), we allow the adversary only one guess. This definition was extended to allow for, say, k guesses for some integer k. This is particularly relevant for privacy problems, for example, if U is a password to some system, then an adversary is typically allowed several wrong guesses before he/she is possibly locked out. We call the modified measure k-maximal leakage, and denote it by L (k) (X→Y ). And the k-Maximal leakage theorem established the equivalence between maximal leakage and k-maximal leakage, and hence its robustness.

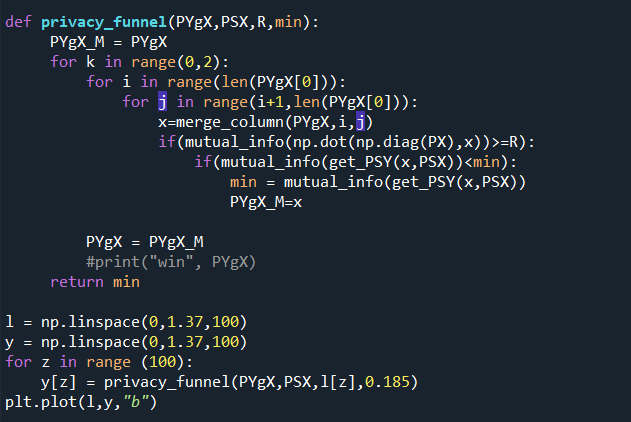
In addition to that I tried to recreate the greedy algorithm for the privacy funnel problem using python. We input R which is the level of informativeness in the PF problem and the joint probability Ps,x. We start with PYgX as an identity matrix and then at each iteration we delete two columns of P (corresponding to yi and yj ) and add their summation as a new column (corresponding to yij ) to PYgX. We need to calculate I(S; Y )−I(S; Y i−j ) at each iteration of Algorithm and take the argument that gives out the maximum difference and proceed with this matrix.



Text

Description automatically generated





Chart

Description automatically generated

Here is some basic explanations for the functions shown above:

The function get\_PYX does not take inputs and return the result of the matrix multiplication of the matrix representing the probability of X and that representing the probability of Y given X.

The function entropy\_term takes as input a certain number and returns the entropy of that term using the formula, H(X) = - X.log(X) , using log base e to perform the calculation.

The function merge\_column takes 3 inputs, ‘a’ which is the matrix, i and j are the columns that need to be merged. It returns the new matrix with the columns merged.

The function entropy\_column takes a matrix representing a joint distribution and returns the marginal entropy of the element that is on the columns.

The function entropy\_row takes a matrix representing a joint distribution and returns the marginal entropy of the element that is on the rows.

The function joint\_entropy takes a matrix representing a joint distribution and returns the joint entropy of that matrix.

The function HrowGivenColumn takes as input a matrix and return conditional entropy of element on rows given that on the columns

The function mutual\_info takes as input a matrix and returns the mutual information of that matrix.

And the final function privacy\_funnel that takes as input the matrices PYgX and PSX, R and min. min being a dummy variable used to get the actual minimum I(S;Y)

We can see from the graph that it is piece-wise linear which is somehow expected due to its nature of being a greedy algorithm.