DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING, SECOND CYCLE, 30 CREDITS

STOCKHOLM, SWEDEN 2018

Utility of Differentially Private Synthetic Data Generation for High-Dimensional Databases

DAAN JOSEPHUS KNOORS

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Abstract

When processing data that contains sensitive information, careful consideration is required with regard to privacy-preservation to prevent disclosure of confidential information. Privacy engineering enables one to extract valuable patterns, safely, without compromising anyone's privacy. Over the last decade, academics have actively sought to find stronger definitions and methodologies to achieve data privacy while preserving the data utility. Differential privacy emerged and became the de facto standard for achieving data privacy and numerous techniques are continuously proposed based on this definition. One method in particular focuses on the generation of private synthetic databases, that mimic statistical patterns and characteristics of a confidential data source in a privacy-preserving manner. Original data format and utility is preserved in a new database that can be shared and analyzed safely without the risk of privacy violation. However, while this privacy approach sounds promising there has been little application beyond academic research. Hence, we investigate the potential of private synthetic data generation for real-world applicability. We propose a new utility evaluation framework that provides a unified approach upon which various algorithms can be assessed and compared. This framework extends academic evaluation methods by incorporating a user-oriented perspective and varying industry requirements, while also examining performance on real-world use cases. Finally, we implement multiple general-purpose algorithms and evaluate them based on our framework to ultimately determine the potential of private synthetic data generation beyond the academic domain.

Abstract

Vid databehandling av känslig information måste särskild hänsyn tas till sekretessbevarande för att undvika oavsiktligt röjande av konfidentiell information. Med skretessingenjörsskap menas möjliggörandet av informationssäker mönsterextraktion utan att kompromissa någons rätt till ett privatliv. Under det senaste decenniet har akademiker aktivt försökt finna starkare definitioner och metodiker för att uppnå ett sekretessbevarande men endå bibehålla datats nytta. Differentielt hemlighållande (eng. Differential Privacy) framkom som en "de facto" standard för att uppnå sekretessbevarande och det föreslås kontinuerligt nya tekniker baserade på denna. I synnerhet en metod fokuserar på generering av privata syntetiska databaser vilka härmar de statistiska mönster och särdrag från en konfidentiell datakälla på ett sekretessbevarande sätt. På grund av detta kan originaldatats format och nytta bibehållas men fortfarande delas och analyseras utan risk för sekretessöverträdelser. Tyvärr har denna metod sett liten tillämpning utanför akademia. Därför undersöker vi härmed dess potential för användande av hemlighållande syntetisk datagenerering i verkliga användarfall. Vi föreslår vidare ett nytt nyttjandegradsutvärderingramverk vilket ger ett enhetligt sätt att utvärdera diverse algorithmer gentemot varandra. Detta ramverk bygger vidare på de typiska akademiska utvärderingsmetoderna genom att inkorporera ett användarorienterat perspektiv och industrikrav samtidigt som prestandautvärdering av verkliga användarfall görs. Slutligen implementerar vi flera algoritmer med allmänt ändamål och utvärderar dem utifrån kriterierna för detta ramverk med syftet att i slutändan bestämma potentialen för hemlighållande syntetisk datagenerering utanför den akademiska domänen.

Acknowledgements

Here, I would like to express my gratitude to the following people who have been essential to the completion of this work. Without their guidance and support this research would not have been possible.

Before the research project commenced, I came into contact with Leila Bahri who agreed to be my KTH supervisor. Her past experience in data privacy helped accelerate my understanding of this field. Her constant guidance, engagement and advice provided me with a clear direction with regard to the structure of this thesis. I am very grateful for her suggestions and constant review of my work.

As part of this research, I conducted an internship at privacy engineering start-up Privitar in London. I am immensely grateful for this opportunity and would like to thank everyone at Privitar for allowing me to work in a such a stimulating environment. They provided me the resources and constant assistance which has been invaluable to this project. Additionally, they allowed me to experience the working atmosphere at a fast-paced, rapidly growing start-up and I truly enjoyed the collaboration with my colleagues.

In particular, my sincerest thanks go to my supervisors at Privitar, Charlie Cabot and Theresa Stadler, who gave me the opportunity to work on such a rewarding research project. They provided the direction, knowledge and vision to make this project a success. They integrated me into their research team and were constantly engaged with my work. Both of them always encouraged me to learn new skills and gave me the freedom to delve into unfamiliar fields, which has been great for my personal development. They inspired me to work as hard as I could and I wanted their efforts to be rewarded with a meaningful outcome. Again, I am profoundly grateful for this experience and the advice they have given me.

Daan Knoors
August, 2018

Contents

List of Figures

List of Tables

List of Abbreviations

ACS	American Community Survey.
AVD	Average Variation Distance.
DB	Database.
DM	Data Mining.
DP	Differential Privacy.
GDPR	General Data Protection Regulation.
JS	Jensen-Shannon (- Distance/Divergence).
MWEM	Multiplicative Weights Exponential Mechanism.
PET	Privacy Enhancing Technologies.
PII	Personal Identifiable Information.
SEP	Scale-Epsilon Pair.
TVD	Total Variation Distance.

Chapter 1

Introduction

A common challenge in research and industry is the protection of privacy-sensitive data, particularly with regard to storage, processing and sharing. As more data gets collected about people, proper safeguards are required to ensure that this data can get analyzed and distributed without the risk of violating anyone's right to privacy [1]. Sensitive information remains a personal issue and what one considers private differs from person to person, whether this is medical history, salary, direct messages, religious beliefs, online behaviour, travel pictures or visited locations. Having the ability to control who has access to our personal information is often imperative for the creation and maintenance of social relationships [2]. Disclosure of such information can have severe consequences on a person's reputation, relationships and well-being. People rely more on services that require personal information for payments, health care, social media, and transportation. As more of this data gets collected and shared, the risk of privacy violation increases drastically. Hence, the field of data privacy has grown to address the important challenge of processing and sharing private information in a protected manner. Fortunately, users and engineers are becoming more aware of the value of their data and what information can be inferred, stimulated further by recent continent-wide regulation changes, such as the General Data Protection Regulation (GDPR)[1]. Industry and research is slowly shifting to the privacy engineering perspective, however real-world implementation remains limited [3]–[5].

1.1 Background

Beyond traditional statistical inference, data-driven approaches are starting to drive technological advancement for diverse research and commercial purposes. However, this data often contains sensitive information that can be analyzed by adversaries or even leaked to the public domain. An important first step is to install strong security measures to limit access to the data and reduce the risk of privacy violation. However in order to truly guarantee strong privacy protection, one should maintain a privacy by design approach by equipping proper safeguards with regard to data collection, storage, analytics and sharing. Moreover recent regulation changes GDPR[1], enforced from the 25th of May 2018, obligates entities storing or processing data of European data subjects to truly evaluate their privacy-preserving practices.

1.1.1 Security vs. Privacy

In terms of data protection, there is an important distinction to be made between security and privacy [6]. Security provides measures that limit unauthorized access. Hence, only a small set of people can view the protected source and thus also protects the confidential information from the public. It is an important first step to minimize the disclosure of sensitive information and does provide some level of privacy. However, this method still implies that a few authorized personnel

[1]General Data Protection Regulation: https://www.eugdpr.org/

can view very sensitive information which in itself could be privacy concern. Additionally, a data breach remains a constant threat, whether this is through an unintentional mistake or malicious intent. Data security thus provides strong measures to protect data sources but ultimately cannot provide strong privacy guarantees.

Privacy is the right for an individual to be free from uninvited attention and scrutiny [1]. Where security provides protection of confidential data through preventing unauthorized access, privacy governs how data is being stored, analyzed or shared. Privacy Enhancing Technologies (PET) are a system of ICT that measure the protection of informational privacy by eliminating or minimizing personal data thereby preventing unnecessary or unwanted processing of personal data, without losing the functionality of the information system [7].

PET enables privacy engineering, i.e. protecting confidential databases by minimizing personal data and thus unnecessary processing of sensitive information while preserving the utility of the information system. Again, sensitivity of information is dependent on the data source and an individual's perception, e.g. medical history records are typically considered more personal information than nationality or gender. This information can only become sensitive when it can be linked to a specific entity. This is accomplished through Personal Identifiable Information (PII), which enable one to distinguish an individual among a set of records. This can either be based on direct identifiers (e.g. name, id number) or quasi-identifiers (e.g. demographics, location history) which even in the absence of direct identifiers can lead to re-identification with prior knowledge or integration of external data sources. Hence, data privacy engineering reduces the likelihood of a person being identified through PII and thus the amount of sensitive information that can be inferred of any data subject. Therefore it not only protects secured data sources from privacy violations in case of unauthorized access, but also enables sharing of data without releasing personal information.

1.1.2 Privacy and Utility

One important aspect in the application of privacy engineering solutions is to measure the consequences it has on the data. Inherently, masking of personal attributes within a database results in some loss of data utility and perhaps even system functionality [8]. Therefore it is imperative to not only evaluate the privacy guarantee one can attain by using PETs, but also examine the balancing effect it has on utility. In general, stronger privacy guarantees result in higher losses of data utility and vice versa. Especially when sensitive data needs to be shared (e.g. health care analytics or transaction fraud analysis) utility needs to be preserved without compromising on privacy. This balancing act is discussed in more detail as part of the theoretic background in the next chapter and needs to be carefully considered when adopting a new privacy engineering perspective.

1.1.3 Privacy Approaches

Simply removing PII from databases is often not enough to provide strong privacy guarantees. Using prior knowledge about certain individuals or external data sources can in combination with these records lead to the identification of certain individuals, e.g. [9], [10]. For instance, in medical health records the patient numbers and names are emitted. However, through the presence of prior knowledge on one or more data subjects, e.g. patient demographics and recent hospital visits, one can still identify certain data records that might belong to a specific individual. As a result, this method does not provide adequate protection and new methods are required to limit the occurrence of privacy violation. One notable instance is the Netflix prize contest, where seemingly anonymized data was released to the public domain, only for the anonymization process to be reversed by two computer scientists [9].

Over the last two decades, academia have proposed various definitions, techniques and architectures for enabling privacy-preserving analytics [11]–[13]. Still, concepts remain complex and

the consequences on data utility are not always clear. There is constant research for new solutions that meet the requirements of real-world confidential data sources and processing tasks, ensuring strong privacy guarantees with minimal loss in utility. Methods with weaker privacy guarantees normally provide higher levels of utility, particularly when confidential data needs to be released. In general, they operate on the principles of suppression and generalization of sensitive information , e.g. k-anonymity [14], [15], l-diversity [16], or t-closeness [17]. All of these methods require data attributes to be distinguished in groups: identifiers (e.g. name, id number), quasi-identifiers (e.g. ZIP code, date of birth), and sensitive variables (e.g. treatment, salary). However sometimes it is not clear how these categories need to be defined, as the notion of private information differs per person. Moreover, in presence of external information about certain individuals in the data or integrating auxiliary databases, the quasi-identifiers can also lead to the re-identification of individuals, known as linkage attacks. This distinction of attribute categories can therefore become purely artificial. Hence, we see cases like the medical records of the governor of Massachusetts being identified when anonymized medical data was integrated with voter registration records [18], and Netflix subscribers were exposed when anonymized viewing histories were released and subsequently linked with records of the Internet Movie Database (IMDb) [9].

Evidently, we require new methods to prevent personal information disclosure with strong formal guarantees. Hence, Differential Privacy (DP) [13] has emerged to become the de facto standard for guaranteeing privacy even in the presence of other data or prior knowledge. Systems built on this definition, prevent attackers from being capable to detect the presence or absence of a certain individual in the database. It operates on a principle of noise addition, which creates uncertainty in released data. DP promises that even when an attacker knows everything about each data subject but one, the attacker still can not determine whether an individual is present in published statistics or even full DP generated databases. In the next chapter we provide the theoretic background for this definition, including technical concepts and the attractive guarantees it provides. However lets briefly examine its utility in various methods of private data release.

1.1.4 Differential Privacy Solutions

Evidently, the release of sensitive information of even a single person could already be considered a privacy violation. Therefore, often interesting information of individuals is combined to form an aggregate statistic (e.g. summary statistics, histograms, and charts) and is therefore considered safer to release. However, even a small amount of released statistics can already lead to the identification of individuals and subsequently learning confidential information. Either this set of aggregate statistics in their combination can reveal information about a particular individual, or can be integrated with external data leading to the inference of personal information. An individual can thus be singled out from a few aggregates statistics, which could result in a privacy breach. Usually this occurs when a person has very distinct information from the other subjects in the database. For instance, when we release health statistics about a particular town and we can break it down by an arbitrary demographic attribute (e.g. race, occupation, gender) this could lead to singling out one individual and learning their sensitive information. This is information the aggregate statistics did not intend to provide. One way to mitigate this issue is through differentially private algorithms by introducing a small amount of randomness in the aggregate statistics. For instance when the aggregate statistic represents a count of people containing a certain attribute (e.g. number of patients receiving treatment X), a small addition of randomness changes the exact number slightly. When attempting to cancel out the aggregate statistics through linear equations, formally known as a differencing attack, or even using prior knowledge one can never be certain whether the statistic is formed due to participation of a particular individual or due to introduction of noise. Hence, insights are preserved due to these small deviations but the individual's information remains protected.

However, today's industry data usage is not restricted to only the release of statistics. Data is used to discover patterns and new insights upon which automatized systems can be build. Hence, DP algorithms can not just be limited towards noisy statistics. They also lend themselves to the generation of synthetic databases, maintaining the properties of a traditional database format, while preventing linkage to auxiliary data sources. These generation schemes are also known as non-interactive privacy mechanisms [13]. Differentially private synthetic data generation is method that attempts to model patterns from confidential data sources, mimicking the statistical characteristics in a private manner. From this model new records can safely be sampled preserving the patterns of the confidential data source, while none of the records directly link towards any individual. The objective behind this generation process is to obtain data that does not contain any personal, re-traceable information, while maintaining the insights and utility of the confidential data sources. Hence, information about the group is preserved and the individual is protected.

Private synthetic data and its generation algorithms could be the catalyst towards transparent data sharing, analysis and integration in a privacy-preserving manner. Potentially, access of data sources that were previously considered too confidential to release, can now be used to extract new value in a safe manner. As the open data movement continues to grow [19], new approaches are needed for a safe and provably private way to produce synthetic data sets for public release. Therefore, in this thesis we examine the potential of private synthetic data generation, particularly when applied in the real-world domain.

1.1.5 Data Generation Landscape

From a privacy and utility standpoint private synthetic data generation seems a particularly attractive approach. However let's also take a few steps back by examining the concept of data generation in general and which place private synthetic data has in this space. In essence, the underlying phenomenon of data generation actually starts before it is even captured in a permanent format. An abstraction of reality, hiding all likely irrelevant details, allows us to make sense of the world and its vast complexity. When these dynamics are captured and collected in databases, while being integrated with other knowledge and analyzed for patterns, new insights can be derived that were previously hidden for most observers. Data, even in its complete simplified view of natural phenomena, can reveal internal and external interdependence of ongoing processes. The manner how these dynamics interplay and in a sense lead to the generation of data is an infinite study on its own and naturally beyond the scope of our research.

We can take one step further and analyze how new data can be generated according to patterns already captured and observed before. In a sense, it involves recreation of data in similar formats. Purpose may vary, however in general it is to grow data sources in a manner that mimics the underlying phenomenon of the input data source while preserving its overall utility. For instance, testing software can benefit from evaluation of a larger set of test cases that have been generated by techniques complying to test requirements and thus reducing the cost in software development and maintenance [20].

When privacy is of no concern, the objective naturally is to represent the data as closely as possible with minimal distribution deviation. In this case, a different set of techniques can be applied that attempt to capture all of the patterns that exist within the data. In this case, the selection of generation schemes is optimized to retrieve and capture most of the data's underlying patterns without needing to protect any individuals privacy. Differentially private synthetic data generation on the other hand operates on a different principle. Again, the objective would be to have minimal distributions deviation but not at the cost of privacy violation. Every pattern that is retrieved using these techniques requires a small addition of noise. Therefore, in constrast to general data generation we now need to be more careful with the amount of information we intend to learn from a sensitive data source. DP synthetic data generation intends to approximate confidential data sources in the most efficient manner by learning only

the most important characteristics in order to minimize noise addition. Therefore while both fields are very similar, this constraint for private data generators shifts the focus of complete distribution deviation to more efficient pattern capturing schemes.

1.2 Research Objective

In this study we examine the potential of differentially private synthetic data generation in the real-world domain. We start by investigating the state of the art of private data release and propose an evaluation framework, which allows us to compare generation schemes and assess their utility on varying data sources. A selection of algorithms are reviewed according to these criteria, from which we can evaluate whether private synthetic data generation can be considered a strong solution for enabling privacy-preserving practices.

Given the limited amount of resources available and defined project requirements, we focus the scope of this thesis on the leading contenders in general-purpose DP synthetic data algorithms and provide a more comprehensive utility assessment than they were originally published with. We intend to assess their general applicability by including other utility metrics, which become relevant in professional environments beyond academic research. Any specialized algorithms, which might perform well on a particular use case (e.g. graph data, continuous data, spatial data), but seem less suited to other tasks are not considered for further extensive analysis. Hence, we only focus on algorithms with consistent performance and general applicability by addressing the following main research question:

Are general-purpose differentially private synthetic data generation algorithms able to provide required data utility on high-dimensional databases with varying number of attributes, sizes and distributions?

Evidently, the research question does not intend to answer all possible high-dimensional databases as this would be unfeasible in the given time frame. However, we examine properties that are prevalent in actual databases and see how these affect our selected generation algorithms.

