Interactive learning modules on Differential Privacy

Oleksandra Klymenko, Gonzalo Munilla Garrido, 24.06.2021, sebis day

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

Motivation
- Why Differential Privacy?
- What is Differential Privacy?
- The Practitioner's View

Learning Nuggets
- Learning Paths
- Example Learning Nugget

Further Activities

Call To Action

Motivation: Why Differential Privacy?

- Governments, researchers and businesses share aggregated sensitive data to facilitate research or create business value
- Data sharing can be risky since malicious actors might misuse the shared data
- Sensitive data must be modified to ensure that no personally identifiable information can be obtained from it
- Challenging trade-off: accuracy and utility vs. privacy loss
  - Overly excessive data modification may result in unusable data

Motivation (1/2): What is Differential Privacy?

Differential Privacy provides a mathematical foundation which allows to quantify the risk of re-identification for a given data modification (such as inserting noise into a dataset).

Main principle

Any information-related risk to a person should not change significantly as a result of that person's information being included, or not, in the analysis.

Goal

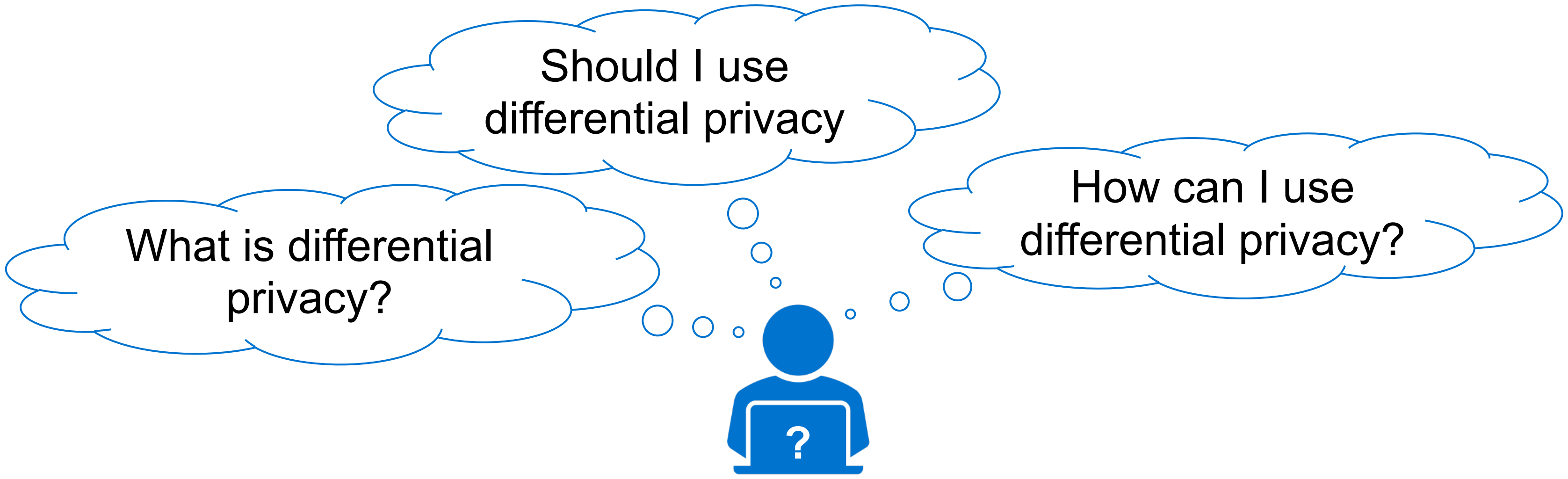
- "**plausible deniability"** giving the users a sufficient level of privacy so that it becomes virtually impossible to identify specific individuals with full certainty
- Achieve trade-off between accuracy and privacy, data should still allow for meaningful and valid statistical analysis ("privacy guarantee" vs. utility of the output)

Two main approaches

- Interactive: The data owner does not know which analysis will occur beforehand
- Non-interactive: The contrary – potentially more utility.

