INSE6180 Jia Yuan Yu

Privacy, Data Mining

Concordia August 9, 2021

What is data mining? Supervised learning (e.g., classifying patients by disease) and unsupervised learning (e.g., discovering a new disease). We can have a whole course on that.

What is privacy and security? Data contains information. Some information is public: constitution, laws, news reports, retail price of goods, etc. Some information is private. Privacy is about private information. For example, when you play a game of high stakes poker, private information includes: what are the cards that you hold, your aversion to risk, your belief on each opponent, etc.

Security issues in data mining include: hacking the computer containing the data, changing your bank account information in a database, stealing your information, etc.

There are security issues that can be addressed by encryption, by wearing camouflage or hoodies, by jamming communications, etc. In this course, we worry only about security issues arising from loss of privacy in a specific context.

1 Context

There's a curator holding a database D. The database contains n rows, each row contains data of a distinct individual. The goal of private data analysis is to find out something useful from the database as a whole, while not finding out much about a given individual. For example, find the winner of an election, but not who voted for whom.

A query on D is a function that maps D to a string. A privacy mechanism is an algorithm that takes as input

1. a database D,
2. a universe $\mathcal{X}$, such that each row of D is an element of $\mathcal{X}$,
3. random bits (review random variables),
4. a set of queries,

and outputs a string that encodes answers to the queries. This output string is sometimes a synthetic database D' of the same composition as D. The queries answers are then obtained by applying the queries on this synthetic database.

Example 1.1. Database D is a record of all transactions in a bank account. The query is the balance after all these transactions.

Example 1.2. There are five candidates and one million voters. Database D is a record of all votes in an election (one column for voter ID number, one column for their vote). An example of a privacy mechanism replaces all elements of the first column by the same number. The output is another database D'.

Example 1.3 (Randomized response)**.** In a study, participants are asked: have you cheated in an exam before? The answers are randomized: with probability 1/2 (a coin flip), the true answer is replaced by a random answer (another coin flip). Let p denote the true fraction of cheaters, the expected fraction of 'yes' answers is $\frac{1}{4}(1-p)+\frac{3}{4}p$.

There are 100 students. A fraction p have cheated. The number of cheaters is $100p$, and the number of non-cheaters is $100(1-p)$. Let's count the average number of 'yes' answers:

- cheaters who get Heads on the first flip: $50p$
- cheaters who get Tails, then Heads on the second flip: $25p$
- non-cheaters who get Tails, then Heads: $25(1-p)$.

Exercise 1. In the above example, we counted the average (expected) number of 'yes' answers. The actual number is a random variable. What is the probability distribution of this random variable? For simplicity, assume that $100p$ is always an integer.

Exercise 2. Following the above randomzation of the anwers, what is the probability that a student who has cheated answers 'no'? what is the probability that a student who has not cheated answers 'yes'?

Exercise 3. Consider an arbitrary student. Let X be a Bernoulli random variable denoting whether this student has cheated before (probability p). Let Y be the random variable denoting the answer given by this student following the randomization above. What is the joint probability distribution of the random variables X, Y ?

1.1 Differential privacy

Let us formalize one notion of privacy and how to achieve it.

Let $D_1, D_2 \in \mathcal{X}^d$ denote two databases that differ in one row[1] (more formal definition later). A randomized algorithm $M : \mathcal{X}^d \to \mathcal{Y}$ is (ϵ, δ)-differentially private if for all $S \subseteq \mathcal{Y}$, and for all pairs $D_1, D_2 \in \mathcal{X}^d$ that differ in one row, we have

$$\mathbb{P}(M(D_1) \in S) \leq \exp(\epsilon) \cdot \mathbb{P}(M(D_2) \in S) + \delta.$$

Example 1.4. In the election example, there are five candidates: Trudeau, Legault, Bieber, Shreya, Trump. In this case, $\mathcal{X}^d$ is an array of votes, with n rows corresponding to the voters, and two columns corresponding to the voter ID and that voter's chosen candidate. Consider an election algorithm M that takes the array of votes as input and outputs the election outcome. In this case, $\mathcal{Y} = \{\text{Trudeau, Legault, Bieber, Shreya, Trump}\}$, and S is one of the possible outcomes of the election, i.e., all non-empty subsets of the five candidates:

[1] We can also make this definition "differ in at most one row."

- Single winners: {Trudeau}, {Legault}, etc.
- Ties with two winners: {Trudeau, Legault}, {Trudeau, Bieber}, etc.
- Ties with three winners
- Ties with four winners
- Ties among all candidates.

Let D_1 be the original array of votes, and D_2 be another array of votes where one voter's (Melania's) vote is changed to a random candidate (uniform distribution). In this way, if the election office releases the database D_2, the private information of Melania is not compromised (up to plausible deniability).

If $\epsilon = \delta = 0$, then $(0,0)$-differential privacy means that for every outcome S and every pair D_1, D_2 (differing in one row), we have

$$\begin{aligned} &\mathbb{P}(M(D_1) \in S) \leq \mathbb{P}(M(D_2) \in S), \\ \text{(and equivalently)} \quad &\mathbb{P}(M(D_2) \in S) \leq \mathbb{P}(M(D_1) \in S). \end{aligned}$$

Observe that $\mathbb{P}(M(D_1) \in S)$ for all S uniquely specifies the probability distribution of the random variable $M(D_1)$. Hence, $(0,0)$-differential privacy means that $D_1 = D_2$.