In order to fulfill this research objective, the main research question is divided in the following sub-questions:

1. *What is the state-of-the-art with regard to synthetic data generation and which algorithms would be good candidates for more extensive evaluation?*

2. *How can these algorithms and their output be evaluated in a fair manner, taking into account data characteristics, privacy-utility trade-offs and real-world requirements?*

3. *How do the selected algorithms perform on the proposed utility framework and what are their inherent strengths and limitations?*

1.3 Purpose

Our purpose for this research is to drive strong privacy practices that can easily be adopted in the industry and research. Synthetic data generation is a particularly attractive solution, enabling value to be extracted safely in a traditional database format requiring minimal adaption in existing systems and data exploration practices. Moreover, data can be shared safely without compromising on privacy, stimulating the open data movement and providing access to people who can actually derive new value of sources that were previously considered too confidential to release. Subsequent developments based on the insights attained from this research can drive practical application of differentialy private architectures within a wide range of industries.

1.4 Methods

In order to assess the potential of general-purpose private synthetic data algorithms, an extensive literature review is required. Therefore, we start by exploring data privacy concepts, particularly differential privacy (introduced in the next chapter), investigate privacy architectures, real-world implementations and decide which algorithms would be suitable candidates for further review. Subsequently, an evaluation framework is proposed which takes into account the diverging requirements in real-world use cases when adopting a new privacy engineering approach. The selected set of algorithms are then implemented to work on actual data sources and subsequently experimented with in order to estimate the performance on each criteria in the evaluation framework. Hence, we are able to assess whether synthetic data algorithms can be used for various applications and what their inherent strengths and limitations are. A more detailed description on the evaluation methods used throughout this study can be found in Chapter 3.

1.5 Delimitations

Our primary aim is to estimate whether privately generated synthetic data can fit the requirements of various applications. These requirements tend to vary drastically between various industries and therefore we can not test all of them. Hence, the intention of this study is to assess whether any form of analysis on generated synthetic data provide similar insights when using the original confidential source. Moreover, as most algorithms in academic literature either do not provide an implementation or are not designed to handle varying data characteristics, we implement them ourselves to evaluate their performance fairly. Given the limited time and resources we can therefore only evaluate a few algorithms, which we select carefully based on our literature study. Hence, we do not provide an exhaustive evaluation of the state of the art, but do assess whether these selected real-world applicable algorithms perform adequately on our evaluation methodology. Moreover, inherently the generation of synthetic data is a slow process due to the amount of patterns to be captured, especially on large data and column domains typical in actual use cases. The number of experiments is thus limited by run-time requirements. As our intention is to evaluate the potential in the real-world domain we do not want to alter data sources significantly to improve performance. An accepted constraint is that any continuous columns can be discretized as long as this does not sacrifice the data's utility. Hence, highly dimensional continuous data domains are not evaluated in-depth and are beyond the scope of our evaluation of general-purpose algorithms. For a more extensive experimental analysis we refer to other benchmark test [5], where the tests are limited to data containing only one or two columns and can thus cover a wider range of algorithms and test experiments. Our experiments rather focus on multidimensional data sources and their inherently varying properties. Nonetheless, our study enables for a deeper understanding of the influence of several generation schemes and diverging data characteristics on real-world use cases.

1.6 Outline

The remainder of the thesis is organized as follows. Chapter 2 reviews important theoretic background regarding differential privacy, which has emerged as the standard for achieving data privacy. In Chapter 3 the utility evaluation framework is proposed, highlighting each criteria and providing a description of the evaluation protocol. Chapter 4 introduces various generation schemes and architectures, including an in-depth discussion of the four selected algorithms that are implemented for more extensive review. Subsequently, we test these algorithms on the evaluation framework and provide the results in Chapter 5. Finally, we conclude this study in Chapter 6 and present directions for future research.

Chapter 2

Theoretic Background

In this chapter we present theoretic background on data privacy. In particular, our research focuses on differentialy private algorithms. Differential Privacy (DP) has become the academic standard for strong data privacy, due to its its provable guarantees, resistance to background knowledge, theorems of composition and accuracy [4], [13].

2.1 Differential Privacy

Differential privacy has emerged as the de facto standard for ensuring privacy and managing significant risks of personal data disclosure. In databases that consist of records corresponding to individuals, differential privacy guarantees that the participation of a single individual does not alter the output of a computation significantly. This guarantee is highly appealing, as it allows one to to share information knowing that the results are not affected by any individuals record's presence or absence. A single record only makes an output slightly more likely and any inference by adversaries would remain unchanged even if the individual was removed from the data. Differential privacy is a definition, not a technique in itself, and is denoted as follows:

Definition 1 (Differential Privacy [13])**.** A randomized computation A is ϵ-differentially private if for all data sets D_1 and D_2 differing by just one individual and all output $S \subseteq Range(A)$:

$$Pr[(A(D_1) \in S] \leq exp(\epsilon) \times Pr[A(D_2) \in S] \tag{2.1}$$

In other words, DP algorithms introduce a small amount of noise into query results, restricted by the parameter ϵ. Hence, output stays relatively close to the original answer, but the introduction of a small amount of randomness ensures that the presence of an individual in the output can not be inferred. As a result, differential privacy guarantees that the same conclusions can be drawn about the population, while the information of a specific individual remains private.

In order to achieve differential privacy one requires mechanisms that introduce noise into aggregate statistics. These statistics can then safely be released. Subsequently. these private statistics can be used to build systems in a private manner and even lend themselves to the generation of new synthetic databases. A privacy mechanism is in essence an algorithm that allows one to query a database, receive slightly noise answers which can then safely be used for further analysis. For example, when one would be interested in the number of patients with disease X, a mechanism would take the real answer add a small amount of randomness bounded by ϵ and output this noisy value. Hence, the answer remains close to the original value but this small amount of uncertainty ensures that one can never be sure whether the value is caused by a specific individual or is caused by noise. For example, when an employee knows the salary of all his male colleagues except one. When the average salary of all males in his department is released with some noise, it would still be impossible to infer the one unknown salary figure.

Most privacy algorithms generate data based on raw data query results, taking private data as an input and using the output for a given task. If each subroutine action is performed privately, the algorithm can be considered differentially private as a whole, which is formally defined as the composition theorem:

Theorem 1 (Composition Theorem [21])**.** Sequential executions of sub-tasks $A_1,\cdots, A_n$ each corresponding to ϵ_i-differential privacy, ensures that the algorithm is ϵ-differentially private for $\epsilon = \sum \epsilon_i$.

According to the composition theorem, ϵ is to be considered the *privacy budget*, indicating that the combination of performed operations should not exceed this parameter setting. The value of ϵ can be released to the public and the choice is essentially a social question, one that to this date has not had a definitive answer. Low values around 0.01 to 0.1 are suggested [22]. However, the effects on actual applications need to be analyzed more extensively, as these low epsilon values are likely impractical to retain the required data utility.

2.2 Privacy/Utility Trade-off

Differential privacy remains a complex definition, where even academics are unable to define an adequate amount of privacy budget. Therefore we would like to take a step back and explain the main concepts again in simpler terms [23]. The goal of differential privacy is to quantify the privacy level for a given operation on a database. A operation could be as simple as a query (e.g. how many patients have disease X), where an answer is privately generated, while staying very close to the real answer. In order to achieve this, noise is added to the result thus making it impossible to infer something about a specific individual. The total amount of noise is captured by the parameter ϵ, which in its essence can be considered a privacy budget. If a certain level of privacy is required, each operation according to the Theorem 1 requires a portion of this privacy budget. A lower value of ϵ, thus reflects a higher level of privacy at the cost of loss in utility due to larger uncertainty. This balancing act is known as the Privacy/Utility trade-off, where one can set the budget lower for more privacy, or higher for increased utility.

2.3 Other Privacy Methods

Other privacy methods including k-anonymity[14] or l-diversity [16], have had wider application compared to differential privacy, however pose several limitations that are easily exploited by a motivated attacker. Their simplicity often makes them favorable over differential privacy. However, it is imperative to note that these heuristics do not provide any guarantees over what someone can infer when the data or statistics are released, in contrast to differential privacy [24]. In essence, differential privacy provides formal guarantees for any well-implemented differentially private algorithm and thus one can quantify the privacy implications of using these algorithms, i.e. notion of information leakage, composition of actions, future-proof release, and preservation of insights regardless of any individual's record presence or absence. Other heuristics can not provide similar guarantees or even ignore these properties. For instance, the notion of composability, i.e. cumulative privacy leakage for each private computation, does exist for any privacy method but is just not quantified.

2.4 Basic DP Mechanisms

In order to allow for a more technical understanding, perhaps it helps to introduce two of the major mechanisms in differential privacy. A mathematically averse reader can safely skip this section. The two most commonly used mechanisms for differential privacy are the Laplace

mechanism [13] and the Exponential mechanism [25], which in most cases act as the building blocks for more sophisticated algorithms. Both require the sensitivity of a function between two neighboring data sets, i.e. the maximum possible change in output value after adding or removing one single individuals from the data set, defined as follows,

Definition 2 (l_1-Sensitivity)**.** For $f : D \rightarrow R^k$, the sensitivity of f is

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

for all D_1, D_2 differing in at most one element.

Hence, l_1-sensitivity captures how much one person's data can affect the output. For instance in the case of counting queries, the sensitivity would be equal to 1.

The Laplace Mechanism achieves differential privacy by adding noise to the function output from the Laplace distribution which in the remainder of this thesis is simply denoted as Laplace(σ) with mean 0 and scale σ.

Definition 3 (Laplace Mechanism [13])**.** Given a function f(I) that outputs a vector in $\mathbb{R}^d$, the Laplace Mechanism L is denoted as

$$L(D) = f(D) + Laplace(\Delta f/\epsilon)$$

The Laplace Mechanism is thus used to compute queries in a private manner, by adding a small amount of Laplacian noise to the answer bounded by sensitivity and privacy budget ϵ.

Some algorithms require the selection of the best of a discrete set of alternatives. Here, the Exponential Mechanism [25] can be used, which is again an ϵ-differentially private mechanism, where the best solution is defined by a score function that relates all the alternatives in the original database. Hence, the score indicates the quality of the result for a given task. For example, which option would result in the highest information gain or lowest error rate. Subsequent to the score computation of each alternative, the mechanism samples an output value based on the probability distribution of the output domain, which can be defined as follows:

Definition 4 (Exponential Mechanism [25])**.** Given a quality function q, with range argument $\Delta q \equiv \max_{r, D_1 \Delta D_2 \equiv 1} ||q(D_1, r) - q(D_2, r)||$, which assigns a real valued score to each outcome in R, the Exponential Mechanism $M_E(x, u, R)$ then selects and outputs an element $r \in R$ with a probability equal to

$$M_E(x, u, R) \sim exp(\epsilon q(x,t)/(2\Delta q))$$

Both the Laplace Mechanism and the Exponential Mechanism are often used in more sophisticated algorithms in conjunction with other operations. They can even be combined to perform numerical and categorical operations, which are necessary to perform certain synthesis tasks, e.g. [26], [27].

2.5 Differentially Private Architectures

Using the concepts and basic mechanisms to achieve differential privacy, we can distinguish three implementation architectures [3]. Each system consists of a Database (DB) and Data Mining (DM) element and differ based on the location of the Differential Privacy (DP) mechanism.

First, *DP Synthetic Data*, where the DP mechanism is positioned on the DB in order to generate new synthetic data upon which the DM services can act without alteration. Hence, all DM services remain in tact, but mining quality is heavily dependent on the data synthesis

scheme. Second, *DP Aggregate Statistics*, introduces a query interface inbetween the DB and DM services, supporting traditional aggregation queries with data privacy guarantees. As a result, higher accuracy is expected compared to the previous method, but data mining is restricted to statistics from traditional aggregation queries. Finally, *DP Data Mining Specific* includes a DP interface in the DM services, allowing for specific queries optimized for a particular data mining task. Generally able to obtain highest levels of accuracy at the cost of added complexity in the database and data mining logics.

Again, in this thesis we only examine the first architecture. While data mining specific algorithms might result in higher accuracy, they lack versatility. For each data mining tasks you would have to identify a suitable algorithm in the state of the art and implement it accordingly. Hence, the solution can not be scaled to various use cases. Therefore our research focuses on general-purpose private synthetic data algorithms, which hopefully can achieve similar levels of accuracy for which the effort to implement complex specialized algorithms becomes redundant. A more detailed description on a comparison of the various architectures can be found in the Appendix A.

2.6 Differentially Private Synthetic Data Generation

Finally, we would like to review the process of DP synthetic data generation again in depth. Traditionally, one starts with a confidential database containing records with sensitive information about individuals that needs to be processed or shared in a private manner. One way is to create a new database that mimics patterns of the confidential data source, but does not contains any private information directly linking to any individual.

As discussed before, a pattern or statistic released without noise can result in a privacy violation when combined with prior knowledge or integration with auxiliary data sources; even when this pattern is hidden in a database format. Therefore, any pattern that needs to be mimicked in a synthetic database requires a small amount of noise as defined by the privacy budget ϵ in differential privacy. A DP synthetic data generation algorithm captures these patterns in a private manner and stores them in a model, e.g. histogram representations [26], Bayesian networks[27], or sparse vector techniques [28]. We can now safely sample new records according to statistical characteristics in the model to ultimately generate new full synthetic databases. As this final step does not require further access to the confidential data source, i.e. all patterns have already been privately computed and stored, we can generate multiple synthetic databases without running the risk of releasing new private information. As a result, we obtain a synthetic database that mimics the statistical characteristics and patterns of the original database where no record directly links to any individual.

Chapter 3

Methodology

In this chapter we present the methodology used throughout this thesis. Specifically we propose a new evaluation framework upon which private synthetic data generation algorithms are tested for general utility and real-world requirements. Subsequently, we present the evaluation procedure throughout our experiments.

3.1 Evaluation Framework

When a new DP synthesis method is proposed in academic literature it is always subjected to some form of evaluation. Various data sources, utility metrics and evaluation methods are used to assess the performance of these algorithms. Upon reading various academic papers, there does not seem to be one unified method for evaluation. Additionally, the type of metrics or data sources used to assess synthetic data utility vary as well, making direct comparison to other literature impossible. Finally, most papers actually do not comment on opportunities for deployment on real-world use cases and the challenges that might arise. Particularly, there seems to be lack of consideration for user interaction, especially by someone without a privacy background.

Hence, we feel it is imperative to define a new evaluation framework that allows us to evaluate DP synthesis methods under various industry requirements and assessing how a potential user might interact with the system. By using this framework, we can assess the potential of synthetic data generation and its applicability to real-world use cases. Perhaps in this manner we can close the gap between the academic and practical domain.

For evaluating private synthetic data generation algorithms we measure each method's strengths and limitations across six components, upon which the deployment of a generator for a specific data source or use case can be justified. As a result, this framework enables comparison between various methods and provides insight into their real-world applicability.

In this section we address each aspect of the utility evaluation framework which consists of the following elements:

1. *Synthetic data output quality*: ability to mimic statistical characteristics and patterns of the original database.

2. *Privacy-Utility Trade-off*: the balancing effect between privacy and utility, analyzing the levels of privacy that can be guaranteed while preserving acceptable utility.

3. *Transparency*: synthetic data generation process and output transparency, referring to whether the method for data generation and the consequences on the output are comprehensible by the user and analyst.

4. *Ease of configuration*: limited expert knowledge required during implementation and subsequent reconfiguration for including new patterns.

5. *Performance*: an algorithm's space and time complexity during the generation process, as well as examining options for parallelization for running on commodity hardware.

6. *Real-world applicability*: using synthetic data for various specific use cases, while evaluating whether any obtained insights from the raw data are retained.

An overview of the full framework is depicted in Figure 3.1, including the method of evaluation. In the upcoming sections we discuss each element of the framework in more detail.

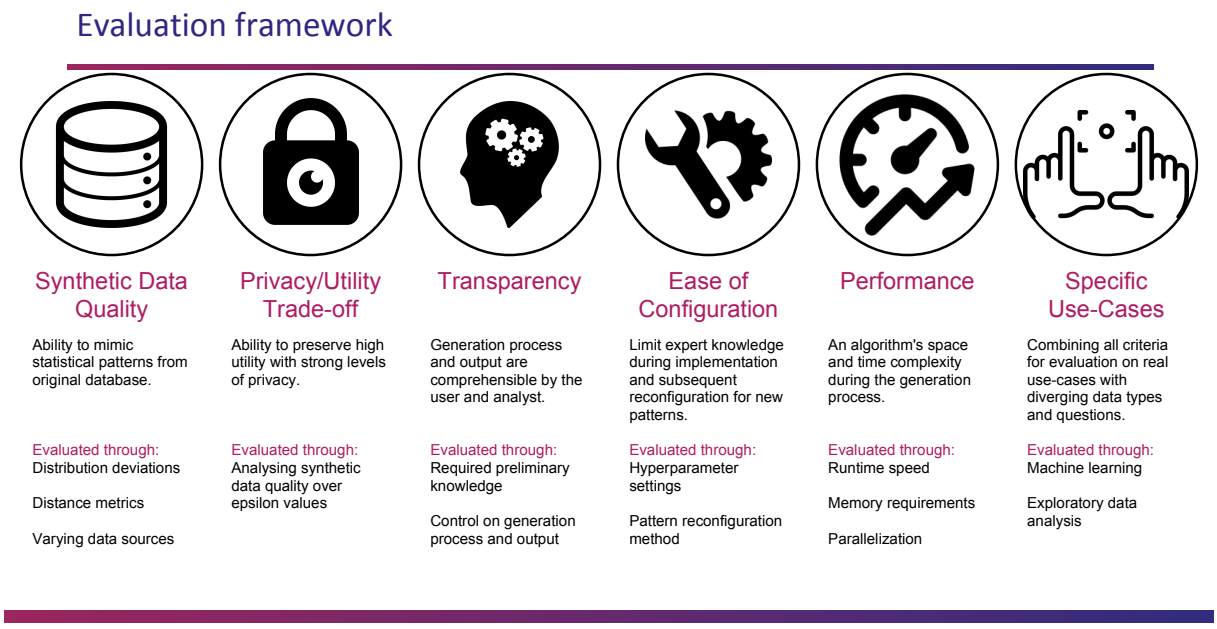


Figure 3.1: Proposed synthetic data utility evaluation framework

3.1.1 Synthetic Data Quality

Output quality of a synthetic data generator refers to the algorithm's capability to consistently generate new databases that closely mimic the statistical characteristics and patterns of an input database. As proposed by Snoke et al [29], in order to estimate whether most of the insights are preserved we need to evaluate general and specific utility measures. This element in the evaluation framework only focuses on general utility measures, which try to give a sense of which patterns are captured adequately and also give a notion of generalizability to other patterns. As each data use case is bound to have varying requirements, some specific use cases are examined in the final element of the evaluation framework. However, general utility results are normally the most informative of the output quality, as only evaluating specific use cases tends to overly reaffirm beliefs an analyst might have [29]. Nevertheless, both general and specific utility require thorough investigation in order to estimate the potential of synthetic data real-world applicability.

