Differential Privacy and its Application in Aggregation

Part 1 — Differential Privacy

presenter: Le Chen

Nanyang Technological University

lechen0213@gmail.com

October 5, 2013

Outline

Reference

- Cynthia Dwork.
  Differential Privacy.
  Invited talk at ICALP, Venice, Italy, July 10-14, 2006.
  Automata, Languages and Programming, Lecture Notes in Computer Science Volume 4052, 2006, pp 1-12.
- Wikipedia.
  <http://en.wikipedia.org/wiki/Differential_privacy>
- Elaine Shi, T-H. Hubert Chan, Eleanor Rieffel, Richard Chow and Dawn Song.
  Privacy-Preserving Aggregation of Time-Series Data.
  In Network and Distributed System Security Symposium (NDSS), 2011.

Reference

T-H. Hubert Chan, Elaine Shi, and Dawn Song
Privacy-Preserving Stream Aggregation with Fault Tolerance.
16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012. Financial Cryptography and Data Security, Lecture Notes in Computer Science Volume 7397, 2012, pp 200-214.

Motivation

- Consider a trusted party that holds a dataset of sensitive information (e.g. medical records, voter registration information, email usage) with the goal of providing global, statistical information about the data publicly available, while preserving the privacy of the users whose information the data set contains. Such a system is called a statistical database.
- The notion of indistinguishability, later termed Differential Privacy, formalizes the notion of "privacy" in statistical databases.

Motivation

- Consider a trusted party that holds a dataset of sensitive information (e.g. medical records, voter registration information, email usage) with the goal of providing global, statistical information about the data publicly available, while preserving the privacy of the users whose information the data set contains. Such a system is called a statistical database.
- The notion of indistinguishability, later termed Differential Privacy, formalizes the notion of "privacy" in statistical databases.

Definition

ϵ-differential privacy

- The actions of the trusted server are modeled via a randomized algorithm $\mathcal{A}$. A randomized algorithm $\mathcal{A}$ is ϵ-differentially private if for all datasets D_1 and D_2 that differ on a single element (i.e., data of one person), and all $S \subseteq \mathrm{Range}(\mathcal{A})$,

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S],$$

where the probability is taken over the coins of the algorithm and $\mathrm{Range}(\mathcal{A})$ denotes the output range of the algorithm $\mathcal{A}$.

Description

- ϵ is the privacy parameter, e.g. if $\epsilon = 0.1$ then $e^{\epsilon} \approx 1.105$, if $\epsilon = 0.5$ then $e^{\epsilon} \approx 1.649$.

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S]$$

$$\Pr[\mathcal{A}(D_2) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_1) \in S]$$

- This means that for any two datasets which are close to one another (that is, which differ on a single element) a given differentially private algorithm $\mathcal{A}$ will behave approximately the same on both data sets.
- The definition gives a strong guarantee that presence or absence of an individual will not affect the final output of the query significantly.

Description

- ϵ is the privacy parameter, e.g. if $\epsilon = 0.1$ then $e^\epsilon \approx 1.105$, if $\epsilon = 0.5$ then $e^\epsilon \approx 1.649$.

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^\epsilon \times \Pr[\mathcal{A}(D_2) \in S]$$

$$\Pr[\mathcal{A}(D_2) \in S] \leq e^\epsilon \times \Pr[\mathcal{A}(D_1) \in S]$$

- This means that for any two datasets which are close to one another (that is, which differ on a single element) a given differentially private algorithm $\mathcal{A}$ will behave approximately the same on both data sets.
- The definition gives a strong guarantee that presence or absence of an individual will not affect the final output of the query significantly.

Description

- ϵ is the privacy parameter, e.g. if $\epsilon = 0.1$ then $e^{\epsilon} \approx 1.105$, if $\epsilon = 0.5$ then $e^{\epsilon} \approx 1.649$.

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S]$$

$$\Pr[\mathcal{A}(D_2) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_1) \in S]$$

- This means that for any two datasets which are close to one another (that is, which differ on a single element) a given differentially private algorithm $\mathcal{A}$ will behave approximately the same on both data sets.
- The definition gives a strong guarantee that presence or absence of an individual will not affect the final output of the query significantly.

Diabetes example

- For example, assume we have a database of medical records D_1 where each record is a pair (**Name**,**X**), where **X**$\in \{0,1\}$ denotes whether a person has diabetes or not. For example:

Name	**Has Diabetes (X)**
Ross	1
Monica	1
Joey	0
Phoebe	0
Chandler	1

- Now suppose a malicious user (often termed an adversary) wants to find whether Chandler has diabetes or not.

