6.889 New Developments in Cryptography May 3, 2011

Lecture 11- Differential Privacy

Lecturer: Salil Vadhan *Scribes: Alan Deckelbaum and Emily Shen*

1 Introduction

In class today (and the next two lectures) we will discuss differential privacy. The plan is as follows:

- Motivation
- Definition
- Examples
- Composition
- Other definitions
- More examples
- Friday (Jon): Many queries
- Tuesday: Connections to cryptography (and finish the Bayesian definition of differential privacy)

A good reference for differential privacy is a CACM survey by Dwork.

2 Motivation

The motivation for differential privacy is to be able to compute and release information about a sensitive database. For example, consider a database with rows $x_1, \ldots, x_n$. Each row corresponds to an individual in the database, and the columns correspond to fields. Some fields, such as "name" and "social security number," contain identifying information. Other fields contain sensitive information such as medical history or census data. We would like to be able to do computation on the sensitive information without revealing personal information about any individual.

The traditional approach is to simply remove from the database any information which obviously identifies an individual. For example, we would remove the "name" and "social security number" fields from the database. This approach isn't always a good idea. Even if the remaining fields don't provide enough information on their own to identify an individual, together they might give too much information when combined.

For example, in the 1990's, insurance databases were made public that removed the individual names but retained information such as birth date. Latanya Sweeney cross-referenced the insurance database with publicly available information from voter registration records in order to identify the medical record of the governor or Massachusetts.

2.1 k-anonymity

The solution to this problem seemed to be to suppress and generalize information. (For example, we could change the "date of birth" to "year of birth," remove the last digit of the zip code, etc.) The goal of this approach would be to achieve k-anonymity- when we restrict our focus to the "identifying columns," every row occurs at least k times. This approach ensures that any cross-referencing with external data won't isolate any individual to a group of size smaller than k.

Unfortunately, k-anonymity by itself does not provide privacy. Consider the function M in the following example, and suppose that M provides k-anonymity:

$$M(x) = \begin{cases} \text{suppress} & \text{if Jon likes Broadway musicals} \\ \text{generalize} & \text{otherwise.} \end{cases}$$

Even if M provides k-anonymity, it does not provide privacy. By looking at the output of M, we can determine personal information about Jon. The problem with M is that, while it provides k-anonymity, it chooses which operations to perform based on the contents of the database.

2.2 Interactive Queries

Consider the model where we have a trusted mediator M which receives queries from a user. The mediator can communicate with the database x, and then returns an answer to the user. Consider the following two queries:

q_1 = "How many people in the database like Broadway musicals?"
q_2 = "How many people in the database other than Jon like Broadway musicals?"

While the questions q_1 and q_2 seem private when taken in isolation, the composition of answers to both questions is non-private, since it reveals information about Jon's preferences.

We will attempt to resolve this issue by giving the trusted mediator M an additional random input, and allowing M to return "noisy" answers to the user.

3 Defining Differential Privacy

We have the following two desiderata in our definition of differential privacy:

1. Strong notion of privacy
2. Utility- "noisy" answers are still useful.

We now have the following definition.

Definition 1. *$M : \mathcal{X}^n \times \mathcal{Q} \to \mathcal{Y}$ is **differentially private** iff for all $q \in \mathcal{Q}$ and for all $x, x' \in \mathcal{X}$ differing only on a single row, the distributions $M(x, q)$ and $M(x', q)$ are "similar."*

In the above definition, $\mathcal{X}$ is the space from which the rows come, $\mathcal{Q}$ is the space of questions, and $\mathcal{Y}$ is the output space. The definition captures the idea that no individual's data has a significant influence on the output. Note that the "similarity" of distributions depends only on the coin tosses of the mechanism, and we need not worry about what supplemental information is available to the user.

We say that a mechanism is **ϵ-differentially private** if the distributions $M(x, q)$ and $M(x', q)$ have "distance" at most ϵ. We now must choose what to use for the notion of "distance" of two distributions.

