IEEE TRANSACTIONS ON
KNOWLEDGE AND
DATA ENGINEERING

Differentially Private Data Publishing and Analysis: a Survey

Tianqing Zhu, *Member, IEEE,* Gang Li, *Senior Member, IEEE,* Wanlei Zhou, *Senior Member, IEEE,* and Philip S. Yu, *Fellow, IEEE*

Abstract—Differential privacy is an essential and prevalent privacy model that has been widely explored in recent decades. This survey provides a comprehensive and structured overview of two research directions: differentially private data publishing and differentially private data analysis. We compare the diverse release mechanisms of differentially private data publishing given a variety of input data in terms of query type, the maximum number of queries, efficiency, and accuracy. We identify two basic frameworks for differentially private data analysis and list the typical algorithms used within each framework. The results are compared and discussed based on output accuracy and efficiency. Further, we propose several possible directions for future research and possible applications.

Index Terms—Differential privacy, Privacy preserving data publishing, Privacy preserving data analysis.

1 Introduction

Over the past two decades, digital information collected by corporations, organizations, and governments has resulted in a vast number of datasets, and the speed of such data collection has increased dramatically over the last few years. Typically, a data collector, also known as a *curator*, is in charge of publishing data for further analysis [1]. However, most collected datasets contain private or sensitive information. Even though curators can apply several simple anonymization techniques, sensitive personal information still has a high probability of being disclosed [2]. Privacy preservation has, therefore, become an urgent issue that needs to be addressed.

Fig. 1 shows how a trusted curator preserves an original dataset that includes personal information. The curator provides aggregated information to public users, who may use this information for further investigation. We divide this process into *data publishing* and *data analysis* based on the purpose of its release. Data publishing aims to share datasets or some query results to the public. In some literature, this scenario is known as data sharing or data release. Another scenario, where a curator provides data models directly to the public, is normally defined as data analysis. The shared models may be associated with particular algorithms, such as data mining or machine learning algorithms.

Research communities have proposed various methods to preserve individual privacy in these two scenarios. The methods and their privacy criteria are defined as a *privacy model*. As shown in Fig. 1, a privacy model sits between a trusted curator and untrusted public users. *Differential privacy* is one such new and promising privacy model. It ensures that the ability of an adversary to inflict harm on any individual in a dataset is essentially the same, independent of whether any individual opts in to, or out of, the dataset [3]. Compared to previous privacy models, differential privacy can successfully resist most of privacy attacks and provide a provable privacy guarantee.

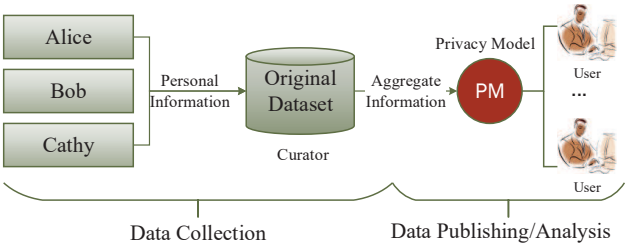


Fig. 1: Privacy model

Interest in differential privacy is very high and its central notion spans a range of research areas, from the privacy community to areas of data science such as machine learning, data mining, statistics, and learning theory. Much work has also been conducted in a number of application domains, including social networks, location privacy, and recommender systems. Fig. 2 shows some of these key elements to be further discussed in this survey.

1.1 Outline and Survey Overview

The initial work on differential privacy was pioneered by Dwork et al. [3] in 2006. Over the last decade, several surveys on differential privacy have been completed:

1) The first survey by Dwork et al. [4] recalled the definition of differential privacy and two of its principle mechanisms with the aimed to of showing how to apply these techniques in data publishing.
2) The report [5] exploited the difficulties that arise when data publishing encounters prospective solutions in the context of statistics analysis. It identified several research issues in data analysis that had not been adequately investigated at that time.

- *Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607*
  E-mail: psyu@uic.edu
- *Tianqing Zhu, Gang Li, Wanlei Zhou are with the School of Information Technology, Deakin University, Australia, Burwood, 3125.*

*Manuscript received ****, ****; revised ****, ****.*

Digital Object Identifier no. 10.1109/TKDE.2017.2697856

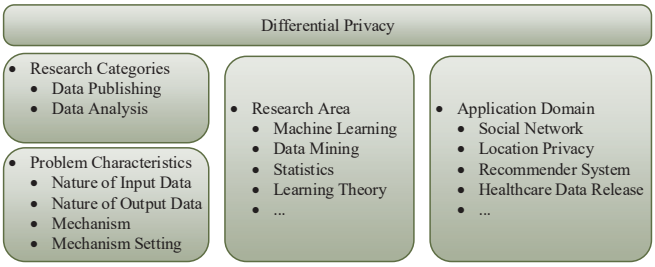


Fig. 2: Key elements associated with differential privacy

3) In a review [6], Dwork et al. provided an overview of the principal motivating scenarios, together with a summary of future research directions.
4) Sarwate et al. [7] focused on privacy preserving for continuous data to solve the problems in signal processing.
5) A book by Dwork et al. [8] presented an accessible starting place for anyone looking to learn about the theory of differential privacy.

Those surveys and the book focus on the concepts and theories of differential privacy; however, the mathematical theories are not easily implemented into applications directly. Yet, after more than ten years of theoretical development, a significant number of new technologies and applications have appeared in this area. We believe that now is a good time to summarize the new technologies and address the gap between theory and application.

Here we attempt to find a clearer way to present the concepts and practical aspects of differential privacy for the data mining research community.

- We avoid detailed theoretical analysis of related differentially private algorithms and instead place more focus on its practical aspects which may benefit applications in the real world.
- We try to avoid repeating many references that have already been analyzed extensively in the above well-cited surveys.
- Even though differential privacy covers multiple research directions, we restrict our observations to data publishing and data analysis scenarios, which are the most popular scenarios in the research community.

Table 1 defines the scope of these two research directions within the survey. Mechanism design for data publishing is normally independent from its publishing targets, as the goals of publishing is to release query answers or a dataset for further usage and, hence, is unknown to the curator. The mechanism design for data analysis aims to preserve privacy during the analysis process. The curator already knows the details of the analysis algorithm, so the mechanism is associated with the analysis algorithm.

TABLE 1: Comparison between differentially private data publishing and analysis

	Differentially private data publishing	Differentially private data analysis
Mechanism	independent mechanism	coupled with a particular algorithm
Input	various data types	transaction dataset (training samples)
Output	query answers or datasets	various models

2 Differential Privacy Preliminaries

2.1 Notation

We consider a finite data universe $\mathcal{X}$ with the size $|\mathcal{X}|$. Let r represent a record with d attributes; a dataset D is an unordered set of n records sampled from the universe $\mathcal{X}$. Two datasets D and D' are defined as neighboring datasets if they differ in one record. A query f is a function that maps dataset D to an abstract range $\mathcal{R}$: $f : D \to \mathcal{R}$. A group of queries is denoted as F. We use the symbol m to represent the number of queries in F.

The aim of differential privacy is to mask the differences in query f between neighboring datasets. The maximal difference in the results of query f is defined as the *sensitivity* Δf. Differential privacy is generally achieved by a mechanism $\mathcal{M}$ which is a randomized algorithm that accesses the database and implements some functionality. Randomized output is denoted by a circumflex over the notation. For example, $\widehat{f}(D)$ denotes the randomized answer of querying f on D. Table 2 summarizes the notations used in the following sections.

TABLE 2: Notations

Notations	Explanation	Notations	Explanation
$\mathcal{X}$	Universe	D	Dataset
D'	Neighbour dataset	$\mathcal{D}$	Data distribution
r	Record	d	Dataset dimension
n	Size of dataset	N	size of histogram
f	Query	F	Query set
m	query number	$\mathcal{M}$	Mechanism
$\widehat{f}$	Noisy output	η	Noise
ϵ	Privacy budget	Δf	Sensitivity
G	Graph data	t, T	Time sequence
$\mathbf{w}$	Output model	$VC(\cdot)$	VC dimension
c	Concept	$\mathcal{C}$	Concept set
h	Hypothesis	H	Hypothesis set
$\ell(\cdot)$	Loss function	θ	Threshold
δ	Confidence parameter	α, β	Accuracy parameter

2.2 Differential Privacy

Definition 1 ((ϵ, δ)-differential privacy). [9] A randomized mechanism $\mathcal{M}$ gives (ϵ, δ)-differential privacy for every set of outputs Ω, and for any neighbouring datasets of D and D', if $\mathcal{M}$ satisfies:

$$Pr[\mathcal{M}(D) \in \Omega] \leq \exp(\epsilon) \cdot Pr[\mathcal{M}(D') \in \Omega] + \delta \quad (1)$$

If $\delta = 0$, the randomized mechanism $\mathcal{M}$ gives ϵ-differential privacy by its strictest definition. (ϵ, δ)-differential privacy provides freedom to violate strict ϵ-differential privacy for some low probability events. ϵ-differential privacy is usually called *pure differential privacy*, while (ϵ, δ)-differential privacy with $\delta > 0$ is called *approximate differential privacy* [10].

2.3 The Privacy Budget Composition

In Definition 1, the parameter ϵ refers to the *privacy budget* [9], which controls the level of privacy guarantee achieved by mechanism $\mathcal{M}$. A smaller ϵ represents a stronger privacy level. Two privacy budget compositions theorems are widely used: *sequential composition* [11] and *parallel composition* [12].

Theorem 1 (Parallel Composition). Suppose we have a set of privacy mechanisms $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_m\}$. If each $\mathcal{M}_i$ provides a ϵ_i-differential privacy guarantee on a disjointed subset of the entire dataset, $\mathcal{M}$ will provide $(\max\{\epsilon_1, ..., \epsilon_m\})$-*differential privacy*.

Theorem 2 (Sequential Composition). Suppose a set of privacy mechanisms $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_m\}$ are sequentially performed on a dataset, and each $\mathcal{M}_i$ provides ϵ-differential privacy guarantee, $\mathcal{M}$ will provide $(m \cdot \epsilon)$-*differential privacy*.

2.4 The Sensitivity

Sensitivity is the parameter that determines how much perturbation is required for a particular query in a mechanism.

Definition 2 (Sensitivity). [6] For a query $f : D \rightarrow \mathcal{R}$, and neighboring datasets D and D', the *sensitivity* of f is defined as

$$\Delta f = \max_{D,D'} ||f(D) - f(D')||_1 \tag{2}$$

Sensitivity Δf is only related to the type of query f. It considers the maximal difference between the query results on neighboring datasets and indicates the extent to which the differences should be hidden.

2.5 The Principle Differential Privacy Mechanisms

Any mechanism meeting Definition 1 can be considered as differentially private. Currently, two basic mechanisms are widely used to guarantee differential privacy: the Laplace mechanism [13] and the exponential mechanism [11].

2.5.1 The Laplace Mechanism

The Laplace mechanism adds independent noise to the true answer. We use $Lap(b)$ to represent the noise sampled from a Laplace distribution with a scaling of b.

Definition 3 (Laplace mechanism). [6] For a function $f : D \rightarrow \mathcal{R}$ over a dataset D, the mechanism $\mathcal{M}$ in Eq. 3 provides the ϵ-differential privacy.

$$\mathcal{M}(D) = f(D) + Lap(\frac{\Delta f}{\epsilon}) \tag{3}$$

2.5.2 The Exponential Mechanism

For non-numeric queries, differential privacy uses an exponential mechanism to randomize the results, and this is paired with a *score function* $q(D, \phi)$ that evaluates the quality of an output ϕ. Defining a score function is application-dependent and different applications lead to various score functions.

Definition 4 (Exponential mechanism). [11] Let $q(D, \phi)$ be a score function of dataset D that measures the quality of output $\phi \in \Phi$, Δq represents the sensitivity of ϕ. The exponential mechanism $\mathcal{M}$ satisfies ϵ-differential privacy if

$$\mathcal{M}(D) = \left(\text{return } \phi \ \propto \exp(\frac{\epsilon q(D, \phi)}{2\Delta q})\right). \tag{4}$$

2.6 Utility Measurement of Differential Privacy

Several utility measurements are used in both data publishing and analysis when privacy level is fixed to ϵ.

- *Noise size measurement*: the easiest approach is to calibrate how much noise is added to the query results. A smaller amount of noise indicates a higher utility. This utility measurement has been widely used in data publishing.
- *Error measurement*: utility can be evaluated by the difference between the non-private output and the private output. The error measurement is normally represented by a bound with accuracy parameters [14]:

  Definition 5 ((α,β)-useful). A mechanism $\mathcal{M}$ is (α,β)-useful if

  $$Pr(\max_{f \in F} |F(D) - \widehat{F}(D)| \leq \alpha) > 1 - \beta, \tag{5}$$

  where α is the accuracy parameter that bounds the error.
  For different publishing scenarios, the error measurement can be interpreted in various ways. For synthetic dataset publishing, Eq. 5 can be interpreted as:

  $$Pr(\max_{f \in F} |F(D) - F(\widehat{D})| \leq \alpha) > 1 - \beta. \tag{6}$$

  For data analysis, the utility measurement normally depends on the analysis algorithms. Suppose the algorithm is denoted by $\mathcal{M}$ and the private algorithm is denoted by $\widehat{\mathcal{M}}$, Eq. 5 can be interpreted as

  $$Pr(|\mathcal{M}(D) - \widehat{\mathcal{M}}(D)| \leq \alpha) > 1 - \beta. \tag{7}$$

  Eq. 7 has several implementations in data analysis, such as risk bound and sample complexity, which are presented in Section 4.

