# A proposal to increase data utility on Global Differential Privacy data based on data use predictions

**Henry C. Nunes**[1], **Marlon P. da Silva**[1], **Charles V. Neu**[2], **Avelino F. Zorzo**[1]

[1]Polytechnic School PUCRS, Porto Alegre, Brazil

[2]School of Computing Newcastle University, Newcastle upon Tyne, UK

{henry.nunes, marlon.pereira}@edu.pucrs.br

charles.neu@newcastle.ac.uk, avelino.zorzo@pucrs.br

***Abstract.*** *This paper presents ongoing research focused on improving the utility of data protected by Global Differential Privacy(DP) in the scenario of summary statistics. Our approach is based on predictions on how an analyst will use statistics released under DP protection, so that a developer can optimise data utility on further usage of the data in the privacy budget allocation. This novel approach can potentially improve the utility of data without compromising privacy constraints. We also propose a metric that can be used by the developer to optimise the budget allocation process.*

## 1. Introduction

Differential Privacy (DP) [Dwork and Roth 2014] is a new and powerful approach to implement privacy in datasets. It has been used, for example, in government systems [1], and important companies such as Microsoft [2]and Google [3]. However, its application may be difficult, as it frequently requires a tailored solution for a problem, and decreases the data utility due to the anonymisation process. Thus, the data utility is a constant problem for DP, being a current research topic the development of approaches that increase data utility after the anonymisation process while the desired privacy is preserved.

Our ongoing research proposes a novel approach to improve data utility in DP. We exploit predictions on how the data will be used by the Analyst. These predictions are defined by the developer, the entity that will build the DP solution. The developer is prone to error, missing its predictions can nullify any benefit or even deteriorate the data utility. However, when correctly predicted our approach permits that during the budget allocation process of DP, several queries are privileged. These queries are more influential in how the data will be further used, yielding better utility. Currently, there is no direct way to find the best budget allocation for our approach. In this paper, we propose a metric that the developer can use to help to find better budget allocations by comparing different

[0] This work has been supported by: the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, UK EPSRC grant EP/S035362/1, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES).

[1]https://www.census.gov/programs-surveys/decennial-census/decade/2020/planning-management/process/disclosure-avoidance/differential-privacy.html

[2]https://azure.microsoft.com/en-us/resources/microsoft-smartnoisedifferential-privacy-machine-learning-case-studies/

[3]https://github.com/google/rappor

allocations. However, it is currently a manual process. Our further research includes developing techniques to find such budget allocations automatically.

Using strategies based on budget allocation to improve utility is not new, although it continues to be a relevant topic with ongoing research. There are several approaches for allocation based on the machine learning algorithm that will use the data, including recent works [Fang et al. 2019] [Fan and Xu 2019] [Hou et al. 2019]. Other approaches not based on the algorithm that will use the data also exist, such as in the work of Luo *et al.* [Luo et al. 2021] inspired on the allocation of resources. However, to the best of our knowledge, there is no other approach that manipulates the budget allocation based on predictions on further use of the data in summary statistics.

## 2. Problem Statement

The **developer** is an entity that wants to release several summary statistics to the public using DP. In such situation, there will be a privacy parameter budget $\epsilon$ that is defined externally which must be respected by the **developer**. This budget needs to be divided among all the statistics that will be made available. The **budget allocation** between different statistics is a pivotal part of our work, and it can be done in different ways [Yan et al. 2020]. By the Sequential Composition property of DP, all ways of sharing the privacy budget keep the same privacy guarantee [Dwork and Roth 2014]. When releasing the data, the **developer** intends to build a DP scenario including its budget allocation that helps in providing the best possible quality of data to an **Analyst** that will use the statistics.