General utility is measured by examining distribution deviations and distance metrics between both the synthetic and original data. In terms of distribution deviations each column domain can be visualized and compared between both databases. It is a qualitative measure that enables an analyst or user of a DP synthetic algorithm to assess whether the generated data closely resembles the original database under the specified privacy guarantee.

Distance metrics quantify the output quality, by measuring how far both distributions deviate. By examining these metrics, a user can determine an acceptable level of privacy that can be guaranteed for a pre-determined level of synthetic data utility. We have chosen to evaluate two distinct metrics in order to not bias our results to one distance metric. Hence, we evaluate both the Average Variation Distance (AVD) (i.e. average of the Total Variation Distance (TVD) in each column, where TVD is equivalent to half of the *L1*-distance between marginal distributions when treated as probability distributions) [30] and Jensen-Shannon (JS) distance (i.e. square root of the Jensen-Shannon divergence metric, which is a symmetrized and smoothed version of the Kullback-Leibler divergence) [31]. The latter metric is chosen over the traditional Kullback-Leiber divergence, as it allows for evaluation of columns with unequal amount of discrete values. This can occur in synthetic data generation when a certain attribute has a very small probability of being sampled, due to very little occurrence in the input, and might therefore not be represented in the output data.

Finally, data characteristics have a strong influence on the output quality, e.g. shape of the data distribution, data types, number of records and columns [5]. Hence, we evaluate multiple different databases with varying characteristics, which have carefully been chosen to reflect common properties in the real-world domain. These are to be discussed in the upcoming Section 4.4. We have specifically chosen these databases to reflect common use cases and database patterns that occur in the real-world, in order to assess whether utility is generalizable to other applications.

Note, measuring synthetic data utility is inherently complex due to amount of randomness that is introduced by the DP mechanisms, as well as the record sampling process itself. Naturally, it is impossible to generate data in a private manner without some loss of utility. What can be defined as an acceptable loss of utility and deviation in insights is naturally subject to the requirements of a specific use case.

3.1.2 Privacy-Utility Trade-off

Intertwined with output quality evaluation is the assessment of the privacy-utility trade-off. Specifically, the way data utility is affected by varying privacy guarantees. As we decrease the privacy budget ϵ, the amount of noise introduced increases and distributions start to deviate further. As academics have left the correct value for ϵ a social question, one often starts by first defining the desired level of data utility. Once this utility level is defined, perhaps through one of the proposed distance metrics, we can identify the level of privacy that can be guaranteed. Therefore, we analyze this trade-off by performing multiple experiments over varying ϵ values for each algorithm and each input database.

As discussed earlier, when the definition of differential privacy emerged the suggested privacy budget ranges from 0.01 to 0.1 [22]. However it is important to evaluate whether this actually delivers an acceptable form of utility that real-world applications require. While actual DP applications are limited, one study on private machine learning on real telecom data suggests that accuracy requirements could only be met with an epsilon of at least 3.0 [3]. More real use cases are needed to get a more realistic perspective on adequate privacy guarantees. Hence, we contribute to a better understanding of the privacy-utility trade-off by examining the effects on output quality of multiple algorithms and varying data characteristics.

3.1.3 Transparency

Moving beyond the academic domain, an algorithm's process, usage and consequences on output need to be comprehensible for a non-privacy expert to ensure correct deployment. A new trade-off appears when it comes to the ability to specify which patterns need to be captured during the generation process. On one hand, control over the output quality for specific set of patterns

is preferred; on the other hand, the full original distribution needs to be well-represented in the synthetic data. Hence we see a trade-off in pattern control - generalized utility.

In some scenarios a user benefits from having control over which patterns should be captured in the output. It ensures that the defined statistical patterns in the input data are optimally approximated in the output and thus also gives an understanding what the data can be used for. However, this method does not guarantee minimal deviation between both data distributions. In this case, generation is only aimed at preserving the insights defined by the user, not reducing overall distribution deviation. One controls which patterns are captured but the synthetic data is less likely to be reflective of the original source. Moreover this requires sufficient preliminary knowledge about the future usage of the data prior to its release.

When the synthetic data's use is not known or requirements change over time, generalizable results are preferred to handle a wide range of operations. The user has less control over which patterns are captured well in the data, but let the algorithm determine the optimal generation scheme to provide a higher general data utility. We therefore do not know which patterns are captured, but the synthetic data distribution will be more reflective of the original data. Hence, depending on the requirements a user can either prefer to control pattern modeling when the data's use is known or limit pattern control for stronger general data utility.

3.1.4 Ease of Configuration

Directly linked to algorithm transparency is the ease of configuration. Specifically, we analyze the steps required towards deployment to function on real use cases, as well as the method for pattern addition or removal. This latter aspect only applies when pattern control is required, as defined in the transparency section. It concerns whether an algorithm easily extends itself to new patterns being added, or if this would require a significant change in the implementation.

Ideally an algorithm can be configured with limited expert knowledge, aligning with the constant change of requirements in the professional domain. Algorithm configuration should not be a burden to the user and thus decisions prior to generation should be limited. Therefore, besides the pattern addition method, we also analyze hyperparameter settings that need to be set by the user, as well as other important decisions to meet the required privacy guarantee and synthetic data utility.

3.1.5 Performance

When DP synthetic data generation algorithms emerged these were considered highly inefficient and inadequate to handle high-dimensional data, e.g. [32]. While algorithms have become more efficient [33], this remains a common challenge. In general, synthetic data generation is not considered a real-time issue [27], where normally the intention is to continue using the data for a prolonged period when shared, analysed or used for test beds. However, it is important to understand the space and time complexity of each algorithm to get a better sense of the hardware requirements. Additionally in order to speed up the process and allocate memory in an efficient manner, we also examine possibilities for parallelization. All these aspects allow us to assess the requirements and limitations of each algorithm during execution.

3.1.6 Real-world Applicability

As already discussed in the first element on general utility, we also evaluate specific use cases to determine if the insights from general utility metrics extend themselves to other scenarios. A wide variety of use cases can be examined, but we focus on one example use case that contains many components typical to the professional domain that allow us to assess the specific utility of synthetic data [29].

One use case resembling these features is machine learning, e.g. classification, prediction and clustering. Prior to the actual machine learning some exploratory data analysis is required as well, e.g. data distribution analysis or attribute correlations. When replicating this use case we can examine whether the insights from privately generated synthetic data are equivalent to the original database. Subsequently, once we have trained multiple models the metrics typically used in machine learning can be compared between both the private and confidential database. From this analysis we aim to evaluate whether the performance metrics and model coefficients are similar and thus if synthetic data can actually be used for training machine learning models.

3.2 Evaluation Procedure

Before commencing with the algorithm experiments, it is imperative to establish a solid evaluation procedure that allows for fair comparison taking into account all factors that might influence a result. Hay et al (2016) [5], encountered the complications that arise during algorithm selection and found that results in various papers often missed sound evaluation procedures that would allow one to assess the performance and utility of an algorithm in a particular context. In their paper they defined ten principles one should adhere to for fair comparison. Unfortunately their evaluation, while being very extensive, was limited to two-dimensional data sets and is thus not reflective of typical use cases in the real-world domain. Nevertheless we aim to incorporate these principles as the foundation to our utility assessment, but extend the process by incorporating the remaining criteria of our utility evaluation framework. As a result, we can evaluate algorithms fairly by taking into account varying input data characteristics, while also examining the applicability to real-world multidimensional data domains.

A full benchmark test equivalent to Hay et al (2016) [5] is beyond the scope of this thesis. Instead, we focus on some of the leading contenders in general-purpose differentially private synthetic data algorithms and provide a more comprehensive utility assessment than they were originally published with. Hence, we aim to compare algorithms properly by adhering to the principles in in the following manner. Principles I to IV on “input diversity” (i.e. epsilon, scale, shape and domain) are realized by performing experiments with multiple parameter settings on dissimilar real-world databases. Used databases and their inherently interesting properties are described in the upcoming chapter. Principle V to VII on “end-to-end private algorithms” are evaluated for each selected algorithm separately, using some of their proposed algorithm alterations to ensure truly private methods. Finally, Principles VIII to X on “sound evaluation of outputs” requires an assessment of algorithm result variance, bias and utility under various privacy levels.

One of our contributions is to extend this evaluation procedure to real-world multidimensional data domains, while also utilizing a wider range of query types and utility measures. Moreover we also incorporated a user-centered focus in our framework to assess adoption likelihood in the real-world domain based around the concerns that might arise during implementation and configuration. Finally, separate particular use cases, including machine learning and exploratory data analysis are investigated by assessing whether insights or model performance are retained after substitution of synthetic data for its raw variant.

Chapter 4

Generating Private Synthetic Data

In this chapter, we introduce four general-purpose DP synthetic data generation algorithms that are evaluated based on the proposed evaluation framework of the previous chapter. Specifically we introduce two basic synthesis approaches, as well as two more complex sophisticated algorithms. Finally, we discuss the implementation details and data sources used throughout the evaluation process.

4.1 Types of Algorithms

As differential privacy has become the standard in privacy engineering, numerous algorithms are continuously developed and proposed. These DP algorithms often focus on specific tasks, e.g. private statistics, machine learning, graphs, or synthetic databases. In this thesis we are particularly interested in the latter and especially how these can be applied in the professional domain. However, as there only have been a few instances of actual applications and academic literature often lacks evaluation on real-world use cases, it remains hard to assess how these algorithms perform on data sources with varying characteristics and requirements. Therefore, we performed an extensive literature study and selected a few of the leading contenders in synthetic data generation to evaluate according to our framework.

When selecting these privacy algorithms, one notices that some perform particularly well for a given task. Whether those tasks concern private machine learning [34], [35], dimensionality reduction [36], or continuous data [37], [38]. Other privacy algorithms seem well-suited for a given data source or use case, e.g. health analytics [39]–[41], transaction data [42], [43], graphs [44], or location data [45]–[47]. While these algorithms seem promising and do actually work on real-world use cases, their applications are focused on a specific task. Hence, we decided to evaluate a more scalable approach that with promising results could be deployed in a variety of industries and use cases. Hence, we evaluate general-purpose DP synthetic data generators that work on multidimensional data sources. One accepted constraint is that continuous variables are generalized, which is already considered a common task in data privacy [27].

Even then given our current resources, it remains difficult to select algorithms which allow us to assess the potential of this new technology. Especially considering the incomplete evaluation methods used in a veriaty of papers, e.g. not comparing to baselines or leading contenders, or varying utility measures [5]. Even their chronological release makes the selection difficult, as older algorithms are typically not compared to their new competitors. Additionally, because of the complexity of differential privacy the privacy analysis in some papers is sometimes incorrect as well. One notable instance claimed to have better performance than most leading contenders [48], where the privacy analysis was later proofed to be erroneous [27], where then a subsequent paper claims that this proof is invalid as well [28]. All these factors complicate the selection process, therefore we have chosen only to select the upcoming few based on their citations, and numerous appearances in evaluation sections of other literature.

4.2 Selected Algorithms

In this section, we review four algorithms for synthetic data generation. We start by proposing two basic approaches: the Laplacian Hist [13] and the Uniform Hist. Afterwards we will detail two of the leading contenders in synthetic data generation: MWEM [26] and PrivBayes [27].

4.2.1 Basic Generators

Both of the upcoming approaches are simple in nature and require little computational cost and implementation effort. Therefore both algorithms function as baselines for more sophisticated algorithms to be compared to. When performance of the latter type is lower than these baselines, the added complexity and computational requirements of the other algorithms become redundant [5]. Hence, we analyze the performance of these basic generators to identify the use cases and data characteristics which might favor a simpler approach.

Laplacian Hist

As the name suggests, the Laplacian Hist [13] is closely related to the Laplacian Mechanism (Definition 3). A confidential data source is represented as a histogram over the full data domain space, i.e. each bin in the histogram is equivalent to a combination of values for each discrete column. Hence, the full domain space is essentially modelled in terms of frequency of occurrence, where each possible combination of attributes is represented. For instance, when a database consists of three columns, e.g. gender, age, and nationality, one bin in the histogram representation could be: female - 23 - Spanish. Subsequently, the Laplacian Mechanism is used to fill in each bin within the histogram by querying the confidential data source and adding a small amount of Laplacian noise to the real answer. Normally the total privacy budget is divided over the amount of counting queries. However, in the case of a histogram, each query is disjoint and can be drawn independently according to the defined privacy budget [22]. In the end, we obtain a new histogram representation with noisy counts for each combination of attributes. Finally, this representation is normalized and one can sample records according to the histogram probabilities.

Uniform Hist

Similarly to the Laplacian Hist, we evaluate another simple histogram sampling method. The Uniform Hist, again represents the full column domain space in a histogram format. However instead of using the Laplacian Mechanism to fill in each bin separately, it uses the total privacy budget ϵ only once to query for the size of the confidential data source and subsequently fills in each bin entry with the uniform distribution. Again after normalization one can sample records according to the probabilities of each combination of attributes.

4.2.2 MWEM

Now that we have defined two naive baselines for comparison, we can introduce two more complex algorithms. We start with the Multiplicative Weights Exponential Mechanism (MWEM) [26]. MWEM approximates a distribution in a differentially private manner through an iterative process. Similar to the Uniform Hist, it starts with an estimate according to the uniform distribution. A user can then define or automatically generate a workload of queries, i.e. patterns, based on the input data characteristics. Each iteration MWEM privately selects a query from the workload through the Exponential Mechanism (Definition 4). Queries are selected based on the highest error between the original and the synthetic distribution, i.e. the error is used as the scoring function in the Exponential Mechanism which samples a query with highest absolute difference between both distributions. Once selected, a bit of noise is added to the query

result through the Laplacian Mechanism (Definition 3). Once the query has been selected and we obtain the noisy query result of the original distribution, the Multiplicative Weights update rule is used to reduce the error. Note this update process does not require further access to the confidential data source and only uses the noisy statistic. Subsequently each iteration all sampled queries are used to update their corresponding bins in the histogram representation and reduce their error. As this process is non-private, the update process of all selected queries can safely be repeated according to user preference. Hence, each iteration we achieve a balancing act where a new query gets added to the previously sampled set and bins are updated repeatedly to approximate the original distribution.

Besides the familiar challenge of setting an adequate privacy budget ϵ, deciding the number of iterations (T) is non-trivial. The original paper [26] suggests a default value of 10 iterations, however benchmark test in [5] propose another method for selecting an optimal value of iterations in a data-independent manner. Various experiments were done and they discovered that based on the epsilon value and scale of the data (i.e. number of records), one can find an optimal number of iterations for a given input database. They defined a list of Scale-Epsilon Pairs (SEP) including their optimal number of iterations (T) in MWEM. Prior to generation these SEPs are used to interpolate T close to the optimal value based on the input data size and defined privacy budget. Supposedly, this can result in a 7.5x increase in data utility in comparison to using a fixed default value. This improved version of the algorithm is thus used in our experiments, in order to remove one difficult decision from the user and obtain results reflecting of MWEM's capabilities.

4.2.3 PrivBayes

PrivBayes is a differentially private method for releasing high-dimensional data [27]. In contrast to the previous three algorithms, PrivBayes does not capture patterns in a histogram representation. As the authors claim, these previous methods encounter difficulties when trying to release high-dimensional data. For one, due to the fact that many data sets have a very large domain size (i.e. the product of cardinalities of the data's attributes), compared to the data size which would make the generation very memory intensive and slow compared to just using the input data. Second, as the histogram representation likely has many bins with low counts, any added noise to these bins will have a much stronger effect and potentially dominate the original signal.

Instead, PrivBayes models correlations among attributes in a Bayesian Network [49], which approximates the original distribution of the data by using low-dimensional distributions, with the intention to maintain high accuracy for any type of (linear or non-linear) query. Using this method we overcome the previous two problems. Observe that, this method is query-independent, unlike MWEM for example, and thus strives to approximate all possible queries. Hence, this approach might be weaker when the objective is to optimize the result for a specific set of queries but generally performs well when minimizing the total distribution deviation.

PrivBayes consists of three steps. First, the algorithm learns a Bayesian Network that approximates the full-dimensional distribution. This Bayesian network consists of nodes and directed edges, where each node represents some attribute in the data. This step uses the Exponential Mechanism from Definition 4 to iteratively select nodes for the graph that result into the highest mutual information when combined with any of the previously selected nodes, i.e. forming parent-child combinations. Second, once the network is learned, the conditional probabilities between attributes is determined through the Laplace Mechanism, Definition 3, which indicates how likely certain attributes are to co-exist in original input data. The final step consists of the actual data synthesis, by using the network configuration and conditional probabilities, PrivBayes samples records obeying the schema and format of the original input.