If $\epsilon = 0.01$ and $\delta = 0$, then the election M being $(0.01, 0)$-differentially private means that for every outcome S and every pair D_1, D_2, we have

$$\mathbb{P}(M(D_1) \in S) \leq 1.01005\ldots \cdot \mathbb{P}(M(D_2) \in S).$$

Exercise 4. Consider the above example, to make it more interesting, we add a $(n+1)$-th voter, who always votes at random (uniform distribution over the five candidates). This new database is denoted D_3. Let M denote the election mechanism that takes the database D_3, selects one row uniformly at random, changes the vote in that row to a random vote (uniformly at random among the candidates), and then declares a winner by majority count (flipping a coin when there are ties). For what values of ϵ and δ is M (ϵ, δ)-differentially private? Brute-force approach: verify this property for a pair of values, then another, etc.

Consider the case where $n = 2$. Let the first voter votes for Trump, and the second voter votes for Legault. There is a third voter who votes at random. Consider a majority mechanism W on these three votes, with tie-breaking at random:

$$\begin{aligned} \mathbb{P}(W(D_3) \in \{\text{Trudeau}\}) &= (1/5) \cdot (1/3) \\ \mathbb{P}(W(D_3) \in \{\text{Trump}\}) &= (1/5) + (1/5) \cdot 0 + (3/5) \cdot (1/3). \end{aligned}$$

Let's consider the randomized mechanism M: if the randomly selected row is the row corresponding to the third voter, then the probabilities are as computed above. Hint: draw a tree for the possible outcomes.

$$\begin{aligned} \mathbb{P}(M(D_3) \in \{\text{Trudeau}\}) = &(1/3) \cdot (1/5) \cdot (1/3) \\ &+ (1/3)((1/5) \cdot (1/5) + (1/5) \cdot (4/5) \cdot (1/3)) \\ &+ (1/3)((1/5) \cdot (1/5) + (1/5) \cdot (4/5) \cdot (1/3)). \end{aligned}$$

In future lectures, we will see how to make a query differential privacy by adding Laplacian noise. Before that, we first make precise the notion of data mining in the next lecture.

1.2 References

- Chapter 1 of ESL.
- Chapter 1 of AFDP.

2 Laplace mechanism for real-valued data

Datasets containing different types of data require different techniques for guaranteeing privacy. In this section, we consider a database $D \in \mathbb{N}^d$ and a query $f : \mathbb{N}^d \to \mathbb{R}^k$ that produces k numbers.

A Laplace random variable with mean 0 and scale b has the following density function:

$$g(z) = \frac{1}{2b} e^{-|z|/b}, \quad z \in \mathbb{R}.$$

Observe that a Laplace random variable with mean μ and scale b has density $g(z - \mu)$.

Exercise 5. Calculate the variance of a Laplace random variable with mean μ and scale b. How does it compare with the variance of an exponential random variable with mean μ and parameter b? Take the derivative of the sigmoid function introduced in the neural network setting, does it decrease exponentially away from its peak?

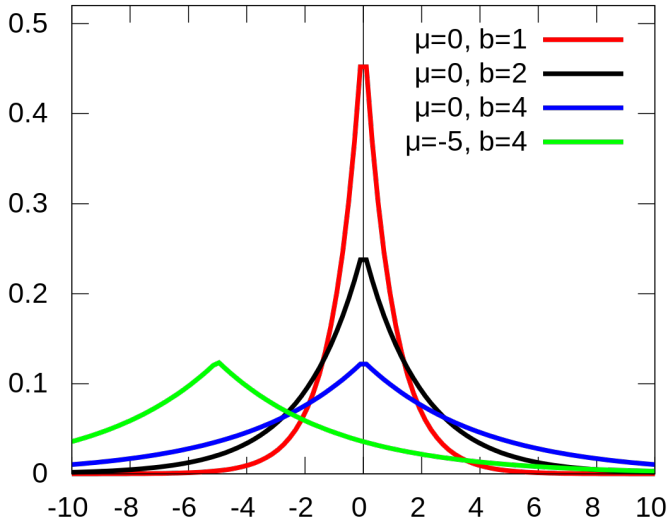

Figure 1: Source: Wikipedia.

Definition 2.1 (Difference between databases)**.** Let $\|x\|_0$ denote the number of nonzero rows in database x, and $\|x - y\|_0$ denote the number of different rows between x and y.

Example: Consider the following databases:

$$r = \begin{matrix} 1 & 2 \\ 5 & 6 \\ 7 & 8 \\ 11 & 5 \end{matrix}, \quad s = \begin{matrix} 1 & 2 \\ 5 & 7 \\ 7 & 8 \\ 5 & 11 \end{matrix}, \quad v = \begin{matrix} 5 & 6 \\ 1 & 2 \\ 7 & 8 \\ 11 & 5 \end{matrix}, \quad w = \begin{matrix} 5 & 6 \\ 1 & 2 \\ 7 & 8 \\ 0 & 0 \end{matrix}$$

then, $\|r - s\|_0 = 2$, $\|r - v\|_0 = 2$, $\|w\|_0 = 3$. Example, let the first column represent the chequing account balance, and the second column the saving account balance, f queries for the highest total balance among customers. Here, $f(r) = 16 = f(s)$, $f(w) = 15$.