The **Analyst** is an entity that intends to consume the statistics that will be released to the public. It will use these statistics in equations to generate new insights about the data. The distinction between **Statistics**, a result from a query in the database with the DP noise added, and the **Equations** that use multiple statistics from a DP solution to create new insights from the data is important. A third entity is the **Curator** that works as intended in DP, an intermediary that will add noise to queries before making them available to the analyst. Here we will treat it as a non-interactive fashion DP solution, the queries are already defined and are released as **statistics** to the **Analyst** which cannot create new queries.

Figure 1 summarises our problem. The developer will build (ii) the DP scenario. For this, it will project which equations the analyst will create when the DP scenario (i) is made available. The Developer then creates one **budget allocation** for all statistics that will benefit these equations by taking into account the allocation that will cause the minimum amount of noise possible. This is a pivotal aspect of our work.

We will develop this notion in further works, but since each statistic carry its noise, the interaction between them caused by mathematical operations in equations change the total amount of noise. Depending on the **budget allocation** this total noise will have a different size. The developer wants to minimize it, which can be done by choosing the optimal **budget allocation**. In the next section, we present a metric to compare budget allocations that can be used by the developer to compare **budget allocations** to find the optimal solution that minimizes the noise.

Having the optimal budget allocation, it can create the DP scenario (ii). After this, the solution works as a normal DP solution (iii) (iv) (v). The Analyst will receive the

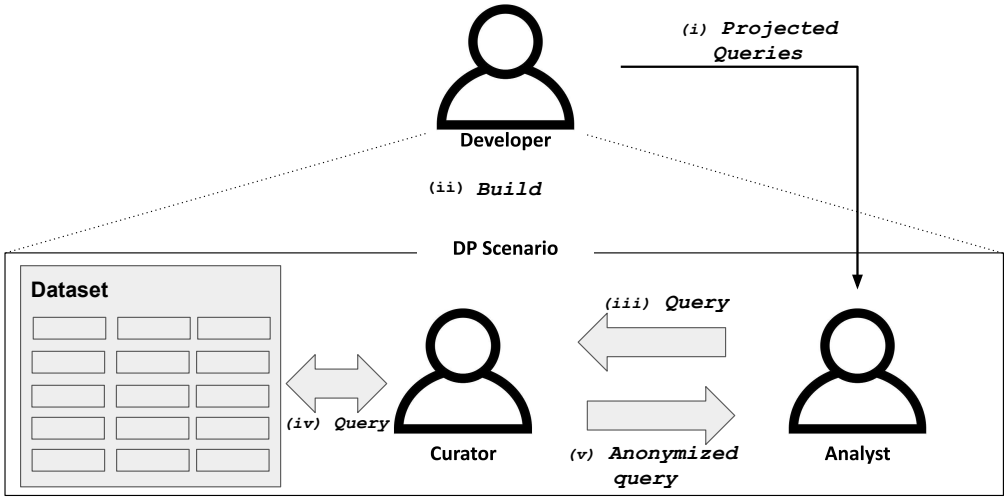


**Figure 1. Scenario**

statistics that the curator did as queries to the database with the added noise. Which the analyst will use in equations. This approach can potentially increase the utility of data without compromising privacy. However, it is highly dependent on the developer's ability to predict the equations that the analyst will use. It improves the utility of the data by prioritizing the predicted equations at the cost of the utility of non-predicted equations. Also, it is reliant on budget allocation, which is the parameter that the developer can fine-tune. In the next section, we propose a metric that allows comparing different **budget allocations** in order to optimise the selection.

## 3. A Metric to support the process of budget allocation

In this section, we propose a metric that can be used to compare different distributions of the privacy budget $\epsilon$. Initially, an array (1) with a total of $nsta$ statistics that will be released by the developer ranging from $sta_1$ to $sta_{nsta}$ is defined.

$$Sta = [sta_1, sta_2, ..., sta_{nsta}] \tag{1}$$

A second array (2) $Sen$ stores the **sensitivity** from $sen_1$ to $sen_{nsta}$. Each statistic in array $Sen$ has an equivalent member in array $Sta$ at the same position in the equivalent array. For example, the sensitivity for element $sta_i$ is $sen_i$.