Diabetes example

- For example, assume we have a database of medical records D_1 where each record is a pair (**Name**,**X**), where **X**$\in \{0, 1\}$ denotes whether a person has diabetes or not. For example:

Name	**Has Diabetes (X)**
Ross	1
Monica	1
Joey	0
Phoebe	0
Chandler	1

- Now suppose a malicious user (often termed an adversary) wants to find whether Chandler has diabetes or not.

What an adversary can do

- As a side information he knows in which row of the database Chandler resides.
- Now suppose the adversary is only allowed to use a particular form of query $Q(i)$ which returns the partial sum of first i rows of column $\mathbf{X}$ in the database.
- In order to find Chandler's diabetes status the adversary simply executes $Q(5) - Q(4)$.
- One striking feature this example highlights is: individual information can be compromised even without explicitly querying for the specific individual information.

What an adversary can do

- As a side information he knows in which row of the database Chandler resides.
- Now suppose the adversary is only allowed to use a particular form of query $Q(i)$ which returns the partial sum of first i rows of column $\mathbf{X}$ in the database.
- In order to find Chandler's diabetes status the adversary simply executes $Q(5) - Q(4)$.
- One striking feature this example highlights is: individual information can be compromised even without explicitly querying for the specific individual information.

What an adversary can do

- As a side information he knows in which row of the database Chandler resides.
- Now suppose the adversary is only allowed to use a particular form of query $Q(i)$ which returns the partial sum of first i rows of column $\mathbf{X}$ in the database.
- In order to find Chandler's diabetes status the adversary simply executes $Q(5) - Q(4)$.
- One striking feature this example highlights is: individual information can be compromised even without explicitly querying for the specific individual information.

What an adversary can do

- As a side information he knows in which row of the database Chandler resides.
- Now suppose the adversary is only allowed to use a particular form of query $Q(i)$ which returns the partial sum of first i rows of column $\mathbf{X}$ in the database.
- In order to find Chandler's diabetes status the adversary simply executes $Q(5) - Q(4)$.
- One striking feature this example highlights is: individual information can be compromised even without explicitly querying for the specific individual information.

Differential privacy in the example

- Now we construct D_2 by replacing (Chandler,1) with (Chandler,0). Let $\mathcal{A} = Q(i)$.
- We say $\mathcal{A}$ is ϵ-differentially private if it satisfies the definition:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S]$$

- S can be thought of as a singleton set (something like $\{3.5\}, \{4\}$ etc.) if the output function of $\mathcal{A}$ is a Discrete Random Variable (i.e. has a probability mass function (pmf)).
- S can also be thought to be a small range of reals (something like $3.5 \leq \mathcal{A}(D_1) \leq 3.7$) if it is a Continuous Random Variable (i.e. has a probability density function (pdf)).

Differential privacy in the example

- Now we construct D_2 by replacing (Chandler,1) with (Chandler,0). Let $\mathcal{A} = Q(i)$.
- We say $\mathcal{A}$ is ϵ-differentially private if it satisfies the definition:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S]$$

- S can be thought of as a singleton set (something like $\{3.5\}, \{4\}$ etc.) if the output function of $\mathcal{A}$ is a Discrete Random Variable (i.e. has a probability mass function (pmf)).
- S can also be thought to be a small range of reals (something like $3.5 \leq \mathcal{A}(D_1) \leq 3.7$) if it is a Continuous Random Variable (i.e. has a probability density function (pdf)).

Differential privacy in the example

- Now we construct D_2 by replacing (Chandler,1) with (Chandler,0). Let $\mathcal{A} = Q(i)$.
- We say $\mathcal{A}$ is ϵ-differentially private if it satisfies the definition:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S]$$

- S can be thought of as a singleton set (something like $\{3.5\}, \{4\}$ etc.) if the output function of $\mathcal{A}$ is a Discrete Random Variable (i.e. has a probability mass function (pmf)).
- S can also be thought to be a small range of reals (something like $3.5 \leq \mathcal{A}(D_1) \leq 3.7$) if it is a Continuous Random Variable (i.e. has a probability density function (pdf)).