One non-trivial parameter is the network degree (k), which determines the amount of parents a new node can have. This parameter is automatically selected by the algorithm, based on

the input data characteristics and privacy budget. Additionally it requires a more intuitive parameter: the notion of θ-usefulness, i.e. the ratio in average scale of information to average scale of noise. In practice, the performance of PrivBayes is not sensitive to the choice of θ according to the authors experiments. Hence, this value is fixed according to their suggestions.

4.3 Experiment Set-up

All of these algorithms have been implemented and tested in various settings. Below you find the implementation details and selected data sources used throughout our evaluation.

4.3.1 Implementation Details

For fair comparison and specific requirements, all of these algorithms have been implemented by the author and contributed to the industry. Each algorithm is programmed in python and designed to handle multidimensional data sources. Other existing open-source implementations were either implemented in a different language or pose stricter requirements of the input data format, e.g. only functioning on one or two-dimensional data domains [5]. Hence, the current implementations are able to handle various data formats, enabling evaluation of various use cases. Important to note, all of these methods operate only on discrete data, hence continuous columns need to be discretized. As suggested in [27], this is a common task in data privacy literature (e.g. k-anonymity) and can often be done intuitively or based on domain knowledge. Still, the method of discretization and chosen granularity is a factor that will influence the generation process, where more granular data requires more patterns to be captured which in return affects data utility and computational complexity.

Once developed we define the hyperparameter settings used throughout our evaluation for each algorithm. Every DP algorithm requires a pre-defined privacy budget ϵ before commencing. Therefore, we experiment with multiple values to assess the performance and synthetic data utility under multiple privacy guarantees, i.e. varying epsilon from 0.01 to 0.1 to 1. Both baselines, Laplacian Hist and Uniform Hist, have no other parameters to set-up prior to generation. As we use the improved version of MWEM the number of iterations (T), i.e. queries selected from the total workload, is automatically determined. Besides that we use the recommended values for the epsilon allocation 50/50 and number of times the sampled queries get repeated at 100. Similarly for PrivBayes, the network degree (k) is automatically determined by the algorithm. Epsilon is allocated 40/60 and the ratio between scale of information and scale of noise has been fixed to 4, according to the authors' suggestions. We have chosen only to evaluate the version with Binary encoding and F score function, as this was the original proposal by the authors. A new version was proposed which might improve performance and does not require variables to be binary encoded. Nevertheless, we feel that the original version suffices to evaluate the potential of this method and consider the development of even more advanced versions as future work.

4.4 Data Sources

When evaluating the potential of private synthetic data generation in the professional domain, we carefully select databases with characteristics that allow for fair evaluation and are reflective of typical industry requirements. Hence, we have tested the algorithms with numerous input sources to evaluate the consequences on the generation process and outcomes. In Table 4.1 we discuss the data sources that are featured in the upcoming results.

Table 4.1: Data sources used during evaluation

Data Source	Description	Interesting characteristics
Adult[1]	Census Income data set for predicting whether an individual's income exceeds $50k/year.	Almost every paper introducing a new DP mechanism includes this database in the evaluation section. Hence, one can easily compare the results with other literature.
American Community Survey (ACS)[2]	The American Community Survey is a yearly survey providing information about the United States and its people.	A subset of the complete data domain is used without any form of pre-processing. As a result, we can evaluate the effect of unclean data on the output utility.
Crimes in Chicago[3]	Reports incidents of crime that occurred in the City of Chicago.	A subset of the complete data domain is used to evaluate the effects of a large data size with over a million records and columns with a large amount of categories.
Facebook Check-ins[4]	Artificial location data for predicting which place a person might check-in to.	Contains some continous attributes, including timestamps, lattitude and longitude, which required heavy generalization. Hence, we obtain a more uniform distribution that allows us to evaluate the effects of discretizing continuous variables.

[1]Adult database: `https://archive.ics.uci.edu/ml/datasets/adult`
[2]ACS database: `https://www.census.gov/programs-surveys/acs`
[3]Crimes in Chicago database: `https://www.kaggle.com/currie32/crimes-in-chicago`
[4]Facebook Check-ins database: `https://www.kaggle.com/c/facebook-v-predicting-check-ins`

Chapter 5

Evaluation Results

In this chapter we present experiment results and discuss how the selected algorithms perform on each element of the evaluation framework, as proposed in Chapter 3. We start by examining general utility results, where we assess the synthetic data output quality and privacy-utility trade-offs. Next, algorithm transparency and configuration methods are evaluated to understand how a user can interact with the system and which consequences this has on the results. Subsequently, we also examine algorithm performance including execution time and memory requirements. Finally, we apply the algorithms to a machine learning use case to assess whether we can build models and draw conclusions on synthetic data sets that are equivalent to models trained on the original data.

5.1 General Utility Results

Measuring general utility of synthetic data is challenging due to the large amount of patterns that need to be captured. Moreover, there is no clear general definition of how far distributions can deviate, while preserving acceptable utility. General utility can be assessed by looking at how closely the synthetic distribution is able to mimic the original distribution, i.e. the number of insights that can be preserved in relative terms. Needless to say, exact distribution equivalence is not expected as any generation scheme already is bound to introduce some randomness during the sampling phase and when the process occurs differentially private even more noise gets introduced. In order to measure this general utility we use qualitative and quantitative measures and evaluate how the privacy-utility trade-off gets affected by varying ϵ values and the characteristics of input databases.

General utility measures can be distinguished in two components: distribution deviations and distance metrics. Distribution deviations measure how closely the original distribution is represented by the synthetic version, which can be assessed by comparing the marginal distributions over the complete discrete columnar domain. For small to medium domain sizes, these can be visualized and give a more qualitative assurance that distributions behave in a similar manner.

For a quantitative comparison of two distributions we use the two distance metrics discussed in the methodology section, i.e. Average Variation Distance (AVD) and Jensen-Shannon distance (JS). We observed that often these metrics tend to overlap in most cases and thus provide the same insights in relative terms.

As generation due to noise introduction and sampling is an inherently random process, we have performed multiple identical experiments and assessed any deviation in output quality measures. We noted that output stayed quite consistent and most insights remained the same, thus to enhance readability we report mean values in the upcoming results. However one should note that when two values seem relatively close to one another we can not make a statement about superiority of one algorithm over another, as due to randomness in generation the ordering

could change slightly. Hence we consider them to be equal in output quality. We have chosen to structure the upcoming general utility results according to typical data properties of real-world applications and only highlight the interesting findings.

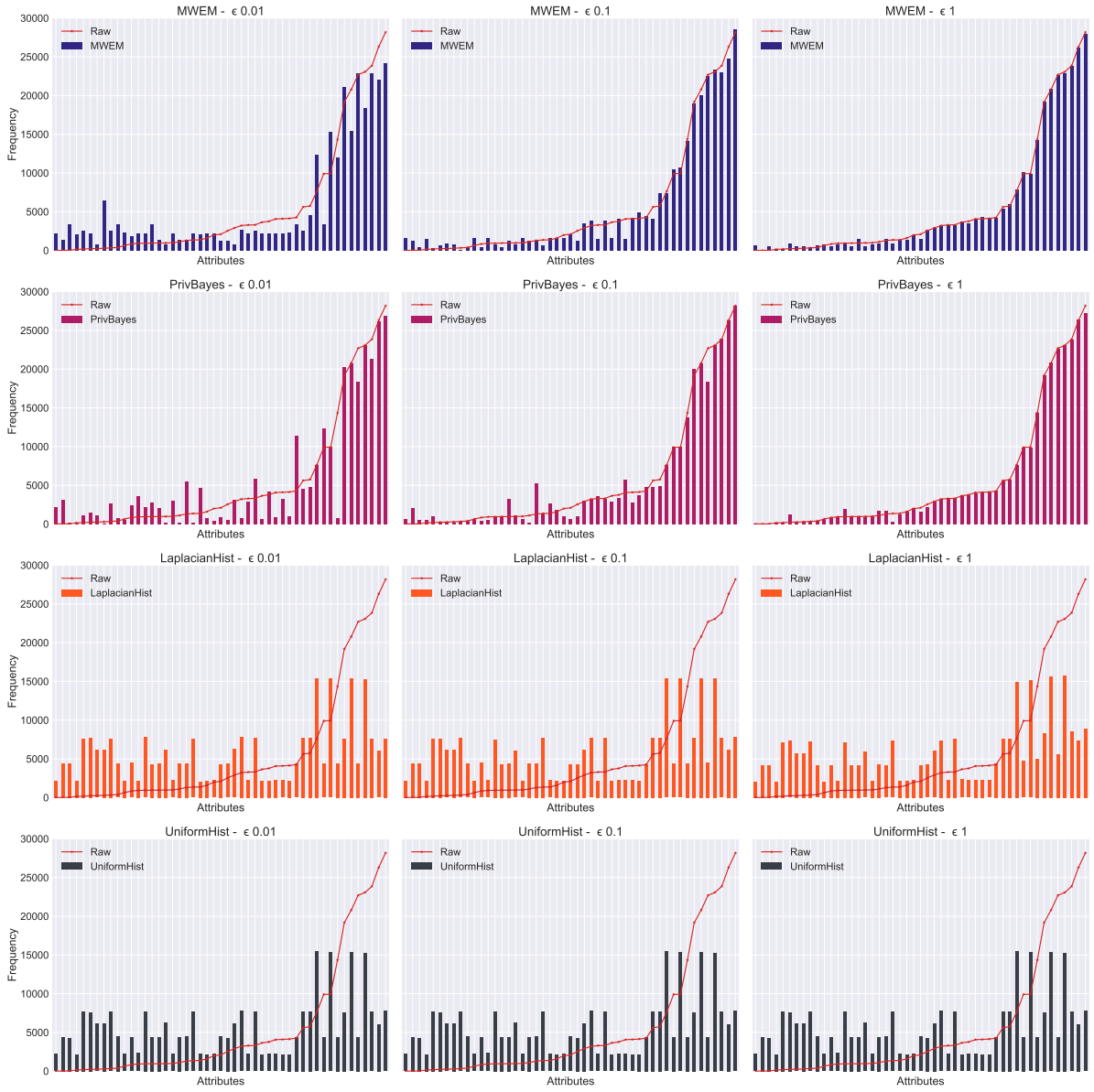


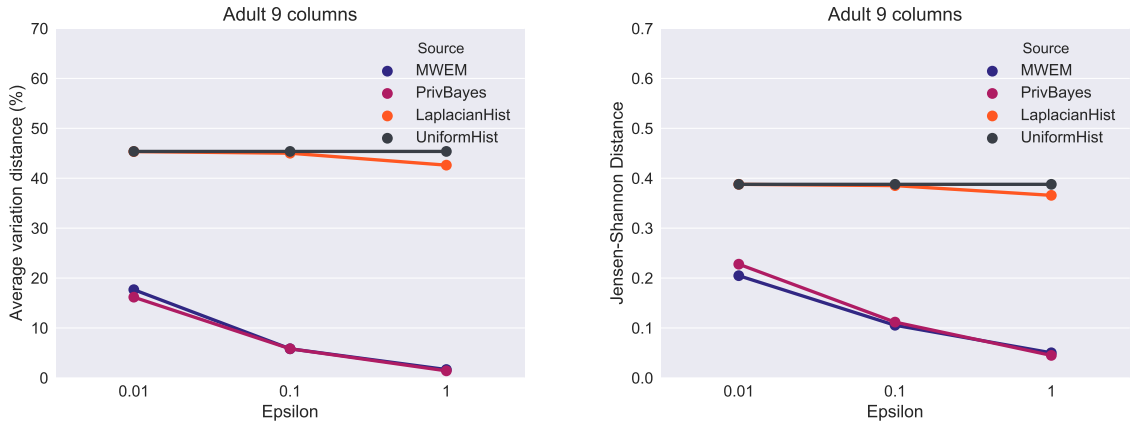
Figure 5.1: Distribution comparison of the complete attribute domain in the Adult database varying the generation over epsilon from 0.01 to 0.1 to 1. Each row and color represents an algorithm, while the columns refer to the used privacy budget. On the x-axis all the possible values in each column are placed and their frequency of occurrence is presented in the colored bins. A small red line indicates the original distribution. Hence, for optimal approximation of the synthetic data the histogram bins should closely resemble the raw distribution line.

5.1.1 Privacy Budget

As we increase privacy budget ϵ, we expect a closer approximation of the synthetic distribution to the original. In Figure 5.1 we see the effects of increasing privacy budget on the approximation for the Adult database. Evidently, the sophisticated algorithms, MWEM and PrivBayes, both seem to improve over increasing privacy budget. Particularly at ϵ of 1 we almost see an identical representation, meaning that the algorithms are able to estimate the data's original distribution quite well. Both baselines do not perform well on this particular dataset, even with increasing

epsilon. As expected the Uniform Hist does not alter its approximation even with relaxing privacy guarantees. Similarly, Laplacian Hist with an increase in ϵ changes only minimally.

While distributions graphs can give us a visual understanding of each algorithm's output quality, in order to compare algorithm performance we examine two distance metrics. In Figure 5.2a and Figure 5.2b we depict the Average Variation Distance (AVD) and Jensen-Shannon (JS) distance respectively. Both metrics seem to provide similar insights and confirm the statements made during the previous assessment of the distribution deviations. Both PrivBayes and MWEM perform equally, where due to randomness the ordering might change. Additionally, we see the effect of increasing privacy budget, where indeed at an epsilon value of 1 we get an almost perfect approximation. Finally, Laplacian Hist seems to perform slightly better when epsilon increases, however does not come close to the more sophisticated algorithms. Perhaps this is due to the data dimensionality and size, therefore we evaluate these data features in the upcoming sections.



(a) average variation distance

(b) Jensen-Shannon distance

Figure 5.2: AVD and JS-distance on the full Adult data set

5.1.2 Data Dimensionality

An important aspect in private synthetic data generation is the dimensionality of data, i.e. the number of columns and possible values. Particularly algorithm time and space complexity is highly dependent on the amount of columns as more patterns need to be captured (i.e. bins in histogram representation and nodes in networks), which we discuss in more detail in Section 5.3. Here we analyze how data dimensionality affects distribution deviation and distance metrics by reducing the number of columns. In Figure 5.3 we depict the complete distribution of a reduced version of the Adult database with only the first four columns, which have been generated at 0.1 epsilon. Notice that both sophisticated algorithms MWEM and PrivBayes seem to perform equally to the full dataset, with some incorrect estimations. Interestingly, the Laplacian Hist has improved significantly in comparison to the full Adult database with epsilon at 0.1. Probably due to the fact that when dimensionality is reduced, less patterns need to be captured and overall less noise is accumulated. Hence, with small data dimensionality and moderate privacy guarantees, simple DP algorithms seem to be able to closely represent the original distribution as well.

Again we depict the distance metrics in Figure 5.4. Both measures again tell a similar story. Observe that, especially under the lowest epsilon value PrivBayes excels and actually does not seem significantly affected by the lower column domain. However, when comparing it to the previous situation with full columnar domain, MWEM seems to perform considerably worse at strong privacy guarantees. Finally, again we confirm the statement above that Laplacian Hist performs roughly equivalent to MWEM at strong privacy guarantees has an almost perfect

approximation as well. On low columnar domains and relaxed privacy guarantees this naive approach already performs relatively well. Hence, we observe that decreasing data dimensionality affects each algorithm in a different manner, where PrivBayes stays relatively consistent, MWEM has reduced output quality with lower epsilon values and Laplacian Hist significantly improves over the original database.

5.1.3 Data Size

Next, we analyze data size and its influence on the generation process and output. During our experiments we noticed a steep increase in time complexity. Space requirements remained practically the same, as conversion to a histogram format or network generation is only dependent on the dimensionality of the data. However, time increases significantly for the sophisticated algorithms, especially when free-parameters are selected. For MWEM the number of iterations increases, while PrivBayes increases the network degree and thus more combinations need to be evaluated. Therefore, in our current set-up we are not able to evaluate MWEM and PrivBayes when these free-parameters are too high. However, both algorithms are highly parallelizable and thus with adequate resources should run on large databases, which we consider part of the future research.

Regardless, if we continue with the reduced Adult database of only 4 columns, we can increase the data size by duplicating all the records 5 times and thus obtain a new database of approximately 150k records. In Figure 5.5 we can see the effects on the distribution distance. Evidently both distance metrics again report similar different results, however MWEM does seem to perform a little worse in JS-distance. We noticed that academic literature has not agreed on one particular metric for utility evaluation, hence it could be a reason for some authors to select one that favors their proposed solution.

When compared to Figure 5.4, the increase in data size actually reduced the distance for all the algorithms and made them perform slightly better. Perhaps this is due to the increase in attribute frequencies, thus when noise is added to these distributions it has a relatively smaller effect. Therefore, similar to an increase in privacy budget, an increase in data size actually improves the output quality as well. This notion was also observed by Hay et al. [5], where an increase in scale (i.e. records) of the input data set and increasing epsilon have equivalent effects on the error. Thus, an algorithm's error remains roughly the same for all scale and epsilon pairs (SEPs) with the same product, which they defined as a scale-epsilon exhangeability.

Finally we observe that Laplacian Hist actually seems to performs almost equally to both sophisticated algorithms from ϵ of 0.1, suggesting that small data dimensionality, moderate privacy budget and especially large data size makes the complexity of the sophisticated algorithms almost redundant.