3 Differentially Private Data Publishing

Differentially private data publishing aims to output aggregate information to the public without disclosing any individual's record. This problem can be presented as follows: if a curator has a dataset D and receives a query set $F = \{f_1, ..., f_m\}$, they are required to answer each query $f_i \in F$ subject to the constraints of differential privacy.

Name	Age	Diabetes
Alen	25	N
Bob	29	N
Cathy	35	Y
David	41	Y
Emily	56	N
...	...	...
Emma	21	Y

TABLE 3: Medical table

Age	No. of patients with diabetes	Variable
60-79	41	x_1
40-59	32	x_2
20-39	8	x_3
0-19	1	x_4

TABLE 4: Frequency table

TABLE 5: Differentially private data publishing problem characteristics

Differentially private data publishing	
The nature of input data	transaction, histogram, graph, stream
The nature of output data	query result, synthetic dataset
Publishing setting	interactive, non-interactive
Publishing mechanism	Laplace/exponential, query separation, transformation, iteration, dataset partitioning
Challenges	query number, accuracy, computational efficiency

Two settings, interactive and non-interactive, are involved in this publishing scenario. In the interactive setting, a query f_i can not be issued until the answer to the previous query f_{i-1} has been published. In the non-interactive setting, all queries are given to the curator at one time, and the curator can provide answers with full knowledge of the query set.

An example to show the difference between the two settings is presented in Table 3. Queries to the curator may presented as follows:

- f_1: *How many patients have diabetes at the age of* $40-79$*?*
- f_2: *How many patients have diabetes at the age of* $40-59$*?*

Suppose the privacy budget ϵ is fixed for each query. In the interactive setting, the curator will first get f_1, then counts the number of patients who have diabetes between the ages of 40 and 79 and adds independent Laplace noise to the number with a sensitivity equal to 1, $Lap(1/\epsilon)$. When f_2 is then submitted to the curator, f_2 will be answered with sensitivity equal to 2, as changing one person in the table may change the results of both queries. The total noise added to the query set is $Lap(1/\epsilon) + Lap(2/\epsilon)$.

In non-interactive settings, both queries are submitted to the curator at the same time. The sensitivity measured for both queries is 2. The total noise added to the query set is $2 * Lap(2/\epsilon)$, which is larger than the interactive setting. The correlation between queries leads to a higher sensitivity. Therefore, non-interactive settings normally incurs more noise than interactive settings.

The above example presents the differences between the two settings and shows that the amount of noise increases dramatically when queries are correlated to each other. In addition, for a dataset with size n, the Laplace mechanism can only answer, at most, sub-linear in n number of queries to a certain level of accuracy [15].

These weaknesses make the Laplace mechanism impractical in the scenarios that require answering large amounts of queries. New mechanisms are required. Table 5 summarizes the problem characteristics of differentially private data publishing, in which mechanism design focuses on the number of queries, the accuracy of the output, and the computational efficiency.

3.1 Publishing Mechanisms

We categorize existing mechanisms into several types: transformation, dataset partitioning, query separation and iteration. Table 3 is again used to show the key ideas.

- *Transformation*: Transformation mechanisms map the original dataset to a new structure to adjust the sensitivity or level of noise. In the above example, the original dataset can be transferred to a frequency dataset as shown in Table 4.
  In the new structure, f_2 can be answered directly by the second row with a sensitivity equal to 1. f_1 can be answered by combining $\widehat{f_2}$ and the noisy result of first row of Table 4. As the result for patients with diabetes from age $60-79$ is listed independently to f_2, the sensitivity of f_1 is still equal to 1. The total noise of two queries will be $3 * Lap(1/\epsilon)$, which is lower than the non-interactive Laplace mechanism. The new structure is used to decompose the correlation between queries, so the sensitivity can be decreased as well. The challenge is finding a new structure.
- *Dataset partitioning*: the original dataset is divided into several parts to decrease noise. In the above example, suppose we need to answer f_1 with Table 4, the noise $Lap(1/\epsilon)$ needs to be added twice: once to the first row and again to the second row. The total noise added is $2 * Lap(1/\epsilon)$. However, if we partition the dataset another way, for example, by rearranging the age range to $40-79$, the total noise will decrease to $Lap(1/\epsilon)$. The challenge here is designing a partition strategy for multiple queries.
- *Query separation*: query separation assumes that a query set can be separated into several groups and that some queries can be answered in the sense of reusing noise. In the above example, if f_2 has been answered, f_1 can be approximately answered by doubling the answer of f_2 as the age range is doubled. Query separation is used to break limitations on the number of queries.
- *Iteration*: iteration is a mechanism that updates a dataset recursively to approximate the noisy answers for a set of queries. For example, we can manually define an initial dataset D_0, in which the number of patients with diabetes in Table 4 at different age ranges are equal, then perform f_1 on D_0, and compare the noise result $\widehat{f_1}(D)$ with $f_1(D_0)$. If the distance between the two answers is smaller than a pre-defined threshold, $f_1(D_0)$ can be published, and D_0 will be used in the next round. Otherwise, $\widehat{f_1}(D)$ will be published and D_0 will be updated by a particular strategy into D_1. As publishing $f_1(D_0)$ does not consume any privacy budget, the iteration mechanism can achieve a higher utility and can answer more queries than the Laplace mechanism. The challenges are designing an update strategy and setting the related parameters such as threshold.

Table 6 compares the various publishing mechanisms. In the following sub-sections, we present how those mechanisms work in both interactive and non-interactive settings.

3.2 Interactive Publishing

Interactive settings operate on various aspects of the input data, including transactions, histograms, streams and graph datasets. In the following subsections, we discuss publishing scenarios involving these types of input data.

3.2.1 Transaction Data Publishing

The most popular representation of D is a transaction dataset, in which every record represents an individual with d attributes.

3.2.1.1 ***Query separation***: The goal of query separation is to design a separation strategy for given types of queries to decrease noise. Roth [16] presented the *median* mechanism and found that, among any set of m queries, there are $O(\log m \log|\mathcal{X}|)$ queries that can determine the answers of all other queries. Based on this observation, all queries are separated into hard and easy queries. Hard queries can be answered directly by the Laplace mechanism, while easy queries are answered by the median values of hard query results. Therefore, easy queries do not consume any privacy budget. By separating the queries, the median mechanism can answer exponentially many more queries with acceptable accuracy; however, it is inefficient and comes with an exponential time complexity corresponding to the dataset size n.

3.2.1.2 ***Iteration***: Hardt et al. [17] proposed *private multiplicative weights* (PMW), which considers datasets as a histogram with positive weight on each bin. By updating the weights, PMW constructs a histogram sequence to answer a set of queries. After the parameters have been calibrated for complexity and accuracy, this mechanism is able to answer each query with a sampling error approximately to $O((\log m)/\sqrt{n})$. This means that the sampling error grows logarithmically with an increase in the number of queries being answered, while the Laplace mechanism's error is linear, increasing by m. In addition, PMW can accurately answer an exponential number of queries.

Similarly, Gupta et al. [18] presented a general iteration framework termed iterative database construct (IDC), which implements other release mechanisms by using the framework. In each round of iteration, when a significant difference between the current dataset and the original dataset is witnessed for a given query, the mechanism updates the current dataset for the next update. IDC is a more general framework that can be incorporated into various other mechanisms, including PMW and the median.

3.2.1.3 ***Discussion***: The query separation and iteration mechanisms can answer more queries than that of the Laplace mechanism, normally an exponential number of n. With an increase of the number of queries, the error bound of the Laplace mechanism increases by $m \log m$, whereas other mechanisms limit the increase to $\log m$, which is a huge improvement when searching an entire data universe. With the exception of the Laplace mechanism, most of the other mechanisms are inefficient, as they do need to traverse the whole data universe. Accordingly, the error bound grows logarithmically with $|\mathcal{X}|$ in these mechanisms.

given iteration mechanisms outperform others in terms of the error bound, many subsequent works, therefore, have followed the iteration mechanism approach. For example, based on *IDC*, Huang et al. [19] considered the distance query defined over an arbitrary metric. Iteration mechanisms are normally used in low-dimensional datasets or histograms; however, publishing high-dimensional datasets still needs further exploration.

3.2.2 Histogram Publishing

It is often convenient to regard transaction data in terms of their histogram representations. A histogram has N bins, and a differential privacy mechanism aims to hide the frequency of each bin. Table 4 can be considered as a histogram presentation of Table 3 with four bins. The advantage of histogram representation is that limits the sensitivity to noise [4]. For example, when the histogram serves to support the range or count queries, adding or removing a single record will affect, at most, one bin. Hence, range or count queries on the histogram have a sensitivity equal to 1, and the magnitude of added noise to each bin will be relatively small.

3.2.2.1 ***Laplace***: This is a direct mechanism which adds Laplace noise to the frequency of each bin that a query covered. When a count of range query covers only small number of bins, this mechanism retains high utility for the query result; however, if the original dataset contains multiple attributes, the combination of these attributes and their related range of values will lead to a large number of bins. The answer of queries are meaningless due to the large amount of error accumulated from having an enormous number of bins.

3.2.2.2 ***Dataset partitioning***: As the number of bins is derived from the partition of attribute values, one method for decreasing error is to optimize the partition strategy. For example, when the query covers a larger amount of bins, several bins can be merged into a single new bin. Laplace noise can be reduced because it is only added once to the new merged bin. When the number of bins covered by a query is relatively small, the curator can either split the large bin into smaller bins, or approximate the query result by estimating proportion of the large bin's frequency. However, splitting the large bin into smaller bins leads to more Laplace noise, while estimating the proportion of the large bin's frequency may introduce estimation error. Therefore, optimizing the partition strategy to obtain a trade-off between splitting and merging bins is a challenge that needs to be addressed.

Xu [20] provided two partition strategies by minimizing the sum of squared error (SSE) of a set of queries. Both strategies set each attribute as a single bin at an initial state and partition the attribute value to create more bins. The first strategy, *NoiseFirst*, injects Laplace noise into each bin before partitioning the attribute values. Another strategy, *StructureFirst*, uses an exponential mechanism to select optimal splitting values of attributes by adopting the SSE as the score function.

Qardaji et al. [21] partitioned the attribute values in a hierarchical way. They also focus on range queries and

TABLE 6: Differentially Private Data Publishing Mechanism Comparison

Mechanism	Description	Advantage	Challenge
Laplace	Add Laplace noise directly to the query	Easy to implement; can answer all types of real-value queries	Answer a sub-linear in n number of queries; introduce large volume of noise
Transformation	Transfers original dataset to a new structure. The sensitivity of the query set will be adjusted.	Noise can be reduced; consistency of results can be maintained	Not easy to find a new structure that suits for queries.
Dataset partitioning	Divides dataset into several parts and adds noise to each part separately.	Sensitivity can be decreased which results in less error in the output.	partition strategy is not easy when answering multiple queries
Iteration	Updates datasets or query answers recursively to approximate the noisy answer.	Only some updates will consume privacy budget, so more queries can be answered (linear to n or exponential number of n) in a fixed privacy budget.	Most iteration mechanisms are computationally inefficient; unsuitable parameters can result in inferior performance.
Query separation	Only need to add noise to small numbers of queries	Some queries do not consume privacy budget, so the mechanism can answer more queries with a fixed privacy budget.	Separating queries is a difficult problem

allocate ranges into a tree. The root of the tree is the full range of values of an attribute or several attributes. Each node in the tree is associated with the union of the ranges of its children. Several unit-length ranges are defined as leaves. On each branch of the tree, a factor controls the accuracy of the query result. These factors are further studied with a mean squared error (MSE) when answering range queries, and the results are optimized by tuning these factors.

3.2.2.3 ***Histogram consistency***: Even though partitioning is a popular mechanism in histogram releases, this mechanism may bring problems with inconsistency. For example, after adding noise, the summation of two bin's value may be less than one bin. To maintain consistency in histogram publishing, Hay et al. [22] defined a *constrained inference* to adjust the publishing output. Two types of consistency constraints were explored. The first, sorted constraints, requires query results to satisfy a particular sequence. The second, hierarchical constraints, predefines the sequence of a hierarchical interval set. A constrained inference step applies a linear combination method to estimate a set of approximate answers that are close to the noisy answers, which satisfies the consistency constraints.

There are some other ways to improve the consistency of the histogram. For example, Lin et al. [23] viewed a set of sorted histograms as a Markov chain and proposed an algorithm that applies ordering constraints on the estimates. Lee et al. [24] added a post-processing step, formulated as a constrained maximum likelihood estimation problem, before publishing the histogram.