Definition 2.2 (Sensitivity)**.** Recall the ℓ_1-norm of linear algebra: for $z = (z_1, \ldots, z_k) \in \mathbb{R}^k$, $\|z\|_1 = |z_1| + \ldots + |z_k|$. Let $f : \mathbb{N}^d \to \mathbb{R}^k$ denote a vector query. The ℓ_1-sensitivity Δ_f of a query f is

$$\Delta_f = \max_{x,y \in \mathbb{N}^d, \|x-y\|_0 = 1} \|f(x) - f(y)\|_1 .$$

Example: $\mathbb{N} = \{0, 1\}$ and $d = 2$. In that case, the sensitivity is calculated over all pairs of databases that differ in one row:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$
$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$
$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$
$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \ldots$$

Example: x is a bank balance database, f is a query for the number of checking accounts, saving accounts, USD accounts, whose balances exceed 100.

Definition 2.3 (Laplace Mechanism)**.** Given a query f on database x, let $\nu_1, \ldots, \nu_k$ denote i.i.d. Laplace random variables with zero mean and scale Δ_f/ε. The Laplace mechanism M_ε^L with parameter ε takes as input a query f and a database x and returns the random vector

$$M_\varepsilon^L(x, f) = f(x) + [\nu_1, \ldots, \nu_k].$$

As the database curator, you can control ε, but Δ_f is given by the query.

Caveat: although some queries f may only generate nonnegative numbers, the output of the Laplace mechanism can be any real number. Hence, it is usually reserved for queries that return (negative and nonnegative) real numbers.

Theorem 2.1 (Privacy of Laplace Mechanism)**.** *The Laplace mechanism M_ε^L with parameter ε is $(\varepsilon, 0)$-differentially private.*

Proof. Let x and y denote two databases such that $\|x - y\|_0 \leq 1$. Let X denote the random variable $M_\varepsilon^L(x, f)$, and let h^X denote the density function and $\mu^X = (\mu_1^X, \ldots, \mu_k^X)$ the mean. Let Y denote the random variable $M_\varepsilon^L(y, f)$, and let h^Y denote the density function and μ^Y the mean. Let $z = (z_1, \ldots, z_k)$ denote an arbitrary output of the Laplace mechanism. Observe that (by independence, and $g^Y(z) > 0$)

$$\begin{aligned}
\frac{h^X(z)}{h^Y(z)} &= \frac{g(z_1 - \mu_1^X) \ldots g(z_k - \mu_k^X)}{g(z_1 - \mu_1^Y) \ldots g(z_k - \mu_k^Y)} \\
&= \frac{\exp(-|z_1 - \mu_1^X|/(\Delta_f/\varepsilon)) \ldots \exp(-|z_k - \mu_k^X|/(\Delta_f/\varepsilon))}{\exp(-|z_1 - \mu_1^Y|/(\Delta_f/\varepsilon)) \ldots \exp(-|z_k - \mu_k^Y|/(\Delta_f/\varepsilon))} \\
&= \exp\left(\frac{|z_1 - \mu_1^Y| - |z_1 - \mu_1^X|}{(\Delta_f/\varepsilon)}\right) \ldots \exp\left(\frac{|z_k - \mu_k^Y| - |z_k - \mu_k^X|}{(\Delta_f/\varepsilon)}\right) \\
\text{(Triangle Ineq.)} &\leq \exp\left(\frac{|\mu_1^Y - \mu_1^X|}{(\Delta_f/\varepsilon)}\right) \ldots \exp\left(\frac{|\mu_k^Y - \mu_k^X|}{(\Delta_f/\varepsilon)}\right) \\
&= \exp\left(\frac{|\mu_1^Y - \mu_1^X| + \ldots + |\mu_k^Y - \mu_k^X|}{(\Delta_f/\varepsilon)}\right) \\
\text{(1-norm)} &= \exp\left(\frac{\left\|\mu^Y - \mu^X\right\|_1}{(\Delta_f/\varepsilon)}\right) \\
&= \exp\left(\varepsilon \frac{\|f(y) - f(x)\|_1}{\Delta_f}\right) \\
\text{(Def of } \Delta_f) &\leq e^\varepsilon.
\end{aligned}$$

Observe that for every $S \subseteq \mathbb{R}^k$

$$\begin{aligned}
\mathbb{P}(M_\varepsilon^L(x, f) \in S) &= \int_S h^X(z) dz \\
&\leq \int_S e^\varepsilon h^Y(z) dz \\
&= e^\varepsilon \mathbb{P}(M_\varepsilon^L(y, f) \in S).
\end{aligned}$$

□

Example 2.1. Let $x = (x_1, \ldots, x_n)$ denote a database with n rows. Let the function h be a binary function, that checks whether a row of x meets a property. A (scalar) counting query $f : \mathbb{N}^d \to \mathbb{R}$ takes the following form:

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} 1_{[h(x_i)=1]}.$$

Clearly, we have $\Delta_f = 1$ in this case. In turn, the Laplace mechanism adds Laplace noise with scale $1/\varepsilon$ to the count $f(x)$. We can form a vector-valued counting query $[f_1, \ldots, f_k]$, which in turn has $\Delta_f = k$ (by definition of the ℓ_1-norm).

Caveat: when $f(x)$ is an integer, or nonnegative, the Laplace mechanism can output a nonsensical value. For example, the query comes from a software that expects integers or nonnegative values as inputs, and throws an exception otherwise. In that case, one hack is to resample the noise. Another hack is rounding real numbers to the nearest integer (think about the effect of rounding on the privacy guarantee).