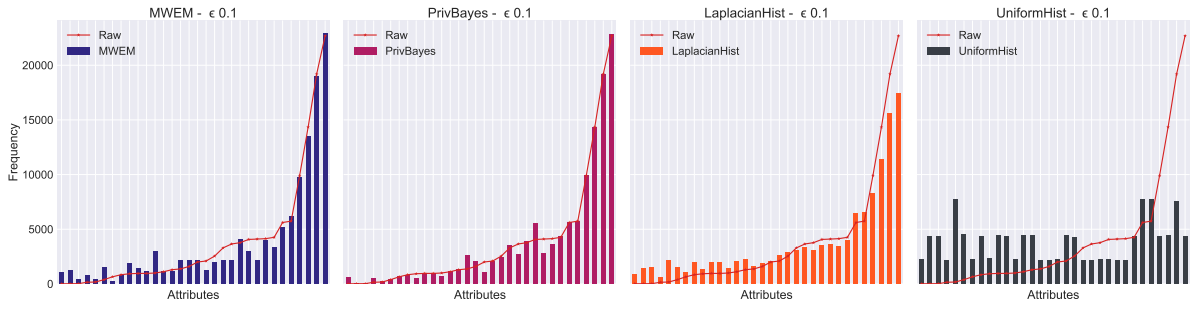
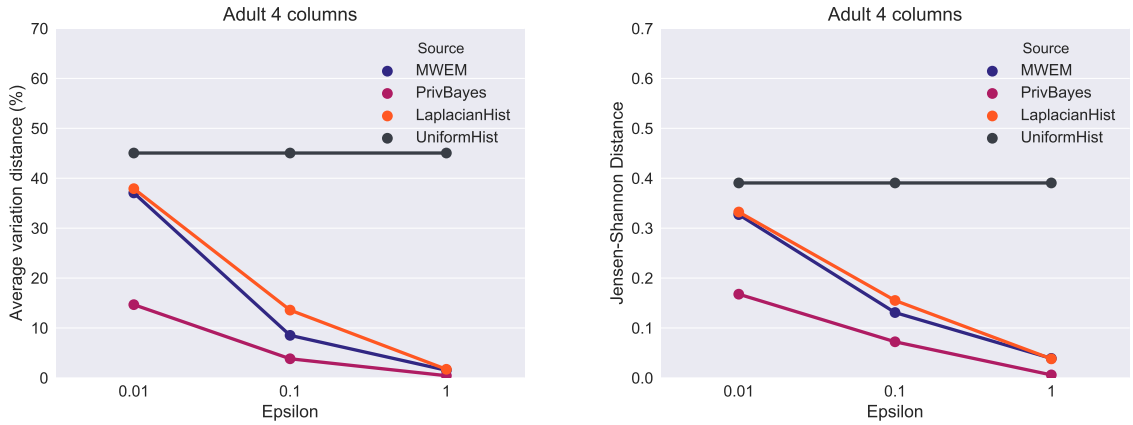
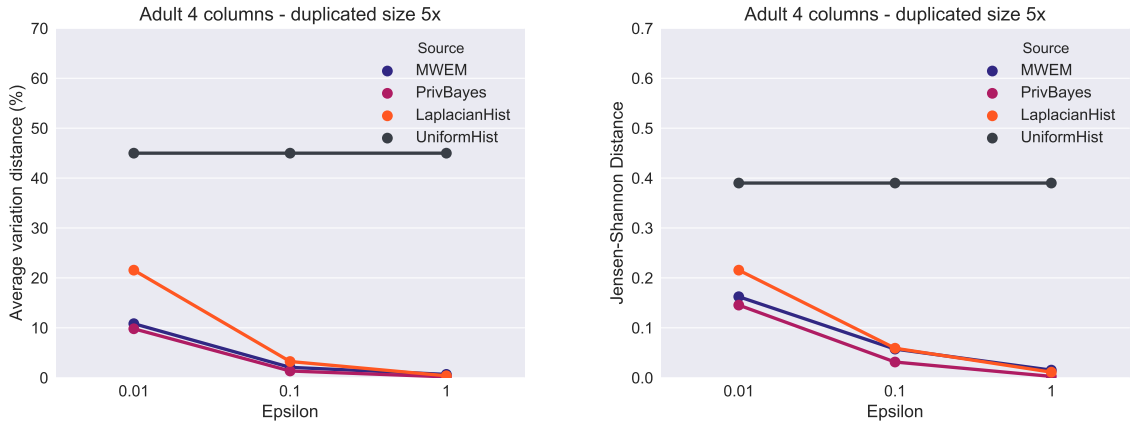


Figure 5.3: Distribution comparison limiting dimensionality by focusing only on four columns in the Adult data set and 0.1 epsilon value.



(a) average variation distance (b) Jensen-Shannon distance

Figure 5.4: AVD and JS-distance on Adult dataset with only 4 columns



(a) average variation distance (b) Jensen-Shannon distance

Figure 5.5: AVD and JS-distance on Adult dataset with only 4 columns and 5x the amount of records

5.1.4 Fixed Free Parameters

Scale-epsilon exchangeability actually influences the automatic free-parameter selection of MWEM and PrivBayes. As an increase in either scale or epsilon actually results in a equivalent increase in the automatically selected free parameter. Thus we cannot evaluate databases with a large number of records and columns, due to the free-parameters becoming too high for our machine to handle. We have therefore chosen not to comment on the performance of each algorithm on very large databases without using optimal parameter settings. However, we can evaluate large data sizes if these free parameters remain fixed, i.e. MWEM T of 70 and PrivBayes k of 4, where at least 140 and 10 are considered optimal respectively in the upcoming experiments.

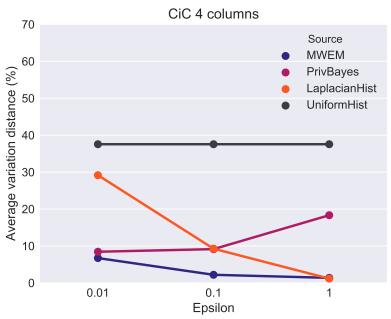


Figure 5.6: AVD on the Crimes in Chicago data set with fixed free parameters.

In Figure 5.6 we illustrate how the AVD changes with an increase of privacy budget and fixed free parameters on the Crimes in Chicago database which consists over 1 million records. MWEM

and PrivBayes actually perform quite well on very strong privacy guarantees. However, the latter actually start to perform considerably worse with larger privacy budget, suggesting that the decision of free-parameters is non-trivial and in order to examine PrivBayes' capability properly the algorithm should run with optimal settings. In contrast, MWEM seems less influenced by incorrect parameter settings and still improves over time. However in other research the automatic free parameter selection over fixed values has achieved an 7.5x decrease in error [5], again suggesting it's importance. Finally, we again observe the effect data size has on the simple Laplacian Hist, where with a privacy budget of 1 the AVD is equivalent to the more complex algorithm MWEM. Both naive approaches do not have any other parameters. Thus with input data of this magnitude, Laplacian Hist can actually provide adequate results with minimal increase in computational complexity.

5.1.5 Missing Values

When data is considered highly confidential, any form of preprocessing prior to the generation of synthetic data could already be a privacy concern. Hence, we evaluate the effects unprocessed data with missing values might have on the output quality. In Figure 5.7 we depict just one column in the ACS database, we can see the effect of missing values. Evidently, the sum of raw distributions (represented as a red faded layer) does not equal the total amount of 90.000 records, indicating that there are values missing in the histogram. Each algorithm tries to approximate the distributions according to the original distribution, but does not take into account that there might be some entries missing. As the probability distributions do not sum up to 1, the approximation will not improve even with a higher privacy budget. Therefore, this experiment suggests that data needs to be pre-processed prior to generation to remove or impute missing values. When the database is too confidential to be shared, methods for automatic data cleaning are necessary to preserve privacy.

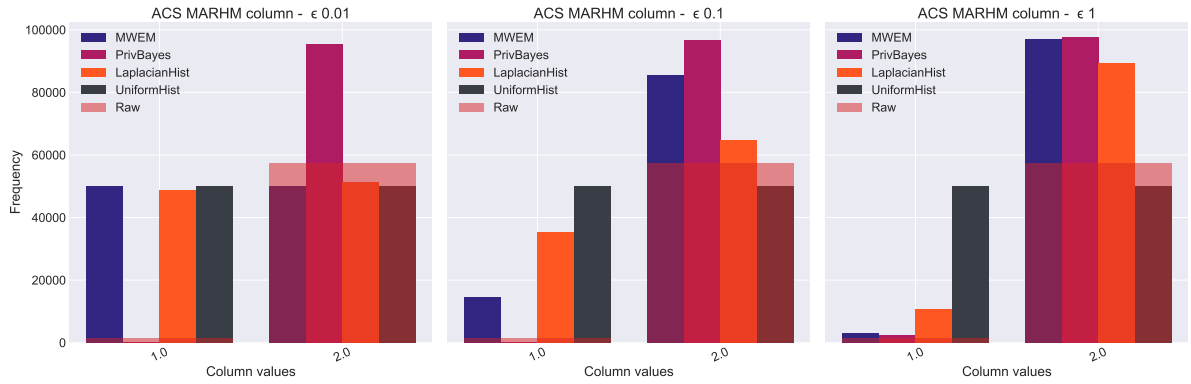


Figure 5.7: How missing values influence the synthetic distribution in Column MAHRM in the American Community Survey data set. With a red faded layer we highlight the distribution of the raw data, which evidently does not sum up to the 90k records of the database.

5.1.6 Discretizing Continuous Data

One of the requirements of all the algorithms used in this thesis, is that continuous data needs to be pre-processed and discretized. Therefore, we evaluate the case where continous data is very common: location data. We used the Facebook Check-Ins dataset, which contains time-stamps, latitudes and longitudes that are heavily discretized. As a result, this dataset is more uniform in character and with 30k rows it is similar in size and dimensionality to the Adult dataset. Hence it gives a sense on how data distribution shape affects the generation output quality.

As we can see in Figure 5.8 the sophisticated algorithms approximate the raw distribution best. However interestingly the Laplacian Hist and Uniform Hist seem almost identical in terms

of distribution and due to the uniform character of the original database actually perform quite well. When analysing the distance metrics in Figure 5.9 both report similar results in relative terms. Most interesting is the fact that under the strongest privacy guarantee the simple Uniform Hist actually outperforms MWEM. PrivBayes seems very adequate at handling uniform data and has very little to improve when epsilon increases. MWEM does catch up in the end when the privacy budget gets relaxed even further.

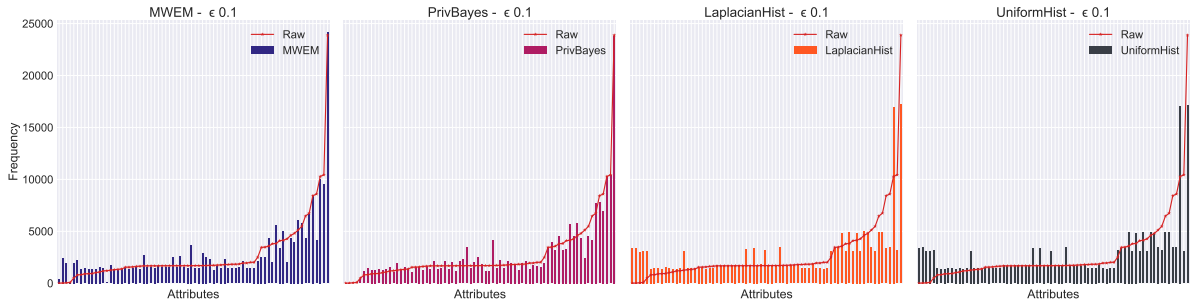
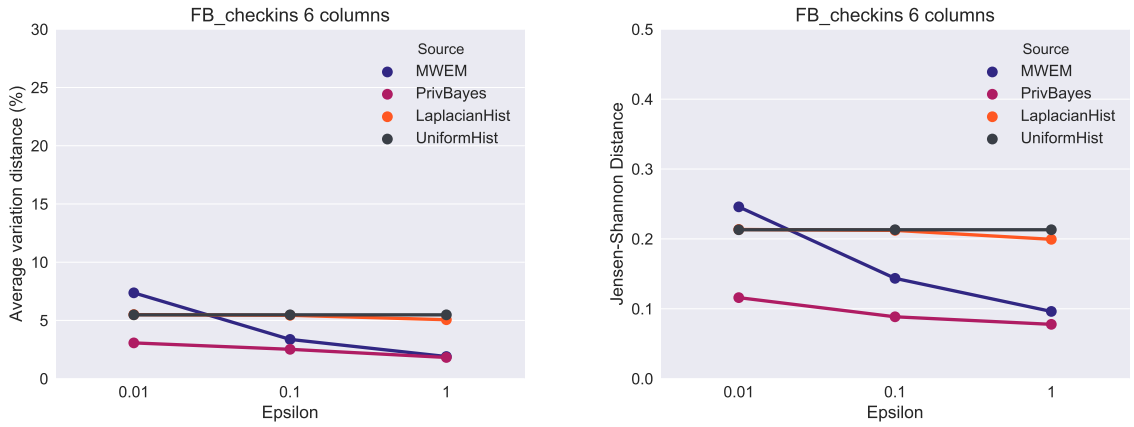


Figure 5.8: Distribution comparison Facebook Check-Ins database and 0.1 ϵ.



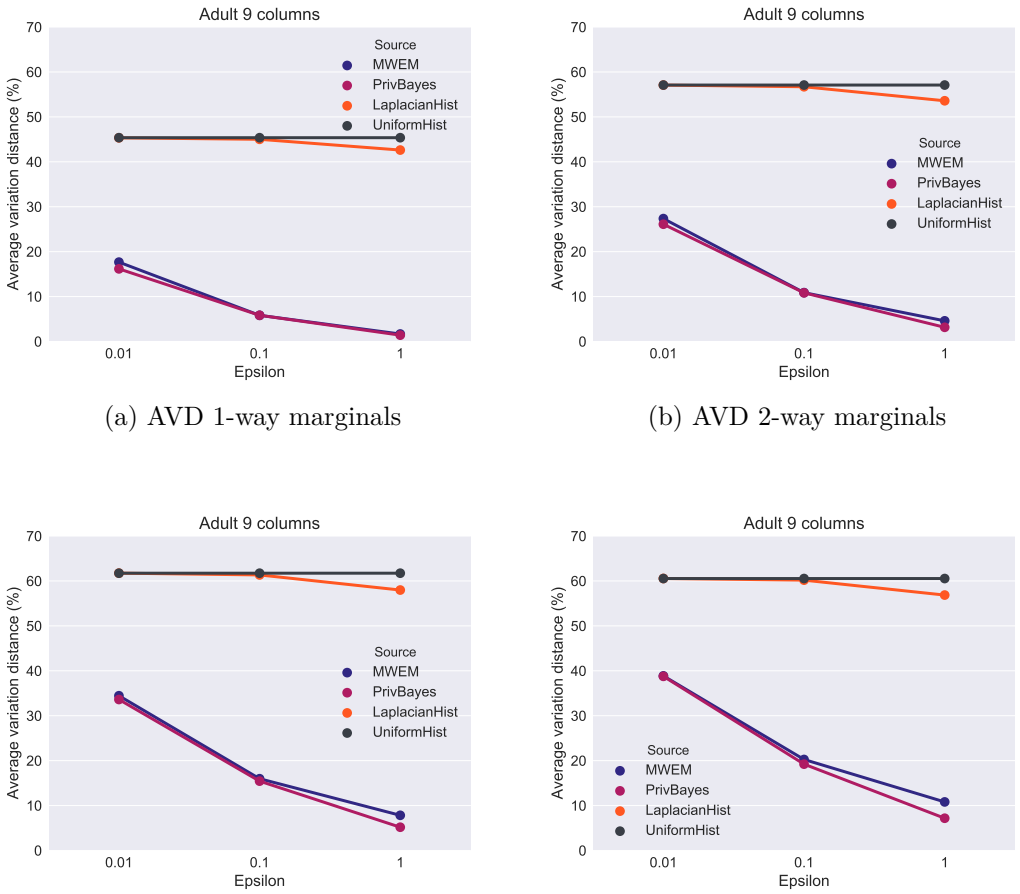
(a) average variation distance

(b) Jensen-Shannon distance

Figure 5.9: AVD and JS-distance on Facebook Check-Ins dataset

5.1.7 J-way Marginals

Up to now all distance measures have been evaluated based on the marginal distributions in each column. We can also evaluate distributions over multiple columns in order to understand their interdependence, formally known as j-way marginals, i.e. the counts for each of the possible settings of the restricted set of attributes [50]. We observed that both distance measures report relatively similar values, thus we only depict the change in AVD when increasing the j-way marginals from 1 to 4 for the Adult database in Figure 5.10. Note that each increase in j-way marginals increases the AVD slightly. However the relative deviations between algorithms over varying ϵ remains identical, indicating that evaluating higher marginals only increases overall error.



(a) AVD 1-way marginals

(b) AVD 2-way marginals

(c) AVD 3-way marginals

(d) AVD 4-way marginals

Figure 5.10: Effect of increasing j-way marginals on average variation distance

5.2 User interaction

Application in the real-world is subject to industry-specific requirements which determine the utility and feasibility of new technology adoption. Besides the previously discussed quality and privacy measures, an analysis of the steps towards actual deployment is needed to assess the challenges users might face. We start by examining the transparency of the algorithm, which enables one to understand clearly what is happening during the generation process and which consequences this process has on the generated output. Subsequently, we evaluate implementation and training scheme configuration methods. These components require an analysis of the steps to implement and deploy an algorithm to see whether it can be used in various use cases with minimal adaptations. Moreover, we also analyze whether the changing user requirements over time, i.e. new patterns that need to be captured by the synthetic data, can easily be incorporated for future private data generation and release.