Differential privacy in the example

- Now we construct D_2 by replacing (Chandler,1) with (Chandler,0). Let $\mathcal{A} = Q(i)$.
- We say $\mathcal{A}$ is ϵ-differentially private if it satisfies the definition:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S]$$

- S can be thought of as a singleton set (something like $\{3.5\}, \{4\}$ etc.) if the output function of $\mathcal{A}$ is a Discrete Random Variable (i.e. has a probability mass function (pmf)).
- S can also be thought to be a small range of reals (something like $3.5 \leq \mathcal{A}(D_1) \leq 3.7$) if it is a Continuous Random Variable (i.e. has a probability density function (pdf)).

Sensitivity

- The sensitivity (Δf) of a function $f : \mathcal{D} \rightarrow \mathbb{R}^d$ is

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1$$

for all D_1, D_2 differing in at most one element, and $D_1, D_2 \in \mathcal{D}$.

- Clearly, if we change one of the entries in the database then the output of the query $Q(i)$ will change by at most one. So, the sensitivity of the query $Q(i)$ is one.
- It so happens that there are techniques (which are described below) using which we can create a differentially private algorithm for functions with low sensitivity.

Sensitivity

- The sensitivity (Δf) of a function $f : \mathcal{D} \to \mathbb{R}^d$ is

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1$$

for all D_1, D_2 differing in at most one element, and $D_1, D_2 \in \mathcal{D}$.

- Clearly, if we change one of the entries in the database then the output of the query $Q(i)$ will change by at most one. So, the sensitivity of the query $Q(i)$ is one.
- It so happens that there are techniques (which are described below) using which we can create a differentially private algorithm for functions with low sensitivity.

Sensitivity

- The sensitivity (Δf) of a function $f : \mathcal{D} \rightarrow \mathbb{R}^d$ is

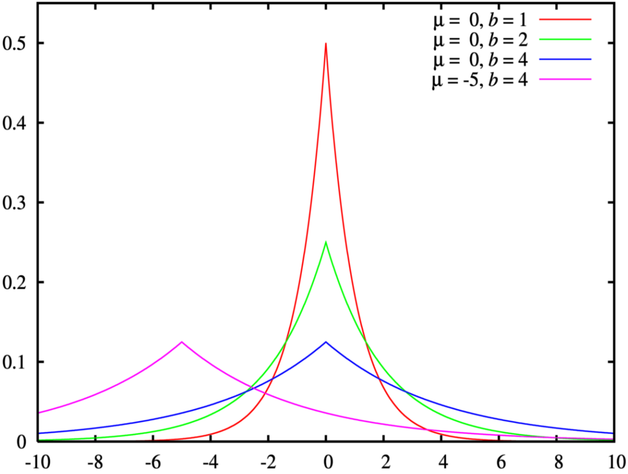
$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1$$

  for all D_1, D_2 differing in at most one element, and $D_1, D_2 \in \mathcal{D}$.
- Clearly, if we change one of the entries in the database then the output of the query $Q(i)$ will change by at most one. So, the sensitivity of the query $Q(i)$ is one.
- It so happens that there are techniques (which are described below) using which we can create a differentially private algorithm for functions with low sensitivity.

Laplace Distribution

- Laplace distribution is a continuous probability distribution named after Pierre-Simon Laplace.
- It is also sometimes called the *double exponential distribution*, because it can be thought of as two exponential distributions (with an additional location parameter) spliced together back-to-back.

Laplace Distribution



Laplace Distribution Probability Density Function

Laplace Distribution

- Probability density function (PDF):

$$f(x|\mu,\lambda) = \frac{1}{2\lambda} e^{\left(-\frac{|x-\mu|}{\lambda}\right)} = \frac{1}{2\lambda} \begin{cases} e^{\left(-\frac{\mu-x}{\lambda}\right)} & \text{if } x < \mu \\ e^{\left(-\frac{x-\mu}{\lambda}\right)} & \text{if } x \geq \mu \end{cases}$$

- Here, μ is a location parameter and $\lambda \leq 0$ is a scale parameter. If $\mu = 0$ and $\lambda = 1$, the positive half-line is exactly an exponential distribution scaled by $1/2$.
- PDF of exponential distribution:

$$f(x;\beta) = \begin{cases} \frac{1}{\beta} e^{-x/\beta}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

Laplace Noise

- Many differentially private algorithms rely on adding controlled noise to functions with low sensitivity.
- We will elaborate this point by taking a special kind of noise (whose kernel is a Laplace distribution i.e. the probability density function noise(y) $\propto e^{-|y|/\lambda}$, mean zero and standard deviation λ).
- Now in our case we define the output function of $\mathcal{A}$ as a real valued function (called as the transcript output by $\mathcal{A}$) $\mathcal{T}_{\mathcal{A}}(x) = f(x) + Y$, where $Y \sim \text{Lap}(\lambda)$ and f is the original real valued query/function we plan to execute on the database.

