entropy

MDPI

Article

Utility–Privacy Trade-Off in Distributed Machine Learning Systems

Xia Zeng [1,2,†], Chuanchuan Yang [2,3,†] and Bin Dai [1,2,*,†]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[2] Peng Cheng Laboratory, Shenzhen 518055, China
[3] Department of Electronics, Peking University, Beijing 100871, China
* Correspondence: daibin@home.swjtu.edu.cn; Tel.: +86-135-480-53724
† These authors contributed equally to this work.

Abstract: In distributed machine learning (DML), though clients' data are not directly transmitted to the server for model training, attackers can obtain the sensitive information of clients by analyzing the local gradient parameters uploaded by clients. For this case, we use the differential privacy (DP) mechanism to protect the clients' local parameters. In this paper, from an information-theoretic point of view, we study the utility–privacy trade-off in DML with the help of the DP mechanism. Specifically, three cases including independent clients' local parameters with independent DP noise, dependent clients' local parameters with independent/dependent DP noise are considered. Mutual information and conditional mutual information are used to characterize utility and privacy, respectively. First, we show the relationship between utility and privacy for the three cases. Then, we show the optimal noise variance that achieves the maximal utility under a certain level of privacy. Finally, the results of this paper are further illustrated by numerical results

Keywords: differential privacy; distributed machine learning; mutual information; Gaussian noise; trade-off

Citation: Zeng, X.; Yang, C.; Dai, B. Utility–Privacy Trade-Off in Distributed Machine Learning Systems. *Entropy* **2022**, *24*, 1299. https://doi.org/10.3390/e24091299

Academic Editors: Diego Oliva and Ali Rıza Yıldız

Received: 27 June 2022
Accepted: 8 September 2022
Published: 14 September 2022

1. Introduction

With the rapid development of data-driven intelligent applications and the increasing attention on data security, distributed machine learning (DML) has been one of the hottest research fields in machine learning. The goal of DML is to deploy tasks with a huge quantity of data and computations to multiple machines in a distributed way, so as to improve the speed and scalability of data computation, reduce the time consumption of tasks, and improve the privacy performance. In Ref. [1],the authors summarized the design principles of a DML platform and algorithm from four aspects: program deployment and execution, task communication mode, and communication content. In Ref. [2], the authors analyzed and summarized the research status of machine learning algorithms and parallel algorithms based on big data. In Ref. [3], the authors compared the scale and availability of the current mainstream DML platforms, analyzed the fault tolerance and bottlenecks of these platforms, and compared their effects on handwritten data sets. In Ref. [4], the authors reviewed the research status and application of parallel machine learning algorithms, and looked forward to its development trend. In Ref. [5], the authors reviewed some popular algorithms and optimization techniques in the field of machine learning, and focused on the current status, applications, and future development trends of related platforms and algorithms for DML.

However, with the problem of privacy leakage caused by data sharing in DML, researchers have tried to study protection schemes in DML. Specifically, Ref. [6] considered the privacy protection in DML in the case of arbitrary worker collusion. Random quantization was used to convert the data set and weight vector of each round into a finite field, and Lagrange coding was used to encode the quantized value and random matrix to protect privacy. Subsequently, Ref. [7] proposed a DML framework for privacy protection, which

eliminated the assumption of trusted servers and provided users with differentiated privacy according to the sensitivity of data and the trust of servers. In addition, Ref. [8] proposed a distributed learning algorithm based on differential privacy, which kept both the data and the model information theoretically private, while allowing an efficient parallelization of training across distributed workers. Recently, Ref. [9] compared the impact of two different privacy protection methods, local differential privacy and federated learning, on DML. The results showed that differential privacy could achieve the best misclassification rate below 20 percent. To sum up, existing discussions about data privacy in DML mainly focus on Lagrange coding, differential privacy, and federated learning. Since differential privacy is promising in the privacy preserving of DML, in this paper, we choose a differential privacy mechanism as our main tool for analyzing the properties of distributed machine learning systems.

In differential privacy, data need to be added by noise to ensure data privacy. Utility is used to characterize the usefulness of the polluted data generated by applying differential privacy to the original data. From an information theory aspect, mutual information characterizes the correlation between two random variables, and [10] used the mutual information as a way to characterize the utility of the polluted data generated by differential privacy. Hence, in this paper, we also used mutual information to define utility in a differential privacy mechanism.

In order to avoid reverse data retrieval in DML, the differential privacy mechanism was introduced to add noise to the parameters uploaded by clients, which protects the privacy of each client data. DP is a privacy protection mechanism proposed by Dwork et al. [11–16]. This mechanism uses random noise to ensure that the public output does not leak the client's privacy. The kinds of added noise generally include Laplacian noise [6], Gaussian noise [17], and exponential noise [18]. However, among the differential privacy mechanism studies, most of them focus on how to reduce the amount of privacy leakage and ignore the utility of the data after noise addition. There is little literature on the relationship between the utility and privacy of the data after noise addition. For example, Ref. [19] used the minimum entropy to quantify the amount of information leakage and calculated the upper bound of information leakage $ulog_2\frac{ve^\epsilon}{v-1+e^\epsilon}$ when the DP conditions were satisfied. Ref. [20] and others defined a formula for information privacy by defining the posterior probability of the same query result of adjacent data sets and proved that if a mechanism satisfies the information privacy with the security parameter, then it also satisfied the differential privacy 2ϵ, and proved the upper limit $\frac{\epsilon}{2n}$ of mutual information between the data sets and the query return value. In [21], the authors proved that in the joint differential privacy of two data sets, the upper bound of mutual information between data sets and query results was further reduced, and the maximum value was $3n\epsilon$. In [22], the authors studied the boundary between the maximum allowable distortion and the privacy budget in the case of noninteractive data release. At the same time, they compared the privacy protection strength of differential privacy with that of reconfigurable privacy and mutual information privacy under the same distortion. The degree of distortion could directly measure the utility of the algorithm mechanism. The optimization problem was established in the paper, which solved the problem of the maximum degree of distortion of different privacy protection mechanisms under the condition of satisfying differential privacy.

As is known to all, utility is one of the important indicators to measure the performance of algorithms in DP. Hence, we aim to find the utility–privacy trade-off of DML from an information-theoretic point of view in this paper. Specifically, three cases including independent clients' local parameters with independent DP noise and dependent clients' local parameters with independent/dependent DP noise are considered. We assume that the local parameters and added noise in distributed machine learning are subject to a Gaussian distribution. This is because Gaussian distribution models are widely used in machine learning. Many machine-learning models with probability distribution as the core mostly assume that the data have Gaussian distributions, e.g., logistic regression models, naive Bayes models, and so on. Why can some data be assumed to follow a Gaussian distribution? The intuitive reason is that real-life examples generally satisfy

Gaussian distribution, such as the distribution of students' grades. Furthermore, a Gaussian distribution has many advantages: (1) It is easy to describe, and only two parameters are needed to describe it, the mean and variance, which are the essential information of the distribution. (2) A Gaussian distribution is easy to calculate. It has some good mathematical properties. The data that obey a Gaussian distribution still obey a Gaussian distribution after some operations. For example, a linear combination of normal random variables is still a normal random variable. (3) Many random variables in reality are formed by the combined influence of a large number of independent random factors, and each of the individual factors plays a small role in the overall impact. Such random variables tend to approximately obey a Gaussian distribution (objective background to the central limit theorem). (4) When the mean and variance are known, the entropy of the Gaussian distribution is the largest among all distributions. When the data distribution is unknown, the model with the largest entropy is usually selected. Therefore, it is reasonable to assume that the local parameters and the added noise in our distributed machine learning follow a Gaussian distribution.

Based on the above three cases, the main research methods of this paper are as follows. First, we establish the utility–privacy trade-off for these three cases. Then, we determine the optimum noise variances that achieve the maximal utility under a certain level of privacy. Finally, we further explain the results of this paper by numerical examples.

The remainder of this paper is organized as follows. Section 2 mainly introduces the background knowledge of DML and DP, gives the framework of DML–DP established in this paper, and uses mutual information and conditional mutual information to characterize utility and privacy. Section 3 analyzes the relationship between utility and privacy in DML based on the DP framework and gives the noise level that can obtain the maximum utility under the condition of privacy with three different cases, including independent clients' local parameters with independent DP noise and dependent clients' local parameters with independent/dependent DP noise. Section 4 summarizes all the results and discusses the limitations of this paper and future work.