3.1 Statistical Difference

A first attempt for our definition is to use statistical difference, where the statistical difference between distributions A and B is defined to be

$$\max_{T \subset Y} |Pr[A \in T] - Pr[B \in T]|.$$

Unfortunately, this is a bad choice for differential privacy. Consider the following two cases:

- Case 1: $\epsilon \leq \frac{1}{10n}$. In this case, we can use a hybrid argument by changing one entry at a time to show that, with probability at least 90%, we gain no useful information from our query. In particular, the statistical difference between $M(x, q)$ and $M(0^n, q)$ is less than 0.1.

- Case 2: $\epsilon \geq \frac{1}{10n}$. Consider a mechanism which releases a random row from the database with probability $\frac{1}{10}$. This mechanism satisfies the statistical difference constraint, but intuitively shouldn't be allowed. (Alternatively, we could look at a mechanism which releases the first row with probability $\frac{1}{10n}$.)

3.2 Actual Choice of Distance Between Distributions

The condition we choose to use for distance between probability distributions A and B being at most ϵ is, for all $T \subset \mathcal{Y}$,

$$Pr[A \in T] \leq e^{\epsilon} Pr[B \in T].$$

We will think of ϵ being in the range $\frac{1}{n} \leq \epsilon \leq 1$. (We still cannot think of ϵ being arbitrarily small, due to critiques similar to case 1 above.) We note that $e^{\epsilon} \approx 1 + \epsilon$, and that the above constraint implies that the statistical difference between distributions is $O(\epsilon)$. This is in fact a stronger condition than merely bounding statistical difference. In particular, the bad mechanisms from case 2 above are eliminated. This definition is more useful for rare events (events which have probability less than ϵ) than the statistical difference definition.

4 Examples

We consider the problem of counting queries. In particular, we are given $q : \mathcal{X} \to \{0, 1\}$, and define $q(x_1, \ldots, x_n) = \sum_{i=1}^{n} q(x_i)$. We consider the mechanism

$$M(x, q) = q(x) + Lap(1/\epsilon)$$

where $Lap(\lambda)$ is the Laplace distribution on $\mathbb{R}$, which has density $h_\lambda(y) \propto e^{-|y|/\lambda}$. (This density function peaks at 0 and decays exponentially on both sides, where the rate of decay depends on λ.)

Claim 1. *M is ϵ-differentially private*

Proof: Let x, x' be neighboring databases. We observe that the difference between $q(x)$ and $q(x')$ is either 0 or ± 1. For all y, we know that the density of $[M(q, x) = y]$ is proportional to $e^{-\epsilon|y-q(x)|}$ while the density of $[M(q, x') = y]$ is proportional to $e^{-\epsilon|y-q(x')|}$. Since the exponents differ by at most ϵ, we see that the densities are within a factor of e^{ϵ}, as desired. □

In general, for $q : \mathcal{X}^n \to \mathbb{R}$,

$$M(x, q) = q(x) + Lap(GS_q/\epsilon)$$

is ϵ-differentially private, where GS_q is the "global sensitivity" of q, defined by

$$GS_q = \max_{\text{neighboring } x,x'} |q(x) - q(x')|.$$

For $q : \mathcal{X}^n \to \mathbb{R}^d$, then

$$M(x, q) = q(x) + Lap(GS_q/\epsilon)^d$$

is ϵ-differentially private, where

$$GS_q = \max_{\text{neighboring } x,x'} |q(x) - q(x')|_1.$$

and the notation $Lap(GS_q/\epsilon)^d$ means that we do d independent samples.

For example, consider $q(x)$ as a histogram into d buckets. In this case, we have $GS_q = 2$, since changing one row will change the value of at most 2 buckets.

4.1 Choice of ϵ

We should think of ϵ as $1/n \leq \epsilon \leq 1$. For example, for $M(x, q) = q(x) + Lap(1/\epsilon)$, if ϵ is a constant, we get $O(1)$ error. If $\epsilon = 100/n$, we get error about $0.01n$. (Compare this to the statistical database sampling error $O(\sqrt{n})$.)

