Partitioning-based mechanisms under personalized differential privacy

Haoran Li[1], Li Xiong[1], Zhanglong Ji[2], Xiaoqian Jiang[2]

[1] Emory University {hli57, lxiong}@emory.edu,
[2] University of California at San Diego {x1jiang, z1ji}@ucsd.edu

Abstract. Differential privacy has recently emerged in private statistical aggregate analysis as one of the strongest privacy guarantees. A limitation of the model is that it provides the same privacy protection for all individuals in the database. However, it is common that data owners may have different privacy preferences for their data. Consequently, a global differential privacy parameter may provide excessive privacy protection for some users, while insufficient for others. In this paper, we propose two partitioning-based mechanisms, privacy-aware and utility-based partitioning, to handle personalized differential privacy parameters for each individual in a dataset while maximizing utility of the differentially private computation. The privacy-aware partitioning is to minimize the privacy budget waste, while utility-based partitioning is to maximize the utility for a given aggregate analysis. We also develop a t-round partitioning to take full advantage of remaining privacy budgets. Extensive experiments using real datasets show the effectiveness of our partitioning mechanisms.

1 Introduction

Differential privacy [6] is one of the strongest privacy guarantees for aggregate data analysis. A statistical aggregation or computation satisfies differential privacy (DP) if the outcome is formally indistinguishable when run with and without any particular record in the dataset. One common mechanism for achieving differential privacy is to inject random noise, that is calibrated by the sensitivity of the computation (i.e. the maximum influence of any record on the outcome) and a global privacy parameter or budget ϵ. A lower privacy parameter requires larger noise to be added and provides a higher level of privacy.

One important limitation of DP is that it provides the same level of privacy protection for all data subjects in a database. This approach ignores the reality that different individuals may have very different privacy requirements for their personal data, as shown in Figure 1. In the medical domain, some patients may openly consent their data for studies or have a low privacy restriction while others may have a high privacy restriction of their medical records. The privacy setting where users in a dataset could set their own privacy preferences is considered as "personalized differential privacy" (PDP) [10]. One possible approach to achieve PDP is to use the minimal privacy budget among all records, called *minimum*

mechanism [10]. But this may introduce an unacceptable amount of noise into the outputs because of under-utilized (wasted) privacy budget for most users, resulting in poor utility. Another possible approach, called *threshold mechanism* [10], is to set a privacy threshold and select records with privacy budgets no less than the threshold as a subset, which is then used for a target DP aggregate computation. However the threshold is difficult to choose due to the tradeoff between the perturbation error and the sampling error. A higher privacy budget threshold will result in less perturbation error but at the cost of fewer number of records and a potentially higher sampling error, and vice versa.

Name	**Age**	**Zip**	**Salary**	**Budget** α_i
Alice	22	02152	70000	0.01
Emily	32	02112	180000	0.02
John	31	02130	105000	0.05
Olga	27	02114	110000	0.07
Frank	36	02232	90000	0.09
Bob	35	01245	140000	0.11
Mark	33	04323	110000	0.14
Cecilia	39	02121	100000	0.15

Fig. 1. Dataset with personalized privacy parameters

Our contributions. This paper investigates two novel partitioning mechanisms for achieving PDP while fully utilizing the privacy budgets of different individuals and maximizing the utility of the target DP computation: privacy-aware and utility-based partitioning. Given any DP aggregate computation M, our partitioning mechanisms group records with various privacy budgets into k partitions, apply M on each partition using its minimum privacy budget, then bag perturbed results from k partitions to compute the final output. To maximally utilize all leftover privacy budgets, we also develop a t-round partitioning and prove its convergence theoretically. The privacy-aware mechanism considers all privacy budgets as a histogram and groups histogram bins with similar values to minimize privacy waste. The utility-based mechanism partitions all privacy parameters with the goal of maximizing the utility of target computation M. In particular, we find that the utility-based mechanism has superior performance for many important DP aggregate analysis, such as count queries, logistic regression and support vector machine. This is because it considers both privacy budget waste and the number of records in each partition, which significantly impact the utility of target DP aggregate mechanisms. Extensive experiments demonstrate the general applicability and superior performance of our methods.