Exercise 6 (How does plausible deniability work with differential privacy?)**.** Suppose that there is a database with n rows corresponding to n customers, and an attacker colludes with $n-1$ of these customers to find out the values of these $n-1$ rows. Suppose that the attacker also obtains an answer $M(D, f) = 0.5$ to a query f and we know that M is $(0.1, 0)$-differentially private. If the attacker tells the remaining customer: "I know the value of your data," what can this customer say to deny that the attacker knows the value of its data? If the attacker queries the mechanism ℓ times with the same query f and obtains a sequence of answers $M_1, \ldots, M_\ell$, what can the last remaining customer say to deny that the attacker has revealed its data? Hint: consider an interval $[f(D) - \gamma, f(D) + \gamma]$.

Plausible deniability is useless if the answer from the mechanism is not accurate. Next, we give an error bound on the difference between $M_\varepsilon^L(x, f)$ and $f(x)$.

Theorem 2.2 (Error of Laplace Mechanism)**.** *Let* $\|x\|_\infty = \max_{i=1,\ldots,k} |x_i|$*. For every* $\alpha > 0$*, we have*

$$\mathbb{P}\left(\left\| M_\varepsilon^L(x, f) - f(x) \right\|_\infty \geq \frac{\Delta_f}{\varepsilon} \log(k/\alpha) \right) \leq \alpha.$$

Remark 1. Observe the effect of ε and Δ_f on the error. As ε decreases, error increases; as Δ_f increases, the error increases. We want both terms $\frac{\Delta_f}{\varepsilon} \log(k/\alpha)$ and α as small as possible.

Proof.

$$\begin{aligned}
\mathbb{P}\left(\left\| M_\varepsilon^L(x, f) - f(x) \right\|_\infty \geq \frac{\Delta_f}{\varepsilon} \log(k/\alpha) \right) &= \mathbb{P}\left(\max\{\nu_1, \ldots, \nu_k\} \geq \frac{\Delta_f}{\varepsilon} \log(k/\alpha) \right) \\
&\leq \mathbb{P}\left(\{\nu_1 \geq \frac{\Delta_f}{\varepsilon} \log(k/\alpha)\} \cup \ldots \cup \{\nu_k \geq \frac{\Delta_f}{\varepsilon} \log(k/\alpha)\} \right) \\
\text{(Union bound)} \quad &\leq \sum_{i=1}^{k} \mathbb{P}\left(\nu_i \geq \frac{\Delta_f}{\varepsilon} \log(k/\alpha) \right) \\
\text{(By identical distribution)} &= k\mathbb{P}\left(\nu_1 \geq \frac{\Delta_f}{\varepsilon} \log(k/\alpha) \right) \\
&= k(\alpha/k) = \alpha.
\end{aligned}$$

□

Exercise 7. Check that $\mathbb{P}\left(\nu_1 \geq \frac{\Delta_f}{\varepsilon}\log(k/\alpha)\right) = \alpha/k$.

Example 2.2 (Firstnames)**.** We have a database of census entries of the USA with 100 million entires. We are told that there are 10000 distinct firstnames. We query the database for the number of individuals with each firstname (a vector of 10000 entries), while guaranteeing $(1, 0)$-differential privacy. What is the error guarantee we expect if we use Laplace mechanism? Observe that $\Delta_f = 2$ when we change one firstname to another among the 10000 firstnames. The Laplace mechanism gives us as output $M_\varepsilon^L(x, f)$ a random vector with $k = 10000$ elements (one for each possible firstname). The error theorem gives us:

$$\mathbb{P}(\left\|M_\varepsilon^L(x, f) - f(x)\right\|_\infty \geq \log(10000/\alpha)) \leq \alpha$$

for all $\alpha > 0$. For instance, if $\alpha = 0.01$, then

$$\mathbb{P}(\left\|M_\varepsilon^L(x, f) - f(x)\right\|_\infty \geq 27.6) \leq 0.01.$$

In other words, for every one of the 10000 firstnames in the census, the output of the Laplace mechanism is within 27.6 of the true firstname count with probability at least 0.99. This doesn't depend on the number of entries in the database! Consider another g which asks only how many people in the database have the name "Albert," then observe that $\Delta_g = 1$. Suppose that attacker hacks the database, knows all rows expect yours, can the attacker deduce your name? How many samples do you need to distinguish between two Laplace distributions with a small difference in their means $f(D)$ versus $f(D) - 1$?

2.1 Noisy Max-count mechanism

Consider the problem of counting k diseases in a patient record database. Each patient can have any subset of these diseases. Observe that the Laplace mechanism for k counting queries has $\Delta_f = k$ (worst case, one patient can have all k diseases), which shows up in the noise variance, and the error guarantee. If, however, we are only interested in the highest count among k counts, then the following mechanism gives a good balance of privacy guarantee and error guarantee.

Definition 2.4 (Noisy Max-count Mechanism)**.** Let $f : \mathbb{N}^d \to \mathbb{R}^k$ denote a vector of k counting queries $(f_1, \ldots, f_k)$. Let $\nu_1, \ldots, \nu_k$ denote i.i.d. Laplace random variables with zero mean and scale $1/\varepsilon$. The Noisy Max-count Mechanism

$$N_\varepsilon(x, f) \in \arg\max_{i=1,\ldots,k} f_i(x) + \nu_i.$$

Notice that the noise parameter $1/\varepsilon$ depend on k or Δ_f!

Theorem 2.3. *The Noisy Max-count Mechanism is $(\varepsilon, 0)$-differentially private.*

Proof in the textbook, section 3.3. See textbook for the error guarantee.

This mechanism can be used to find the most common medical condition in a database of patients where each patient may have multiple conditions.