$$Sen = [sen_1, sen_2, ..., sen_{nsta}] \tag{2}$$

A third array that we will be used is the budget allocation array (3) $Bud$ with $nsta$ elements. This array represents the privacy budget distribution, for each element in the original array $Sta$ there is an element in $Bud$ in the same position. This represents the privacy budget allocated for that specific statistic. For example, the privacy budget allocated for element $sta_i$ is $bud_i$.

$$Bud = [bud_1, bud_2, ..., bud_{nsta}] \tag{3}$$

There are two limitations (4) (5) to the values in this array. The sum of all the elements in $Bud$ must be equal to the privacy budget $\epsilon$.

$$\sum_{i=1}^{nsta} Bud[i] = \epsilon \tag{4}$$

Also, all the elements in the array need to have a value greater than 0.

$$\forall i(Bud[i] > 0) \tag{5}$$

To help organize our data for use in further functions we will consolidate all these three arrays ($Sta$, $Sen$, and $Bud$) in one array of tuples (6) $Tup$. Each tuple will aggregate the statistic, sensitivity, and budget $t = (sta, sen, bud)$. We will also define a function (7) $os$ that retrieves the value $sta$ from a tuple.

$$Tup = [(sta_1, sen_1, bud_1), (sta_2, sen_2, bud_2), ..., (sta_{nsta}, sen_{nsta}, bud_{nsta})] \tag{6}$$

$$os((sta, sen, bud)) = sta \tag{7}$$

In the previous section, we described what are **Equations**. We will now define them as a mathematical function $eq() \Rightarrow R$. An equation uses multiple specific statistics to calculate its output value. The equation $eq$ will receive all tuples with their statistics from array $Tup$, although it will use just a few specific ones.

Equations are defined by the developer on a case-by-case base. Thus, it is impossible to define the operation beyond the function signature. However, we will show two examples of equations using lambda functions. In our example scenario, we have an array $Tup = [t1, t2, t3, t4]$. The developer will define two operations: The first one (8) $eq1$ will receive the statistic value from $t2$, and $t3$ to sum up their values. In the second one (9) $eq2$, the value of $t1$, $t2$ will be added, and then divided by $t4$.

$$eq1(Tup) = ((\lambda x\ y.os(x) + os(y))Tup[2])Tup[3] \tag{8}$$

$$eq2(Tup) = (((\lambda x\ y\ z.\frac{os(x) + os(y)}{os(z)})Tup[1])Tup[2])Tup[4] \tag{9}$$

Finally, there will exist a fourth array (10) $Eqs$, that will hold tuples with two values $te = (eq, sen)$. The first value is the function of an equation, defined by the developer. The second one is the sensitivity of that operation if it was a statistic directly retrieved from the database instead of an equation composed of multiple statistics. Further expanding, $eq1$ previously described, is the sum of the statistics from $Tup[2]$, and $Tup[3]$. However, it is possible to get the result of this equation by directly querying the database, which would be the same as retrieving a new statistic. This alternative way of retrieving the value of $eq1$ would have a sensitivity, the value of $sen$ is the sensitivity if instead of

using $eq$ we would query the database for a new statistic with the same result as $eq$. For each equation defined by the developer a tuple in $Eqs$ will be created. The size of this array depends on how many equations will be defined by the developer, we define the size as $neq$.

$$Eqs = [te_1, te_2, ...te_{neq}] \tag{10}$$

To quantify the utility of a single statistic we will create a function $us(t)$ that receives a single tuple from $Tup$. We will not give details about what means utility and how to measure it in this work. It will be approached in further work, we make a brief comment in Section 4. The important consideration for this work is that it will output a positive real number representing the loss of utility, and a higher number indicating less utility.

Similar to $us$, we will create another function to quantify the utility of an equation. This function $ue(te)$ receives a tuple from $Eqs$. It outputs a positive real number representing the loss of utility, a higher number means less utility.