2 Related Work

Differential privacy has attracted increasing attention in recent years as one of the strongest privacy guarantees for statistical data analysis [6]. Alaggan et al. [1] proposed *heterogeneous differential privacy*, which to our knowledge is the first work to consider various privacy preferences of data subjects. They proposed a "stretching" mechanism, based on the Laplace mechanism by rescaling

the input values due to corresponding privacy parameters. But it cannot be applied to many commonly used functions (e.g. *median*, *min/max*), and counting queries which count the number of non-zero values in a dataset. Jorgensen et al. [10] proposed two PDP mechanisms. The first one, sampling mechanism, samples a subset of original dataset by assigning each record a weight determined by its own privacy budget and a predefined threshold, then uses the sampled subset for DP aggregate mechanisms. The second one, PDP-exponential mechanism, is based on the exponential mechanism [14], and develops a special utility function for a given aggregate analysis to satisfy PDP particularly. While the PDP-exponential mechanism provides better utility for simple count queries, it is not easily applicable to remove for complex aggregate computations (e.g. logistic regression). In our experiments, we compare our methods with the sampling mechanism [10].

3 Preliminaries

Personalized differential privacy. A mechanism is differentially private if its outcome is not significantly affected by the removal or addition of a single user. An adversary thus learns approximately the same information about any individual, irrespective of his/her presence or absence in the original dataset. We give formal definition of differential privacy as below:

Definition 1 (ϵ-differential privacy [5]). *A randomized mechanism $\mathcal{A}$ gives ϵ-differential privacy if for any dataset D and D' differing in at most one record, and for an arbitrary set of possible outputs of $\mathcal{A}$, we have $Pr[\mathcal{A}(D) \in \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in \mathcal{O}]$.*

The privacy parameter ϵ, also called the privacy budget, specifies the privacy protection level. A common mechanism to achieve differential privacy is the Laplace mechanism [5] that injects a small amount of independent noise to the output of a numeric function f to fulfill ϵ-differential privacy. The noise is drawn from $Lap(b)$ with pdf $Pr[\eta = x] = \frac{1}{2b} e^{-\frac{|x|}{b}}$, and $b = \Delta_f/\epsilon$, where Δ_f is the sensitivity defined as the maximal L_1-norm distance between the outputs of f over D and D'. A lower value of ϵ requires a larger perturbation noise with less accuracy, and vice versa.

Personalized differential privacy allows each individual in a database to set their own privacy parameter ϵ of their data. We assume in this paper that the personalized privacy parameters are public and not correlated with any sensitive information. For example, in Figure 1, a sensitive attribute Salary is not correlated with the privacy budget. We give formal definition of PDP as below:

Definition 2 (Personalized Differential Privacy [10]). *For a privacy preference $\phi = (\epsilon_1, \ldots, \epsilon_n)$ of a set of users U, a randomized mechanism $\mathcal{A}$ gives ϕ-PDP if for any dataset D and D' differing in at most one arbitrary user u, and for an arbitrary set of possible outputs of $\mathcal{A}$, we have $Pr[\mathcal{A}(D) \in \mathcal{O}] \leq e^{\phi^u} Pr[\mathcal{A}(D') \in \mathcal{O}]$, where ϕ^u is the privacy preference corresponding to user $u \in U$.*

Sampling mechanism. The sampling mechanism [10] for PDP first samples a subset D' due to privacy preference vector, then applies DP aggregate computations on D'. Consider a function $f : D \to R$, a dataset D with n records of n individual data owners, and a privacy preference vector $\phi = (\epsilon_1, \ldots, \epsilon_n)$. Given ϵ_T ($\epsilon_{min} \leq \epsilon_T \leq \epsilon_{max}$), the sampling mechanism selects each record $x_j \in D$ ($1 \leq j \leq n$) with probability $p_j = 1$ if $\epsilon_j \geq \epsilon_T$, and samples other records i.i.d. with probability $p_j = \frac{e^{\epsilon_j}-1}{e^{\epsilon_T}-1}$ if $\epsilon_j < \epsilon_T$.

4 Partitioning mechanisms

In this section, we propose two partitioning mechanisms to fully utilize the privacy budget of individuals and maximizing the utility of target DP computations. The general partitioning mechanism includes: (1) partition records of D horizontally into k groups $(D_1, \ldots, D_k)$ due to various privacy budgets; (2) compute noisy output q_i of target aggregate mechanism M for each D_i with ϵ_i-differential privacy, and (3) ensemble $(q_1, \ldots, q_k)$ to compute q. We define the general partitioning mechanism as below:

Definition 3 (The General Partitioning Mechanism). *For an aggregate function $f : D \to R$, a dataset D with n records of n individual users, and a privacy preference $\phi = (\epsilon_1, \ldots, \epsilon_n)$ ($\epsilon_1 \leq \ldots \leq \epsilon_n$). Let $Partition(D, \phi, k)$ be a procedure that partitions the original dataset D into k partitions $(D_1, \ldots, D_k)$. The partitioning mechanism is defined as $PM = B(DP^f_{\epsilon_1}(D_1), \ldots, DP^f_{\epsilon_k}(D_k))$ where $DP^f_{\epsilon_i}$ is any target ϵ_i-differentially private aggregate mechanism for f, B is an ensemble algorithm.*

The partitioning mechanisms have no privacy risk because it is computed directly from public information, privacy budget of each record. The target aggregate mechanism guarantees ϵ_i-DP for each partition, with ϵ_i as the minimum privacy parameter value of the records in that partition.