5.2.1 User Interaction: Baselines

Both baselines Laplacian Hist and Uniform Hist operate on very similar principles and thus can be evaluated together. These methods work based on histogram representations of the data, where the Laplacian Hist will query each bin separately and store the noisy statistic, and the Uniform Hist just queries for the dataset size resulting in an estimation based on the uniform distribution. Hence, there is very little user interaction required for both these methods. One only needs to determine the optimal privacy budget, like any other DP method, which should

be done by a trusted analyst through experimentation or by examining the data characteristics taking into account the insights of this study.

One can visualize the process through a histogram graph, where each bin will be filled in according the respective method. When the data is highly dimensional, the visualization of the complete distribution can become a bit cluttered. Thus, it will also be harder to assess which patterns have been captured well by synthetic data. Moreover the user has no control on the prioritization of patterns and can also not add or remove specific insights. In short, these simplicity of these methods allow them to work in a variety of settings easily, but lack a form control by the user.

5.2.2 User Interaction: MWEM

MWEM actually improves on transparency at the cost of additional configuration settings. Again, the histogram representation allows for an intuitive understanding, but now the user has more control through being able to specify a set of patterns they would like to see captured well. Nevertheless if minimal distribution deviation is the objective, one can easily generate a complete list of patterns automatically and let MWEM control the optimal generation scheme. Moreover, one can view which patterns have actually been selected by the algorithm and thus what queries the new synthetic data is trained for. It is however still hard to assess how generalizable the results are beyond the selected patterns. Nevertheless in terms of transparency, the generation process and consequences on the output quality is more comprehensible.

These additional benefits of a more sophisticated approach like MWEM do make the configuration and deployment a bit more difficult. Not only does one have to decide between pattern control and generalized utility, but there are some other settings that need to be considered prior to generation as well. For one, if very specific patterns are to be captured, beyond the scope of the traditional queries, these queries need to be programmed separately in an adequate manner taking into account the sensitivity. Thus, some expert knowledge is required to ensure that the generation process continues to be ϵ-differentially private. Finally, when the algorithm is ready for deployment additional parameter settings need to be set. In the original paper, the number of iterations (i.e. number of patterns to be selected) is considered a non-trivial decision but ultimately left to the user [26]. However subsequent academic research found a way to let the algorithm automatically select the optimal number of iterations based on the input data and set privacy budget [5]. Hence, this burden is removed from the user. There remain some additional parameters that can be set, including the privacy budget allocation and number of times the selected query list gets repeated in each step. These numbers can be fixed to the default settings, i.e. 50/50 allocation and query set repetition of 100, but the implications on the output need to be tested over a variety of use cases. All in all, MWEM adds quite a few extra decisions to the user at the benefit of a more transparent process.

5.2.3 User Interaction: PrivBayes

Unlike the previous methods, PrivBayes models the data properties in a Bayesian network. In terms of transparency, one can easily visualize the network as long as the data dimensionality is not too large. In combination with the conditional probabilities one can assess how variables relate to one another and derive the probability a certain tuple with a specific set of attributes exists in the input. Hence, the user can have an overview of how data will be sampled by examining the interdependence of variables. A visualization of the complete network enhances process and output transparency. However, once the number of columns and distinct values becomes quite large the network might become a bit too cluttered.

With PrivBayes a user typically has very little control over which patterns are captured and thus this method aims to provide high generalizable utility. This lack of control makes the generation process a lot easier as less user decisions are required with regard to pattern selection

and there is no need for future inclusion of non-traditional queries. Needless to say, this does however mean that a user has no option for optimizing certain patterns and also has no notion of how well their preferred queries are modeled. There are however some hyperparameter and pre-deployment decisions to be made. For one there are 4 different variants of encoding methods, i.e. binary, gray, hierarchical and vanilla, each vary in terms of complexity and require distinct configurations. Second, while the network degree can automatically be selected, one still needs to decide on the epsilon allocation and more importantly the ratio scale of information to the scale of noise. Particularly the latter does not seem very intuitive to select and while the original paper suggests a level of 4 [27], more experiments are needed to understand whether this would be optimal for most real use cases. In conclusion, the additional pre-deployment decisions seem rather complex and non-trivial thus requiring more experimentation, while the Bayesian network itself enables a transparent process but lacks the ability to target a specific pattern set.

5.3 Computational Complexity

Synthetic data generation typically is considered a time-consuming and resource intensive process. Hence an analysis of algorithms is required to measure the computational complexity of each generation scheme in terms of time, storage, and other resources required for execution. From the literature study and our experiments we observed the influence of data size and dimensionality on the computational complexity and present an overview of our findings in Table 5.1. Besides the loading and processing time, these data features also affect the automatic parameter selection of the sophisticated algorithms and thus their space and time complexity considerably. Nevertheless, all the selected algorithms are easily processed in parallel. Where the histogram representations can be split and divided over multiple resources, and the bayesian network calculation of parent-child pairs can be distributed as well.

Table 5.1: Influence of data characteristics on the algorithm's computational complexity

	Algorithms			
	Laplacian Hist	Uniform Hist	MWEM	PrivBayes
Number of records	Negligible influence on runtime as histogram size retains its shape.	Negligible influence on runtime as histogram size retains its shape	Histogram size again retained, but runtime increases significantly as the algorithm selects a larger amount of iterations.	Influences the computation of joint distributions and increases the automatically selected network degree. Hence more network configuration need to be evaluated.
Number of columns	Increases histogram size, hence adding another column can increase the runtime exponentially.	Increases histogram size, hence adding another column can increase the runtime exponentially.	Increases histogram size, hence adding another column can increase the runtime exponentially.	Increases the amount of nodes to be modeled in the graph, and thus the amount of node combinations that need to be evaluated.

5.4 Specific Use Case: Machine Learning

While general utility results and an analysis of user interaction are essential to assess the potential of private synthetic data, it is equally important to evaluate how these algorithms would perform on real-world use cases. Particularly whether they can even be applied, under which constraints, and what we can actually expect from the output. Naturally we can not assess each possible business requirement, data type, or resource capacity within the given time frame. However we can evaluate one typical use case to give insights into the capabilities of private synthetic data generation. We feel its important to first present our insights regarding the use case selection process, as not every use case will lend itself to synthetic data generation or our current limited resources. However, we did manage to perform an analysis of a use case typical to most data-driven industries: machine learning classification. Below we detail the process of exploratory data analysis and subsequent modeling by comparing the effects of using synthetic data instead of its confidential variant. This analysis should therefore function as an example of how we can actually apply these private methods in practice.

5.4.1 Use case Selection

During the selection process, several use cases came to mind which would allow us to assess the potential of privately generated synthetic data. The objective is to substitute private generated data for its original variant and analyze whether we would obtain equivalent insights and performance. We provide more detail on this process in the next section.

We are particularly interested in evaluating a use case which requires private data and would benefit from the traditional relational database format, as most DP methods in the industry focus solely on release of aggregate information. Therefore we decided to examine a machine learning use case, particularly classification, which allows us to assess whether we can develop a system on data that can predict which class a certain entity belongs to. For instance, banks which handle highly confidential transaction data could perhaps use synthetic data to develop transaction fraud systems. Another example would be a hospital which intends to create a system for medical treatment outcome prediction.

Before we commence, we feel it is important to also discuss which use cases did not suit private synthetic data very well, or were beyond the capacity of our resources to evaluate. It also helps to understand which data sources might have problems preserving the data utility when privately generated. For instance another typical machine learning use case is prediction. In contrast to classification which is based on discrete variables, prediction models focus on continuous numbers. All the algorithms selected for evaluation in this thesis require continuous variables to be discretized. Hence, we can not perform a prediction use case without post-processing of the synthetic data. As an example, you could discretize a salary column into ranges of 20.000$ per year and generate new data. Afterwards for each record one could uniformly sample the salary within the given range to again retrieve a continuous number which can be used for prediction purposes. However this method would heavily depend on the granularity of the categories and will likely result in some data utility loss. Moreover another trade-off appears, where fine-grained granularity reduces the amount of lost information, it also naturally increases the computational cost of the sophisticated algorithms as there are now more patterns to be captured.

Similarly, when looking for data sources that would be interesting to evaluate as well as being reflective of true industry use cases you often find that the categorization of the continuous variables would yield too much loss of information. For instance, in time series analysis or location clustering, we would need to discretize timestamps and locations in regions. It is nearly impossible to categorize these attributes without either increasing the computational cost significantly or losing the data's utility. Finally, data that requires the possibility to integrate with other data sources based on unique identifiers, e.g. market baskets with customers, or patient records with treatment methods, are also not applicable with the current selection of

algorithms. Again, these variables would be too highly dimensional. However, more importantly when the goal is to enhance private practices we do not want the synthetic data to contain any records that directly link towards real individuals. Hence, as a first example we have chosen the classification use case and detail the objective below.

5.4.2 Synthetic Machine Learning Objective

As an exemplary machine learning use case we perform a classification task through logistic regression on the Adult data set to determine whether an individual earns more or less than $50.000 per year. Typically prior to modeling one starts of with an exploratory data analysis by for instance examining the data distributions and prepossess the data accordingly. As our objective is to analyze the effects of using synthetic data these steps are not described in the upcoming section. Instead we need to examine how synthetic data can actually substitute its confidential variant and the effects on the output results, as illustrated in Figure 5.11. In the simplest case one would give an analyst access to (confidential) data. After pre-processing, the analyst splits the data in training data to create the model, and test data to evaluate the model's classification performance. Subsequently the model can be used to classify new data and gather insights.

Now there is one clear privacy concern in this process when the data used contains sensitive information. When we intend to release the data to perhaps an untrusted analyst or to the public, we cannot provide direct access to this private data. Here, synthetic data can provide an answer. This data can be distributed safely and used to train a machine learning model. However, when we intend to use the model in production, its classification performance needs to be evaluated on real data. Hence, we let an analyst examine and train models on synthetic data, but test the performance on the real data without the need to disclose the sensitive information.

If results are similar to the non-private approach, synthetic machine learning can finally provide access to new people to create systems and gather insights, while also ensuring that these insights exist in the real confidential data sources. If these models actually do operate well on both data sources, these can be used in a production environment on real data sources without the need for any human analyst to access the sensitive information.

In the upcoming sections we aim to provide some insights from this analysis. A complete guide towards machine learning and the preprocessing steps prior to modeling is beyond the scope of this thesis. The creation of the logistic regression model has been done on both the original data source and the synthetic variants generated by the various algorithms and privacy settings. We compare the results and aim to provide insights on the effects of using private synthetic data for machine learning purposes.

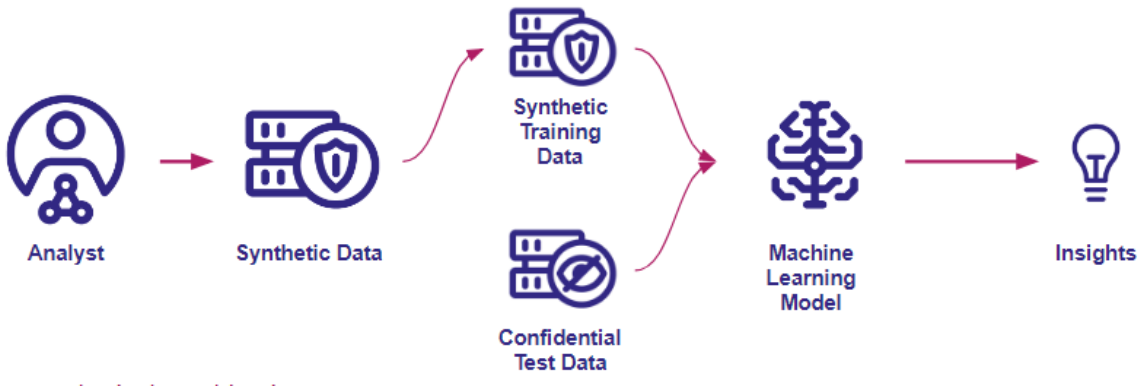


Figure 5.11: Machine learning on synthetic data where we release synthetic data for an analyst to train their model, which can then be tested and subsequently run in production on real confidential data.

5.4.3 Correlations

Before training the logistic regression model, we evaluate whether variable correlations are similar between the original data source and the synthetic variants. If the correlations are very dissimilar, we do not expect the synthetic trained models to perform well on real data. Additionally, we investigate how this might differ per algorithm and varying privacy budgets.

In Figure 5.12 we depict the variable correlations of the original data source. In principle, we would like to see similar patterns in the correlations of the synthetic data. These are illustrated in Figure 5.13, where the correlations have been ordered according to the original figure. Particularly for the two sophisticated algorithms, MWEM and PrivBayes, we observe a very close approximation to the original and notice a considerable improvement when the privacy budget increases. This indicates that the synthetic data probably is quite similar to the original at high epsilon values. As expected from the previous section the two baselines, Laplacian Hist and Uniform Hist, do not perform particularly well on this database likely due to the low amount of records and non-uniform distribution respectively. We do see a small improvement for the former when privacy budget increases, but when compared to the original correlations there do seem to be some dissimilarities.

Finally, we would like to note that its important to evaluate whether all attributes from the original data source are actually present in the generated synthetic file. If a certain attribute, e.g. occupation 'farming-fishing', has only very little occurrence in the original data source, the probability of it being sampled in the output data is quite low as well. If this attribute is missing, it will also not be included in the synthetic correlations figure. Thus, any model trained on synthetic data with missing attributes can not be tested on the raw source, as it cannot classify an unknown attribute.

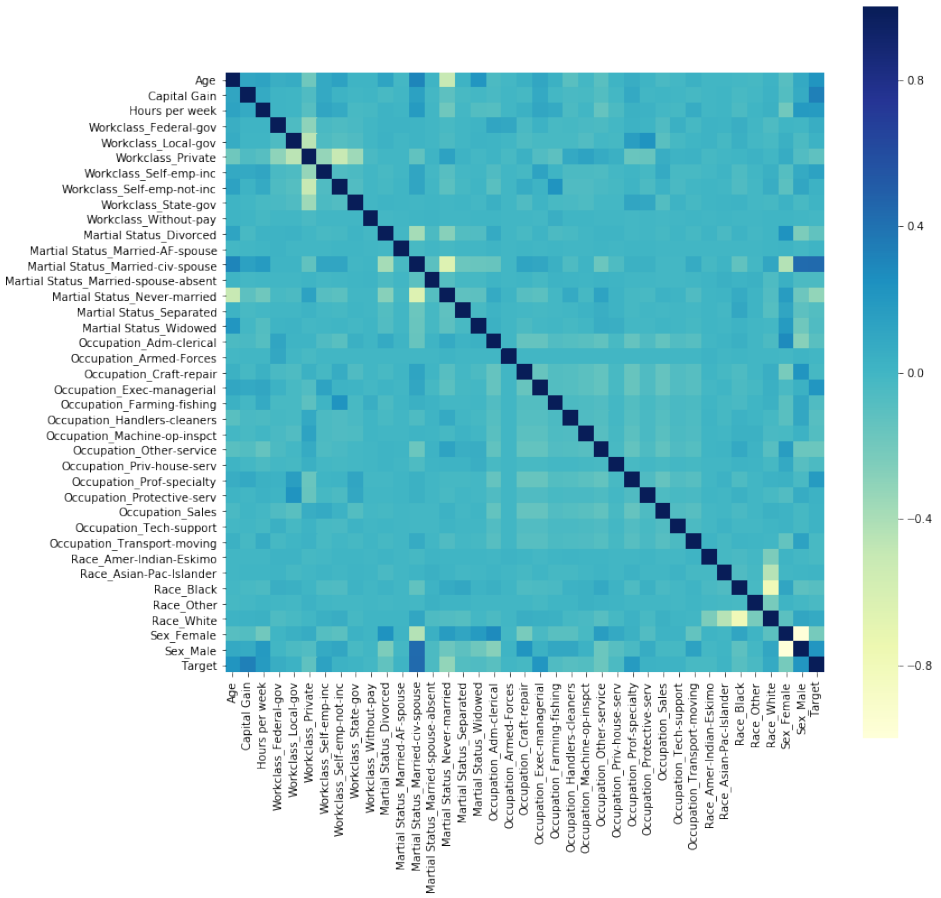


Figure 5.12: Correlations in the original Adult database

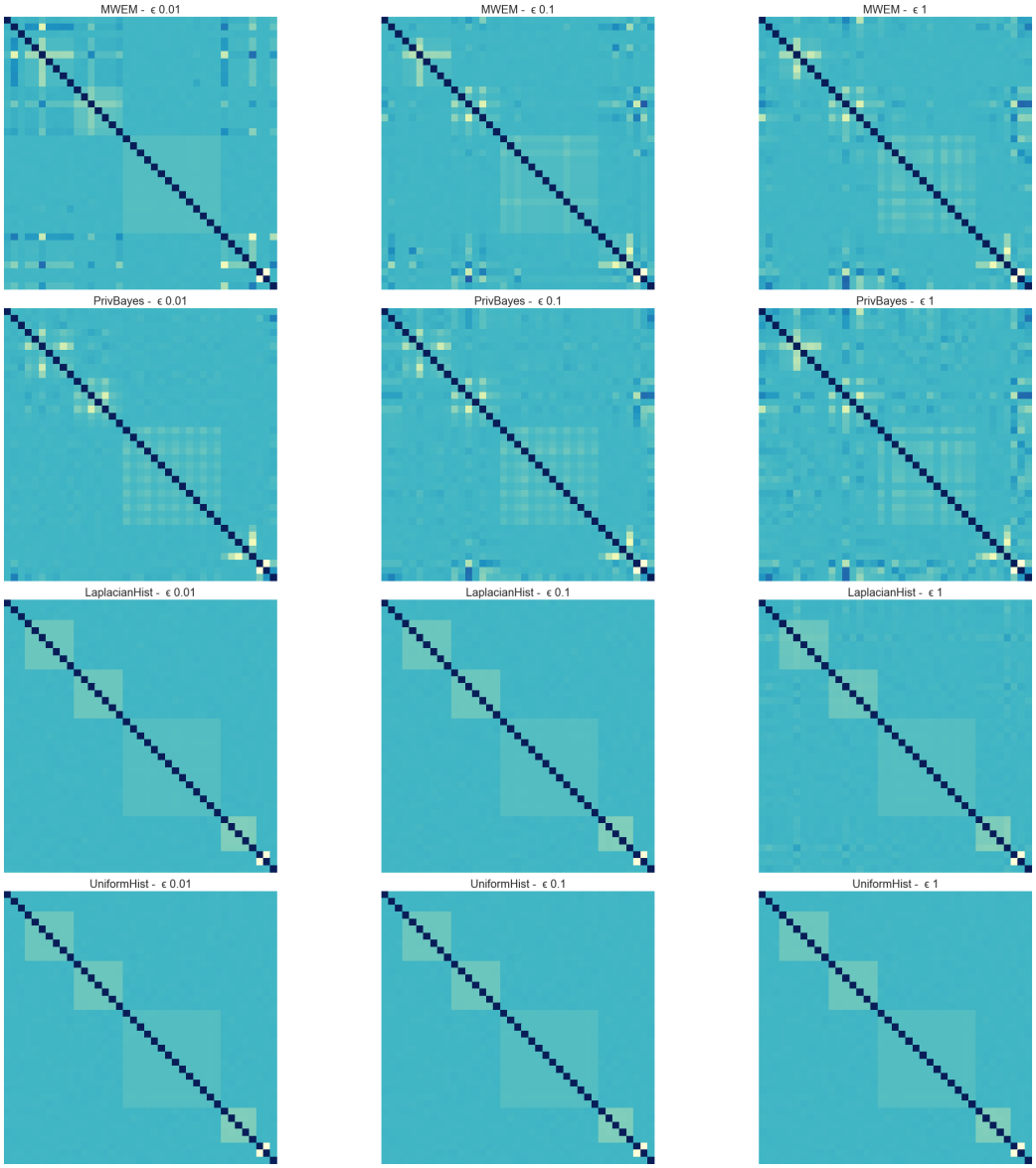


Figure 5.13: Correlations in the synthetic Adult databases. On every row the correlations of each algorithm's synthetic data are displayed by varying the privacy budget from 0.01 to 0.1 to 1.