3.2.3 Stream Data Publishing

In real world scenarios, it is more practical to publish continuously updated data. This type of data can be simplified as a bit string $\{0,1\}^n$ and each 1 in the stream represents the occurrence of an event [25]. A differentially private mechanism releases one bit at every time step.

Fig. 3 illustrates an example. Suppose there is a binary bit stream $D \in \{0,1\}^T$, where T represents a time sequence $T = \{t_k : k = 0, ...\}$. The bit $\sigma(t_k) \in \{0,1\}$ denotes whether an event has occurred at time t_k. At each time step t_k, the

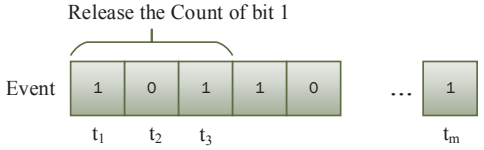


Fig. 3: Stream data

count of $1s$ is denoted as $f(t_k)$ and the noisy output of the mechanism is $\widehat{f}(t_k)$.

3.2.3.1 ***Dataset partitioning***: Partitioning the dataset in stream data aims to recursively split the stream into sub-domains and evaluate a noisy count for each sub-domain. The problem lies in defining the sub-domain and how many times the stream can be divided.

Chan et al. [26] presented a *p-sums* mechanism which computes the partial sum of consecutive bits in a stream. Each sub-domain is defined by p-sum, which is an intermediate result estimating the count at each time interval. Laplace noise is added to a p-sum result rather than an individual count answer. This guarantees an error bound of $O(\frac{\log^{3/2}|T|}{\epsilon})$, which decreases the linear complexity of noise to *log* complexity.

Zhang et at. [27] created a quadtree for spatial datasets. They defined a threshold to determine the minimum of a sub-domain and another threshold to limit the height of a quadtree. Using these two thresholds, the amount of noise added to the quadtree can be limited to a constant.

3.2.3.2 ***Iteration***: Dwork et al. [28] proposed a continual output transformation algorithm and developed an iteration release mechanism to output the continual count at each time step. The error is decreased to $O(\frac{\log^{3/2}|T|}{\epsilon})$.

Georgios et al. [29] proposed the notion of *w-event* privacy to achieve a balanced privacy target. They formulated the iteration mechanism using a sliding window methodology. This w-event privacy aims to mask any event sequence within w time steps, which is suitable for infinite stream releases.

3.2.3.3 ***Discussion***: Even though continual publishing is a popular topic in the data mining community, there

are still many unsolved problems in the privacy preserving area. For example, releasing a multi-dimensional dataset periodically, and dealing with other statistical queries needs further exploration.

3.2.4 Graph Data Publishing

With the significant growth of online social networks (OSNs), graph data has raised concern among OSN participants. An OSN dataset can be modeled as an undirected graph $G = (V, E)$. We use V to denote nodes and use $E \subseteq V \times V$ to denote edges. In an OSN, nodes normally represent individuals while edges denote their relationships. Two concepts of differential privacy in graph data have been defined: *node differential privacy* and *edge differential privacy*.

3.2.4.1 ***Edge Differential Privacy***: Edge Differential Privacy ensures that a query's answer does not reveal the inclusion or removal of a particular edge in the graph. Two graphs are neighboring if they differ in one edge. The first differential privacy research into graph data was conducted by Nissim [30], who showed how to evaluate the number of triangles in a social network with edge differential privacy. They use local sensitivity, which limits the sensitivity tightly with an upper bound, and show how to efficiently calibrate the noise for subgraph counts accordingly. The results of this technique were investigated by Karwa et al. [31] to release counts on k-triangles and k-stars. They achieved ϵ-differential privacy for k-star counting, and (ϵ, δ)-differential privacy for k-triangle counting.

Zhang et al. [32] claim that if one can find an isomorphic graph with proper statistical properties that is similar to the original graph, the isomorphic graph can be used to generate accurate query answers. Given a subgraph G', they adopted an exponential mechanism to search a number of isomorphic copies of G' to answer subgraph queries.

3.2.4.2 ***Node differential privacy***: In node differential privacy, two graphs are considered to be neighboring if one of them is generated by removing one node and all the edges related to this node from the other graph. Node differential privacy is more strict than edge differential privacy. The sensitivities resulting from the change of one node are proportional to the graph size.

With the observation that many useful statistics have low sensitivities on graphs G_θ with a small degree θ, a common approach to achieve *node differential privacy* is to transform the graph G to a θ-degree-bounded graph G_θ, in which the nodes with degrees of more than θ are deleted [33], [34].

Chen et al. [35] proposed an iteration mechanism for node differential privacy. Given a graph G and any real-valued function f, they defined a sequence of real-valued functions $0 = f_0(G) \leq f_1(G) \leq \cdots \leq f_m(G) = f(G)$ with a *recursive monotonicity* property. The recursive approach will return subgraph counting for any kind of subgraph with node differential privacy. However, constructing the sequence of functions $f_i(G)$ is usually NP-hard, and efficiently implementing it remains an open problem.

3.2.4.3 ***Discussion***: Existing methods work reasonably well with edge differential privacy or even node differential privacy for basic graph statistics. However, releasing specific statistics such as `cuts`, pairwise distances between nodes, or on hyper-graphs, still remain open issues.

3.3 Non-interactive Publishing

Non-interactive settings mean all queries are given to the curator at one time. The key challenge for non-interactive publishing is the sensitivity measurement. Correlation between queries will dramatically increase the sensitivity. Two possible methods are proposed to fix this problem: one is decomposing the correlation between batch queries, which is presented in Sub-section 3.3.1, another is publishing a synthetic dataset with the constraint of differential privacy to answer those proposed queries. Related methods are presented in the synthetic dataset publishing sub-section.

3.3.1 Batch Queries Publishing

Batch query release refers to the most common non-interactive scenario in which a fixed set of m queries $F = \{f_1, ..., f_m\}$ are answered in a batch.

Using Table 4 again to show the batch query problem, suppose a curator would like to release a batch of range queries for Table 4, and Table 7 contains all possible `range` queries $F = \{f_1, ..., f_{10}\}$. Deleting any record in D will change at most 6 query results (column containing x_2 or x_3 in Table 7) in F. According to the definition of sensitivity, the sensitivity of F is 6, which is much higher than the sensitivity of a single query. Therefore, most research focus on how to decrease the sensitivity of F.

TABLE 7: Batch query release example

Query							
f_1	x_1	+	x_2	+	x_3	+	x_4
f_2	x_1	+	x_2	+	x_3		
f_3		+	x_2	+	x_3	+	x_4
f_4	x_1	+	x_2				
f_5		+	x_2	+	x_3		
f_6					x_3	+	x_4
f_7	x_1						
f_8			x_2				
f_9					x_3		
f_{10}							x_4

3.3.1.1 ***Transformation***: One possible means of decreasing noise is to re-calibrate the sensitivity in a new data structure A.

Xiao et al. [36] proposed a wavelet transformation, called *Privelet*, which applies a wavelet transformation on the frequency dataset D to generate a wavelet coefficient A. Each entry of A is considered as a linear combination of the entries in D. Range queries can be generated with a linear combination of basic queries. Table 8 shows the basic queries. For example, the answer of $range(x_2, x_3) = x_2 + x_3$ can be generated by $range(x_2, x_3) = 0.5f_1 - 0.5f_3 - 0.5f_4$, and the sensitivity will be decreased from 6 to 3. In this way, the sensitivity of the wavelet coefficient is estimated as $1 + \log_2 n$ and the variance of noise per answer is $O((\log_2 n)^3/\epsilon^2)$, which is much smaller than that in the Laplace mechanism.

Li et al. [37] proposed a *matrix* mechanism which can answer sets of linear counting queries. The set of queries, defined as a workload, is transformed into a matrix A, where each row contains the coefficients of a linear query. The essential element of the matrix mechanism is to select

TABLE 8: m Range queries

query							
f_1	x_1	+	x_2	+	x_3	+	x_4
f_2	x_1	+	x_2	-	x_3	-	x_4
f_3	x_1	-	x_2				
f_4					x_3	-	x_4

A to represent the set of queries. Based on the selection of A, the matrix mechanism can be extended to a variety of approaches. For example, if A is an identity matrix, this mechanism can be considered to be a normal Laplace mechanism for batch queries. If A is selected using a Haar wavelet, it can be extended as a Privelet [36]. The error bound of the matrix mechanism was analyzed in their subsequent paper [38]. They proved that when batch queries are denoted in a matrix manner, the minimum error for the result can be evaluated according to the spectral properties of this query matrix.

Similarly, Huang et al. [39] transformed the query sets into a set of orthogonal queries to reduce the correlation between queries. Yuan et al. [40] formulated searches of A as a constrained optimization problem. This approach is challenging, however, as the optimization objective is non-convex. They formulated searches of A as a low-rank matrix factorization problem using an iteration method to approximate a suitable A. Later, they considered approximate differential privacy and transformed the optimization objective into a convex program to minimize the overall error in the results [41].

3.3.1.2 ***Dataset partitioning***: Kellaris et al. [42] decomposed the dataset columns into disjoint groups and added Laplace noise to these groups' counts. The final result is generated using the new column count. Because the maximum number of original counts in a group affected by a user is limited, the sensitivity of each group is decreased, and the Laplace noise required for ϵ-differential privacy also diminishes. The advantage of this mechanism is that it can limit the sensitivity for numerical datasets.

3.3.1.3 ***Iteration***: Iteration has also been used in batch query releases. By recursively approximating the true answer, the noise in the output can be effectively diminished. Xiao et al. [43] aimed to decrease errors in the released output. They argued that the Laplace mechanism adds noise with a constant range to every query answer without considering the true value of the answer. Thus, queries with small answers have a much higher error than expected, defined as relative error. In practice, larger answers can tolerate more noise. In some applications, relative errors are more important than absolute errors. To decrease the relative error, Xiao et al. [43] proposed a mechanism named *iReduct*, which initially obtains rough error estimations for query answers and subsequently uses this information to iteratively refine these error evaluations.

3.3.1.4 ***Discussion***: The key problem in batch query publishing is how to decrease the sensitivity between correlated queries. Currently, transformation is the most popular way to tackle this problem. Current works mainly focus on range queries and develop appropriate structures to answer linear combinations of those queries. More types of structures need to be developed to answer various types of queries. Iteration or partitioning the dataset may not be effective for correlation decomposition, but they have the potential to answer more types of queries.

3.3.2 Synthetic Dataset Publishing

Synthetic dataset publishing investigates publishing datasets with the public instead of sharing answers to queries. Suppose the input dataset is D with d attributes, and the curator would like to publish a synthetic $\widehat{D}$ that can be used to answer a query set F. Two different methods to achieve this goal exist. One method applies anonymization techniques to publish an anonymized dataset, which retains the same records as the original dataset. Another method takes samples from the data universe to build a synthetic dataset, which follows the original datasets distribution but may not necessarily keep the same records.

3.3.2.1 ***Synthetic dataset publishing based on anonymization***: This line of works observes that if the anonymization process follows the requirement of differential privacy at each step, the published synthetic dataset will satisfy differential privacy. Based on this observation, Mohammed et al. [44] proposed the anonymized algorithm *DiffGen* to preserve privacy for data mining purposes. The anonymized procedure consists of two major steps: partitioning and perturbation. Every attribute of the original dataset is generalized to its top-most state. The partitioning step then splits these attributes into more specific groups according to the attribute taxonomy trees. It applies an exponential mechanism to choose a candidate for the specialization. After that, random noise is added to the true count of each records group by perturbation.

3.3.2.2 ***Synthetic dataset publishing based on learning theory***: Kasiviswanathan et al. [45] and Blum et al. [14] claim that if a query on a published dataset is limited to a particular concept set $\mathcal{C}$, the learning process can ensure the accuracy of the query answer in the constraint of differential privacy.

Kasiviswanathan et al. [45] proposed an exponential based mechanism to search a synthetic dataset from the data universe that can accurately predict $\mathcal{C}$. Fig. 4 shows that after creating multiple candidate datasets from the data universe, the mechanism will search the most suitable $\widehat{D}$ based on the exponential mechanism. The authors claimed that for any $\mathcal{C}$ and any $D \geq \{0,1\}^d$, if the size of the dataset satisfies

$$n \geq O(\frac{dVC(\mathcal{C})\log(\frac{1}{\alpha})}{\epsilon\alpha^3} + \frac{\log\frac{1}{\beta}}{\epsilon\alpha}),$$

α accuracy can be achieved with a probability of $1-\beta$.

Subsequent works made progress toward improving accuracy. Dwork et al. [46] used *boosting* to improve the lower bound of accuracy to $O(\frac{\sqrt{n\log|\mathcal{X}|}\log^{2/3}m}{\epsilon})$. Hardt et al. [47] combined an exponential mechanism with a multiplicative weight iteration approach to achieve a nearly optimal accuracy guarantee.