Laplace Noise

- Many differentially private algorithms rely on adding controlled noise to functions with low sensitivity.
- We will elaborate this point by taking a special kind of noise (whose kernel is a Laplace distribution i.e. the probability density function noise$(y) \propto e^{-|y|/\lambda}$, mean zero and standard deviation λ).
- Now in our case we define the output function of $\mathcal{A}$ as a real valued function (called as the transcript output by $\mathcal{A}$) $\mathcal{T}_{\mathcal{A}}(x) = f(x) + Y$, where $Y \sim \text{Lap}(\lambda)$ and f is the original real valued query/function we plan to execute on the database.

Laplace Noise

- Many differentially private algorithms rely on adding controlled noise to functions with low sensitivity.
- We will elaborate this point by taking a special kind of noise (whose kernel is a Laplace distribution i.e. the probability density function noise$(y) \propto e^{-|y|/\lambda}$, mean zero and standard deviation λ).
- Now in our case we define the output function of $\mathcal{A}$ as a real valued function (called as the transcript output by $\mathcal{A}$) $\mathcal{T}_{\mathcal{A}}(x) = f(x) + Y$, where $Y \sim \text{Lap}(\lambda)$ and f is the original real valued query/function we plan to execute on the database.

Laplace Noise

- Now clearly $\mathcal{T}_{\mathcal{A}}(x)$ can be considered to be a continuous random variable, where

$$\frac{\mathrm{pdf}(\mathcal{T}_{\mathcal{A},D_1}(x)=t)}{\mathrm{pdf}(\mathcal{T}_{\mathcal{A},D_2}(x)=t)} = \frac{\mathrm{noise}(t-f(D_1))}{\mathrm{noise}(t-f(D_2))}$$

which is at most $e^{\frac{|f(D_1)-f(D_2)|}{\lambda}} \leq e^{\frac{\Delta(f)}{\lambda}}$.

- We can consider $\frac{\Delta(f)}{\lambda}$ to be the privacy factor ϵ. Thus $\mathcal{T}$ follows a differentially private mechanism (as can be seen from the definition).
- If we try to use this concept in our diabetes example then it follows from the above derived fact that in order to have $\mathcal{A}$ as the ϵ-differential private algorithm we need to have $\lambda = 1/\epsilon$.

Laplace Noise

- Now clearly $\mathcal{T}_{\mathcal{A}}(x)$ can be considered to be a continuous random variable, where

$$\frac{\mathrm{pdf}(\mathcal{T}_{\mathcal{A},D_1}(x)=t)}{\mathrm{pdf}(\mathcal{T}_{\mathcal{A},D_2}(x)=t)}=\frac{\text{noise}(t-f(D_1))}{\text{noise}(t-f(D_2))}$$

  which is at most $e^{\frac{|f(D_1)-f(D_2)|}{\lambda}} \leq e^{\frac{\Delta(f)}{\lambda}}$.
- We can consider $\frac{\Delta(f)}{\lambda}$ to be the privacy factor ϵ. Thus $\mathcal{T}$ follows a differentially private mechanism (as can be seen from the definition).
- If we try to use this concept in our diabetes example then it follows from the above derived fact that in order to have $\mathcal{A}$ as the ϵ-differential private algorithm we need to have $\lambda = 1/\epsilon$.

Laplace Noise

- Now clearly $\mathcal{T}_{\mathcal{A}}(x)$ can be considered to be a continuous random variable, where

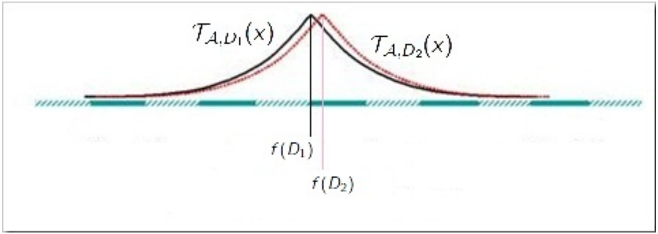
$$\frac{\mathrm{pdf}(\mathcal{T}_{\mathcal{A},D_1}(x)=t)}{\mathrm{pdf}(\mathcal{T}_{\mathcal{A},D_2}(x)=t)}=\frac{\text{noise}(t-f(D_1))}{\text{noise}(t-f(D_2))}$$

  which is at most $e^{\frac{|f(D_1)-f(D_2)|}{\lambda}} \leq e^{\frac{\Delta(f)}{\lambda}}$.
- We can consider $\frac{\Delta(f)}{\lambda}$ to be the privacy factor ϵ. Thus $\mathcal{T}$ follows a differentially private mechanism (as can be seen from the definition).
- If we try to use this concept in our diabetes example then it follows from the above derived fact that in order to have $\mathcal{A}$ as the ϵ-differential private algorithm we need to have $\lambda = 1/\epsilon$.