2. Preliminaries and Model Formulation

In this section, the preliminary background knowledge of DML and DP is introduced. In addition, we present the distributed machine learning–(mutual information-differential privacy) (DML–(MI-DP)) model that is discussed in the next section.

2.1. Preliminaries

Distributed machine learning: The goal of DML is to solve how to coordinate and utilize a large number of GPU clusters and massive data to complete the training of a deep learning model and obtain good convergence, so as to achieve relatively high performance. DML involves how to allocate training tasks, how to allocate computing resources, and coordinate various functional modules to achieve the balance between training speed and accuracy. A DML system usually includes the following main modules: data model partition module, single machine optimization module, communication module, and model and data aggregation module. Each module has a variety of implementations, and each implementation method can also be arranged and combined, which makes the methods of DML diverse.

In this paper, we mainly studied the privacy disclosure problem from the DML framework of data partitioning. In order to visualize the problems studied, this paper adopts the following framework, as shown in Figure 1. The main learning process is as follows: There is a central server and n clients. An active client inputs the locally owned data set into the model, and after the model is trained, the model parameter is obtained by the client and then uploaded to the server. On the server side, after it has received the model parameters (also called local parameters) provided by each client, it integrates the local parameters into a global parameter in some way. Here, note that D_i represents the data set owned by the client C_i, and X_i, $i \in \{1, 2, \ldots, n\}$ is the gradient variable of the model trained locally.

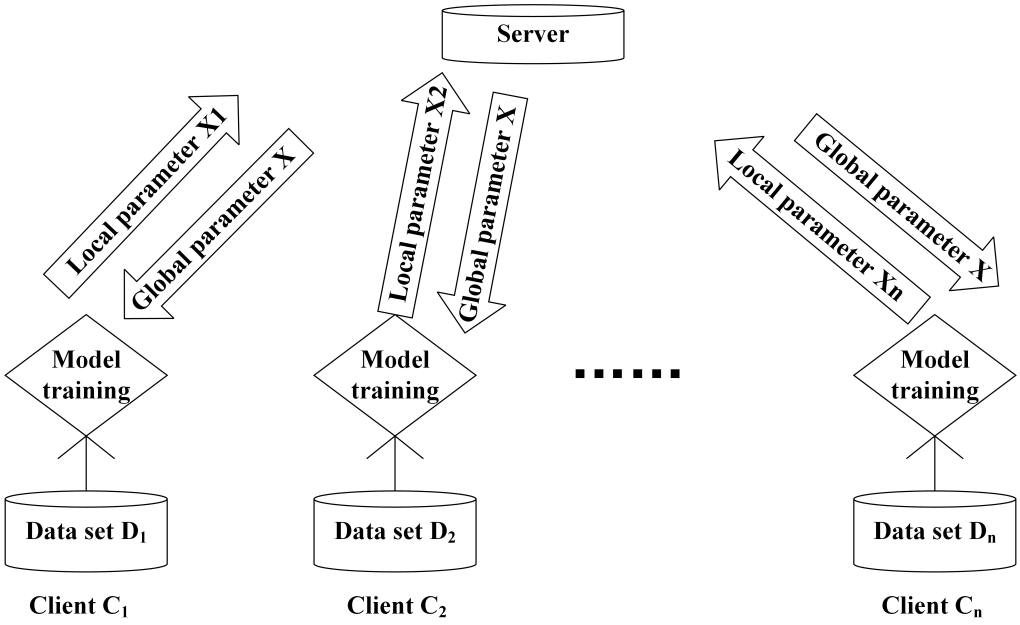

Figure 1. A general distributed machine learning framework.

Differential privacy: DP prevents differential attacks. The goal of DP is to protect the privacy of each entry in the database while answering queries about the total quantity of data. There are several definitions of DP, such as traditional DP [23] and Renyi DP [24–26]. Since Shannon's definition of mutual information has been widely adopted in DP, we also used Shannon's definition of mutual information in differential privacy (MI-DP) in this paper. In [10], the authors proposed the concept of MI-DP by defining similarity and demonstrated the relationship between MI-DP and the two types of traditional DP in terms of security strength. In fact, the MI-DP is sandwiched between ϵ-differential privacy and (ϵ, δ)-differential privacy in terms of its strength [27]. MI-DP is fundamentally related to conditional mutual information. The conceptual advantage of using mutual information, aside from yielding a simpler and more intuitive definition of differential privacy, is that its properties are well understood. Several properties of differential privacy are easily verified for the conditional mutual information [27].

Definition 1 (Mutual-Information Differential Privacy [10])**.** *A randomized mechanism $P_{R|M^n}$ satisfies ϵ-mutual-information differential privacy if*

$$\max_i I(M_i; R|M^{-i}) \leq \epsilon, \tag{1}$$

where R is the output of randomized mechanism $P_{R|M^n}$, $M^n = (M_1, \ldots, M_n)$ is a database, M^{-i} denotes the other data in the database except for the M_i element, and $\epsilon > 0$ represents the privacy budget: the larger ϵ, the lower the privacy requirements, and the smaller ϵ, the stronger the privacy.

This definition clearly reveals what kind of privacy is guaranteed by DP and what kind of privacy is not, which is easy to understand intuitively. For example, we can suppose that an adversary already knows about all except a certain data element, and they want to use the randomized mechanism to analyze the remaining data information. This is also known as the strong adversary hypothesis. This hypothesis is clearly revealed in MI-DP by conditional mutual information [10].

By adding random noise, DP ensures that the public output results will not be significantly changed due to an entity being in the data set and gives a quantitative model for the degree of privacy leakage. Different kinds of noise can be added to this model. For example, Laplace noise, exponential noise, or Gaussian noise can be chosen.

2.2. Model Formulation

Our framework, a general distributed machine learning framework based on differential privacy: DML can train large quantities of data locally due to its distributed structure. However, in the process of DML, an attacker can analyze the model parameters X_i ($i \in \{1, 2, \ldots, n\}$) uploaded by each client to obtain the client's sensitive information [28,29]. Therefore, we use the method of combining DP with Gaussian noise and distributed machine learning to deal with the risk of such privacy leakage. The model architecture is shown in Figure 2. In fact, it adds random noise on the basis of the general DML framework to complete DP. After the clients have trained the model locally, the model parameters X_i ($i \in \{1, 2, \ldots, n\}$) are not directly uploaded to the server, but are handed over to a trusted third party. We assume that the channel through which the clients transmit the local parameters to the trusted third party is absolutely safe and reliable. The third party adds random Gaussian noise Z_i to each local parameter X_i ($i \in \{1, 2, \ldots, n\}$), and finally, after adding noise, the local parameters are transmitted to the server by a trusted third party to complete the aggregation and obtain the global parameter.

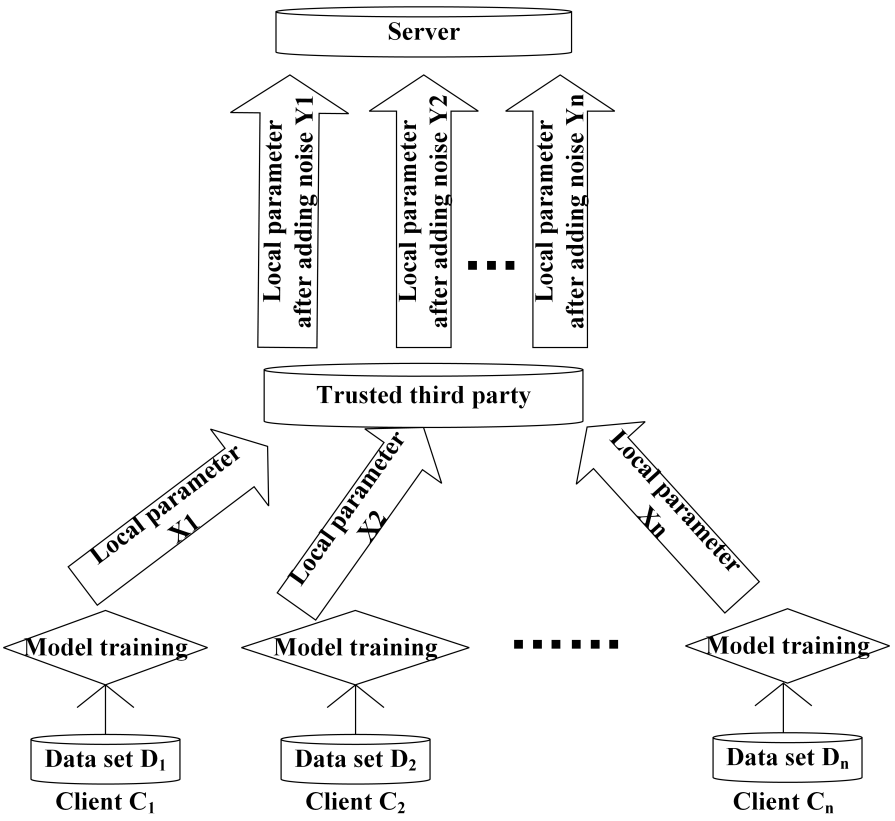

Figure 2. The general distributed machine learning framework based on differential privacy.