Although these mechanisms developed tight bounds on accuracy, they suffer from inefficient computation time. A normal time cost is exponential in the size of universe $\mathcal{X}$ and the size of concept class $\mathcal{C}$. Blum et al. [14] claim that if polynomial time is required, the definition of privacy has to

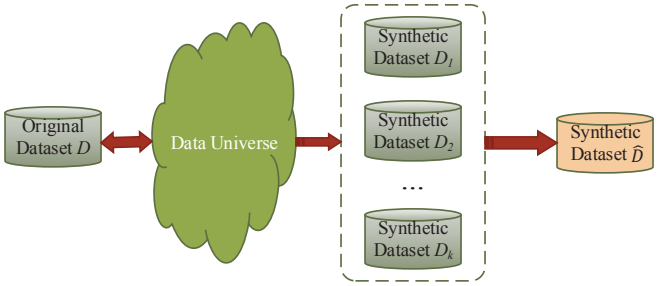


Fig. 4: Synthetic Dataset Publishing with Learning Theory

be relaxed. This claim was confirmed by Dwork et al. [48], who proved that only after relaxing the notion of ϵ to (ϵ, δ) can computational time be polynomial.

Moreover, Ullman et al. [49] showed that among all computationally efficient algorithms, the Laplace mechanism is almost optimal. In fact, no algorithm has the ability to answer more than $O(n^2)$ queries randomly in polynomial time. This result suggests that it is difficult to design mechanisms to answer arbitrary queries efficiently.

Table 9 summarizes the key works on ynthetic datasets. For simplicity, sensitivity is predefined as 1, the dependence on β is omitted and the run-time only considers the size of the universe.

TABLE 9: Synthetic dataset publishing with learning theory

Mechanism	Accuracy	Efficiency	Privacy
Blum et al. [14]	$O(\frac{n^{2/3}\log^{1/3} m \log^{1/3}\mathcal{X}}{\epsilon^{1/3}})$	inefficient	ϵ
Dwork et al. [48]	$O(\frac{n^{1/2}\log^{1/2}\lvert\mathcal{X}\rvert m}{\epsilon})$	inefficient	(ϵ, δ)
Dwork et al. [46]	$O(\frac{n^{1/2}\log^{1/2}\lvert\mathcal{X}\rvert\log^{3/2} m}{\epsilon})$	inefficient	(ϵ, δ)
Hardt et al. [47]	$O(\frac{\log\lvert\mathcal{X}\rvert\log m}{\epsilon n})^{1/3}$	inefficient	ϵ

Learning theory extends the research work on synthetic data releases, proving it is possible to maintain acceptable utility while preserving differential privacy. Nevertheless, the issue of reducing the computational complexity remains a challenge.

3.3.2.3 ***Synthetic dataset publishing for high dimensional datasets***: Neither anonymized dataset publishing nor learning theory-based publishing can effectively handle high-dimensional datasets. In anonymized methods, when the input dataset contains many attributes, existing anonymized methods will inject a prohibitive amount of noise, which leads to inferior utility. In learning theory-based methods, computational complexity is exponential to the dimension of the dataset, making the publication infeasible for high-dimensional datasets. One promising way to address high dimensionality is to disassemble the dataset into a group of lower dimensional marginal datasets, and then apply methods to infer the joint data distribution from these marginal datasets.

Zhang et al. [50] followed the above rationale and used a Bayesian network to deal with high dimensionality. They assumed some correlations between attributes exist and, if these correlations can be modeled, the model can be used to generate a set of marginal datasets to simulate the distribution of the original dataset. The disadvantage of this solution is that it consumes too much of the privacy budget during network construction and, hence, makes the approximation of the distribution inaccurate.

Chen et al. [51] addressed this disadvantage by proposing a clustering method. They disclose the pairwise correlation of all the attributes and generate a dependency graph. A junction tree algorithm is applied to the graph to identify a set of attribute clusters, which are used to generate noisy marginals. As a final step, an inference model is used to create a synthetic dataset. As opposed to [50], they have limited access to the dataset, saving the privacy budget to obtain a better result.

3.4 Summary of Differentially Private Data Publishing

3.4.1 Summary of the Interactive Setting

The interactive setting has attracted attention due to advances in statistical databases. In interactive settings, the privacy mechanism receives a user's query and replies with a noisy answer to preserve privacy. Traditional Laplace mechanisms can only answer $O(n)$ queries, which is insufficient in many scenarios. Researchers have to provide different mechanisms to fix this essential weakness.

The proposed interactive publishing mechanisms improve performance in terms of query type, the maximum number of queries, accuracy, and computational efficiency. Upon analysis, we conclude that these measurements in interactive releases are associated with one another. For example, given a fixed privacy budget, a higher accuracy usually results in a smaller number of queries. On the other hand, with a fixed accuracy, a larger number of queries normally leads to computationally inefficient mechanisms. Therefore, the goal of data publishing mechanism design is to achieve a better result that can balance the above mentioned measurements. The choice of mechanism depends on the requirement of the application.

3.4.2 Summary of the Non-interactive Setting

High sensitivity presents a big challenge in non-interactive settings. Batch query publishing methods can only publish limited types of queries. Publishing a synthetic dataset seems appealing because, in some scenarios, people require details of the attributes to determine further analysis methods. The research on synthetic data publishing, however, is still in its early stages and there are many open problems in this area. The essential problem is efficiency. Given most publishing mechanisms need to sample datasets from the entire data universe, it is hard to search for a suitable dataset in polynomial time.

Another problem is that synthetic dataset publishing can only publish datasets for particular purposes. For example, an anonymization dataset focuses on a decision tree algorithm, the published dataset obtains an acceptable result for decision tree tasks, yet the proposed method does not guarantee the performance for other types of tasks. Learning-based methods have the same disadvantage, or worse, as they only guarantee learning performance for a particular class. Publishing a dataset for multiple purposes needs further investigation.

The third problem is dealing with high-dimensional datasets. Even though [50] and [51] have undertaken some

TABLE 10: Differentially private data analysis problem characteristics

Differentially Private Data Analysis	
The nature of input data	Transaction
The nature of output data	Analysis models/algorithms
Analysis framework	Laplace/exponential framework, private learning framework
Analysis mechanism	Laplace/exponential mechanism for Laplace/exponential framework, learning process for private learning framework
Challenges	Accuracy, computational efficiency

initial work, they both consume too much of the privacy budget when building the distribution model, making the results less accurate than that of lower-dimensional datasets.

4 Differentially Private Data Analysis

The essential task of differentially private data analysis is extending the current non-private algorithms to differentially private algorithms. This extension can be realized by several frameworks, roughly categorized into Laplace/exponential frameworks and private learning frameworks. The Laplace/exponential framework incorporates Laplace or exponential mechanisms into non-private analysis algorithms directly. For example, adding Laplace noise to the count steps in the algorithm or by employing perform exponential mechanisms when making selections.

Private learning frameworks consider the data analysis as learning problems in terms of optimization. The learning problems are solved by defining a series of objective functions. Compared with the Laplace/exponential framework, a private learning framework has a clear target, and the results produced by this framework are easier to compare in terms of risk bound or sample complexity. But private learning frameworks can only deal with limited learning algorithms, while nearly all types of analysis algorithms can be implemented in a Laplace/exponential framework.

Table 10 shows the characteristics of the differentially private data release problems. Researchers are concerned with the accuracy and computational efficiency of these two frameworks. As different papers use diverse terms to describe the output, the terms "model" and "algorithm" are interchangeable in this section.

4.1 Laplace/exponential Framework

The most common extension method is to incorporate Laplace or exponential mechanisms into non-private analysis algorithms. These algorithms are usually associated with specific machine learning or data mining tasks, which are separated into the supervised learning, unsupervised learning, and frequent pattern mining categories.

4.1.1 Supervised Learning

Supervised learning refers to the prediction methods that extract models describing data classes via a set of labeled training records [52]. As one of the most popular supervised learning algorithms, *decision tree* learning, has been extensively studied in Laplace/exponential Frameworks.

Decision trees are iterative processes that recursively partition the training sample to build a tree with each label representing a leaf. Assuming there is an input dataset D with d categorical attributes $\{a_1, ...a_d\}$, a decision tree is constructed from the root that holds all the training records then the algorithm chooses the attribute a_i that maximizes the information gain to partition the records into child nodes. The procedure is performed recursively on each subset of the training records until a stop criteria is meet.

The first differentially private decision tree algorithm was developed on the SuLQ platform [53]. Noise is added to the information gain, and an attribute a_i with noisy information gain that is less than a specified threshold is chosen to partition a node. However, as information gain is evaluated separately for each attribute in each iteration, the privacy budget is consumed several times in each iteration, which results in a large volume of noise. In addition, SuLQ fails to deal with continuous attributes. If those attributes are simply discretized into intervals, the basic concept of differential privacy is violated, because the split values in continuous attributes would reveal information about the records.

To overcome the SuLQ platform's disadvantages, Friedman et al. [54] improved the algorithm in two ways. First, they implemented an exponential mechanism in the attribute selection step. The score function is defined by the information gain or the gain ratio. The attributes with a top score have a higher probability of being selected. In this way, less of the privacy budget is consumed than by SuLQ. Second, the proposed method can deal with continuous attributes. An exponential mechanism is employed to select every possible splitting value and the continuous attributes domain is divided into these intervals. Compared to SuLQ, they obtain better performance.

Jagannatham et al. [55] provided an algorithm for building random private decision trees, which randomly select attributes to create nodes. The algorithm first creates a tree in which all the leaves are on the same level and then builds a leaf count vector. Once the independent Laplace noise is added to the count vector, a differentially private random decision tree can be generated from the noisy leaf vector. The algorithm iteratively produces multiple random decision trees and uses the ensemble method to combine these trees. As the attribute is randomly selected, this step saves the privacy budget; however, as each tree's magnitude is scaled up with the number of trees in the final ensemble step, the utility of the ensemble remains a problem.

Rana et al. [56] proposed a practical approach to ensemble decision trees in a random forest. They do not strictly follow the notion of differential privacy, which keeps the neighboring data distribution approximately invariant. Instead, they only keep the statistic features invariant. A privacy attack model is defined to prove the reliability of the proposed decision tree. As less budget has been consumed in the ensemble process, this relaxation of differential privacy can lead to higher utility compared to other algorithms.

Discussion: Differentially private decision tree algorithms were the earliest algorithms to be investigated in the differential privacy community. The advantage of this series of methods is that they are concise and easy to implement. However, because tree-based algorithms need to select split

Algorithm 1 Basic k-means Algorithm

Require: points r_1, ... ,r_n.
Ensure: k centers $\{\nu_1, ..., \nu_k\}$ with arranged points
repeat
1. Create k clusters by assigning each point to its closest center;
2. Re-calculate the center of each cluster;
until centers do not change

attributes multiple times, the privacy budget is quickly consumed, which incurs a huge utility loss. This drawback stems from decision tree building and is not easy to deal with. Nowadays, one of the most popular ways to design differentially private supervised learning algorithms is to apply a private learning framework, which is discussed in Sub-section 4.2.

4.1.2 Unsupervised Learning

As a typical unsupervised learning method, *clustering* algorithms group unlabeled records into clusters to ensure that all records in the same cluster are similar to each other. Assume the input is a set of points, r_1,...,r_n, the clustering output is k centers $\{\nu_1, ..., \nu_k\}$ and assigned points.

A basic k-means algorithm is formulated by Algorithm 1. The aim of differential privacy clustering is to add uncertainty into the center ν and the number of records in each center. To achieve this goal, noise is added in Step 1 in Algorithm 1. In fact, adding noise to cluster centers is impractical because the sensitivity of the cluster center will be quite large as deleting one point will totally change the center. Therefore, the challenge for clustering is to evaluate and minimize the sensitivity of the cluster centers.

Nissim et al. [30] used local sensitivity with k-means clustering to circumvent the large sensitivity problem, by relying on the following intuition. In a well-clustered scenario, a point with noise should have approximately the same center as its previous center. In addition, moving a few "well-clustered" records would not ultimately change the centers. As such, they define a local sensitivity to measure the record-based sensitivity of the cluster center, which is much lower than the traditional global sensitivity. Since the value of local sensitivity is difficult to measure, they provide a sample aggregate method to approximate local sensitivity.

Based on local sensitivity, Wang et al. [57] implemented subspace clustering algorithms. They introduce Laplace mechanism into an agnostic k-means clustering, and an exponential mechanism into Gibbs sampling subspace clustering algorithm. Their subsequent paper, [58] adopted a Johnson-Lindenstrauss transform to guarantee differential privacy in a subspace clustering algorithm. The Johnson-Lindenstrauss transform can reduces the dimensions of the dataset, making the clustering problem practical, as well as preserving the distance between each record.

Discussion: In general, even some differentially private platforms such as SuLQ [53], PINQ [12], Privgene [59], Gupt [60] can automatically implement clustering algorithms. They all assume that the sensitivity of the cluster centers has been predefined. Even though local sensitivity can partially solve the problem, further reducing sensitivity still remains a challenge.