Laplace Noise

- If $Y \sim \text{Lap}(0, \lambda)$, then

$$\mathcal{T}_{\mathcal{A},D_1}(x) = f(D_1) + Y \sim \text{Lap}(f(D_1), \lambda)$$
$$\mathcal{T}_{\mathcal{A},D_2}(x) = f(D_2) + Y \sim \text{Lap}(f(D_2), \lambda)$$

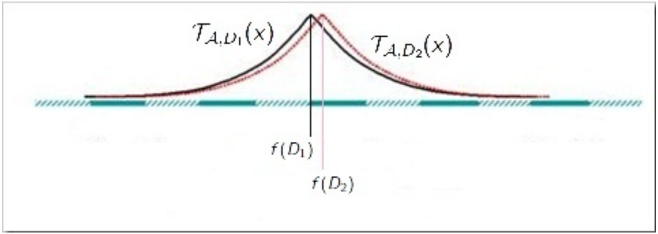


- Thus we can achieve differential privacy by add adding Laplace noise to the output.

Laplace Noise

- If $Y \sim \text{Lap}(0, \lambda)$, then

$$\mathcal{T}_{\mathcal{A},D_1}(x) = f(D_1) + Y \sim \text{Lap}(f(D_1), \lambda)$$
$$\mathcal{T}_{\mathcal{A},D_2}(x) = f(D_2) + Y \sim \text{Lap}(f(D_2), \lambda)$$



- Thus we can achieve differential privacy by add adding Laplace noise to the output.

Sequential composition

- If we query an ϵ-differential privacy mechanism t times, and the randomization of the mechanism is independent for each query, then the result would be ϵt-differentially private.

- In the more general case, if there are n independent mechanisms: $\mathcal{M}_1, \ldots, \mathcal{M}_n$, whose privacy guarantees are $\epsilon_1, \ldots, \epsilon_n$ differential privacy, respectively, then any function g of them: $g(\mathcal{M}_1, \ldots, \mathcal{M}_n)$ is $(\sum_{i=1}^{n} \epsilon_i)$-differentially private.

Sequential composition

- If we query an ϵ-differential privacy mechanism t times, and the randomization of the mechanism is independent for each query, then the result would be ϵt-differentially private.
- In the more general case, if there are n independent mechanisms: $\mathcal{M}_1, \ldots, \mathcal{M}_n$, whose privacy guarantees are $\epsilon_1, \ldots, \epsilon_n$ differential privacy, respectively, then any function g of them: $g(\mathcal{M}_1, \ldots, \mathcal{M}_n)$ is $(\sum_{i=1}^{n} \epsilon_i)$-differentially private.

Parallel composition

- Furthermore, if the previous mechanisms are computed on disjoint subsets of the private database then the function g would be $(\max_i(\epsilon_i))$-differentially private instead.

Group privacy

- In general, ϵ-differential privacy is designed to protect the privacy between neighboring databases which differ only in one row. This means that no adversary with arbitrary auxiliary information can know whether one particular participant submitted his information.
- However this is also extendable if we want to protect databases differing in c rows, which amounts to adversary with arbitrary auxiliary information can know whether c particular participants submitted their information.
- This can be achieved because if c items change, the probability dilation is bounded by $e^{\epsilon c}$ instead of e^{ϵ}, i.e. for D_1 and D_2 differing on c items:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{(\epsilon c)} \times \Pr[\mathcal{A}(D_2) \in S]$$

Group privacy

- In general, ϵ-differential privacy is designed to protect the privacy between neighboring databases which differ only in one row. This means that no adversary with arbitrary auxiliary information can know whether one particular participant submitted his information.
- However this is also extendable if we want to protect databases differing in c rows, which amounts to adversary with arbitrary auxiliary information can know whether c particular participants submitted their information.
- This can be achieved because if c items change, the probability dilation is bounded by $e^{\epsilon c}$ instead of e^{ϵ}, i.e. for D_1 and D_2 differing on c items:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{(\epsilon c)} \times \Pr[\mathcal{A}(D_2) \in S]$$