Previous work: Researched and applied by Facebook, Microsoft, Google and the US Census Bureau

The Practitioner's View

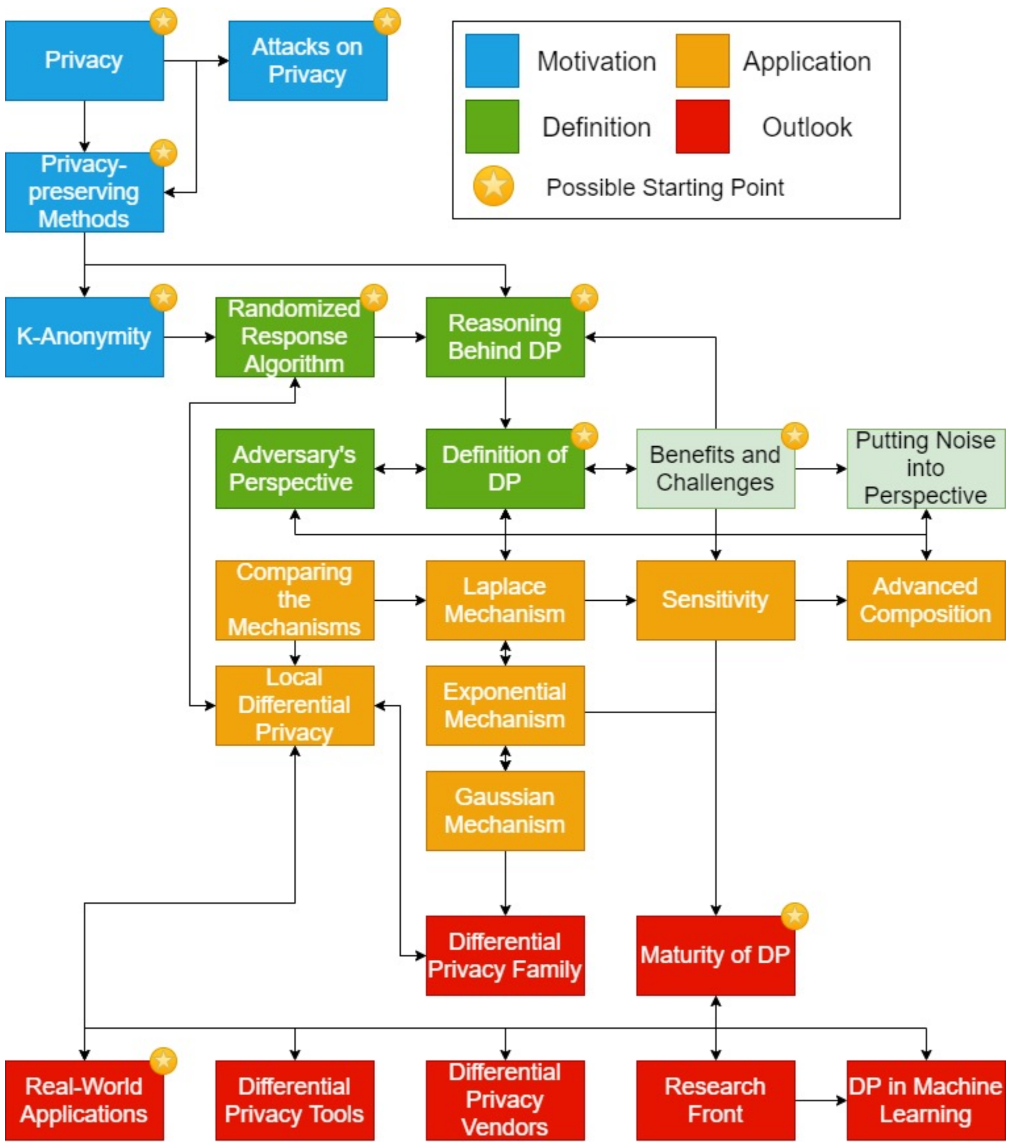


Learning Paths

- The graph on the right shows some of the main learning nuggets and how they are linked to each other
- The chart below illustrates possible learning paths

I want to...