Finally, our metric (11) will receive as input the array $Tup$, and $Eqs$. The output is a score that represents the utility for a budget allocation $Bud$, lower means a better utility. The metric function is defined as follows.

$$Metric(Tup, Eqs) = \sum_{i=1}^{nsta} us(Tup[i]) + \sum_{i=1}^{neq} ue(Eqs[i]) \tag{11}$$

Array $Sta$, and $Eqs$ are based on the information that will be disclosed and equations that the developer predicts the analyst will use. This implies that it will not change after being defined. Although, array $Bud$ represents a single instance of all possible divisions of the privacy budget $\epsilon$ to the statistics. The objective of the developer is to find the combination of $Bud$ that yields the lowest metric value. Which means the highest utility.

One final consideration is why sensitivity is included in all tuples. The sensitivity can be used to relativize the values of function $us$, and $ue$. A counting query in the database would create a statistic with a sensitivity of one, which would result in a small noise in absolute values when used in a mechanism in DP. While other queries would create a statistic with higher values for sensitivity that could result in bigger noise in absolute values. Both statistics are of equal importance, but the higher noise in absolute value would create a higher return in the function $us$. We plan to use the sensitivity in $us$ and $ue$ to balance all statistics and equations giving them equal importance. This will be developed in further works with the definition of $us$, and $ue$.

## 4. Conclusion and Further Work

This work presented a novel approach to improve the utility of DP scenarios by predicting equations that the analyst will do with the released statistics and benefit those equations in the budget allocation. To support this approach we proposed a new metric system that can be used to measure the utility of a specific budget allocation. The approach is designed to be used with summary statistics.

The potential improvement in utility comes from the budget allocation. Certain allocations increase or decrease the amount of noise that a predicted equation will generate. With our proposed metric, a developer may choose the allocation, considering all predicted equations. Since it works just on the budget allocation it does not affect the solution's privacy.

As a next step, we are focusing on the measurement technique to be used with the functions $us$, and $uo$. Thus, we are using the amount of expected noise added to a statistic and equation through the DP mechanism as a way to measure utility. The reason to use such approach is that we believe that it can be used to help find the optimal budget allocation. Other techniques, such as the one used by Luo *et al.* [Luo et al. 2021] could also be evaluated. The study aims at evaluating the impact of different budget allocations in simple equations using basic operations, such as sum, multiplication, and division, using the proposed measurement. Preliminary results are promising support the current work by showing that different allocations have indeed an impact on the data usability.

From this point, we have three axes of research. The first one is the application of our metric and budget allocation. This axis includes an evaluation of the budget allocation method, after the current work previously described this will be the next focus of our efforts. The second axis is expanding the metric and budget allocation to other properties similar to DP, such as approximate DP, which would make our research more applicable in real scenarios. Finally, the third axis is the optimisation of the process of finding the best budget allocation. In this axis we have two approaches, the first one is an analytical approach where we find a closed formula to find the best budget allocation for a problem, the measurement that we are using may help with this approach. The second one is a numerical approach, using techniques such as gradient descent.

## References


Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Fan, Z. and Xu, X. (2019). Apdpk-means: A new differential privacy clustering algorithm based on arithmetic progression privacy budget allocation. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1737–1742.

Fang, X., Yu, F., Yang, G., and Qu, Y. (2019). Regression analysis with differential privacy preserving. *IEEE Access*, 7:129353–129361.

Hou, J., Li, Q., Meng, S., Ni, Z., Chen, Y., and Liu, Y. (2019). Dprf: A differential privacy protection random forest. *IEEE Access*, 7:130707–130720.

Luo, T., Pan, M., Tholoniat, P., Cidon, A., and Geambasu, R. (2021). Privacy budget scheduling.

Yan, Y., Gao, X., Mahmood, A., Zhang, Y., Wang, S., and Sheng, Q. Z. (2020). An arithmetic differential privacy budget allocation method for the partitioning and publishing of location information. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1395–1401.