5.4.4 Model Coefficients

Now that we have confirmed that the correlations do seem quite similar for some of the synthetic data variants, we train our models and compare the logistic regression coefficients. These coefficients indicate which attributes have a positive, neutral or negative influence on the classification task. Again, ideally we would like to see similar patterns in the coefficients when comparing the models trained on the original and synthetic variants. In Figure 5.14, we can see the coefficients for the model trained on the original Adult database. In Figure 5.15 the same ordering is used and the red line indicates the influence of each coefficient of the original model. The bars represent the coefficients of the synthetic model and ideally these should be quite similar to the original. Observe that indeed as privacy budget increases the algorithms do in the end come closer to the original coefficients, especially for MWEM and PrivBayes. The latter seems a bit more adequate at fitting the original line, however there is still quite some improvement to be made. The two baselines actually do not report any high negative or positive coefficients, hence none of the variables seem to strongly influence the classification in any direction.

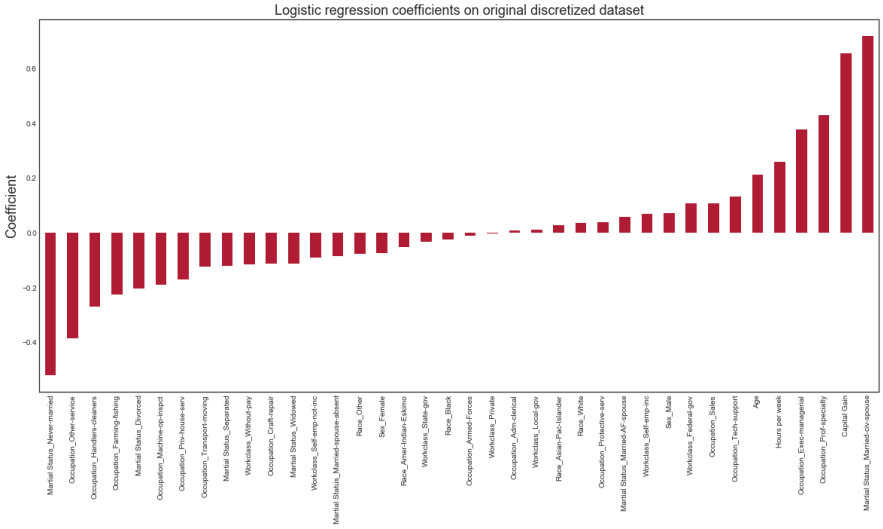


Figure 5.14: Coefficients of the logistic regression model trained on the original Adult database

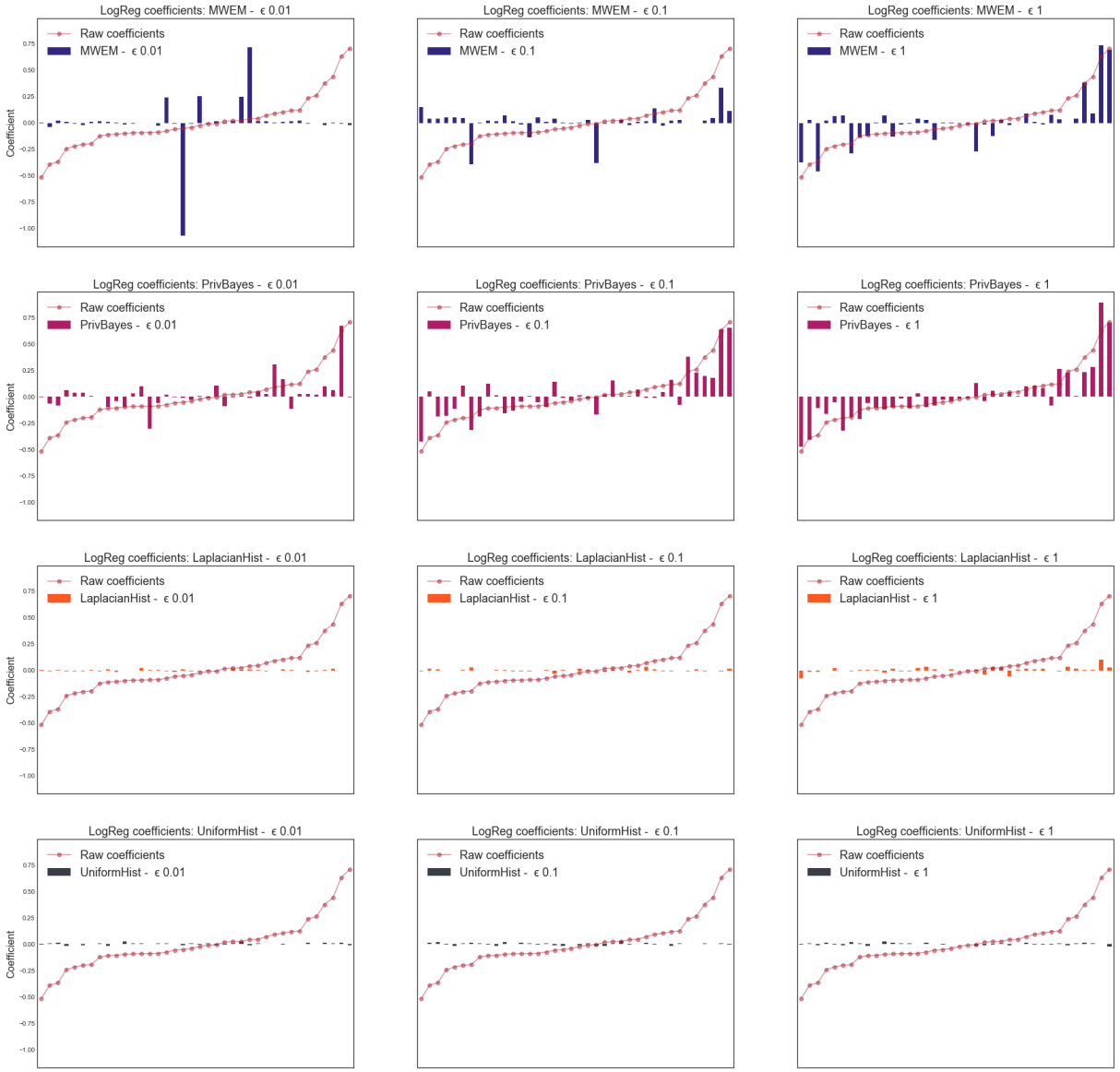


Figure 5.15: Coefficients of the logistic regression model trained on the private synthetic Adult databases. Each row represents an algorithm, where the columns indicate the different privacy budgets. A red line highlights the coefficients of the model trained on the original data and the bins represent the coefficients of the synthetic model.

5.4.5 Classification Scores

After we observed that some of the synthetic data variants are approximating the original data source in this classification task quite well, it is time to investigate some performance measures. These measures indicate how well the original and synthetic models are able to predict unseen test cases. In order to mitigate the chance of overfitting and assess the generalizability of our results, we cross-validate based on 5 slices of test data from the original data source. Additionally to not bias our model comparison on just one score measure, we will examine two common measures in classification tasks: the misclassification rate and F-score. Both measures are displayed in Figure 5.16 respectively. A red line indicates the average score when trained and tested on the original data through again 5-fold cross-validation. Note that to improve model performance, we'd like to maximize the F score and minimize the misclassification rate.

Interestingly, both measures seem to vary a bit in relative terms regarding the model performance, thus indicating that we should not bias our results to just one measure. Moreover, the baseline Uniform Hist actually performs quite well. This is rather unexpected when analyzing the model coefficients, which seem quite far off the original distribution. Perhaps the fact that each coefficient only has a marginal influence on the classification outcome resulted in this peculiar outcome. We note that MWEM and Laplacian Hist behave more randomly when comparing both measures. It seems that an increase in privacy budget does not naturally yield a better performance, perhaps due to the introduction of bias as noted in [5]. Only synthetic data generated by PrivBayes has a predictable pattern which actually comes closer to the original model with increased privacy budget. Remarkably, it seems that with still quite acceptable privacy guarantees, i.e. ϵ of 1, we are able to achieve almost equivalent scores to the original model. Hence, when using the PrivBayes and reasonable privacy guarentees the results seem quite satisfactory and perhaps indicate that there is indeed potential for synthetic data to substitute its confidential variant in a machine learning classification task. More experiments are needed to confirm this statement and assess its generalizability on other use cases, however the results do seem promising.

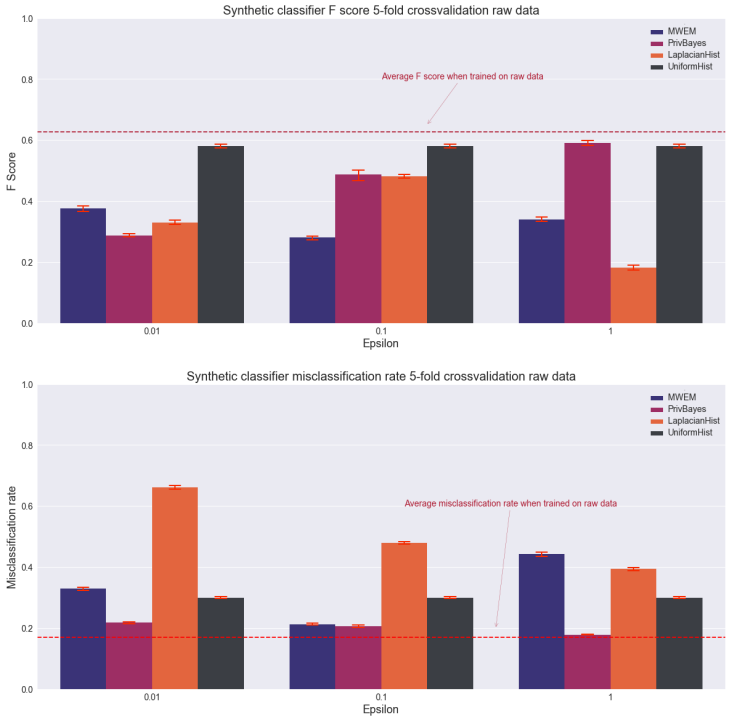


Figure 5.16: Logistic regression model classification scores tested over 5-folds of the original database. On top the F-score is displayed, at the bottom the misclassification rate. A red line indicates the average score of the original model.

Chapter 6

Conclusion

In this thesis we investigated the potential of private synthetic data generation algorithms in a real-world context. We reviewed the state of this technology and selected a few algorithms for extensive review. Subsequently, we proposed an evalution framework that provides a guideline to assess the potential of synthetic data, not only from an academic perspective, but also taking into account potential users and specific application requirements. Finally, we experimented with a selection of algorithms on various data sources and tasks to determine which factors might influence the generation and utility of privately generated data.

We conclude our work by re-examining the research objective and assess whether private synthetic data generation actually has the potential to be used in a real-world enviroment. We evaluate the strenghts and limitations of this privacy solution and discuss its relevance and utility. Finally we provide insights into future research opportunities.

6.1 Review of Research Objective

Our experiments demonstrated that synthetic data generation algorithms are able to closely approximate a sensitive database in a private manner with minimal loss of utility. Even with moderate values of epsilon we observed minimal deviation between input and output. It is however up to the user to define the level of privacy that needs to be guaranteed and what can be considered an acceptable range of utility loss. Additionally we also observed that certain data characteristics, including the data size, columnar domain and distribution shape, have a strong influence on the generation process itself and its outcomes. Large data sizes and columnar domains increase the computational cost considerably for the sophisticated approaches. In terms of outcomes, the output quality of some algorithms is highly dependent on certain input data features. For instance, when data has a high columnar domain, the large amount of patterns within the data can affect the obtained utility. Higher data sizes are naturally favorable, as some of the methods are prone to lose the original signal in lower data sizes as noise addition has a relatively stronger effect. Finally, some data distributions shapes, perhaps obtained post-discretization, do suit particular algorithms more than others. Hence, we noticed that sometimes even the most basic approaches can outperform more sophisticated methods under certain data characteristics and privacy guarantees.

Evaluation metrics and the slight randomness typical in DP mechanisms and sampling methods, do have a strong effect on the output quality as well. Hence, we observed that various academic papers proposing new algorithms used different metrics and data sources that might be particularly suited to their generation method and thus perceived outcomes. Hence, to truly get a better understanding of the state of synthetic data generation a unified approach is needed for proper evaluation. Moreover, to stimulate industry adoption the user needs to be considered as well by estimating the level of control one typically needs and the ease of implementation. Hence, we proposed an evaluation framework that takes these factors into account and gives

insight into the application of these algorithms and the expected outcomes.

More experiments are required to determine how generalizable these results are. Especially concerning the strong influence of input data features and their intended use cases. Certain tasks and data sources seemed more suited than others. Others actually complicated the process and would lose too much utility with our current selection of algorithms. For instance, columns containing continuous values or high column domains. Thus, not every data task and source is suited for synthetic data generation. Nevertheless, we focused solely on general-purpose algorithms with the constraint that continuous variables are generalized. Perhaps for these cases a more specialized algorithm could provide the required performance. While general-purpose algorithms do function very well on some data sources and even tasks beyond their training scheme, e.g. machine learning, we do have to evaluate the performance of methods focused on one specific task to determine if we can overcome the barriers of the general-purpose mechanisms.

When considering the fact that there has been very little adoption so far of DP algorithms and especially data generation methods, one starts to question why this is the case. Whether its due to the overall complexity and novelty of data privacy methods, or that there was previously a lack of necessity for privacy engineering in the absence of data protection regulations. Additionally, privacy engineering can be seen as an unnecessary burden that can partly be solved by proper security measures. Perhaps the new opportunities private practices bring are not completely clear. For one, it enables safe and responsible extraction of value without the need to identify any sensitive information and protect the rights of data subjects. When we see this shift in enhanced private practices, data can finally be shared between various entities or to the public and reach people that can actually benefit and learn new things from sources that were previously considered too confidential to release. With this change in perception one can only expect that the future of privacy engineering and its various methods, including synthetic data generation, are bound to grow.

6.2 Future Work

Variability in algorithm performance indicates that more experiments are required to determine how input data and its intended task affect the output quality. The influence of these features need to be studied in-depth to improve our understanding of their influence. Hence, more use cases need to be evaluated, including distinct data sources and use cases, to truly assess the potential of privately generated synthetic data. We focused our study primarily on data sources with minimal loss of utility when generalized and containing low columnar domain sizes. Hence, we require an evaluation of algorithms targeting continuous and highly dimensional data and assess their utility on typical industry sources, e.g. location data or transaction data. Even with our current selection of algorithms, more specific use cases need to be evaluated, e.g. prediction, clustering, exploratory data analysis, or time series forecasting. Additionally, since we noticed the variability in performance based on input data, these features need to be studied in-depth. Finally, a meta-algorithm can be developed to privately select the optimal generation scheme based on input data [51] and evaluate whether the performance does increase significantly on our proposed utility evaluation framework.

Bibliography

[1] S. D. Warren and L. D. Brandeis, "The right to privacy", *Harvard Law Review*, vol. 4, no. 5, pp. 193–220, 1890, ISSN: 0017-811X. DOI: 10.2307/1321160. [Online]. Available: http://www.jstor.org/stable/1321160 (visited on 06/20/2018).

[2] J. Rachels, "Why privacy is important", *2017*, Privacy, pp. 11–21, [Online]. Available: http://www.jamesrachels.org/CEPA10.pdf (visited on 06/20/2018).