In this model, the key step is to design the noise Z_i ($i \in \{1, 2, \ldots, n\}$), which should not only meet the MI-DP condition, but also maximize the utility of local parameters after adding noise. The relationship between X^n, Z^n and Y^n is represented by Figure 3,

$$Y^n = X^n + Z^n. \tag{2}$$

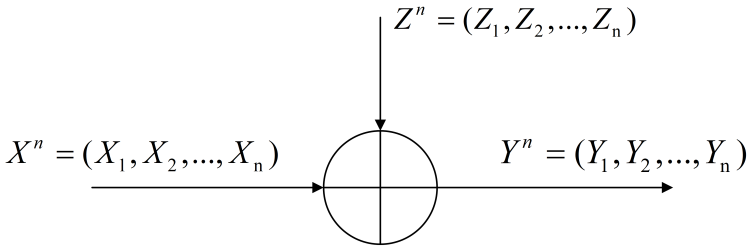

Figure 3. The relationship between X^n, Z^n and Y^n.

Here, note that $X^n = (X_1, \ldots, X_n)$, $Z^n = (Z_1, \ldots, Z_n)$, $Y^n = (Y_1, \ldots, Y_n)$, X^n is the set of local parameters obtained from the local training model of all clients, and Z^n is the set of Gaussian random noise added to the local parameters. Y^n is the set of all local parameters after adding noise.

Through the definition of MI-DP, we know that X_i, Z_i and Y_i ($i \in \{1, 2, \ldots, n\}$) must meet the following condition:

$$\max_i I(X_i; Y^n | X^{-i}) \leq \epsilon, \tag{3}$$

where $X^{-i} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$. In this paper, we regard the left side of (3) as the expression of satisfying privacy.

By the meaning of mutual information, the expression that measures the utility of the noisy local parameters is denoted by,

$$U = \frac{I(X^n; Y^n)}{n}, \tag{4}$$

which is regarded as the expression of utility.

In the next section, we mainly introduce the two important tasks of this paper: The first part focuses on how to design the variance of the noise to make the noisy local parameters have the best utility (the maximum value of (4) while satisfying the Equation (3)). The second part focuses on exploring the relationship between the utility and privacy of the noisy local parameters, in other words, the relationship between U and ϵ.

3. Analysis of the Amount of Noise Added and Exploration of the Relationship between Utility and Privacy

In this section, the local parameter X_i ($i \in \{1, 2, \ldots, n\}$) obtained by the client's local training is a Gaussian random variable, and the added noise Z_i ($i \in \{1, 2, \ldots, n\}$) is also a random Gaussian variable. We used the conditional expression of privacy and the expression of utility so that the most suitable noise variance could be designed after calculation, and the designed noise could not only meet the definition of DP, but also optimize the utility of noisy local parameters.

Therefore, this problem could be mathematized and expressed as: How should the variance of the added noise be designed to make $U = \frac{I(X^n; Y^n)}{n}$ maximum under the conditions $\max_i I(X_i; Y^n | X^{-i}) \leq \epsilon$? The three cases are described below:

Case 1: independent noise added to the independent local parameters.

Case 2: independent noise added to the dependent local parameters.

Case 3: dependent noise added to the dependent local parameters.

After solving this problem, the most suitable noise variances were designed. Next, we studied theoretically the relationship between the utility and privacy of the noisy local parameters in the DML based on the DP framework.

3.1. Case 1: Independent Noise Added to Independent Local Parameters

In Case 1, we assumed that the parameters of each client were independent of each other. Consequently, this case corresponded to the actual application scenario, which could

be used for training data with little or no correlation between them, e.g., word recognition, spam classification, etc.

The clients' local parameters $X_i \sim \mathcal{N}(0, \sigma_i^2)$ were independent of each other, where $\sim$ denoted "distributed as", and σ_i^2 varied with i ($i \in \{1, 2, \ldots, n\}$), that is to say, the distribution of local parameters of each client was different. The noise added to each local parameter was also independent and is distributed as $Z_i \sim \mathcal{N}(0, \sigma^2)$.

Theorem 1. *The optimum Gaussian noise variance σ^2 in Case 1 is given by*

$$\sigma^2 = \frac{\sigma_{xmax}^2}{2^{2\epsilon} - 1},$$

where $\sigma_{xmax}^2 = \max\{\sigma_1^2, \ldots, \sigma_n^2\}$, and σ^2 achieves the maximal utility $U = \frac{I(X^n;Y^n)}{n}$ under a certain secrecy level $\max_i I(X_i; Y^n | X^{-i}) \leq \epsilon$.

Proof of Theorem 1. Since the relationship between Y_i, X_i and Z_i is $Y_i = X_i + Z_i$ ($i \in \{1, 2, \ldots, n\}$), we easily obtain that Y_i are independent of each other and distributed as $Y_i \sim \mathcal{N}(0, \sigma_i^2 + \sigma^2)$. From the definition of (3), we have

$$\begin{aligned}
&I(X_i; Y^n | X^{-i}) \\
&= h(Y^n | X^{-i}) - h(Y^n | X^n) \\
&= h(X_i + Z_i, Z^{-i})) - h(Z^n) \\
&= \frac{1}{2} \log[(2\pi e)^n (\sigma_i^2 + \sigma^2) \sigma^{2(n-1)}] - \frac{1}{2} \log[(2\pi e)^n \sigma^{2n}] \\
&= \frac{1}{2} \log(1 + \frac{\sigma_i^2}{\sigma^2}) \leq \epsilon.
\end{aligned} \tag{5}$$

From (5) we deduce that the variance σ^2 of the designed noise Z_i should satisfy the following inequality

$$\sigma^2 \geq \frac{\sigma_{xmax}^2}{2^{2\epsilon} - 1}, \tag{6}$$

where $\sigma_{xmax}^2 = \max\{\sigma_1^2, \ldots, \sigma_n^2\}$. After clarifying the value range of the noise variance under the privacy condition, the next step is to select the most suitable variance value in the value range to get the best utility (make $U = \frac{I(X^n;Y^n)}{n}$ maximal). From (4), we have

$$\begin{aligned}
U &= \frac{1}{n} I(X^n; Y^n) \\
&= \frac{1}{n} [h(Y^n) - h(Y^n | X^n)] \\
&= \frac{1}{n} [h(Y^n) - h(Z^n)] \\
&= \frac{1}{2n} \log[(2\pi e)^n (\sigma_1^2 + \sigma^2) * \ldots * (\sigma_n^2 + \sigma^2)] \\
&- \frac{1}{2n} \log[(2\pi e)^n \sigma^{2n}] \\
&= \frac{1}{2n} \sum_{i=1}^{n} \log(1 + \frac{{\sigma_i}^2}{\sigma^2}).
\end{aligned} \tag{7}$$

From (8), it can be clearly observed that $U = \frac{I(X^n;Y^n)}{n}$ is a monotonically decreasing function of σ^2. Therefore, when σ^2 takes the minimum value

$$\sigma^2 = \frac{\sigma_{xmax}^2}{2^{2\epsilon} - 1}, \tag{8}$$

the expression of utility U takes the maximum value

$$U_{max} = \frac{1}{2n}\sum_{i=1}^{n}\log\left[1+\frac{\sigma_i{}^2(2^{2\epsilon}-1)}{\sigma_{xmax}^2}\right]. \tag{9}$$

□

In summary, the problem of how to design the size of the noise variance in Case 1 was solved and the first part of the work of Case 1 completed.