Group privacy

- In general, ϵ-differential privacy is designed to protect the privacy between neighboring databases which differ only in one row. This means that no adversary with arbitrary auxiliary information can know whether one particular participant submitted his information.
- However this is also extendable if we want to protect databases differing in c rows, which amounts to adversary with arbitrary auxiliary information can know whether c particular participants submitted their information.
- This can be achieved because if c items change, the probability dilation is bounded by $e^{\epsilon c}$ instead of e^{ϵ}, i.e. for D_1 and D_2 differing on c items:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{(\epsilon c)} \times \Pr[\mathcal{A}(D_2) \in S]$$

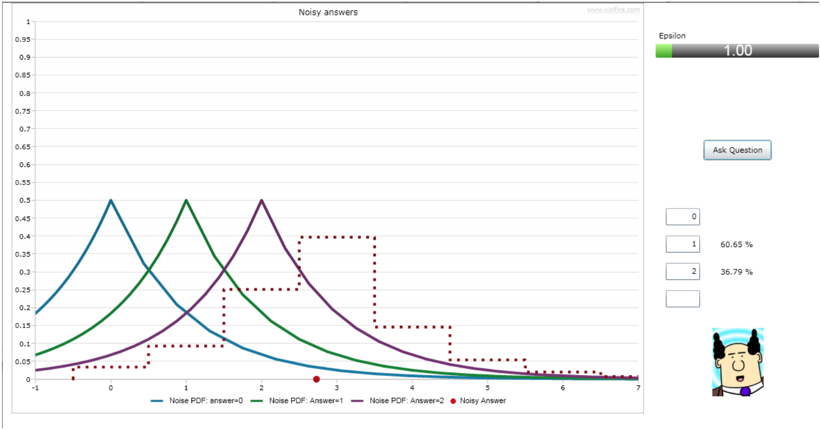
Group privacy

- Thus setting ϵ instead to ϵ/c achieves the desired result (protection of c items). In other words, instead of having each item ϵ-differentially private protected, now every group of c items is ϵ-differentially private protected (and each item is (ϵ/c)-differentially private protected).
- What's the drawback?

Group privacy

- Thus setting ϵ instead to ϵ/c achieves the desired result (protection of c items). In other words, instead of having each item ϵ-differentially private protected, now every group of c items is ϵ-differentially private protected (and each item is (ϵ/c)-differentially private protected).
- What's the drawback?

Group privacy



Group privacy

Proof idea:

- For three datasets $D1, D2$, and $D3$, such that $D1$ and $D2$ differ on one item, and $D2$ and $D3$ differ on one item (implicitly $D1$ and $D3$ differ on at most 2 items), the following holds for an ϵ-differentially private mechanism $\mathcal{A}$:

$$\Pr[\mathcal{A}(D_1) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{A}(D_2) \in S],$$

and

$$\Pr[\mathcal{A}(D_2) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{A}(D_3) \in S],$$

hence:

$$\begin{aligned} \Pr[\mathcal{A}(D_1) \in S] &\leq \exp(\epsilon) \times (\exp(\epsilon) \times \Pr[\mathcal{A}(D_3) \in S]) \\ &= \exp(2\epsilon) \times \Pr[\mathcal{A}(D_3) \in S] \end{aligned}$$

The proof can be extended to c instead of 2.

Conclusion & Discussion

- According to the sensitivity (Δf) of the statistical function, ϵ-differential privacy can be achieved in the statistical database by choosing appropriate λ of Laplace noise.
- In the next talk we will discuss how to achieve differential privacy in discrete functions, and how to use the technique in privacy-preserving aggregation.
- Discussion?

Conclusion & Discussion

- According to the sensitivity (Δf) of the statistical function, ϵ-differential privacy can be achieved in the statistical database by choosing appropriate λ of Laplace noise.
- In the next talk we will discuss how to achieve differential privacy in discrete functions, and how to use the technique in privacy-preserving aggregation.
- Discussion?

Conclusion & Discussion

- According to the sensitivity (Δf) of the statistical function, ϵ-differential privacy can be achieved in the statistical database by choosing appropriate λ of Laplace noise.
- In the next talk we will discuss how to achieve differential privacy in discrete functions, and how to use the technique in privacy-preserving aggregation.
- Discussion?