[3] X. Hu, M. Yuan, J. Yao, Y. Deng, L. Chen, Q. Yang, H. Guan, and J. Zeng, "Differential privacy in telco big data platform", *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1692–1703, Aug. 1, 2015, ISSN: 21508097. DOI: 10.14778/2824032.2824067. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2824032.2824067 (visited on 04/06/2018).

[4] C. Dwork, "Differential privacy: A survey of results", in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds., vol. 4978, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19, ISBN: 978-3-540-79227-7 978-3-540-79228-4. DOI: 10.1007/978-3-540-79228-4_1. [Online]. Available: http://link.springer.com/10.1007/978-3-540-79228-4_1 (visited on 05/07/2018).

[5] M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang, "Principled evaluation of differentially private algorithms using DPBench", ACM Press, 2016, pp. 139–154, ISBN: 978-1-4503-3531-7. DOI: 10.1145/2882903.2882931. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2882903.2882931 (visited on 04/15/2018).

[6] D. J. Solove, *Nothing to Hide: The False Tradeoff Between Privacy and Security*. Yale University Press, May 31, 2011, 258 pp., Google-Books-ID: aaJdKX8avUAC, ISBN: 978-0-300-17231-7.

[7] G. W. Van Blarkom, J. J. Borkling, and J. G. E. Olk, *Handbook Privacy and Privacy-Enhancing Technologies*. College bescherming persoonsgegevens, 2003. [Online]. Available: http://www.andrewpatrick.ca/pisa/handbook/Handbook_Privacy_and_PET_final.pdf (visited on 06/20/2018).

[8] J. Brickell and V. Shmatikov, *The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing*. 2008.

[9] A. Narayanan and V. Shmatikov, "How to break anonymity of the netflix prize dataset", *arXiv:cs/0610105*, Oct. 18, 2006. arXiv: cs/0610105. [Online]. Available: http://arxiv.org/abs/cs/0610105 (visited on 05/08/2018).

[10] Michael Barbaro and T. Zeller Jr., "A face is exposed for AOL searcher no. 4417749 - new york times", p. 3, Aug. 9, 2006.

[11] M. Klein, T. Mathew, and B. Sinha, "A comparison of statistical disclosure control methods: Multiple imputation versus noise multiplication", pp. 1–148, Feb. 2013.

[12] S. De Capitani Di Vimercati, S. Foresti, G. Livraga, and P. Samarati, "Data privacy: Defintions and techniques", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 20, no. 6, pp. 793–817, Dec. 2012, ISSN: 0218-4885, 1793-6411. DOI: 10.1142/S0218488512400247. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218488512400247 (visited on 06/13/2018).

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis", p. 20, 2006.

[14] L. Sweeney, "K-anonymity: A model for protecting privacy", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, Oct. 2002, ISSN: 0218-4885, 1793-6411. DOI: 10.1142/S0218488502001648. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218488502001648 (visited on 06/13/2018).

[15] L. Sweeney, "Achieving k-anonymity: Privacy protection using generalization and suppression.", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, Oct. 2002, ISSN: 0218-4885, 1793-6411. DOI: 10.1142/S021848850200165X. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S021848850200165X (visited on 06/13/2018).

[16] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L -diversity: Privacy beyond k-anonymity", *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1–52, Mar. 1, 2007, ISSN: 15564681. DOI: 10.1145/1217299.1217302. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1217299.1217302 (visited on 06/13/2018).

[17] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and i-diversity", *2007*, p. 10,

[18] D. Barth-Jones, *The 're-identification' of governor william weld's medical information: A critical re-examination of health data identification risks and privacy protections*, 2012. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2076397 (visited on 08/20/2018).

[19] M. Janssen, Y. Charalabidis, and A. Zuiderwijk, "Benefits, adoption barriers and myths of open data and open government", *Information Systems Management*, vol. 29, no. 4, pp. 258–268, Sep. 2012, ISSN: 1058-0530, 1934-8703. DOI: 10.1080/10580530.2012.716740. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10580530.2012.716740 (visited on 06/20/2018).

[20] R. P. Pargas, M. J. Harrold, and R. R. Peck, "Test-data generation using genetic algorithms", *Journal of Software Testing*, p. 19, 1999. [Online]. Available: https://www.cc.gatech.edu/~harrold/6340/cs6340_fall2009/Readings/pga.pdf (visited on 09/01/2018).

[21] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis", p. 12, 2010.

[22] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3, pp. 211–407, 2013, ISSN: 1551-305X, 1551-3068. DOI: 10.1561/0400000042. [Online]. Available: http://www.nowpublishers.com/articles/foundations-and-trends-in-theoretical-computer-science/TCS-042 (visited on 05/07/2018).

[23] K. Nissim, T. Steinke, A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, D. R. O'Brien, and S. Vadhan, "Differential privacy: A primer for a non-technical audience", p. 41, 2018.

[24] A. Narayanan, J. Huey, and E. W. Felten, "A precautionary approach to big data privacy", in *Data Protection on the Move*, S. Gutwirth, R. Leenes, and P. De Hert, Eds., vol. 24, Dordrecht: Springer Netherlands, 2016, pp. 357–385, ISBN: 978-94-017-7375-1 978-94-017-7376-8. DOI: 10.1007/978-94-017-7376-8_13. [Online]. Available: http://link.springer.com/10.1007/978-94-017-7376-8_13 (visited on 06/22/2018).

[25] F. McSherry and K. Talwar, "Mechanism design via differential privacy", p. 10, 2007.

[26] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release", p. 13, 2012.

[27] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via bayesian networks", *ACM Transactions on Database Systems*, vol. 42, no. 4, pp. 1–41, Oct. 27, 2017, ISSN: 03625915. DOI: 10.1145/3134428. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3155316.3134428 (visited on 04/15/2018).

[28] M. Lyu, D. Su, and N. Li, "Understanding the sparse vector technique for differential privacy", *arXiv:1603.01699 [cs]*, Mar. 5, 2016. arXiv: 1603.01699. [Online]. Available: http://arxiv.org/abs/1603.01699 (visited on 06/22/2018).

[29] J. Snoke, G. Raab, B. Nowok, C. Dibben, and A. Slavkovic, "General and specific utility measures for synthetic data", *arXiv:1604.06651 [stat]*, Apr. 22, 2016. arXiv: 1604.06651. [Online]. Available: http://arxiv.org/abs/1604.06651 (visited on 03/25/2018).

[30] A. B. Tsybakov, *Introduction to nonparametric estimation*, ser. Springer series in statistics. New York ; London: Springer, 2009, 214 pp., OCLC: ocn300399286, ISBN: 978-0-387-79051-0 978-0-387-79052-7.

[31] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding", IEEE, 2004, pp. 30–30, ISBN: 978-0-7803-8280-0. DOI: 10.1109/ISIT.2004.1365067. [Online]. Available: http://ieeexplore.ieee.org/document/1365067/ (visited on 06/14/2018).

[32] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy ", p. 9, 2008.

[33] C. Dwork, M. Naor, O. Reingold, S. Vadhan, and G. N. Rothblum, "When and how can data be eciently released with privacy?", p. 41, 2008.

[34] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: A survey and review", *arXiv:1412.7584 [cs]*, Dec. 23, 2014. arXiv: 1412.7584. [Online]. Available: http://arxiv.org/abs/1412.7584 (visited on 08/16/2018).

[35] B. Stoddard, Y. Chen, and A. Machanavajjhala, "Differentially private algorithms for empirical machine learning", *arXiv:1411.5428 [cs]*, Nov. 19, 2014. arXiv: 1411.5428. [Online]. Available: http://arxiv.org/abs/1411.5428 (visited on 08/16/2018).

[36] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal components", p. 9, 2013.

[37] F. Liu, "Model-based differentially private data synthesis", *arXiv:1606.08052 [stat]*, Jun. 26, 2016. arXiv: 1606.08052. [Online]. Available: http://arxiv.org/abs/1606.08052 (visited on 08/16/2018).

[38] H. Li, L. Xiong, and X. Jiang, "Differentially private synthesization of multi-dimensional data using copula functions", *Advances in database technology : proceedings. International Conference on Extending Database Technology*, vol. 2014, pp. 475–486, 2014. DOI: 10.5441/002/edbt.2014.43. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4232968/ (visited on 08/16/2018).

[39] F. K. Dankar and K. El Emam, "The application of differential privacy to health data", in *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, ser. EDBT-ICDT '12, New York, NY, USA: ACM, 2012, pp. 158–166, ISBN: 978-1-4503-1143-4. DOI: 10.1145/2320765.2320816. [Online]. Available: http://doi.acm.org/10.1145/2320765.2320816 (visited on 08/16/2018).

[40] F. K. Dankar and K. E. Emam, "Practicing differential privacy in health care: A review", p. 33, 2013.

[41] A. Gkoulalas-Divanis, G. Loukides, and J. Sun, "Publishing data from electronic health records while preserving privacy: A survey of algorithms", *Journal of Biomedical Informatics*, vol. 50, pp. 4–19, Aug. 2014, ISSN: 15320464. DOI: 10.1016/j.jbi.2014.06.002. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1532046414001403 (visited on 08/16/2018).

[42] N. Li, W. Qardaji, D. Su, and J. Cao, "PrivBasis: Frequent itemset mining with differential privacy", *arXiv:1208.0093 [cs]*, Jul. 31, 2012. arXiv: 1208.0093. [Online]. Available: http://arxiv.org/abs/1208.0093 (visited on 08/16/2018).

[43] G. Loukides, A. Gkoulalas-Divanis, and J. Shao, "Efficient and flexible anonymization of transaction data", *Knowledge and Information Systems*, vol. 36, no. 1, pp. 153–210, Jul. 1, 2013, ISSN: 0219-1377, 0219-3116. DOI: 10.1007/s10115-012-0544-3. [Online]. Available: https://link.springer.com/article/10.1007/s10115-012-0544-3 (visited on 08/16/2018).

[44] E. Shen and T. Yu, "Mining frequent graph patterns with differential privacy", *arXiv:1301.7015 [cs]*, Jan. 29, 2013. arXiv: 1301.7015. [Online]. Available: http://arxiv.org/abs/1301.7015 (visited on 08/16/2018).

[45] R. Chen, B. C. M. Fung, and B. C. Desai, "Differentially private trajectory data publication", *arXiv:1112.2020 [cs]*, Dec. 9, 2011. arXiv: 1112.2020. [Online]. Available: http://arxiv.org/abs/1112.2020 (visited on 08/16/2018).

[46] J. Zhang, X. Xiao, and X. Xie, "PrivTree: A differentially private algorithm for hierarchical decompositions", *arXiv:1601.03229 [cs]*, Jan. 13, 2016. arXiv: 1601.03229. [Online]. Available: http://arxiv.org/abs/1601.03229 (visited on 08/16/2018).

[47] D. Zhang and M. Hay, "Challenges of visualizing differentially private data", p. 4, 2016.

[48] R. Chen, Q. Xiao, Y. Zhang, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference", ACM Press, 2015, pp. 129–138, ISBN: 978-1-4503-3664-2. DOI: 10.1145/2783258.2783379. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2783258.2783379 (visited on 06/22/2018).

[49] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*, ser. Adaptive computation and machine learning. Cambridge, MA: MIT Press, 2009, 1231 pp., ISBN: 978-0-262-01319-2.

[50] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar, "Privacy, accuracy, and consistency too: A holistic solution to contingency table release", in *In: Proc. of the 26th Symposium on Principles of Database Systems (pods*, 2007, pp. 273–282.

[51] I. Kotsogiannis, A. Machanavajjhala, M. Hay, and G. Miklau, "Pythia: Data dependent differentially private algorithm selection", ACM Press, 2017, pp. 1323–1337, ISBN: 978-1-4503-4197-4. DOI: 10.1145/3035918.3035945. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3035918.3035945 (visited on 04/15/2018).

Appendix A

DP Architectures Comparison

In this section we delve into the three most common differentially private architectures introduced in Chapter 2. We examine their strengths and weaknesses, from which we can infer the types of use cases each system would work for best.

A.1 Comparison of the Three Architectures

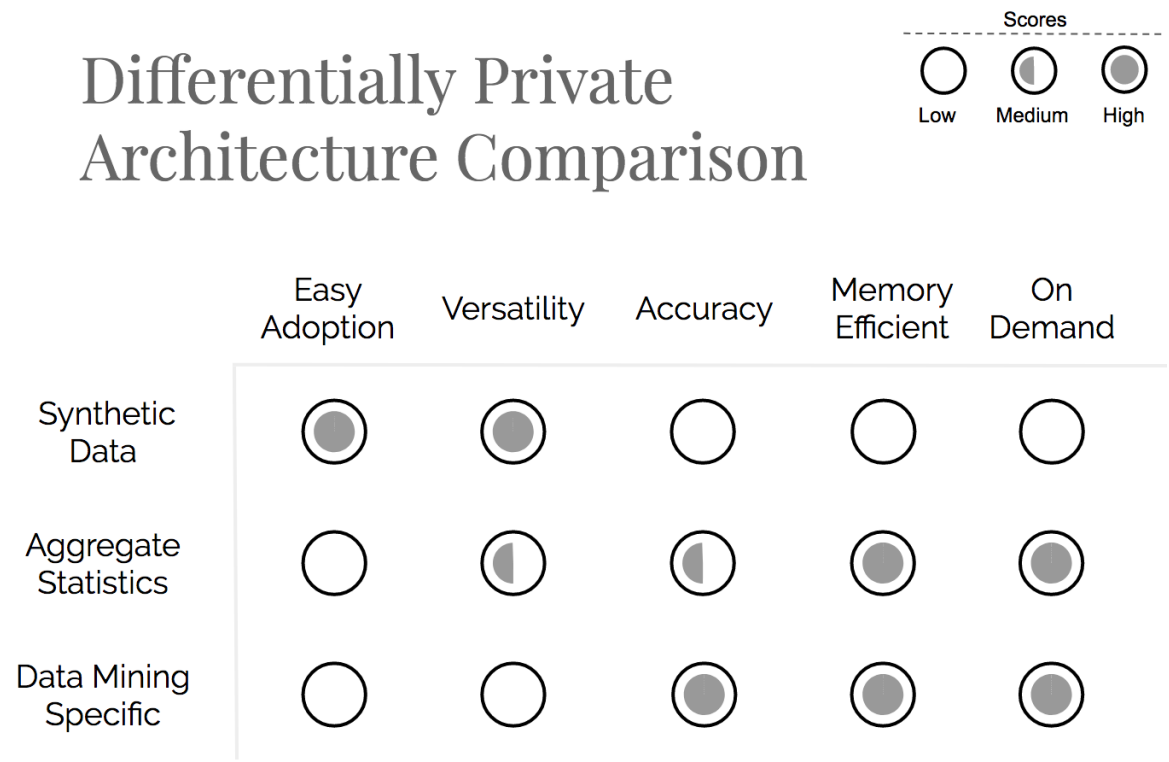


Figure A.1: Architecture comparison of differential privacy engineering solutions

In Figure A.1, we evaluate each architecture on a set of criteria. One major concern which inhibits adoption rates of data privacy architectures are the amount of alterations required within the system and current internal practices with regard to data mining. Ideally, one prefers to limit these changes, while still providing strong privacy guarantees. Here, synthetic data is the strongest contender, as database services and data mining tasks do not require any change only the substitution of raw for synthetic data. Query interfaces and data mining specific systems are more restrictive and require additional changes in the logic of the database management.

In terms of deployment, the versatility of the solution becomes an important factor. Data mining specific architectures only focused on one task, therefore are unable to provide privacy for any other usages of the data. Aggregate statistics are a bit more versatile, but limited towards towards the supported queries. Synthetic data can be used in a traditional format and is therefore again the favorable choice. It can be used for any purpose, except linking and external data integration, which normally in itself is something one wants to avoid in a private setting.

Accuracy in general improves with more tailored approaches that only introduce noise for specific queries and thus best suits the last architecture. Private aggregate statistics provides high accuracy for traditional aggregation queries, however complex data mining tasks will operate only on these statistics and thus might need to combine several noisy answers in order to achieve its goal. Synthetic data tends tends to have lower accuracy for a data mining task, but is heavily dependent on the generation scheme. Normally, a complete distribution needs to be modeled, thus requiring larger cumulative noise. Still accuracy drops can be minimal compared to the other methods if a proper scheme is applied.

Finally, on demand access and memory efficiency represent the computational complexity of the algorithms. Aggregate statistics and data mining specific architectures normally only retrieve a set of statistics, which in general is fast and requires limited memory. Synthetic data on this front is clearly inferior, as it requires considerable time and memory to generate a full database, although dependent on the complexity of the synthesis method. Often these methods are highly parallelizable and thus does not become a major issue. Moreover, generating data is normally not considered a real-time issue.

A.2 Architecture Applications

Evidently, the solutions cater to different data mining requirements, although could coexist within one system. Stand-alone they provide different guarantees. Aggregate statistics are an interesting solution when traditional operations are required on the data, or an organization intends to release statistics safely. For instance, a social media platform would like to send useful information to organizations, while also ensuring that their internal analysts can not learn anything specific about an individual.

Data mining specific is limited to one task, but does this particularly well in terms of accuracy. Therefore when a certain system is highly dependent on strong performance, this would be the favorable solution, even with the added complexity. For instance, in the case of fraud detection where a system attempts to predict if certain transactions are suspicious. Clearly this task needs to be fast, private, and highly accurate so the person affected can be contacted immediately, while also limiting the amount of false alarms.

Synthetic data tailors to the need of open access data in traditional format. Particularly appealing when one wants to retain internal practices or share data externally, where it can be used for varied purposes. For instance, when a health care institution wants to share their patient records with external researches of an university. Patient records are generated privately, however the patterns remain. As a result, research can be done on highly confidential data to discover new insights without exposing any individual.

TRITA TRITA-EECS-EX-2018:595