After obtaining the optimum noise variance, the next step was to study the relationship between utility and privacy in Case 1 (that is, the relationship between U_{max} and ϵ), where (9) is the functional relationship between U_{max} and ϵ.

Figure 4 plots the relationship between U_{max} and ϵ based on (9). In this figure, we assumed that $n = 101$, σ_i^2 ($i \in \{1, 2, \ldots, 101\}$) to be a random value between 0 and 1. It can be seen from the figure that when the local parameters of clients are independent of each other, and the noise is also designed to be independent, the amounts of the privacy budget and utility are proportional. The larger the privacy budget value (the greater the risk of privacy leakage), the bigger the value of utility.

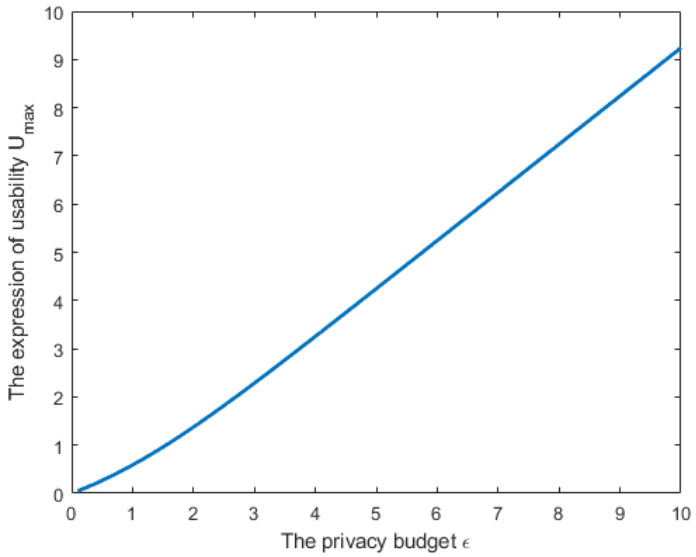

Figure 4. The relationship between utility and privacy in Case 1.

3.2. Case 2: Independent Noise Added to Dependent Local Parameters

Case 2 was different from Case 1, as we assumed that the clients' local parameters were dependent of each other. This case is used for practical application scenarios of correlation between training data. For example, machine learning can be applied to explore the impact of age on health in humans. It is well known that there is a certain correlation between age and human health indicators.

The clients' local parameters $X_i \sim \mathcal{N}(0, \sigma_m^2)$ were dependent of each other. When $i \neq j$, $E[X_i X_j] = \sigma_k^2$ ($j \in \{1, 2, \ldots, n\}$). The distributions of local parameters of each client were the same. The noise added to each local parameter $Z_i \sim \mathcal{N}(0, \sigma^2)$ ($i \in \{1, 2, \ldots, n\}$) was independent.

Theorem 2. *The optimum Gaussian noise variance in Case 2 is given by*

$$\sigma^2 = \frac{\sigma_m^2}{2^{2\epsilon}-1} + \frac{(n-1)\sigma_k^4}{(2^{2\epsilon}-1)[\sigma_m^2+(n-2)\sigma_k^2]},$$

which achieves the maximal utility $U = \frac{I(X^n;Y^n)}{n}$ *under a certain secrecy level* $\max_i I(X_i; Y^n|X^{-i}) \leq \epsilon$.

Proof of Theorem 2. Since $Y_i = X_i + Z_i$ ($i \in \{1, 2, \ldots, n\}$) and X_i are dependent of each other, $Y_i \sim \mathcal{N}(0, \sigma_m^2 + \sigma^2)$ are dependent of each other. We assume that when $i \neq j$, $E[Y_i Y_j] = \sigma_k^2$ ($j \in \{1, 2, \ldots, n\}$). We put the value of each variable into the expression of satisfying privacy, and it is calculated as

$$\begin{aligned}
&I(X_i; Y^n | X^{-i}) \\
&= h(Y^n | X^{-i}) - h(Y^n | X^n) \\
&= h(X_i + Z_i, Z^{-i} | X^{-i}) - h(Z^n) \\
&= h(X_i + Z_i | X^{-i}) + h(Z^{-i}) - h(Z^n) \\
&= h(X_i + Z_i, X^{-i}) - h(X^{-i}) + h(Z_i) \\
&= \frac{1}{2} \log[(2\pi e)^n |COV(X_i + Z_i, X^{-i})|] - \frac{1}{2} \log[(2\pi e)^{n-1} |COV(X^{-i})|] - \frac{1}{2} \log(2\pi e \sigma^2) \\
&= \frac{1}{2} \log(\frac{|COV(X_i + Z_i, X^{-i})|}{|COV(X^{-i})| \sigma^2}) \leq \epsilon.
\end{aligned} \tag{10}$$

for $|COV(X_i + Z_i, X^{-i})|$ and $|COV(X^{-i})|$, we have

$$\begin{aligned}
&|COV(X_i + Z_i, X^{-i})| \\
&= \begin{vmatrix} E(X_i + Z_i)^2 & EX_i X_1 & \cdots & EX_i X_n \\ EX_1 X_i & EX_1^2 & \cdots & EX_1 X_n \\ \vdots & \vdots & \ddots & \vdots \\ EX_n X_i & EX_n X_1 & \cdots & EX_n^2 \end{vmatrix}_{n \times n} \\
&= \begin{vmatrix} \sigma_m^2 + \sigma^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\ \sigma_k^2 & \sigma_m^2 & \cdots & \sigma_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_m^2 \end{vmatrix}_{n \times n} \\
&= (\sigma_m^2 - \sigma_k^2)^{n-1} (\sigma_m^2 + \sigma^2) - (n-1)(\sigma_k^2 - \sigma_m^2 - \sigma^2) \sigma_k^2 (\sigma_m^2 - \sigma_k^2)^{n-2},
\end{aligned} \tag{11}$$

$$\begin{aligned}
&|COV(X^{-i})| \\
&= \begin{vmatrix} EX_1^2 & EX_1 X_2 & \cdots & EX_1 X_n \\ EX_2 X_1 & EX_2^2 & \cdots & EX_2 X_n \\ \vdots & \vdots & \ddots & \vdots \\ EX_n X_1 & EX_n X_2 & \cdots & EX_n^2 \end{vmatrix}_{(n-1) \times (n-1)} \\
&= \begin{vmatrix} \sigma_m^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\ \sigma_k^2 & \sigma_m^2 & \cdots & \sigma_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_m^2 \end{vmatrix}_{(n-1) \times (n-1)} \\
&= [\sigma_m^2 + (n-2)\sigma_k^2](\sigma_m^2 - \sigma_k^2)^{n-2}.
\end{aligned} \tag{12}$$

According to (11), (12) and (10) can be expressed as

$$\begin{aligned}
&I(X_i; Y^n | X^{-i}) \\
&= \frac{1}{2} \log(\frac{(\sigma_m^2 - \sigma_k^2)^{n-1} (\sigma_m^2 + \sigma^2)}{[\sigma_m^2 + (n-2)\sigma_k^2](\sigma_m^2 - \sigma_k^2)^{n-2} \sigma^2} - \frac{(n-1)(\sigma_k^2 - \sigma_m^2 - \sigma^2) \sigma_k^2 (\sigma_m^2 - \sigma_k^2)^{n-2}}{[\sigma_m^2 + (n-2)\sigma_k^2](\sigma_m^2 - \sigma_k^2)^{n-2} \sigma^2}) \\
&\leq \epsilon.
\end{aligned} \tag{13}$$

From (13), we deduce that the variance σ^2 of the designed noise Z_i should satisfy the following inequality

$$\sigma^2 \geq \frac{\sigma_m^2}{2^{2\epsilon}-1} + \frac{(n-1)\sigma_k^4}{(2^{2\epsilon}-1)[\sigma_m^2+(n-2)\sigma_k^2]}. \tag{14}$$