4.1.3 Frequent Itemset Mining

Frequent itemset mining aims to discover itemsets that frequently appear in a dataset D. Suppose I is the set of items in D, and an *itemset* refers to a subset of I. Let each record $r_i \in D$ denote as a transaction that contains a set of items from I. Given an itemset I_j, if transaction r_i contains I_j, we say r_i supports I_j, and the proportion of supporting transactions in the dataset is defined as the *support* of I_j. An itemset with a frequency larger than the predefined support threshold is called a frequent itemset or a frequent pattern.

Let U represent all frequent itemsets, where the $topk$ most frequent itemsets in U should be released under differential privacy guarantee. Laplace noise is normally added to the frequency; however, the main challenge is that the total number of itemsets is exponential to the number of items: If I contains n items, the number of all possible itemsets is $|U| = \sum_{i=1}^{k} \binom{n}{i}$. Decreasing the number of candidate itemsets is a major research issue in differentially private frequent itemset mining.

Bhaskar et al. [61] solved the problem by providing a truncated frequency to reduce the number of candidate itemsets. They proposed an algorithm that uses the exponential mechanism to choose the top-k itemsets. The score function of each candidate is the frequency defined as $\widehat{p(U)} = max(p(U), p_k - \gamma)$, where p_k is the frequency of the k-th most frequent itemsets and $\gamma \in [0, 1]$ is an accuracy parameter. Every itemset with a frequency greater than $p_k - \gamma$ is computed as its normal frequency $p(U)$ while the frequency of the rest is truncated to $p_k - \gamma$. All the itemsets with frequencies of $p_k - \gamma$ are grouped into one set, and the algorithm uniformly selects itemsets from this set. In this way, the computational cost is significantly reduced.

The advantage of truncating frequency is that it can significantly decrease the size of candidate itemsets. However, it is only applicable when k is small. Another weakness is that the length of top-k itemsets needs to be predefined, which affects the flexibility of frequent itemset mining.

To address these weaknesses, Li et al. [62] constructed a basis set $\mathcal{B} = \{B_1, B_2, ...\}$ in which any itemset with a frequency higher than a threshold is a subset of the basis set B_i. The algorithm generated candidate itemsets based on $\mathcal{B}$ and the algorithm could release itemsets with arbitrary length. However, generating basis sets $\mathcal{B}$ is not easy; furthermore, when the length of the itemset and the number of basis sets are large, the cardinality of the candidate itemsets is still too big to handle. Efficiently decreasing the number of the candidate itemsets is still a challenge.

Zeng et al. [63] proposed an algorithm that randomly truncates transactions in a dataset according to a predefined maximal cardinality. The algorithm iteratively generates candidate itemsets and perturbs the support of those candidate itemsets.

Lee et al. [64] proposed an FP-tree based frequent itemset mining algorithm. The algorithm first identifies all frequent itemsets without knowing their exact supports, but only that their supports are above a predefined threshold. The proposed algorithm injects noise into the data structure at the intermediate (FP-tree) step. The final output can be further refined through an optional post-processing step. The advantage of the algorithm is that information disclosure

TABLE 11: Comparison of supervised learning methods

Difficulty	Key methods	Typical papers	Advantages	Disadvantages
Privacy budget has to be consumed multiple times	Add noise to information gain	[53]	Easy to implement	Noise will be high due to privacy budget arrangement in attribute selection
	Use exponential mechanism to select attribute	[54]	Save part of privacy budget in the attribute selection	Still has high noise
	Select attribute randomly	[56], [55]	Does not consume privacy budget during attribute selection	Privacy budget will be largely consumed in the ensemble process

TABLE 12: Comparison of unsupervised learning methods

Difficulty	Key methods	Typical papers	Advantages	Disadvantages
High sensitivity of clustering centers	Use local sensitivity	[30], [57]	Decreases the level of the sensitivity	Local sensitivity may not be easy to estimate
	Johnson-Lindenstrauss transform	[58]	Guarantees differential privacy while retaining distance between points	Johnson-Lindenstrauss transform is only valid for norm 2 distance measurement.

affecting differential privacy occurs only for count queries above the predefined threshold; negative answers do not count against the privacy budget.

Shen et al. [65] applied a Markov chain Monte Carlo (MCMC) sampling method to deal with the challenge of large candidate itemsets. They claim that an MCMC random walk method can bypass the problem, and the exponential mechanism can then be applied to select frequent itemsets. They use this method to mine frequent graph patterns.

Xu et al. [66] applied a binary estimation method to identify all possible frequent itemsets and then use an exponential mechanism to privately select frequent itemsets. The number of candidates is eventually a logarithm of the original number using a binary search.

Discussion: frequent itemset mining is a typical data mining task, which suffers from searching large candidate sets. Differential privacy makes the problem worse. [61], [63], [64] and [62] tried to merge some candidates into groups using different methods. [65] adopted a sampling method to search the candidate itemsets. [66] applied a binary search method. All of those methods decreased the searching space from exponential to polynomial, which is a big achievement. However, the noise still remains high, which needs to be decreased further.

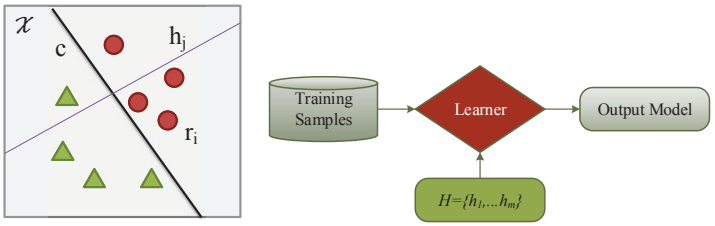
4.1.4 Discussion on the Laplace/exponential Framework

The Laplace/exponential frameworks can introduce Laplace and exponential mechanisms freely to various types of algorithms. However, the utility of the analysis results is a challenge for this framework. When adding noise to algorithm steps, it is unclear how large the utility loss will be and the analysis result is not easy to compare.

One possible way to tackle the difficulty is solving the data analysis problem through the view of optimization and taking advantage of some existing theories, such as the learning theory [45], so that the utility loss can be estimated and compared. Based on this intuition, researchers proposed the private learning framework.

4.2 Private Learning Framework

Another line of differential privacy research investigates learning problems from the perspective of machine learning theory [67]. A learning problem is shown in Fig. 5a. Suppose $D = \{r_1, ..., r_n\}$ is a set of samples drawn from a universe $\mathcal{X}$. Dots and triangles in the figure denote two labels $y \in \{0, 1\}$. The function that separates one label from the other is defined as concept c. Suppose there are groups of functions (hypotheses) H and h_j, the goal of the learning process L is to find a hypothesis h that satisfies with c on almost all of universe $\mathcal{X}$. Fig. 5b illustrates a typical learning process. By using the input training samples, the learner selects the most suitable $h \in H$ as the output model $\mathbf{w}$.



(a) Learning Problem (b) Learning Process

Fig. 5: Learning problem and process description

The purpose of a private learning framework is to design a private learner that outputs an approximately accurate model and preserves the differential privacy of training samples. Private learning frameworks are concerned with the following issues:

- *How to choose an optimal model in terms of differential privacy?*
- *How many samples are needed to achieve a bounded accuracy?*

The first question can be answered by combining an Empirical Risk Minimization (ERM) technique with differential privacy. The utility is measured by a risk bound. The second question can be tackled by analyzing the sample complexity of a private learner via PAC learning theory. Both risk bound and sample complexity can be considered as implementations of Eq. 7.

4.2.1 Private Learning in ERM

TABLE 13: Comparison of frequent itemset mining methods

Difficulty	Key methods	Typical papers	Advantages	Disadvantages
Large candidate itemsets	Merge candidates into groups	[61] [62] [63] [64]	Easy to implement	Merge strategy has high impact on the output. Some important patterns may be missed.
	Binary search in the candidate set.	[66]	The frequent itemsets can be quite accurate	The search time can be decreased, but still inefficient.
	Sampling from candidate set	[65]	Highly efficient	Some important patterns may be missed.

4.2.1.1 The foundation of ERM: ERM is used to select an optimal model from a set of hypotheses by minimizing the expected loss over the collected samples [68]. Suppose $h \in H$ is a hypothesis and $\mathbf{w}$ is the output model. We define a *loss function* $\ell(h(\mathbf{w}, r), y)$ to estimate the expected risk of the hypothesis. The goal of ERM is to identify a $\mathbf{w}$ that minimizes the *empirical risk* $R_n(\mathbf{w})$ on training sample D. Eq. 8 shows the empirical risk minimization.

$$R_n(\mathbf{w}) = min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{w}, r_i), y_i) + \lambda\mu(\mathbf{w}) \quad (8)$$

where the regularizer $\lambda\mu(\mathbf{w})$ prevents over-fitting.

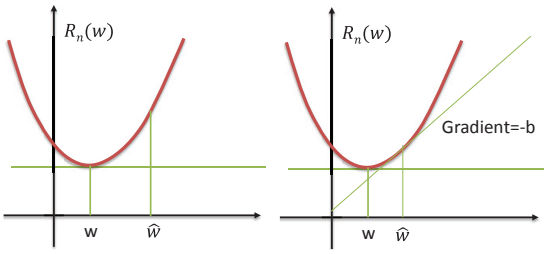
By choosing different loss functions, ERM can be implemented for certain learning tasks, such as linear regression [69], in which the loss function is defined as maximum likelihood estimation (MLE); logistic regression [68], in which the loss function is defined as logistic loss; and the kernel method [70], [71], in which the loss function is set to hinge loss. To make the ERM solution tractable, people assumed that 1) the loss function and regularizer are convex and 2) the loss function is *L-Lipschitz*. Under these two assumptions, Eq. 8 can be considered to be a d dimensional convex optimization problem.

The utility of private ERM is measured by the difference between the real risk $R(\mathbf{w})$ and the private $R(\widehat{\mathbf{w}})$, defined as a *risk bound*. A lower risk bound leads to a higher utility.

4.2.1.2 *Perturbation methods*: When incorporating differential privacy into the current learning process, current works apply two methods via an ERM technique: *output perturbation* and *objective operation*. The output perturbation inserts noise into the output $\mathbf{w}$; while the objective operation adds noise to the objective function prior to learning.

Chaudhuri et al. [68] proposed output perturbation solutions on *logistic regression*. Fig. 6a shows that the $\widehat{\mathbf{w}}$ is obtained by moving the original $\mathbf{w}$ on the horizontal axis. The sensitivity is $\frac{2}{n\lambda}$, which is associated with the regularizer parameter λ. With analysis, Chaudhuri et al. argued that the learning performance of this private algorithm degrades with a decreasing of λ. Their subsequent paper, [69], trains classifiers on disjoint subsets of the data with different parameters and uses an exponential mechanism to privately choose parameters.

The objective perturbation adds noise to the loss function $\ell(D)$ [68], [69], which is illustrated in Fig. 6b and shows that the calibration of noise is associated with the gradient of $R_n(\mathbf{w})$. Chaudhuri et al. conclude that objective perturbation outperforms output perturbation when the loss function is convex and doubly differentiable. This is because regularization already changes the objective to protect against over-fitting, and changing the objective will not significantly impact performance. This claim is confirmed in terms of a risk bound in the next subsection.



(a) Output Perturbation (b) Objective Perturbation

Fig. 6: Output and Objective Perturbation

4.2.1.3 ***Risk bound in different learning algorithms***: Many machine learning algorithms can be implemented in privacy learning frameworks, including linear regression, online learning, deep learning, etc. With the exception of the learning process and the privacy budget, the risk bound is related to the dimension and the size of training samples. Bassily et al. [72] showed that the risk bound depends on $O(\sqrt{d}/n)$ under (ϵ, δ)-differential privacy and $O(d/n)$ under (ϵ)-differential privacy. These results show that the larger the dimension and the size, the lower utility that ERM can achieve. Table 14 lists several typical learning algorithms and their risk bounds.

Nearly all types of learning problems can be analyzed in terms of their risk bound. For example, most of the existing papers solve regression problems that have a single optimal objective. Ullman [73] considered the regression problem as a multiple objective optimization when the datasets are examined with different aims. Multiple convex minimization is considered as a set of queries that can be answered by a prevalent multiplicative weights method. Ullman implemented several single objective regression results in multiple objectives platform and their risk bounds are consistent with those original papers.

Kasiviswanathan et al. [74] considered private incremental regression in terms of streaming data. They combined continual release [28] with ERM technology to analyze the risk bound of several algorithms. Taking linear regression as an example, they continuously update a noisy version of the gradient function to minimize the MLE loss function. The risk bound depends on the $\sqrt{d}$ and the length of the stream: $O(\min\{\sqrt{d}, T\})$, where this bound is quite close to the normal linear regression private learner when the stream length T is large.

In addition, some papers consider private learning in a

higher dimension dataset. Kifer et al. [75] gave the first results on private sparse regression with high dimensions. The authors designed the algorithm based on sub-sampling stability for support recovery using a LASSO estimator. Their following work [76] extended and improved the results with an algorithm based on a sample efficiency test of stability. Jain et al. [77] proposed an entropy regularized ERM using a sampling technique, This algorithm provides a risk bound that has a logarithmic dependence on d. Kasiviswanathan et al. [78] considered random projections in ERM frameworks. They provided a new private compress learning method with the risk bound related to the Gaussian width of the parameter space C in the random projection.