3 Exponential mechanism for categorical data

What happens when you want to find out the most popular movie title in a Netflix database where each row contains the top 10 favorite movies of a customer. Unfortunately, we cannot add Laplace noise to movie titles. What happens when the query is not $f : \mathbb{N}^d \to \mathbb{R}^k$, but a function $f : \mathbb{N}^d \to \{1, \dots, m\}$? Or when the dataset x contains categorical data? In this case, the Laplace mechanism gives an invalid output. If the output of the query is a categorical set $\mathbb{F}$ (e.g., blood type, election winner, country of origin, favorite food), it is not clear what it means to "add" noise random variable.

Suppose that we construct a quality scoring function $u : \mathbb{N}^d \times \mathbb{O} \to \mathbb{R}$ that assigns to each database-output pair (x, o) a score that represents the correctness of the output $o \in \mathbb{O}$ for the database $x \in \mathbb{N}^d$. Let x^N denote the Netflix database of favorite movies. $\mathbb{O}$ is the set of all titles. Suppose that Titanic is the most favorited, Star Wars the second, etc. For example, we could have a scoring function with values:

$$\begin{aligned} u(x^N, \text{Titanic}) &= 100, \\ u(x^N, \text{Star Wars}) &= 95, \\ \dots u(x^N, \text{House of the Dead}) &= 0, \end{aligned}$$

Example 3.1. Note that k-nearest-neighbour classifiers can also give a scoring function when k is large enough. Linear classifiers also give a scoring function.

Definition 3.1 (Exponential Mechanism)**.** The exponential mechanism takes as input a database x, a parameter $\varepsilon > 0$, a set $\mathbb{O}$, and a scoring function u. It computes $\Delta_u = \max_{w \in \mathbb{O}, x, y : \|x-y\|_0 \leq 1} |u(x, w) - u(y, w)|$, and then outputs a $\mathbb{O}$-valued random variable $M_\varepsilon^E(x, u)$ with distribution:

$$\mathbb{P}(M_\varepsilon^E(x, u) = o) = \frac{\exp(\frac{u(x,o)}{2\Delta_u/\varepsilon})}{\sum_{w \in \mathbb{O}} \exp(\frac{u(x,w)}{2\Delta_u/\varepsilon})}, \quad o \in \mathbb{O}.$$

Example 3.2. What is a good scoring function for an election database, for the query "who won"? The score should be proportional to the number of votes. What is Δ_u?

Exercise 8. Check that $\sum_{o \in \mathbb{O}} \mathbb{P}(M_\varepsilon^E(x, u) = o) = 1$.

Exercise 9. Consider a computer that takes one time unit to perform addition, multiplication, and exponentiation. Assume that generating one random variable takes m time units. Compare the computational cost and the storage cost of running the Laplace and exponential mechanisms on a database $x \in \mathbb{Z}_+^{n \times 2}$ of the same size containing a bank's customers' account balances, and on a query of the highest balance versus the customer with the highest balance. How do computers generate random variables? Given a uniformly distributed random variable, how do we generate another random variable with an arbitrary distribution F?

Similar to the Laplace mechanism, the exponential mechanism is $(\varepsilon, 0)$-differentially private. The proof is similar. The main difference is the Δ term.

There is also a similar guarantee on how likely the output of the exponential mechanism is similar to the true query answer:

$$\arg\max_{o \in \mathbb{O}} u(x, o).$$

However this notion of similarity is determined by the scoring function u.

Example 3.3. Consider an election example with two candidates A, B. One example of score $u(D, A)$ is the fraction of votes who are satisfied when candidate A wins.

Theorem 3.1 (Accuracy of Exponential Mechanism)**.** *For every $\lambda > 0$, we have*

$$\mathbb{P}\left(\max_{o \in \mathbb{O}} u(x, o) - u(x, M_\varepsilon^E(x, u)) \geq \frac{2\Delta_u}{\varepsilon}(\log|\mathbb{O}| + \lambda)\right) \leq e^{-\lambda}.$$

Observe that the choice of value for λ gives rise to a tradeoff between the two constant terms above.

Remark 2. There is a nice interpretation of the theorem on accuracy. Plot the probability mass function for the random output $M_\varepsilon(x, u)$. Plot the probability mass function for the random output $u(x, M_\varepsilon(x, u))$. The exponential bound of the theorem bounds the tail probability of this PMF.

A challenge for this mechanism is when d is large and $\mathbb{O}$ is a large set.

Example 3.4. Consider a database of favorite movies with $n = 4$ customers, their favorite movies are Forest Gump, Revenant, Forest Gump, and Harry Potter. The set $\mathbb{O}$ contains 1000 different movies. Suppose that the score function is simply the count $u(D, o) = \sum_i 1_{[x_i = o]}$, such that $u(D, FG) = 2$, $u(D, HP) = u(D, Rev) = 1$, and $u(D, o) = 0$ for other movies o. What is the probability of outputing the true answer $\mathbb{P}(M(D, u) = FG)$? If one customer uses plausible deniability to claim to that his favorite is Revenant, when in reality it is Forest Gump, what is the probability $\mathbb{P}(M(D', u) = FG)$ associated with the hypothetical alternative database D'? What if we change the scoring function to $\tilde{u}(D, o) = 50 \sum_i 1_{[x_i = o]}$?

Exercise 10. Suppose that we have a database D with all votes in an election. How do we query for the candidate with the fewest votes? What scoring function u can we use such that $\arg\max u(D, o)$ gives the answer?