After clarifying the value range of the noise variance under the privacy condition, the next step is to select the most suitable variance value in the value range to get the best utility (make $U = \frac{I(X^n;Y^n)}{n}$ maximal).

$$\begin{aligned} U &= \frac{1}{n} I(X^n;Y^n) \\ &= \frac{1}{n}[h(Y^n) - h(Y^n|X^n)] \\ &= \frac{1}{n}[h(Y^n) - h(Z^n)] \\ &= \frac{1}{2n}\log[(2\pi e)^n |COV(Y_1,\ldots,Y_n)|] - \frac{1}{2n}\log[(2\pi e)^n \sigma^{2n}] \\ &= \frac{1}{2n}\log(\frac{|COV(Y_1,\ldots,Y_n)|}{\sigma^{2n}}). \end{aligned} \tag{15}$$

For $|COV(Y_1,\ldots,Y_n)|$, we have

$$\begin{aligned} &|COV(Y_1,\ldots,Y_n)| \\ &= \begin{vmatrix} E(Y_1)^2 & EY_1Y_2 & \cdots & EY_1Y_n \\ EY_2Y_n & EY_2^2 & \cdots & EY_2Y_n \\ \vdots & \vdots & \ddots & \vdots \\ EY_nY_1 & EY_nY_2 & \cdots & EY_n^2 \end{vmatrix}_{n\times n} \\ &= \begin{vmatrix} \sigma_m^2+\sigma^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\ \sigma_k^2 & \sigma_m^2+\sigma^2 & \cdots & \sigma_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_m^2+\sigma^2 \end{vmatrix}_{n\times n} \\ &= [\sigma_m^2+\sigma^2+(n-1)\sigma_k^2](\sigma_m^2+\sigma^2-\sigma_k^2)^{n-1}, \end{aligned} \tag{16}$$

From (16) and (15), this can be expressed as

$$\begin{aligned} U &= \frac{1}{n} I(X^n;Y^n) \\ &= \frac{1}{2n}\log(\frac{\sigma_m^2+\sigma^2+(n-1)\sigma_k^2}{\sigma^{2n}}) + \frac{1}{2n}\log(\frac{(\sigma_m^2+\sigma^2-\sigma_k^2)^{n-1}}{\sigma^{2n}}). \end{aligned} \tag{17}$$

From (17), it can be clearly observed that $U = \frac{I(X^n;Y^n)}{n}$ is a monotonically decreasing function of σ^2. Therefore, when σ^2 takes its minimum value ($\sigma^2 = \frac{\sigma_m^2}{2^{2\epsilon}-1} + \frac{(n-1)\sigma_k^4}{(2^{2\epsilon}-1)[\sigma_m^2+(n-2)\sigma_k^2]}$ is denoted as σ_{min}^2), the expression of utility U takes its maximum value

$$U_{max} = \frac{1}{2n}\log[\sigma_m^2+\sigma_{min}^2+(n-1)\sigma_k^2] + \frac{1}{2n}\log(\frac{(\sigma_m^2+\sigma_{min}^2-\sigma_k^2)^{n-1}}{\sigma_{min}^{2n}}). \tag{18}$$

□

In summary, the problem of how to design the size of the noise variance in Case 2 was solved and the first part of the work of Case 2 completed.

Next, we carried out the second part of the work: After obtaining the optimal noise variance, the next step was to study the relationship between utility and privacy in Case 2

(the relationship between U_{max} and ϵ), where (18) is the functional relationship between U_{max} and ϵ.

Figure 5 plots the relationship between U_{max} and ϵ based on (18). In this figure, we assumed that $\sigma_m^2 = 2$, $\sigma_k^2 = 1.8$, and $n = 101$. It can be seen from the figure that when the local parameters of clients are dependent of each other, and the added noise is designed to be independent, the amounts of the privacy budget and utility are proportional. We conclude that the larger the privacy budget value (the greater the risk of privacy leakage), the bigger the value of utility.

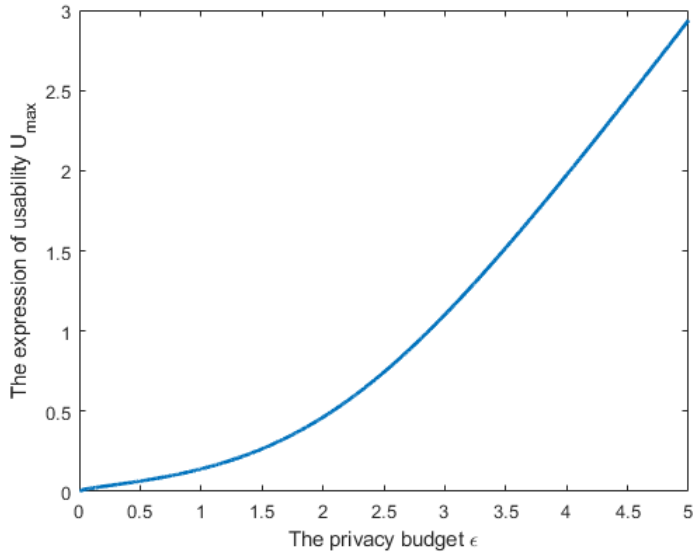

Figure 5. The relationship between utility and privacy in Case 2.

3.3. Case 3: Dependent Noise Added to Local Parameters

Case 3 was different from Case 2, as we assumed that the noise added to each local parameter was dependent. In order to check whether dependent noise performed better than independent noise, we studied Case 3. The application scenario of Case 3 is still that the parameters are correlated, for example, the study of the correlation between human lifespan and gender.

In Case 3, the noise added to each local parameter $Z_i \sim \mathcal{N}(0, \sigma^2)$ $(i \in \{1, 2, \ldots, n\})$ was dependent. When $i \neq j$, $E[Z_i Z_j] = \sigma_e^2$, $j \in \{1, 2, \ldots, n\}$. The clients' local parameters $X_i \sim \mathcal{N}(0, \sigma_m^2)$ were also dependent of each other. When $i \neq j$, $E[X_i X_j] = \sigma_k^2$, $j \in \{1, 2, \ldots, n\}$. The distributions of local parameters of each client were the same.

For the problem to be solved, we made the following analysis. Case 3 was different from Cases 1 and 2. There were two noise parameters in Case 3. We had to design the most suitable noise to make $U = \frac{I(X^n;Y^n)}{n}$ maximum under the conditions $\max_i I(X_i; Y^n | X^{-i}) \leq \epsilon$. Thus, the computational difficulty was also much higher than in Cases 1 and 2. As we know, $Y_i = X_i + Z_i$, $i \in \{1, 2, \ldots, n\}$, so $Y_i \sim \mathcal{N}(0, \sigma_m^2 + \sigma^2)$ are dependent of each other. We assumed that when $i \neq j$, $E[Y_i Y_j] = \sigma_k^2 + \sigma_e^2$, $E[X_i Y_j] = \sigma_k^2$, $j \in \{1, 2, \ldots, n\}$. We put the value of each variable into the expression of satisfying privacy, which could be calculated as

$$
\begin{aligned}
& I(X_i; Y^n | X^{-i}) \\
&= h(Y^n | X^{-i}) - h(Y^n | X^n) \\
&= h(Y^n, X^{-i}) - h(X^{-i}) - h(Z^n) \\
&= \frac{1}{2} \log[(2\pi e)^{2n-1} |COV(Y^n, X^{-i})|] - \frac{1}{2} \log[(2\pi e)^{n-1} |COV(X^{-i})|] - \frac{1}{2} \log(2\pi e)^n |COV(Z^n)|) \\
&= \frac{1}{2} \log(\frac{|COV(Y^n, X^{-i})|}{|COV(X^{-i})||COV(Z^n)|}) \leq \epsilon.
\end{aligned} \tag{19}
$$

Lemma 1. *For $|COV(Y^n, X^{-i})|$, even if $i \in \{1, 2, \ldots, n\}$ takes different values, the result of determinant $|COV(Y^n, X^{-i})|$ is the same.*