One of the most promising directions is the deep learning. Recent research focus has been devoted to the design of deep learning mechanisms. Abadi et al. [79] applied objective perturbation in a deep learning algorithm by defining the loss function as the penalty for mismatching training data. As the loss function is non-convex, they adapted a mini-batch stochastic gradient descent algorithm to solve the problem. The noise is added to every step of the stochastic gradient descent. In another pilot work [80], Shokri et al. designed a distributed deep learning model training system that enables multiple parties to jointly learn an accurate neural network. They implemented private stochastic gradient descent algorithms to achieve (ϵ, δ)-differential privacy within multiple parties. In the work of Phah et al. [81], the authors perturbed the objective functions of the traditional deep auto-encoder and designed the deep private auto-encoder algorithm by incorporating a Laplace mechanism.

4.2.1.4 ***Discussion***: Private learning applies ERM to estimate the hypothesis set to select an optimal output $\mathbf{w}$. It uses the output perturbation and the objective perturbation to ensure that the output satisfies differential privacy. A series of works have proven that this private learning is tractable for certain learning tasks, such as linear regression, SVM, and deep learning, etc.

Risk bounds are highly associated with the dimension of the dataset. Current research has decreased dependence on dimension to $O(d)$. Under certain assumptions, the dimension dependence could be further relaxed.

4.2.2 Sample Complexity in Private Learning

The second problem: *how many samples are needed in bounded accuracy?* is associated with sample complexity analysis. Sample complexity interprets the distance utility measurement in another way to show how many samples are needed to at least achieve a particular accuracy α. Probably approximately correct (PAC) learning [83] helps to measure the sample complexity of learning algorithms. Based on this theory, Blum et al. [14] and Kasiviswanathan et al. [45] proved that every finite concept class can be learned privately using a generic construction with a sample complexity of $O(VC(\mathcal{C})\log|\mathcal{X}|)$ (we omit the other parameters). This is a higher sample complexity than a non-private learner who only needs a constant number of samples. Improving sample complexity is an essential issue, and its study can be categorized into the following three groups: relaxing the privacy requirement, relaxing the hypothesis, and semi-supervised learning.

4.2.2.1 ***Relaxing privacy requirement***: Relaxing the privacy requirement is an effective way to close the gap of sample complexity. Beimel et al. [10] showed that when relaxing ϵ-differential privacy to (ϵ, δ)-differential privacy, sample complexity can be significantly decreased. Their follow-up paper decreases sample complexity to $O(\log(\sqrt{(d \cdot 1/\delta)}))$ [84].

Another way to relax privacy requirements is to preserve the privacy of sample labels, rather than all sample attributes. Chaudhuri et al. [85] assumed that with the exception of labels, the attributes of the samples are insensitive. Label privacy aims to decrease the dimension of sensitivity information. This may provide enough protection in scenarios where the content of the underlying samples is publicly known except for their labels. In other scenarios, however, these attributes may be highly sensitive, and Chaudhuri et al.'s relaxation may not be applicable in these cases.

4.2.2.2 ***Relaxing hypothesis***: This gap can also be closed by relaxing the requirement of the hypothesis. If the output hypothesis is selected from the learning concept $H \subseteq \mathcal{C}$, the learning process is defined as a proper learning. Otherwise, it is called an improper learning. For proper learning, the sample complexity is approximately $\Omega(d)$. If choosing improper learning, the sample complexity can be further decreased.

Beimel et al. [86] confirmed that when selecting a hypothesis that is not in $\mathcal{C}$, the sample complexity can be decreased to the constant. Their subsequent paper [87] proposed a probabilistic representation of $\mathcal{C}$ to improve the sample complexity. They considered a list of hypothesis collections $\{H_1, ..., H_m\}$ rather than just one collection of H to represent $\mathcal{C}$. The authors assumed that, when sampling H_i from the hypothesis list, there will be $h \in H$ close to c in high probability. The sample complexity can be reduced to $O(\max(\ln|H_i|))$.

This improvement in sample complexity comes at the cost of an increased workload on evaluation, however. The learner will have to exponentially evaluate many points that are far from the concept set $\mathcal{C}$. In general, for a private learner, if $H = \mathcal{C}$, the sample complexity is $O(d)$ and the time for evaluation is constant. If $H \neq \mathcal{C}$, there is constant sample complexity but $O(\exp(d))$ time for evaluation.

4.2.2.3 ***Semi-supervised learning***: Semi-supervised learning is a useful method for reducing the complexity of labeled samples. Beimel et al. [88] proposed a private learner by introducing semi-supervised learning to active learnings. The method starts with an unlabeled dataset to create a synthetic dataset for a class $\mathcal{C}$. This synthetic dataset is then used to choose a subset of the hypotheses with a size of $2^{O(VC(\mathcal{C}))}$. In the last step the authors apply $O(VC(\mathcal{C}))$ labeled examples to choose the target synthetic dataset according to the hypotheses set.

In this process, the sample complexity of the labeled samples is $O(\frac{VC(\mathcal{C})}{\alpha^3\epsilon})$ while for the unlabeled samples it is $O(\frac{d \cdot VC(\mathcal{C})}{\alpha^3\epsilon})$. Comparing the sample complexity of labeled and unlabeled samples, this private learner uses a constant number of labeled samples and $O(d)$ unlabeled samples.

4.2.2.4 ***Discussion***: Table 15 compares the different methods in terms of their sample complexity. To make the results clear, we use a *VC dimension* representation and omit

TABLE 14: Private learning risk bound

Learning algorithm	References	Perturbation method	Loss function	Risk bound
Regression	Chaudhuri et al. [69]	output/objective perturbation	regularized MLE	depends on $d \log d$/ depend on d
SVM	Jain et al. [70]	output/objective perturbation	arbitrary kernel, for example, polynomial kernels	depends on $d^{1/3}/n^{2/3}$
	Benjamin et al. [82]	output perturbation	hinge-Loss function with invariant kernel, such as Gaussian kernel	depends on $\sqrt{d}$
Online learning	Kasiviswanathan et al. [74]	objective perturbation	regularized MLE	depends on $\sqrt{d}$ and time T
Deep Learning	Abadi et al. [79] [81]	objective perturbation	average of unmatched sample	N/A

the other parameters such as α, β and ϵ.

4.3 Summary of Differentially Private Data Analysis

As the most prevalent framework in differentially private data analysis, the Laplace/exponential framework has been widely used. The most prominent advantages are its flexibility and simplicity; it can freely introduce Laplace and exponential mechanisms into various types of algorithms. For the non-experts on privacy, it proposed a possible way to ensure that the results satisfy the requirement of differential privacy. However, the essential challenge for this framework is accuracy of the results, especially for those algorithms whose operations have high sensitivities. They lead to a large volume of noise in the analysis results. The Laplace/exponential framework is widely used in current applications but there is still room for further improvement.

Private learning frameworks combine differential privacy with diverse learning algorithms to preserve the privacy of the learning samples. The foundation theories are ERM and PAC learning. ERM helps to select the optimal learning model by transferring the learning process into a convex minimization problem. Differential privacy either adds noise to the output models or to the convex objective functions. PAC learning estimates the relationship between the number of learning samples and the models accuracy. The more samples an algorithm can obtain, the more accurate the result will be. As privacy learning results in higher sample complexity, researchers are currently trying to narrow the sample complexity gap between private and non-private learning processes so that private learning is feasible with an acceptable number of samples.

Private learning frameworks, however, have some constraints. ERM requires that the objective function should be convex and L-Lipschitz. PAC learning can only be applied when the algorithm is PAC learnable. These constraints hinder the practical development of privacy learning frameworks and make them an actively developing theoretical proposition, yet still impractical for real applications.

5 Applications of Differential Privacy

5.1 Differential privacy in Location-based Services

Advances in sensor-enabled devices, such as mobile phones and wearable gadgets, allow location information to be available on social media. This means that the places that people visit are able to disclose extremely sensitive information about their behaviors, home and work locations, preferences, and habits. In addition, the continual release of ones location can be used as a trajectory. Location and trajectory privacy are emerging issues that need to be tackled [89].

Location based services involve several privacy concerns. For example, a location based server would like to hide the number of people in a particular region, and this is a typical range query that differential privacy can deal with. Cormode [90] applied spatial decomposition methods, which is a type of dataset partitioning mechanism, to decrease noise. They instantiated a hierarchical tree structure to decompose a geometric space into smaller areas with data points partitioned among the leaves. Noise is added to the count for each node. Similarly, Zhang et al. [91] applied spatial decomposition methods in the problem of private location recommendations, which is the extension of range queries. Quadtree is used to partition the region and noise is added to protect users' historical trajectories.

Some location based applications need to hide the exact locations of individuals. Chatzikokolakis et al. [92] proposed a new notion, *geo-indistinguishability*, which protects an individual's exact location, while disclosing enough location information to obtain the desired service. Its main idea relates to a differential privacy level of the radius that the individual has visited: for any radius $\vartheta > 0$, an individual will have (ϵ, ϑ)-privacy.

Chen et al. [93] presented a data-dependent solution for sanitizing large-scale trajectory data. They developed a constrained inference technique to increase the resulting utility. He et al. [94] presented a system to synthesize mobility data based on raw GPS trajectories. They discretize raw trajectories using hierarchical reference systems to capture individual movements at differing speeds. They then propose an adaptive mechanism to select a small set of reference systems to construct prefix tree counts. Lastly, a direction-weighted sampling is applied to improve utility.

Even though previous works provide various solutions to location privacy, there are still some open questions. For example, most applications require the system to release a synthetic location dataset rather than query answers, and this is a tough problem in differential privacy. Moreover, privacy issues in spatial crowdsourcing is another emerging research area [95].

5.2 Differentially Private Recommender Systems

Recommender systems are one of the most popular applications in e-commerce and online social networks. They are capable of recommending products users will probably

TABLE 15: Comparison of the sample complexity of different methods

Method	References	Description	Sample complexity	Privacy level
Original	Kasiviswanathan et al. [45], Blum et al. [14]	Uses an exponential mechanism to search h	$O(VC(\mathcal{C})log\|\mathcal{X}\|)$	ϵ
Relaxing privacy level	Beimel et al. [10]	from ϵ to ϵ, δ	$O(\log(1/\delta))$	(ϵ, δ)
	Steinke et al. [84]	from ϵ to ϵ, δ	$O(\log(\sqrt{d \cdot 1/\delta}))$	(ϵ, δ)
	Chaudhuri et al. [85]	Only preserve privacy for labels	$\Omega(d')$ (d' was the adjusted dimension which is lower than d)	(ϵ)
Relaxing hypothesis	Beimel et al. [87] [86]	$H \neq \mathcal{C}$ and set a group of H to privately select a h.	$O(\max(ln\|H_i\|))$	ϵ
Semi-supervised Learning	Beimel [88]	use labeled data to search h	$O(dVC(\mathcal{C}))$ (labeled) $O(VC(\mathcal{C}))$ (unlabeled)	ϵ

like. Collaborative filtering (CF) is one of the most popular recommendation techniques as it is insensitive to the product details. However, a risk of potential privacy leaks exist in the recommendation process. For example, continual observation of the recommendations with some background information enables an adversary to infer the individuals rating or purchase history [96].

McSherry et. al. [97] introduced a differential privacy mechanism to traditional recommender algorithms, adding Laplace noise into the covariance matrix. Zhu et al. [98] proposed a private k nearest neighbor (KNN) algorithm for neighborhood-based recommender systems while minimizing the accuracy loss of the recommendations. The algorithm can resist attacks on collaborative filtering and has the potential to be extended to other scenarios that need to perform top-k private selection from candidates, such as top-k queries. Friedman et al. [99] presented a differentially private SVD algorithm to deal with the privacy problem in matrix factorization based recommender systems.

The majority of work on private recommender systems assume attackers are either public users or recommender servers. Liu et al. [100] proposed a hybrid approach which combines differential privacy with randomized perturbation not only to hide users private data from servers but also prevent privacy inference from pubic users. In practice, randomized perturbation can sometime improve the accuracy of the recommendations.

5.3 Differential Privacy in Genetic Data

Given the advances in genome sequencing technology, highly detailed genetic data are being generated inexpensively at exponential rates. The collection and analysis of these data may accelerate biomedical discoveries and support various applications, including personalized medical services. Despite all the benefits, Naveed et al. [101] reviewed the mitigation strategies for a wide variety of attacks and contextualized them from the perspective of medicine and public policy.

Several initial genome-wide association studies (GWAS) have explored the potential of differentially private data publishing by shifting the original location of the variants. Johnson et al. [102] developed a framework for *ab initio* exploration of case-control GWAS. This framework provides privacy-preserving answers to some key GWAS queries. They designed the operators to output the differentially private number of SNPs associated with the disease and other related statistical results. By considering protection against set membership disclosure, Tramer et al. [103] proposed a relaxation of the adversarial model of differential privacy and showed that this weaker setting achieves higher utility.