5 Max-Divergence

We define the max-divergence of A and B by

$$D_\infty(A\|B) = max_{T\subset\mathcal{Y}} \ln\left(\frac{Pr[A\in T]}{Pr[B\in T]}\right) = max_{y\in\mathcal{Y}} \ln\left(\frac{Pr[A=y]}{Pr[B=y]}\right).$$

A mechanism M is ϵ-differentially private iff for all $q\in\mathcal{Q}$, for all $x, x'\in\mathcal{X}^n$ differing on 1 row,

$$D_\infty(M(x,q)\|M(x',q)) \le \epsilon.$$

Compare this with the KL Divergence

$$D(A\|B) = \mathbb{E}_{y\in A}\left[\ln\left(\frac{Pr[A=y]}{Pr[B=y]}\right)\right].$$

Max-divergence is a worst-case analog of KL divergence, similar to the way that min-entropy is a worst-case analog of Shannon entropy.

6 Composition

We now consider the composition of a differentially private mechanism.

Definition 2. *Given $M:\mathcal{X}^n\times\mathcal{Q}\to\mathcal{Y}$, the* t-fold composition $M^t:\mathcal{X}^n\times\mathcal{Q}^t\to\mathcal{Y}^t$ *is defined by*

$$M^t(x,q_1,\ldots,q_t) = (M(x,q_1),\ldots,M(x,q_t)),$$

where the randomness is independent for each query.

Claim 2. *If M is ϵ-differentially private, then M^t is $t\epsilon$-differentially private.*

Proof: If $A_1,\ldots,A_t$ are independent and $B_1,\ldots,B_t$ are independent, then

$$D_\infty(A_1,\ldots,A_t\|B_1,\ldots,B_t) = \Sigma_{i=1}^t D_\infty(A_i\|B_i).$$

□

If we want $t\epsilon\le 1$, then our initial ϵ should be $\le 1/t$. This means the noise is $\approx t$ for each query.

The composition theorem tells us that we can answer $o(\sqrt{n})$ queries with noise much less than the sampling error, and we can answer $o(n)$ queries with nontrivial noise.

However, we can get better composition with a slightly different definition called (ϵ,δ)-*differential privacy*, where the distance condition is defined by

$$Pr[A\in T] \le e^\epsilon Pr[B\in T] + \delta.$$

(We can think of $\delta = 2^{-k}$ where k a security parameter (e.g., 50 or 100).)

Claim 3. *If M is (ϵ,δ)-differentially private, then M^t is $\approx(\sqrt{t}\epsilon\log(1/\delta'), t\delta+\delta')$-differentially private, for any δ'.*

Now we can make $t = o(n/k^2)$ queries with noise $o(\sqrt{n})$. This is nearly optimal, as Dinur-Nissim have shown that with $\tilde{O}(n)$ counting queries and error $o(\sqrt{n})$ one can recover most of the database.

7 Other Definitions

We now consider other equivalent definitions of differential privacy.

7.1 Simulation-Based Definition

As in cryptography, we can consider a simulation-based definition.

Definition 3. *A polynomial-time mechanism M is* ϵ-simulation-differentially private *if there exists a polynomial-time simulator S s.t. $\forall q \in \mathcal{Q}, x \in \mathcal{X}^n, i \in [n]$,*

$$D_\infty(M(q,x) \| S(q, x_{-i})) \leq \epsilon$$

and

$$D_\infty(S(q,x_{-i}) \| M(q,x)) \leq \epsilon,$$

where x_{-i} denotes x without the ith row.

(The simulator will work by putting anything (e.g., all zeroes) for the ith row and running M. Note that whatever can be learned about an individual from the other rows is not protected.)

Claim 4. *If M is ϵ-simulation-differentially private, then M is 2ϵ-differentially private.*

7.2 Bayesian Definition

We will see this next time. The Bayesian definition captures how beliefs about an individual in the database change after seeing the output of a query.

8 Remarks

Remark 1. *The composition results we have seen say roughly that we can answer $\approx n$ counting queries. This allows implementation of statistical query learning.*

Note that if M is ϵ-differentially private, then $\forall f, f \circ M$ is ϵ-differentially private. Similarly for (ϵ, δ)-differential privacy. This means we can produce differentially private discrete objects.

For example, we can implement learning algorithms such as singular value decomposition (SVD), principle component analysis (PCA), and decision trees.

Remark 2. *The composition theorems also hold for adaptive queries, but these are more complicated.*