... know if I should use it	... know how to use it	... be able to discuss it	... understand it thoroughly	... have a better utility
Attacks on Privacy	Definition of DP	Definition of DP	Randomized Response Algorithm	Putting Noise into Perspective
Privacy-preserving Methods	Laplace Mechanism	Benefits and Challenges	Reasoning Behind DP	Sensitivity
K-Anonymity	Comparing the Mechanisms	Laplace Mechanism	Definition of DP	Advanced Composition
Definition of DP	Differential Privacy Tools	Real-World Applications	Adversary's Perspective	Differential Privacy Family
Benefits and Challenges			Laplace Mechanism	
Maturity of DP			Comparing the Mechanisms	



Example Learning Nugget – Introduction

Prerequisites

- Necessary: the definition of differential privacy, the Laplace mechanism
- Beneficial: the randomized response algorithm

Introduction

- The Laplace mechanism can only be used for queries that are robust to perturbation and relatively insensitive to changes in the data of a single individual. It is not suitable for optimization problems or when dealing with non-numeric values
- In this unit, you will learn how differential privacy can be achieved in these settings, using the exponential mechanism
- We will cover the motivation, the formal definition, the proof and the accuracy of this mechanism

Learning Objectives

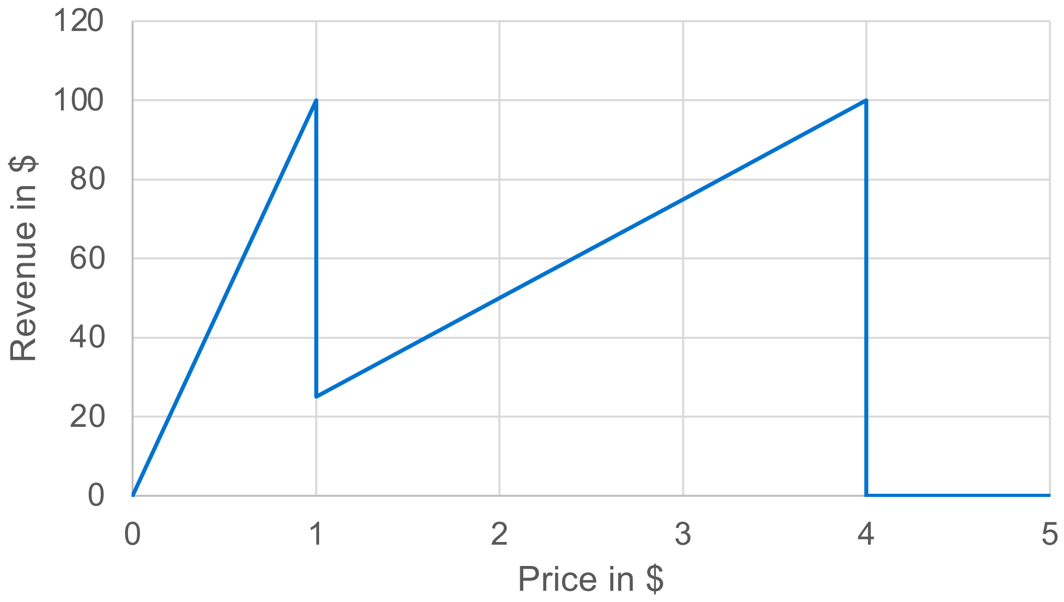
- You can explain the exponential mechanism and why it is ε-differentially private
- You can identify cases in which the exponential mechanism should be used
- You can apply the exponential mechanism to a simple case
- You can determine the accuracy of the exponential mechanism
- You can model other differentially private mechanisms using the exponential mechanism

Example Learning Nugget – Content

- In 2007 Frank McSherry and Kunal Talwar explored how differential privacy extends beyond disclosure limitation and can also give broad game theoretic guarantees[1]
- Intuitively, if the result of an analysis remains the same, regardless of an individual contributing their data or not, there is no incentive for individuals to misreport their data to influence the result in their favor
- As the guarantee of differential privacy extends to groups as well, it is also resistant to collusion
- However, when dealing with optimization problems, even adding a small amount of noise to the result could render it completely useless