6 Future Directions

6.1 Adaptive Data Analysis

Adaptive data analysis creates a connection between differential privacy and machine learning. Machine learning theory is well developed, but is based on the assumption that a learning algorithm operates on a freshly sampled dataset. However, data samples are often reused, and the practice of learning is naturally adaptive. This adaptivity violates standard generalization guarantees. It is easy to over-fit the data when applying adaptive data analysis. Differential privacy can solve this adaptive problem.

Dwork et al. [104], [105] proposed a method that can perform arbitrary adaptive data with solid generalization guarantees. In this method, the algorithm's stability is assumed to prevent over-fitting and differential privacy is considered to be an algorithmic stability guarantee. Thus, a quantitative error bound is provided in terms of generality.

This line of research proves that differential privacy has a generalization property that can be applied in statistical analysis and machine learning [106], [107]. Although, at this early stage, many new possibilities for generalization need to be explored.

6.2 Local Differential Privacy

The majority of work on differential privacy assumes that the curator is trustworthy; however, as more organizations have resorted to distributed systems for data acquisition, storage, and analysis, it has been noted that curators are less reliable in this distributed context.

Recently, a Local Differential Privacy (LDP) model has been proposed to address the issue. Noise is added on the user side before submitting the results to an un-trusted data curator. The curator then conducts post-processing on them to obtain acceptable statistics, which can be further shared with the public. Here in the statistics types, which are available for the data curator to obtain would be limited to the population statistics and depend on both the design of the local perturbation mechanism and the post-processing mechanism.

The local privacy model was first formalized in [45], and then a theoretic upper bound under the LDP model was

TABLE 16: Three aspects of the data lifecycle in local differential privacy

Original data type	Uploaded data type	Post-process target
single numeric or categorical attribute	noisy attribute value	value estimation [109]; distribution estimation [110]
multiple numeric or categorical attribute	noisy attributes values	value estimation [109]; mean and frequency estimation, complex machine learning tasks [111]; multi-dimensional joint distribution estimation [112]
set-valued data	a random bit of noisy set-valued data [113]; noisy set-valued data [114]	frequent item set estimation [113]; discrete distribution estimation [114]
encoding data	a random bit of the encoding of location	count estimation [115]

provided by Duchi et al. [108]. Typically two main research questions have been investigated:

1) How to design acceptable LDP mechanisms for different original data types generated by distributed users?
2) How to design acceptable LDP mechanisms that can achieve different analysis targets such as value estimation, distribution estimation and more complex machine learning tasks?

In the past few years, related LDP works have focused on three aspects: the existing data types in user nodes (original data type), the data types submitted to data curators by users (uploaded data type), and the targets to achieve by analyzing the uploaded data (post-process target). Table 16 gives an overview of these three data types.

6.3 Differential Privacy for a Coupled Information

Existing research assumes that records are sampled independently from a data universe. However, in real-world applications, records are rarely independent. The relationships among records are referred to as coupled information. A differential privacy technique performed on coupled information will disclose more information than expected, and this indicates a serious privacy violation [116], [117].

To deal with this problem, Kifer et al. proposed two privacy frameworks, *Pufferfish* [118] and *Blowfish* [117], to allow application domain experts to add an extra data relationship to develop a customized privacy definition. Based on *Pufferfish*, Yang et al. [119] proposed Bayesian differential privacy to deal with correlation problems. Wang et al. [120] observed that *non-independent reasoning* (NIR) allows the information about one record in the data to be learned from the information of other records in the data. Chen et al. [121] considered social networks and dealt with the coupled problem by multiplying the sensitivity with the number of coupled records. Zhu et al. [122] proposed a coupled sensitivity and designed a coupled data release mechanism.

Generally, when addressing privacy issues in a coupled dataset, the key problem is identifying which information is coupled, then modeling the extra background information introduced by coupled records. These are still open issues that need to be explored further.

6.4 Differential Privacy and Mechanism Design

Mechanism design is a kind of game in which inputs are delivered by multiple agents. However, agents may manipulate their inputs, so the mechanisms design should ask agents to provide true inputs [123].

McSherry [11] first proposed a design for auction mechanisms using differentially private mechanisms as the building blocks. This private mechanism is only approximately truthful. Later, Nissim et al. [124] converted these private mechanisms into fully truthful mechanisms. Cummings et al. [125] studied the multi-dimensional aggregative games and solved the equilibrium selection problem. Barthe et al. [123] introduced a relational refinement system for verifying a mechanism's design in terms of differential privacy.

This series of research focuses on mechanism design based on differential privacy. They are good representatives for the interdisciplinary area of differential privacy and game theory.

6.5 Relaxation of Differential Privacy

Differential privacy is indeed an excellent definition to guarantee strict privacy levels. The most attractive feature is that the privacy level degrades smoothly under composition theory. As pure differential privacy is too strict in practical terms, a relaxation of differential privacy is highly desired. Approximate or (ϵ, δ)-differential privacy is widely used as a method of relaxation and has been discussed heavily in this paper. However, the bound analysis for the composition of approximate differential privacy is hard [126], even for simple computations, making it inconvenient for use in real-world applications.

Dwork et al. [127] developed a new relaxation technique called concentrated differential privacy, in which the privacy loss in multiple computations has a small mean and is sub-Gaussian. Based on this new technique, Bun [128] proposed zero-concentrated differential privacy to minimize the mean of privacy losses to zero, to improve the bound of δ. These relaxation definitions need to be further explored. For example, designing randomized algorithms to achieve the new definitions and analyzing the sample complexity of those algorithms still remain problems. Generally, previous work on differential privacy can be re-visited through the perspective of new definitions, which may lead to new insights and results.

7 Conclusions

This paper presents a multi-disciplinary survey of work on differential privacy including an overview of the huge amount of literature in two major differential privacy research streams: data publishing and data analysis. We identified different publishing mechanisms for data publishing and compared various types of input and output data. In addition, we presented two basic dataset publishing

methods: anonymized and learning-based. We discussed two basic frameworks for data analysis and illustrated their respective analysis scenarios. The basic techniques in differential privacy look simple and intuitively appealing, and when combined with specific problems, differential privacy demonstrates itself as a powerful and useful tool for a diverse range of applications.

Differential privacy still has much unknown potential, and the summarized literature in this paper is intended as a starting point for exploring new challenges in the future. Our goal is to provide an overview of existing works on differential privacy to show their use to newcomers, as well as experienced practitioners, in various fields. We also hope this review will help to prevent redundant or ad hoc efforts for both researchers and industry.

REFERENCES

[1] C. C. Aggarwal and P. S. Yu, Eds., *Privacy-Preserving Data Mining - Models and Algorithms*, ser. Advances in Database Systems. Springer, 2008, vol. 34.

[2] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, 2010.

[3] C. Dwork, "Differential privacy," in *ICALP*, 2006, pp. 1–12.

[4] ——, "Differential privacy: a survey of results," in *TAMC'08*, 2008, pp. 1–19.

[5] ——, "Differential privacy in new settings," in *SODA '10*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2010, pp. 174–183.

[6] ——, "A firm foundation for private data analysis," *Commun. ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[7] A. D. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 86–94, 2013.

[8] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, Aug. 2014.

[9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*, 2006, pp. 486–503.

[10] A. Beimel, K. Nissim, and U. Stemmer, "Private learning and sanitization: Pure vs. approximate differential privacy," *CoRR*, vol. abs/1407.2674, 2014.

[11] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *FOCS*, 2007, pp. 94–103.

[12] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," *Commun. ACM*, vol. 53, no. 9, 2010.

[13] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006, pp. 265–284.

[14] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," in *STOC*, 2008, pp. 609–618.

[15] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *PODS*, 2003, pp. 202–210.

[16] A. Roth and T. Roughgarden, "Interactive privacy via the median mechanism," in *STOC*, 2010, pp. 765–774.

[17] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *FOCS*, 2010, pp. 61–70.

[18] A. Gupta, A. Roth, and J. Ullman, "Iterative constructions and private data release," in *TCC*, 2012, pp. 339–356.

[19] Z. Huang and A. Roth, "Exploiting metric structure for efficient private query release," in *SODA*, 2014, pp. 523–534.

[20] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, 2013.

[21] W. Qardaji, W. Yang, and N. Li, "Understanding hierarchical methods for differentially private histograms," *Proc. VLDB Endow.*, vol. 6, no. 14, pp. 1954–1965, 2013.

[22] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proc. VLDB Endow.*, vol. 3, no. 1, pp. 1021–1032, 2010.

[23] B. Lin and D. Kifer, "Information preservation in statistical privacy and bayesian estimation of unattributed histograms," in *SIGMOD*, 2013, pp. 677–688.

[24] J. Lee, Y. Wang, and D. Kifer, "Maximum likelihood postprocessing for differential privacy under consistency constraints," in *KDD*, 2015, pp. 635–644.

[25] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin, "Pan-private streaming algorithms," in *Innovations in Computer Science*, 2010, pp. 66–80.

[26] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 3, pp. 26:1–26:24, 2011.

[27] J. Zhang, X. Xiao, and X. Xie, "Privtree: A differentially private algorithm for hierarchical decompositions," in *SIGMOD*, 2016, pp. 155–170.

[28] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *STOC*, 2010, pp. 715–724.

[29] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proc. VLDB Endow.*, vol. 7, no. 12, pp. 1155–1166, 2014.

[30] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *STOC*, 2007, pp. 75–84.

[31] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *ACM Trans. Database Syst.*, vol. 39, no. 3, pp. 22:1–22:33, 2014.

[32] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Private release of graph statistics using ladder functions," in *SIGMOD*, 2015, pp. 731–745.

[33] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with node differential privacy," in *TCC*, 2013, pp. 457–476.

[34] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "Differentially private data analysis of social networks via restricted sensitivity," in *ITCS*, 2013, pp. 87–96.

[35] S. Chen and S. Zhou, "Recursive mechanism: Towards node differential privacy and unrestricted joins," in *SIGMOD*, 2013, pp. 653–664.

[36] X. Xiao, G. Wang, and J. Gehrke, "Differential privacy via wavelet transforms," *IEEE Trans. on Knowl. and Data Eng.*, vol. 23, no. 8, pp. 1200–1214, 2011.

[37] C. Li, M. Hay, G. Miklau, and Y. Wang, "A data- and workload-aware query answering algorithm for range queries under differential privacy," *Proc. VLDB Endow.*, vol. 7, no. 5, pp. 341–352, 2014.

[38] C. Li, G. Miklau, M. Hay, A. McGregor, and V. Rastogi, "The matrix mechanism: optimizing linear counting queries under differential privacy," *The VLDB Journal*, vol. 24, no. 6, pp. 1–25, 2015.

[39] D. Huang, S. Han, X. Li, and P. S. Yu, "Orthogonal mechanism for answering batch queries with differential privacy," in *SSDBM*, 2015, pp. 24:1–24:10.

[40] G. Yuan, Z. Zhang, M. Winslett, X. Xiao, Y. Yang, and Z. Hao, "Optimizing batch linear queries under exact and approximate differential privacy," *ACM Trans. Database Syst.*, vol. 40, no. 2, pp. 11:1–11:47, 2015.

[41] G. Yuan, Y. Yang, Z. Zhang, and Z. Hao, "Convex optimization for linear query processing under approximate differential privacy," in *SIGKDD*, 2016, pp. 2005–2014.

[42] G. Kellaris and S. Papadopoulos, "Practical differential privacy via grouping and smoothing," in *PVLDB*, 2013, pp. 301–312.

[43] X. Xiao, G. Bender, M. Hay, and J. Gehrke, "ireduct: differential privacy with reduced relative errors," in *SIGMOD*, 2011, pp. 229–240.

[44] N. Mohammed, R. Chen, B. C. Fung, and P. S. Yu, "Differentially private data release for data mining," in *SIGKDD*, 2011, pp. 493–501.

[45] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" in *FOCS*, 2008, pp. 531–540.

[46] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *FOCS*, 2010, pp. 51–60.

[47] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," in *NIPS*, 2012, pp. 2348–2356.

[48] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: efficient algorithms and hardness results," in *STOC*, 2009, pp. 381–390.

[49] J. Ullman, "Answering n2+O(1) counting queries with differential privacy is hard," in *STOC*, 2013, pp. 361–370.

[50] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: private data release via bayesian networks," in *SIGMOD*, 2014, pp. 1423–1434.

[51] R. Chen, Q. Xiao, Y. Zhang, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference," in *SIGKDD*, 2015, pp. 129–138.

[52] H. Jiawei and M. Kamber, "Data mining: concepts and techniques," *San Francisco, CA, itd: Morgan Kaufmann*, vol. 5, 2001.

[53] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the sulq framework," in *PODS*, 2005, pp. 128–138.

[54] A. Friedman and A. Schuster, "Data mining with differential privacy," in *SIGKDD*, 2010, pp. 493–502.

[55] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," *Transactions on Data Privacy*, vol. 5, no. 1, pp. 273–295, 2012.

[56] S. Rana, S. K. Gupta, and S. Venkatesh, "Differentially private random forest with high utility," in *ICDM*, 2015, pp. 955–960.