Example 3.5. A database x contains patient records relative to two medical conditions A and B. In other words, $\mathbb{O} = \{A, B\}$. The query is: which of these conditions is more common? Suppose that in x, there are $a = 0$ patients with A, and $b > 0$ patients with B. The quality score is

$$u(x, B) = b,$$
$$u(x, A) = a = 0.$$

Moreover, $\Delta_u = 1$. By the above theorem, we have

$$\mathbb{P}(b - u(x, M_\varepsilon^E(x,u)) \geq (2/\varepsilon)(\log 2 + \lambda)) \leq e^{-\lambda}.$$

Choose λ as large as possible (while still keeping the probability non-trivial: $(2/\varepsilon)(\log 2 + \lambda) \leq b$, i.e., $\lambda = \frac{b\varepsilon}{2} - \log 2$, we get

$$\mathbb{P}(b - u(x, M_\varepsilon^E(x,u)) \geq b) \leq e^{-\frac{b\varepsilon}{2} + \log 2} = 2e^{-b\varepsilon/2}.$$

The score for outputting the correct answer is b, whereas the score for outputting the wrong answer is 0. Hence, the probability of the exponential mechanism outputting the wrong answer A is at most $2e^{-b\varepsilon/2}$.

Exercise 11. Repeat the above example for the case where the true numbers of patients are $a > 0$ and $b > 0$.

What about privacy mechanisms for nonnegative data?

4 Concatenation of mechanisms, multiple queries

The Laplace mechanism is private if we do not abuse it: if we make repeated queries, and average the outputs, then the Central Limit Theorem kills the privacy.

What if we concatenate the outputs of two mechanisms? How much privacy do we lose?

Theorem 4.1. *Let $f_1 : \mathbb{N}^d \to \mathcal{X}_1$ and $f_2 : \mathbb{N}^d \to \mathcal{X}_2$ denote two queries. Let M_1 denote an $(\varepsilon_1, 0)$-differentially private mechanism for f_1. Let M_2 denote an $(\varepsilon_2, 0)$-differentially private mechanism for f_2. The randomization used in the two mechanisms is independent: the outputs of the two mechanisms are independent random variables. The joint mechanism (M_1, M_2) is $(\varepsilon_1 + \varepsilon_2, 0)$-differentially private.*

Proof. Let x, y denote two databases such that $\|x - y\|_0 \leq 1$. Consider the case where $\mathcal{X}_1$ and $\mathcal{X}_2$ are finite sets. Observe that for every (o_1, o_2),

$$\begin{aligned}
& \mathbb{P}((M_1(x, f_1), M_2(x, f_2)) = (o_1, o_2)) \\
\text{(Indep. noise RVs)} \quad & = \mathbb{P}(M_1(x, f_1) = o_1)\mathbb{P}(M_2(x, f_2) = o_2) \\
\text{(By definition)} \quad & \leq e^{\varepsilon_1}\mathbb{P}(M_1(y, f_1) = o_1)e^{\varepsilon_2}\mathbb{P}(M_2(y, f_2) = o_2) \\
& = e^{\varepsilon_1 + \varepsilon_2}\mathbb{P}((M_1(y, f_1), M_2(y, f_2)) = (o_1, o_2)).
\end{aligned}$$

□

Exercise 12. Repeat the proof above with $\mathbb{P}((M_1(x, f_1), M_2(x, f_2)) \in S_1 \times S_2)$ for arbitrary $S_1 \subseteq \mathcal{X}_1$ and $S_2 \subseteq \mathcal{X}_2$.

There are more interesting compositions of queries discussed in the literature, but outside the scope of this course. These queries involve different subsets of a complete database, and queries that adapt to answers to previous queries.

Exercise 13. How does concatenation affect the accuracy? What is the accuracy guarantee for the vector query $f = (f_1, f_2)$ and concatenation (M_1, M_2) of mechanism outputs.

5 Gaussian mechanism

Consider an alternative mechanism that employs normally distributed noise instead of Laplace noise. Let $\nu_1, \ldots, \nu_k$ be i.i.d. normal random variables with zero mean, and variance σ^2.

Let $\|z\|_2 = (|z_1|^2 + \ldots + |z_k|^2)^{1/2}$. Let

$$\tilde{\Delta}_f = \max_{x,y:\|x-y\|_0=1} \|f(x) - f(y)\|_2 .$$

The Gaussian mechanism with variance

$$\sigma^2 = 2\log(1.25/\delta)\tilde{\Delta}_f^2/\varepsilon^2$$

is (ε, δ)-differentially private.

Exercise 14. Plot the variance σ^2 as a function of ε (keeping $\delta = 0.1$), and plot σ^2 as a function of δ (keeping $\varepsilon = 0.1$).

What makes normal noise easier to analyze? Sum of normal RVs is normal.

Exercise 15. Consider two Gaussian mechanisms M_1 with query f_1 and M_2 with query f_2. Suppose that f_1 and f_2 are linear functions. Fix a database x, show that there exists a third Gaussian mechanism M_3 such that $M_2(M_1(x))$ is equal to $M_3(x)$. What is the relation between the variances of these mechanisms?

6 Answering large numbers of queries

Answer queries as a whole instead of individually. Consider two queries: How many people in the database have condition A? How many people in the database have condition A, excluding individual X? Such combinations of questions are a form of privacy attack. By answering these questions together, we can avoid privacy loss due to combinations of queries.

Suppose that X has condition A, then the outputs to the two queries through a Laplace mechanism are:

$$f(x) + \nu_1$$
$$f(x) - 1 + \nu_2.$$

Note that ν_1 and ν_2 are i.i.d. zero-mean. The question becomes, can we distinguish to some extent whether two random variables arise from two distinct distributions?