Proof. When $i = 1$,

$$
\begin{aligned}
&|COV(Y^n, X^{-i})| = |COV(Y^n, X^{-1})| \\
&= \left|\begin{array}{cccc|cccc}
EY_1^2 & EY_1Y_2 & \cdots & EY_1Y_n & EY_1X_2 & EY_1X_3 & \cdots & EY_1X_n \\
EY_2Y_1 & EY_2^2 & \cdots & \cdots & EY_2X_2 & EY_2X_3 & \cdots & \cdots \\
\cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \ddots & \cdots \\
EY_nY_1 & \cdots & \cdots & EY_n^2 & EY_nX_2 & \cdots & \cdots & EY_nX_n \\
\hline
EX_2Y_1 & EX_2Y_2 & \cdots & EX_2Y_n & EX_2^2 & EX_2X_3 & \cdots & EX_2X_n \\
EX_3Y_1 & EX_3Y_2 & \cdots & \cdots & EX_3X_2 & EX_3^2 & \cdots & \cdots \\
\cdots & \ddots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\
EX_nY_1 & \cdots & \cdots & EX_nY_n & EX_nX_2 & \cdots & \cdots & EX_n^2
\end{array}\right|_{(2n-1)\times(2n-1)} \\
&= \left|\begin{array}{cccc|cccc}
\sigma_m^2+\sigma^2 & \sigma_k^2+\sigma_e^2 & \cdots & \sigma_k^2+\sigma_e^2 & \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2+\sigma_e^2 & \sigma_m^2+\sigma^2 & \cdots & \cdots & \sigma_m^2 & \sigma_k^2 & \cdots & \cdots \\
\cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \ddots & \cdots \\
\sigma_k^2+\sigma_e^2 & \cdots & \cdots & \sigma_m^2+\sigma^2 & \sigma_k^2 & \cdots & \cdots & \sigma_m^2 \\
\hline
\sigma_k^2 & \sigma_m^2 & \cdots & \sigma_k^2 & \sigma_m^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2 & \sigma_k^2 & \cdots & \cdots & \sigma_k^2 & \sigma_m^2 & \cdots & \cdots \\
\cdots & \ddots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\
\sigma_k^2 & \cdots & \cdots & \sigma_m^2 & \sigma_k^2 & \cdots & \cdots & \sigma_m^2
\end{array}\right|_{(2n-1)\times(2n-1)}
\end{aligned}
\tag{20}
$$

When $i = 2$,

$$
\begin{aligned}
&|COV(Y^n, X^{-i})| = |COV(Y^n, X^{-2})| \\
&= \left|\begin{array}{cccc|cccc}
EY_1^2 & EY_1Y_2 & \cdots & EY_1Y_n & EY_1X_1 & EY_1X_3 & \cdots & EY_1X_n \\
EY_2Y_1 & EY_2^2 & \cdots & \cdots & EY_2X_1 & EY_2X_3 & \cdots & \cdots \\
\cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \ddots & \cdots \\
EY_nY_1 & \cdots & \cdots & EY_n^2 & EY_nX_1 & \cdots & \cdots & EY_nX_n \\
\hline
EX_1Y_1 & EX_1Y_2 & \cdots & EX_1Y_n & EX_1^2 & EX_1X_3 & \cdots & EX_1X_n \\
EX_3Y_1 & EX_3Y_2 & \cdots & \cdots & EX_3X_1 & EX_3^2 & \cdots & \cdots \\
\cdots & \ddots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\
EX_nY_1 & \cdots & \cdots & EX_nY_n & EX_nX_1 & \cdots & \cdots & EX_n^2
\end{array}\right|_{(2n-1)\times(2n-1)} \\
&= \left|\begin{array}{cccc|cccc}
\sigma_m^2+\sigma^2 & \sigma_k^2+\sigma_e^2 & \cdots & \sigma_k^2+\sigma_e^2 & \sigma_m^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2+\sigma_e^2 & \sigma_m^2+\sigma^2 & \cdots & \cdots & \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\cdots & \cdots & \ddots & \cdots & \sigma_k^2 & \sigma_m^2 & \cdots & \sigma_k^2 \\
\cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \ddots & \cdots \\
\sigma_k^2+\sigma_e^2 & \cdots & \cdots & \sigma_m^2+\sigma^2 & \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_m^2 \\
\hline
\sigma_m^2 & \sigma_k^2 & \sigma_k^2 \cdots & \sigma_k^2 & \sigma_m^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2 & \sigma_k^2 & \sigma_m^2 \cdots & \sigma_k^2 & \sigma_k^2 & \sigma_m^2 & \cdots & \\
\sigma_k^2 & \sigma_k^2 & \cdots \ddots & \cdots & \cdots & \ddots & \cdots & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \\
\sigma_k^2 & \sigma_k^2 & \cdots & \sigma_m^2 & \sigma_k^2 & \cdots & \cdots & \sigma_m^2
\end{array}\right|_{(2n-1)\times(2n-1)} \\
&= (-1)^2 \left|\begin{array}{cccc|cccc}
\sigma_m^2+\sigma^2 & \sigma_k^2+\sigma_e^2 & \cdots & \sigma_k^2+\sigma_e^2 & \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2+\sigma_e^2 & \sigma_m^2+\sigma^2 & \cdots & \cdots & \sigma_m^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\cdots & \cdots & \ddots & \cdots & \sigma_k^2 & \sigma_m^2 & \cdots & \sigma_k^2 \\
\cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \ddots & \cdots \\
\sigma_k^2+\sigma_e^2 & \cdots & \cdots & \sigma_m^2+\sigma^2 & \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_m^2 \\
\hline
\sigma_k^2 & \sigma_m^2 & \sigma_k^2 \cdots & \sigma_k^2 & \sigma_m^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2 & \sigma_k^2 & \sigma_m^2 \cdots & \sigma_k^2 & \sigma_k^2 & \sigma_m^2 & \cdots & \\
\sigma_k^2 & \sigma_k^2 & \cdots \ddots & \cdots & \cdots & \ddots & \cdots & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \\
\sigma_k^2 & \sigma_k^2 & \cdots & \sigma_m^2 & \sigma_k^2 & \cdots & \cdots & \sigma_m^2
\end{array}\right| \\
&= |COV(Y^n, X^{-1})|. \qquad (21)
\end{aligned}
$$

Consequently, $|COV(Y^n, X^{-1})| = |COV(Y^n, X^{-2})|$. Similarly, we can prove that $|COV(Y^n, X^{-2})| = |COV(Y^n, X^{-3})| = \ldots = |COV(Y^n, X^{-n})|$. That is to say, no matter what value $i \in \{1, 2, \ldots, n\}$ takes, the result of the determinant $|COV(Y^n, X^{-i})|$ is the same. So the proof of Lemma 1 is completed. □

Therefore, we could calculate $|COV(Y^n, X^{-i})|$ as $|COV(Y^n, X^{-1})|$,

$$
\begin{aligned}
&|COV(Y^n, X^{-i})| = |COV(Y^n, X^{-1})| = \\
&= \left|\begin{array}{cccc|cccc}
\sigma_m^2+\sigma^2 & \sigma_k^2+\sigma_e^2 & \cdots & \sigma_k^2+\sigma_e^2 & \sigma_k^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2+\sigma_e^2 & \sigma_m^2+\sigma^2 & \cdots & \cdots & \sigma_m^2 & \sigma_k^2 & \cdots & \cdots \\
\cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \ddots & \cdots \\
\sigma_k^2+\sigma_e^2 & \cdots & \cdots & \sigma_m^2+\sigma^2 & \sigma_k^2 & \cdots & \cdots & \sigma_m^2 \\
\hline
\sigma_k^2 & \sigma_m^2 & \cdots & \sigma_k^2 & \sigma_m^2 & \sigma_k^2 & \cdots & \sigma_k^2 \\
\sigma_k^2 & \sigma_k^2 & \cdots & \cdots & \sigma_k^2 & \sigma_m^2 & \cdots & \cdots \\
\cdots & \ddots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\
\sigma_k^2 & \cdots & \cdots & \sigma_m^2 & \sigma_k^2 & \cdots & \cdots & \sigma_m^2
\end{array}\right|_{(2n-1)\times(2n-1)} \\
&= (\sigma_m^2+\sigma^2)(\sigma^2-\sigma_e^2)^{n-1}(\sigma_m^2-\sigma_k^2)^{n-1} + (-1)^{n+1}(\sigma_e^2-\sigma_m^2-\sigma^2)(n-1)\sigma_e^2(\sigma_e^2-\sigma^2)^{n-2}(\sigma_m^2-\sigma_k^2)^{n-1} \\
&+\sigma_k^2[(n-1)\sigma_e^2 + \frac{(\sigma_e^2-\sigma^2)\sigma_k^2}{-\sigma_k^2-\sigma_e^2+\sigma_m^2+\sigma^2}](\sigma_e^2-\sigma^2)^{n-2}[(n-1)(-\sigma_k^2-\sigma_e^2+\sigma_m^2+\sigma^2)](\sigma_k^2-\sigma_m^2)^{n-2}. \qquad (22)
\end{aligned}
$$