[57] Y. Wang, Y. Wang, and A. Singh, "Differentially private subspace clustering," in *NIPS*, 2015, pp. 1000–1008.

[58] ——, "A theoretical analysis of noisy sparse subspace clustering on dimensionality-reduced data," *CoRR*, vol. abs/1610.07650, 2016.

[59] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett, "Privgene: Differentially private model fitting using genetic algorithms," in *SIGMOD*, 2013, pp. 665–676.

[60] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler, "Gupt: Privacy preserving data analysis made easy," in *SIGMOD*, 2012, pp. 349–360.

[61] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in *SIGKDD*, 2010, pp. 503–512.

[62] N. Li, W. Qardaji, D. Su, and J. Cao, "Privbasis: Frequent itemset mining with differential privacy," *Proc. VLDB Endow.*, vol. 5, no. 11, pp. 1340–1351, 2012.

[63] C. Zeng, J. F. Naughton, and J.-Y. Cai, "On differentially private frequent itemset mining," *Proc. VLDB Endow.*, vol. 6, no. 1, pp. 25–36, 2012.

[64] J. Lee and C. W. Clifton, "Top-k frequent itemsets via differentially private fp-trees," in *SIGKDD*, 2014, pp. 931–940.

[65] E. Shen and T. Yu, "Mining frequent graph patterns with differential privacy," in *SIGKDD*, 2013, pp. 545–553.

[66] S. Xu, S. Su, L. Xiong, X. Cheng, and K. Xiao, "Differentially private frequent subgraph mining," in *ICDE*, 2016, pp. 229–240.

[67] Y. Wang, J. Lei, and S. E. Fienberg, "Learning with differential privacy: Stability, learnability and the sufficiency and necessity of ERM principle," *CoRR*, vol. abs/1502.06309, 2015.

[68] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *NIPS 2014*, 2008, pp. 289–296.

[69] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, no. 2, pp. 1069–1109, 2011.

[70] P. Jain and A. Thakurta, "Differentially private learning with kernels," in *ICML*, 2013, pp. 118–126.

[71] R. Hall, A. Rinaldo, and L. Wasserman, "Differential privacy for functions and functional data," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 703–727, 2013.

[72] R. Bassily, A. D. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *FOCS*, 2014, pp. 464–473.

[73] J. Ullman, "Private multiplicative weights beyond linear queries," in *PODS*, 2015, pp. 303–312.

[74] S. P. Kasiviswanathan, K. Nissim, and H. Jin, "Private incremental regression," *CoRR*, vol. abs/1701.01093, 2017.

[75] D. Kifer, A. D. Smith, and A. Thakurta, "Private convex optimization for empirical risk minimization with applications to high-dimensional regression," in *COLT*, 2012, pp. 25.1–25.40.

[76] A. G. Thakurta and A. Smith, "Differentially private feature selection via stability arguments, and the robustness of the lasso," in *Conference on Learning Theory*, 2013, pp. 819–850.

[77] P. Jain and A. G. Thakurta, "(near) dimension independent risk bounds for differentially private learning," in *ICML*, 2014, pp. 476–484.

[78] S. P. Kasiviswanathan and H. Jin, "Efficient private empirical risk minimization for high-dimensional learning," in *ICML*, 2016, pp. 488–497.

[79] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *CCS*, 2016, pp. 308–318.

[80] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *SIGSAC*, 2015, pp. 1310–1321.

[81] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: an application of human behavior prediction," in *AAAI*, 2016, pp. 1309–1316.

[82] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for SVM learning," *CoRR*, vol. abs/0911.5708, 2009.

[83] M. J. Kearns and U. V. Vazirani, "An introduction to computational learning theory," vol. 8, no. 2001, pp. 44–58, 1994.

[84] T. Steinke and J. Ullman, "Between pure and approximate differential privacy," *CoRR*, vol. abs/1501.06095, 2015.

[85] K. Chaudhuri and D. Hsu, "Sample complexity bounds for differentially private learning," in *COLT*, 2011, pp. 155–186.

[86] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," in *TCC*, 2010, pp. 437–454.

[87] A. Beimel, K. Nissim, and U. Stemmer, "Characterizing the sample complexity of private learners," in *ITCS*, 2013, pp. 97–110.

[88] ——, "Learning privately with labeled and unlabeled examples," in *SODA*, 2015, pp. 461–477.

[89] L. Stenneth and P. S. Yu, "Mobile systems privacy: 'mobipriv' A robust system for snapshot or continuous querying location based mobile systems," *Transactions on Data Privacy*, vol. 5, no. 1, pp. 333–376, 2012.

[90] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in *ICDE*, April 2012, pp. 20–31.

[91] J. D. Zhang, G. Ghinita, and C. Y. Chow, "Differentially private location recommendations in geosocial networks," in *MDM*, vol. 1, July 2014, pp. 59–68.

[92] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "Location privacy via geo-indistinguishability," *ACM SIGLOG News*, vol. 2, no. 3, pp. 46–69, 2015.

[93] R. Chen, B. C. Fung, B. C. Desai, and N. M. Sossou, "Differentially private transit data publication: a case study on the montreal transportation system," in *SIGKDD*, 2012, pp. 213–221.

[94] X. He, G. Cormode, A. Machanavajjhala, C. M. Procopiuc, and D. Srivastava, "Dpt: Differentially private trajectory synthesis using hierarchical reference systems," *Proc. VLDB Endow.*, vol. 8, no. 11, pp. 1154–1165, 2015.

[95] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proc. VLDB Endow.*, vol. 7, no. 10, pp. 919–930, 2014.

[96] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, ""you might also like: " privacy risks of collaborative filtering," in *SP'11*, 2011, pp. 231–246.

[97] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the net," in *SIGKDD*, 2009, pp. 627–636.

[98] T. Zhu, Y. Ren, W. Zhou, J. Rong, and P. Xiong, "An effective privacy preserving algorithm for neighborhood-based collaborative filtering," *Future Generation Comp. Syst.*, vol. 36, pp. 142–155, 2014.

[99] A. Friedman, S. Berkovsky, and M. A. Kaafar, "A differential privacy framework for matrix factorization recommender systems," *User Modeling and User-Adapted Interaction*, vol. 26, no. 5, pp. 425–458, Dec. 2016.

[100] X. Liu, A. Liu, X. Zhang, Z. Li, G. Liu, L. Zhao, and X. Zhou, "When differential privacy meets randomized perturbation: A hybrid approach for privacy-preserving recommender system," in *DASFAA*, vol. 1, 2017, pp. 576–591.

[101] M. Naveed, E. Ayday, E. W. Clayton, J. Fellay, C. A. Gunter, J.-P. Hubaux, B. A. Malin, and X. Wang, "Privacy in the genomic era," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 6:1–6:44, 2015.

[102] A. Johnson and V. Shmatikov, "Privacy-preserving data exploration in genome-wide association studies," in *SIGKDD*, 2013, pp. 1079–1087.
[103] F. Tramèr, Z. Huang, J.-P. Hubaux, and E. Ayday, "Differential privacy with bounded priors: Reconciling utility and privacy in genome-wide association studies," in *CCS*, 2015, pp. 1286–1297.
[104] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. L. Roth, "Preserving statistical validity in adaptive data analysis," in *STOC*, 2015, pp. 117–126.
[105] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth, "Generalization in adaptive data analysis and holdout reuse," in *NIPS*, 2015, pp. 2350–2358.
[106] K. Nissim and U. Stemmer, "On the generalization properties of differential privacy," *CoRR*, vol. abs/1504.05800, 2015.
[107] R. Bassily, A. Smith, T. Steinke, and J. Ullman, "More general queries and less generalization error in adaptive data analysis," *CoRR*, vol. abs/1503.04843, 2015.
[108] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *FOCS*, 2013, pp. 429–438.
[109] Y. Wang, X. Wu, and D. Hu, "Using randomized response for differential privacy preserving data collection," in *EDBT/ICDT Workshops*, 2016.
[110] G. C. Fanti, V. Pihur, and Ú. Erlingsson, "Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries," *PoPETs*, vol. 2016, pp. 41–61, 2016.
[111] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," *CoRR*, vol. abs/1606.05053, 2016.
[112] X. Ren, C. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and P. S. Yu, "Lopub: High-dimensional crowdsourced data publication with local differential privacy," *CoRR*, vol. abs/1612.04350, 2016.
[113] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *SIGSAC*, 2016, pp. 192–203.
[114] S. Wang, L. Huang, P. Wang, Y. Nie, H. Xu, W. Yang, X. Li, and C. Qiao, "Mutual information optimally local private discrete distribution estimation," *CoRR*, 2016.
[115] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *ICDE*, 2016, pp. 289–300.
[116] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *SIGMOD*, 2011, pp. 193–204.
[117] A. Machanavajjhala and D. Kifer, "Designing statistical privacy for your data," *Commun. ACM*, vol. 58, no. 3, pp. 58–67, 2015.
[118] D. Kifer and A. Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions," *ACM Trans. Database Syst.*, vol. 39, no. 1, pp. 3:1–3:36, 2014.
[119] B. Yang, I. Sato, and H. Nakagawa, "Bayesian differential privacy on correlated data," in *SIGMOD*, 2015, pp. 747–762.
[120] K. Wang, C. Han, A. W. Fu, R. C. Wong, and P. S. Yu, "Reconstruction privacy: Enabling statistical learning," in *EDBT*, 2015, pp. 469–480.
[121] R. Chen, B. C. Fung, P. S. Yu, and B. C. Desai, "Correlated network data publication via differential privacy," *The VLDB Journal*, vol. 23, no. 4, pp. 653–676, 2014.
[122] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: Hiding information in non-iid data set," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 229–242, 2015.
[123] G. Barthe, M. Gaboardi, E. J. Gallego Arias, J. Hsu, A. Roth, and P.-Y. Strub, "Higher-order approximate relational refinement types for mechanism design and differential privacy," in *POPL*, 2015, pp. 55–68.
[124] K. Nissim, R. Smorodinsky, and M. Tennenholtz, "Approximately optimal mechanism design via differential privacy," in *Innovations in Theoretical Computer Science*, 2012, pp. 203–213.
[125] R. Cummings, M. Kearns, A. Roth, and Z. S. Wu, "Privacy and truthful equilibrium selection for aggregative games," *CoRR*, vol. abs/1407.7740, 2014.
[126] J. Murtagh and S. P. Vadhan, "The complexity of computing the optimal composition of differential privacy," in *TCC*, 2016, pp. 157–175.
[127] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *CoRR*, vol. abs/1603.01887, 2016.
[128] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *TCC*, 2016, pp. 635–658.

Tianqing Zhu Tianqing Zhu received her BEng and MEng degrees from Wuhan University, China, in 2000 and 2004, respectively, and a PhD degree from Deakin University in Computer Science, Australia, in 2014. Dr Tianqing Zhu is currently a teaching scholar in the School of Information Technology, Deakin University, Australia. Before joining Deakin University, she served as a lecturer in Wuhan Polytechnic University, China from 2004 to 2011. Her research interests include privacy preserving, data mining and network security. She has won the best student paper award in PAKDD 2014.

Gang Li Gang Li received his PhD in computer science from Deakin University (Australia) in 2005, and currently an associate professor in the school of IT, Deakin University. His research interests are in the area of data mining, machine learning and business intelligence. He has co-authored four papers that won best paper prizes, including the PAKDD2014 best student paper, ACM/IEEE ASONAM2012 best paper award, the 2007 Nightingale Prize by Springer journal Medical and Biological Engineering and Computing. He served on the Program Committee for over 100 international conferences in artificial intelligence, data mining and machine learning, tourism and hospitality management.

Wanlei Zhou Professor Wanlei Zhou received the B.Eng and M.Eng degrees from Harbin Institute of Technology, Harbin, China in 1982 and 1984, respectively, and the PhD degree from The Australian National University, Canberra, Australia, in 1991, all in Computer Science and Engineering. He also received a DSc degree from Deakin University in 2002. He is currently the Alfred Deakin Professor and Chair of Information Technology, School of Information Technology, Deakin University. Professor Zhou has published more than 300 papers in refereed international journals and refereed international conferences proceedings. He has also chaired many international conferences and has been invited to deliver keynote address in many international conferences.

Philip S. Yu Philip S. Yu received the B.S. Degree in E.E. from National Taiwan University, the M.S. and Ph.D. degrees in E.E. from Stanford University, and the M.B.A. degree from New York University. He is a Distinguished Professor in Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. Before joining UIC, Dr. Yu was with IBM, where he was manager of the Software Tools and Techniques department at the Watson Research Center. His research interest is on big data, including data mining, data stream, database and privacy. He has published more than 1,000 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. Dr. Yu is a Fellow of the ACM and the IEEE. He is the Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data. Dr. Yu is the recipient of ACM SIGKDD 2016 Innovation Award for his influential research and scientific contributions on mining, fusion and anonymization of big data, the IEEE Computer Societys 2013 Technical Achievement Award for pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining and anonymization of big data.