Exercise 16 (Bonus)**.** Consider the case where $f(x) = 1$, $\Delta_f = 1$, and $\varepsilon = 0.1$. For fixed a and b, what is the probability of observing $f(x) + \nu_1 = a$ and $f(x) - 1 + \nu_2$? You can evaluate it by integrating or by plotting on a computer.

If we had multiple samples of these random variables, we can make the inference with even higher confidence. Hence, we need a different approach than i.i.d. noise for large numbers of queries.

6.1 General release mechanism

One approach is to answer a query f by generating a database $\hat{D}(f)$, and then answering $f(\hat{D}(f))$. If $\hat{D}$ is generated in a deterministic manner, then an attacker cannot exploit the query mechanism by querying multiple times: the answer will always be the same.

This section is based on "A learning theory approach to noninteractive database privacy" by Blum, Ligett, and Roth, and uses the notion of VC dimension, which motivates the second part of this course on supervised learning.

Consider a database $x \in \mathbb{N}^d$ with n rows, and normalized counting queries of the form

$$Q(x) = \frac{1}{n} \sum_{i=1}^{n} 1_{[g(x_i)=1]},$$

where g checks a property.

In this section, we consider a new type of mechanism A. This mechanism restricts the number of possible queries to those of a set C. This mechanism answers query $Q \in C$ by returning a smaller version x' of x. The query is then answered by computing $Q(x')$.

Definition 6.1 (Usefulness)**.** The mechanism A is (α, δ)-useful with respect to a class of queries C if for every database $x \in \mathbb{N}^d$,

$$\mathbb{P}\left(\max_{Q \in C} |Q(A(x)) - Q(x)| \leq \alpha\right) \geq 1 - \delta.$$

Definition 6.2. Given a class of queries C, a set of databases $N_\alpha(C)$ is a minimal set such that for every $x \in \mathbb{N}^d$, there exists $x' \in N_\alpha(C)$ such that

$$\max_{Q \in C} |Q(x') - Q(x)| \leq \alpha.$$

The Net mechanism builds upon the exponential mechanism:

1. Input: database x, ε, class of queries C, α
2. Set $\mathbb{O} = N_{\alpha/2}(C)$ (a set of databases).
3. Set quality score function:

$$u(x, x') = -\max_{Q \in C} |Q(x) - Q(x')|.$$

4. Output $M_\varepsilon^E(x, u)$ (exponential mechanism with score function u), which is a database in $\mathbb{O}$.

As before, we have a privacy guarantee and an error guarantee:

- Net mechanism is $(\varepsilon, 0)$-differentially private.

- For any class of counting queries C, the Net mechanism is (α, δ)-useful if

$$\alpha \geq \frac{4}{\varepsilon n} \log(|N_\alpha(C)|/\delta).$$

  Moreover, $|N_\alpha(C)| \leq d^{\log|C|/\alpha^2}$.

As a consequence, if

$$\alpha \geq \frac{4}{\varepsilon n} \log(d^{\log|C|/\alpha^2}/\delta) = \frac{4\log|C|}{\varepsilon n \alpha^2} \log(d/\delta),$$

then the Net mechanism is (α, δ)-useful.

6.2 How to construct $N_{\alpha/2}(C)$

7 Supervised Learning

Two random variables Z_1 and Z_2 are independent if for every pair of intervals[2] on the real line S_1, S_2, we have

$$\mathbb{P}(Z_1 \in S_1, Z_2 \in S_2) = \mathbb{P}(Z_1 \in S_1) \cdot \mathbb{P}(Z_2 \in S_2).$$

Conditional probability is informally defined as

$$\mathbb{P}(Z_1 \in S_1 \mid Z_2 \in S_2) = \frac{\mathbb{P}(Z_1 \in S_1, Z_2 \in S_2)}{\mathbb{P}(Z_2 \in S_2)},$$

as long as the event $\{Z_2 \in S_2\}$ has non-zero probability. Let $x_1, \ldots, y_n$ denote a possible realization of the random variables $X_1, \ldots, Y_n$, then

$$\mathbb{P}(g_n(X_{n+1}) \neq Y_{n+1} \mid x_1, \ldots, y_n) = \mathbb{P}(g_n(X_{n+1}) \neq Y_{n+1} \mid X_1 = x_1, \ldots, Y_n = y_n).$$

In turn, $\mathbb{P}(g_n(X_{n+1}) \neq Y_{n+1} \mid X_1, \ldots, Y_n)$ is a random variable corresponding to the random realizations.

Consider the sequence of random variables

$$L(g_1), L(g_2), \ldots$$

If the expected values of this sequence converges to $L(g^*)$, i.e.,

$$\mathbb{E}L(g_1), \mathbb{E}L(g_2), \ldots$$

converges to $L(g^*)$, then this sequence of classifiers is consistent.

[2] An interval is a set of points $[a, b]$ or (a, b) or $[a, b)$ or $(a, b]$, with $a < b$.

7.1 Empirical performance

The empirical performance of a classifier g on a dataset $(X_1, Y_1), \ldots, (X_n, Y_n)$ is the rate of error (empirical frequency of errors):

$$L_n(g) = \frac{1_{[g(X_1)\neq Y_1]} + \ldots + 1_{[g(X_n)\neq Y_n]}}{n}.$$

For example, (X_j, Y_j) can be the medical record of patient j and his or her coronavirus diagnosis. Suppose that we have a set $\mathcal{C}$ of classifiers (e.g., doctors in Canada who classify patients according to coronavirus infection). We can use the rate of error to determine the best classifier in the set $\mathcal{C}$. The rate of error as a random variable has a nice property. Since (X_1, Y_1) is independent of and identically distributed to (X_2, Y_2), and so on, the random variables $1_{[g(X_1)\neq Y_1]}, \ldots, 1_{[g(X_n)\neq Y_n]}$ are independent and identically distributed as well.