For $|COV(Z^n)|$, we had

$$
\begin{aligned}
&|COV(Z^n)| \\
&= \left|\begin{array}{cccc}
EZ_1^2 & EZ_1Z_2 & \cdots & EZ_1Z_n \\
EZ_2Z_1 & EZ_2^2 & \cdots & EZ_2Z_n \\
\vdots & \vdots & \cdots & \vdots \\
EZ_nZ_1 & EZ_nZ_2 & \cdots & EZ_n^2
\end{array}\right|_{n\times n} \\
&= \left|\begin{array}{cccc}
\sigma^2 & \sigma_e^2 & \cdots & \sigma_e^2 \\
\sigma_e^2 & \sigma^2 & \cdots & \sigma_e^2 \\
\vdots & \vdots & \cdots & \vdots \\
\sigma_e^2 & \sigma_e^2 & \cdots & \sigma^2
\end{array}\right|_{n\times n} \\
&= [\sigma^2+(n-1)\sigma_e^2](\sigma^2-\sigma_e^2)^{n-1}. \qquad (23)
\end{aligned}
$$

From (12), we calculated $|COV(X^{-i})| = [\sigma_m^2+(n-2)\sigma_k^2](\sigma_m^2-\sigma_k^2)^{n-2}$. Thus, (19) could be expressed as

$$
\begin{aligned}
&I(X_i; Y^n|X^{-i}) \\
&= \frac{1}{2}\log\left(\frac{A+B+C+D}{[\sigma_m^2+(n-2)\sigma_k^2](\sigma^2+(n-1)\sigma_e^2)(\sigma^2-\sigma_e^2)}\right) \\
&\le \epsilon, \qquad (24)
\end{aligned}
$$

where $A = (\sigma_m^2+\sigma^2)(\sigma^2-\sigma_e^2)(\sigma_m^2-\sigma_k^2)$, $B = (-1)^{n+1}(n-1)\sigma_e^2(\sigma_e^2-\sigma_m^2-\sigma^2)(\sigma_m^2-\sigma_k^2)$, $C = (n-1)\sigma_k^4(\sigma_e^2-\sigma^2)$, and $D = (n-1)^2\sigma_e^2\sigma_k^2(\sigma_m^2+\sigma^2-\sigma_k^2-\sigma_e^2)$

As shown, Equation (24) is very complicated, and it was difficult for us to directly derive the range of values of σ^2 and σ_e^2, the two parameters of the noise. Instead, we calculated the utility expression $U = \frac{I(X^n;Y^n)}{n}$, and then we used the nonlinear constraint optimization function to obtain the optimal values of σ^2 and σ_e^2 which could maximize the U.

Equation (4) could be further computed as

$$\begin{aligned} U &= \frac{1}{n} I(X^n; Y^n) \\ &= \frac{1}{n}[h(Y^n) - h(Y^n|X^n)] \\ &= \frac{1}{n}[h(Y^n) - h(Z^n)] \\ &= \frac{1}{2n}\log[(2\pi e)^n |COV(Y_1, \ldots, Y_n)|] \\ &- \frac{1}{2n}\log[(2\pi e)^n |COV(Z_1, \ldots, Z_n)|] \\ &= \frac{1}{2n}\log(\frac{|COV(Y_1, \ldots, Y_n)|}{|COV(Z_1, \ldots, Z_n)|}). \end{aligned} \tag{25}$$

For $|COV(Y_1, \ldots, Y_n)|$, we had

$$\begin{aligned} &|COV(Y_1, \ldots, Y_n)| \\ &= \begin{vmatrix} EY_1^2 & EY_1Y_2 & \cdots & EY_1Y_n \\ EY_2Y_n & EY_2^2 & \cdots & EY_2Y_n \\ \vdots & \vdots & \cdots & \vdots \\ EY_nY_1 & EY_nY_2 & \cdots & EY_n^2 \end{vmatrix}_{n\times n} \\ &= \begin{vmatrix} \sigma_m^2 + \sigma^2 & \sigma_k^2 + \sigma_e^2 & \cdots & \sigma_k^2 + \sigma_e^2 \\ \sigma_k^2 + \sigma_e^2 & \sigma_m^2 + \sigma^2 & \cdots & \sigma_k^2 + \sigma_e^2 \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_k^2 + \sigma_e^2 & \sigma_k^2 + \sigma_e^2 & \cdots & \sigma_m^2 + \sigma^2 \end{vmatrix}_{n\times n} \\ &= [\sigma_m^2 + \sigma^2 + (n-1)(\sigma_k^2 + \sigma_e^2)](\sigma_m^2 + \sigma^2 - \sigma_k^2 - \sigma_e^2)^{n-1}. \end{aligned} \tag{26}$$

We calculated $|COV(Z^n)| = [\sigma^2 + (n-1)\sigma_e^2](\sigma^2 - \sigma_e^2)^{n-1}$. Thus, (25) could be expressed as

$$\begin{aligned} U &= \frac{1}{n} I(X^n; Y^n) \\ &= \frac{1}{2n}\log[\sigma_m^2 + \sigma^2 + (n-1)(\sigma_k^2 + \sigma_e^2)] + \frac{1}{2n}\log\frac{(\sigma_m^2 + \sigma^2 - \sigma_k^2 - \sigma_e^2)^{n-1}}{[\sigma^2 + (n-1)\sigma_e^2](\sigma^2 - \sigma_e^2)^{n-1}}. \end{aligned} \tag{27}$$

The next step was to combine (24) with (27). The aim was to find the most suitable values of σ^2 and σ_e^2 so that (27) could obtain the maximum value under the condition of (24). In order to solve this problem, we used MATLAB's nonlinear optimization function for the simulation. We varied the privacy budget ϵ values and searched for σ^2 and σ_e^2 that would maximize utility.

Figure 6 shows the simulation results when the privacy budget ϵ takes different values ($\epsilon = \{1, 3, 4, 5, 8, 10\}$). In this figure, we assumed that $\sigma_m^2 = 30$, $\sigma_k^2 = 2$, and $n = 101$. We found that when the privacy budget ϵ increased within a certain range, the value for measuring utility also increased. However, there was an upper bound, that is, when the privacy budget ϵ was out of range, the value of U_{max} remained unchanged. Figure 7 plots the relationship between U_{max} and ϵ. In Figure 7, we obtained the value of the expression of utility when the privacy budget $\epsilon = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. In this figure, we assumed that $\sigma_m^2 = 30$, $\sigma_k^2 = 2$, and $n = 101$. The ten points ($\epsilon = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$) were connected into a line, and finally we obtained the relationship trend between U_{max} and ϵ.