Example

- Consider a digital goods auction. A company has an unlimited supply of a certain item, such as a digital movie, and would like to determine the optimal price to sell it at
- They survey 100 potential customers on how much they would be willing to pay for it (from 0 to 5$). But they want to conduct this analysis in a differentially private manner to preserve the privacy of the respondents and to discourage strategic lying
- According to their findings, 75 of the respondents would pay at most 1$, whereas the remaining 25 would pay up to 4$ for the movie
- Thus, the optimal price would either be 1$ or 4$ as both yield a revenue of 100$. But adding noise to this result could drastically lower the revenue. With a price of 4,01$ not a single movie would be sold, whereas a price of 1,01$ only yields a revenue of 25,25$



Example Learning Nugget – Exercises

1. Implement the Laplace mechanism in a programming language of your choice. The function should take as input parameters a query result (single number), epsilon and the sensitivity

   We will sketch a solution written in Python.
   The package NumPy already comes with a built-in functionality to draw random values from the Laplace distribution

```
import numpy as np

def laplace_mechanism(query_result, epsilon, sensitivity):
    return query_result+np.random.laplace(loc=0, scale=sensitivity/epsilon)

print(laplace_mechanism(8,1,1))
```

   Note that, as in cryptography, it is advisable to rely on existing solutions rather than implementing them yourself.
   The differential privacy guarantee can be undermined if the noise introduced is not truly random.

Key Takeaways

- Differential Privacy extends beyond disclosure limitation and can give broad game theoretic guarantees
- The exponential mechanism can be used in cases where the result of an analysis is non-numeric or not robust to additive perturbations
- Moreover, it is a general framework that captures any mechanism that gives differential privacy
- It operates based on a utility function u that assigns each possible result r for a database D a utility score and outputs results with a good utility with an exponentially higher probability:

$$\Pr[M(D) = r] = \frac{\exp\left(\frac{\varepsilon * u(D,r)}{2 * \Delta u}\right)}{\sum_{r' \in R} \exp\left(\frac{\varepsilon * u(D,r)}{2 * \Delta u}\right)}$$

- The biggest drawback of the exponential mechanism is its computability, as one must iterate over all possible results r

Outlook

- Visit comparing the mechanisms to see when to choose other mechanisms over the exponential mechanism

Further Activities

- Benchmark of five DP libraries: diffprivlib, SmartNoise, Google DP (PyDP), diffpriv and Chorus

Features	diffprivlib	SmartNoise	Google DP (PyDP)	diffpriv	Chorus
Contributor	IBM	Microsoft	Google (OpenMined)	B. Rubinstein et al.	J. P. Near et al.
Programming Language	Python	Python wrapper over Rust runtime	Google DP: C++, Java, Go (PyDP: Python wrapper over C++)	R	Scala
Primary use	Data science facing operations (Notebooks)	Data science facing operations (Notebooks), and large-scale systems	Google DP: Production-ready applications (PyDP: Data science)	Data science	Large-scale systems
Unique value proposition	Numerous machine learning algorithms, and DP mechanisms for experimentation	Blend of data science and operations	Google DP: Deployment of applications, e.g., in mobile phones (PyDP: Data science)	Flexibility for data scientists: User-defined functions and empirical calculation of sensitivity	Scalability via cooperation with existing database; extensibility
License	MIT	MIT	Apache-2.0	MIT	MIT
Benchmarked version	0.4.0	0.2.2	1.0.1	0.4.2	0.1.3

- Blog posts in OpenMined:

https://blog.openmined.org/differential-identifiability/
https://blog.openmined.org/choosing-epsilon/
https://blog.openmined.org/global-sensitivity/
https://blog.openmined.org/local-sensitivity/

Call To Action

- Get in contact with us to:
  - Get the slide decks
  - Try it out!
  - Provide feedback
  - Participate in workshop

Gonzalo Munilla Garrido
gonzalo.munilla-garrido@tum.de

Oleksandra Klymenko
alexandra.klymenko@tum.de

Sascha Nägele
sascha.naegele@tum.de

MSc.

Oleksandra Klymenko

Gonzalo Munilla Garrido

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business
Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289. 17132
Fax +49.89.289.17136
alexandra.klymenko@tum.de
gonzalo.munilla-garrido@tum.de
wwwmatthes.in.tum.de