Exercise 17. Check that

$$\mathbb{P}(1_{[g(X_1)\neq Y_1]} = 0, 1_{[g(X_2)\neq Y_2]} = 0) = \mathbb{P}(1_{[g(X_1)\neq Y_1]} = 0)\mathbb{P}(1_{[g(X_2)\neq Y_2]} = 0).$$

The Central Limit Theorem says that for large n, the rate of error is approximately a normal (Gaussian) random variable. Hence, there is only very small (exponentially small) probability that $L_n(g)$ takes values away from the expectation

$$\mathbb{E}1_{[g(X_1)\neq Y_1]} = \mathbb{P}(g(X_1) \neq Y_1).$$

Moreover, by the Law of Large Numbers, the rate of error $L_n(g)$ converges to this expectation.

7.2 Nearest Neighbour Classifiers

Given the training data $(X_1, Y_1), \ldots, (X_n, Y_n)$, we can compute distances between a new observation X_{n+1} and the n previous observations:

$$|X_{n+1} - X_1|, \ldots, |X_{n+1} - X_n|.$$

The nearest neighbour to X_{n+1} is the X_i that minimizes the above distance. We can classify X_{n+1} with the same label as X_i, which is Y_i.

Likewise, we can find the k-nearest neighbours, that have the k smallest distances. These neighbours have corresponding labels $Y_{i_1}, \ldots, Y_{i_k}$. What label should be give to X_{n+1}? It is natural to give a label corresponding to the majority label among $Y_{i_1}, \ldots, Y_{i_k}$.

When the size of the training dataset n is small, we can start by choosing k small, and as n increases, it is a good idea to increase k as well.

8 Maximum Likelihood Classifiers

Suppose that we reorder and split the samples $(X_1, Y_1), \ldots, (X_n, Y_n)$ into two: $\tilde{X}_1, \tilde{Y}_1, \ldots, \tilde{X}_m, \tilde{Y}_m$, where $\tilde{Y}_1 = \tilde{Y}_2 = \ldots = \tilde{Y}_m = 0$, and $\tilde{X}_{m+1}, \tilde{Y}_{m+1}, \ldots, \tilde{X}_n, \tilde{Y}_n$, where $\tilde{Y}_{m+1} = \ldots = \tilde{Y}_n = 1$. If we do not know the joint distribution of $\tilde{X}_1, \tilde{Y}_1$, nor the marginal distribution f_0 of $\tilde{X}_1$, can we find f_0 from the samples? Suppose that we know that f_0 takes a parametric form f_0^θ, one approach is to find:

$$\hat{f}_0 = \arg\max_\theta f_0^\theta(\tilde{X}_1) \cdot \ldots \cdot f_0^\theta(\tilde{X}_m).$$

Example 8.1. Consider the case where $m = 1$ and $m = 2$, and where f_0^θ is normal with $\sigma^2 = 1$ and unknown mean θ. Given $\tilde{X}_1$ and $\tilde{X}_2$, plot $f_0^\theta(\tilde{X}_1)$ and $f_0^\theta(\tilde{X}_1) \cdot f_0^\theta(\tilde{X}_2)$, and find the value of θ that maximizes these values.

Observe that f_0 is the conditional distribution of X_1 given that $Y_1 = 0$, then, we can write the joint distribution of X_1, Y_1 as:

$$f_y(x)\mathbb{P}(Y_1 = y), \quad y \in \{0, 1\}.$$

Each of the components f_0, $\mathbb{P}(Y_i = 0)$, and $\mathbb{P}(Y_i = 1)$ can be estimated separately from the data.

Since we assume that the sequence $(X_1, Y_1), \ldots, (X_n, Y_n)$ is i.i.d., then the joint distribution is obtained by taking the product of the above $f_y(x)\mathbb{P}(Y_1 = y)$, evaluated at the data points, or:

$$\prod_{i=1}^{n} [f_0(X_i)\mathbb{P}(Y_i = 0)]^{Y_i} [f_1(X_i)\mathbb{P}(Y_i = 1)]^{1-Y_i}.$$

The maximum likelihood approach maximizes this product for f_0^*, f_1^* and $\mathbb{P}(Y_i = 0) = p^*$. From this solution, and a new datapoint X_{n+1}, we give the label $g(X_{n+1}) = y$ corresponding to the highest probability:

$$g(X_{n+1}) = \arg\max_{y=0,1} f_y(X_{n+1})\mathbb{P}(Y_1 = y).$$

8.1 Neural nets

For a fixed neural network architecture (number of hidden layers, choice of sigmoid function), we can construct a class of neural network classifiers $\mathcal{C}$. Then, the training of a neural network classifier is simply the search over $\mathcal{C}$ for the classifier with the lowest rate of error on the training dataset.

Exercise 18. Consider a neural network g_n that is already trained. Suppose that it takes an input observation in $\mathbb{R}^d$, and that it has m hidden layers with k neurons in each hidden layer. The sigmoid function is fixed and given. How many additions, multiplications, and applications of the sigmoid function does it take to output $g_n(X_{n+1})$? How much memory approximately does it take to run if you implement it in C and all variables are stored as 'double' variables?

9 VC theory, regression, clustering

Boxer analogy: estimation error, approximation error, n is training duration.

10 Private learning algorithms