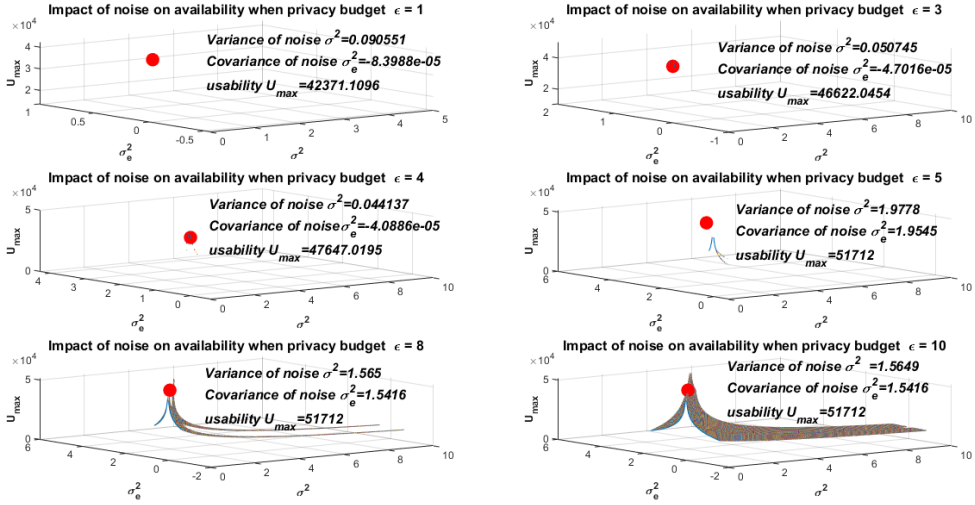

Figure 6. The impact of noise level on utility with different privacy budget ϵ.

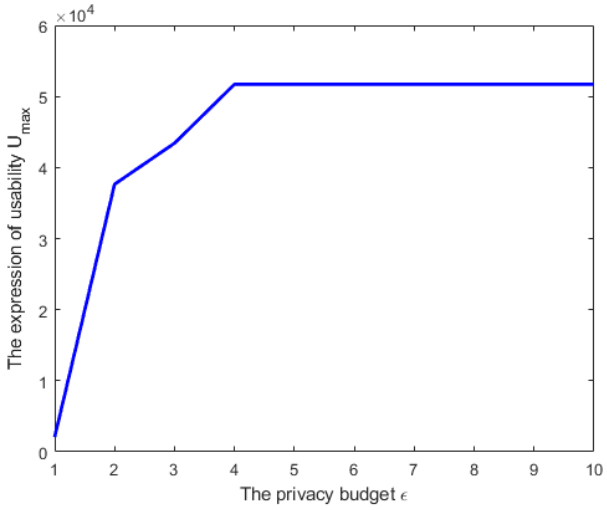

Figure 7. The relationship between utility and privacy in Case 3.

4. Conclusions

In this paper, we used the DP mechanism to protect the clients' local parameters. From an information-theoretic point of view, we studied the utility–privacy trade-off in DML with the help of the DP mechanism. Specifically, three cases including independent clients' local parameters with independent DP noise and dependent clients' local parameters with independent/dependent DP noise were considered. Mutual information and conditional mutual information were used to characterize utility and privacy, respectively. First, we showed the relationship between utility and privacy for the three cases. Then, we show the optimal noise variance that achieved the maximal utility under a certain level of privacy. Finally, the results of this paper were further illustrated by numerical results.

The limitations of this paper are that the local parameters and the added noise of the client were only assumed to be Gaussian distributed, and multiround model training was not considered, which we will in our future work.

Author Contributions: X.Z. did the theoretical work, performed the experiments, analyzed the data and drafting the work; C.Y. performed the conceptualization, theoretical work, experiments, analyzed the data. B.D. reviewed, revised, validated, and supervised, and administered the work. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported in part by the National Key R&D Program of China under Grant 2020YFB1806405; in part by the National Natural Science Foundation of China under Grants 62071392, U21A20454; in part by the Natural Science Foundation of Sichuan under Grant 2022NSFSC0484; in part by the central government to guide local scientific and technological development under Grant No. 2021ZYD0001; in part by the 111 Project No.111-2-14; in part by the NSFC-STINT under grant 62011530134, and in part by the Major Key Project of PCL (PCL2021A04).

Data Availability Statement: The data used in this work are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xing, E.P.; Ho, Q.; Xie, P.; Wei, D. Strategies and Principles of Distributed Machine Learning on Big Data. *Engineering* **2016**, *2*, 179–195.
2. He, Q.; Li, N.; Luo, W.J.; Shi, Z.Z. A survey of machine learning algorithms for big data. *Pattern Recognit. Artif. Intell.* **2014**, *27*, 327–336.
3. Alqahtani, S.; Demirbas, M. A Comparison of Distributed Machine Learning Platforms. In Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017.
4. Liu, B.; Jinrong, H.E.; Geng, Y.; Wang, Z. Recent advances in infrastructure architecture of parallel machine learning algorithms. *Comput. Eng. Appl.* **2017**, *16*, 123–143.
5. Wang, Z.; Liao, J.; Cao, Q.; Qi, H.; Wang, Z. Friendbook: A Semantic-Based Friend Recommendation System for Social Networks. *IEEE Trans. Mob. Comput.* **2016**, *14*, 538–551.
6. Phan, N.H.; Wu, X.; Hu, H.; Dou, D. Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning. In Proceedings of the International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017.
7. Wang, X.; Ishii, H.; Du, L.; Cheng, P.; Chen, J. Privacy-Preserving Distributed Machine Learning via Local Randomization and ADMM Perturbation. *IEEE Trans. Signal Process.* **2020**, *68*, 4226–4241.
8. So, J.; Guler, B.; Avestimehr, A.S. CodedPrivateML: A Fast and Privacy-Preserving Framework for Distributed Machine Learning. *IEEE J. Sel. Areas Inf. Theory* **2021**, *2*, 441–451.
9. Zheng, H.; Hu, H.; Han, Z. Preserving User Privacy for Machine Learning: Local Differential Privacy or Federated Machine Learning? *IEEE Intell. Syst.* **2020**, *35*, 5–14.
10. Cuff, P.; Yu, L. Differential Privacy as a Mutual Information Constraint. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016.
11. Dwork, C. Differential Privacy. In Proceedings of the 33rd international conference on Automata, Languages and Programming—Volume Part II, Venice, Italy, 10–14 July 2006 .
12. Dwork, C. Differential Privacy: A Survey of Results. In Proceedings of the International Conference on Theory and Applications of Models of Computation, Changsha, China, 18–20 October 2008.
13. Dwork, C. The Differential Privacy Frontier. In Proceedings of the Theory of Cryptography Conference, San Francisco, CA, USA, 15–17 March 2009.
14. Dwork, C. A firm foundation for private data analysis. *Commun. ACM* **2011**, *54*, 86–95.
15. Dwork, C. The Promise of Differential Privacy: A Tutorial on Algorithmic Techniques. In Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Washington, DC, USA, 22–25 October 2011.
16. Dwork, C.; Jing, L. Differential privacy and robust statistics. In Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC, Bethesda, MD, USA, 31 May–2 June 2009.
17. Balle, B.; Wang, Y.X. Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
18. Mcsherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007.
19. Alvim, M.S.; Andres, M.; Chatzikokolakis, K.; Degano, P.; Palamidessi, C. Differential Privacy: on the trade-off between Utility and Information Leakage. In Proceedings of the International Workshop on Formal Aspects in Security and Trust, Leuven, Belgium, 12–14 September 2011.
20. Calmon, F.; Fawaz, N. Privacy Against Statistical Inference. In Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 1–5 October 2012.
21. De, A. Lower bounds in differential privacy. In Proceedings of the 9th international conference on Theory of Cryptography, Nerja, Spain, 7–10 June 2011.
22. Wang, W.; Lei, Y.; Zhang, J. On the relation between identifiability, differential privacy, and mutual-information privacy. *EEE Trans. Inf. Theory* **2015**, *62*, 5018–5029.
23. Dwork, C.; Mcsherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In Proceedings of the Third Conference on Theory of Cryptography, New York, NY, USA, 4–7 March 2006.

24. Mironov, I. Renyi Differential Privacy. In Proceedings of the 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–25 August 2017.
25. Mironov, I.; Talwar, K.; Li, Z. Renyi Differential Privacy of the Sampled Gaussian Mechanism. *arXiv* **2019**, arXiv:1908.10530.
26. Wang, Y.X.; Balle, B.; Kasiviswanathan, S. Subsampled Renyi Differential Privacy and Analytical Moments Accountant. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Naha, Japan, 16–18 April 2019.
27. Dwork, C.; Kenthapadi, K.; Mcsherry, F.; Mironov, I.; Naor, M. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In Proceedings of the International Conference on Advances in Cryptology-Eurocrypt, St. Petersburg, Russia, 28 May–1 June 2006.
28. Melis, L.; Song, C.; Cristofaro, E.D.; Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In Proceedings of the Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019 .
29. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017.