4.1 Privacy-aware partitioning mechanism

We develop privacy-aware partitioning mechanism with the goal of grouping records with similar privacy budgets, such that the amount of wasted budget is minimized. Formally, we formulate the privacy budget waste of a partition D_i as $W_i = W(\epsilon_{i,1}, \ldots, \epsilon_{i,n_i}) = \sum_{j=1}^{n_i}(\epsilon_{i,j} - min(\epsilon_{i,j}))^2$, where n_i is number of records in D_i, $\epsilon_{i,j}$ is the privacy budget of jth-record of D_i, and $min(\epsilon_{i,j})$ ensures ϵ_i-DP for D_i. We define privacy-aware partitioning algorithm as follows:

Definition 4 (Privacy-aware partitioning). *In a sorted privacy budget vector $\phi = (\epsilon_1, \ldots, \epsilon_n)$, where $\epsilon_1 \leq \epsilon_2 \leq \ldots \leq \epsilon_n$, we want to split ϕ into k partitions such that $W(\phi) = \sum_{i=1}^{k} W_i$ is minimized, where $W_i = \sum_{j=1}^{n_i}(\epsilon_{i,j} - min(\epsilon_{i,j}))^2$.*

With a predefined k, we find the optimal k-partitioning using dynamic programming and present the privacy-aware partitioning algorithm in Algorithm 1.

Before running Algorithm 1, we first sort all privacy budgets in ascending order. Sorting records in the descending order of privacy budgets generates the same partition. When we sort privacy budgets, the sequence of corresponding data records follows the order of privacy budgets. Therefore, we

Algorithm 1 Privacy-aware partitioning mechanism $W*$ of finding the optimal k-partition of $(\epsilon_1, \ldots, \epsilon_n)$ for a given definition of the function W

Require: Sorted $\phi = (\epsilon_1, \ldots, \epsilon_n)$ and k
Ensure: k partitions of original dataset
1. if $k = 0$ then return 0
2. $minW = \inf$
3. foreach $j \in \{k-1, \ldots, n\}$ do
   $currentW = W^*((\epsilon_1, \ldots, \epsilon_j), k-1) + W(\epsilon_{j+1}, \ldots, \epsilon_n)$
   if $currentW < minW$ **then**
      $minW = currentW$
      $partitions[k-1] = (\epsilon_{j+1}, \ldots, \epsilon_n)$
4. return $minW$ and indexes of k partitions

know which records are included in which partition. To simplify the algorithm, we do not include representation of data records. In step 3, we use dynamic programming to find the optimal partition for a given definition of the function W. The goal is to minimize the waste of privacy budgets in each partition by computing the distance between individual budget and the minimum budget of the current partition. Note that we represent Algorithm 1 as W^*, and $currentW = W^*((\epsilon_1, \ldots, \epsilon_j), k-1) + W(\epsilon_{j+1}, \ldots, \epsilon_n)$ means that we recursively use Algorithm 1 to compute $k-1$ partitions.

Optimal number of partitions. Algorithm 1 finds an optimal k-partitioning given a predefined k. To choose an optimal k, let us consider two extreme cases: (i) we can have n partitions where each record is its own partition and no privacy budget is wasted, or (ii) all data records can be grouped as one partition to maximize the number of records in the partition. The amount of generated noise could be significant in the previous case, while large amount of privacy budget waste may be incurred in the latter case. We need to consider the tradeoff between n and ϵ to find the optimal k by building the following objective function:

$$\min_k \sum_{i=1}^{k} [\frac{1}{n_i} \sum_{j=1}^{n_i} (\epsilon_{i,j} - min(\epsilon_{i,j}))^2] \quad (1)$$

Equation (1) implies a tradeoff between the partition size and privacy budget waste. Due to equation (1), neither extreme case (i) nor (ii) can lead to optimal value of equation (1). If we set a minimum threshold T of partition size n_i for the target differentially private mechanism, we can search different number of partitions from 1 to $\frac{n}{T}$, and find the optimal partition number. The minimum number of records n_i required in one partition is reasonable because many aggregate mechanisms (e.g. logistic regression, support vector machine) require a minimum training data size to ensure acceptable performance, due to machine learning theory. For example, Shalev-Shwartz et al. [16] show that for a given classifier with expected loss defined on a differentiable loss function, the excess loss of the classifier will be upper bounded if training data size is larger than a threshold.

Complexity. Sorting all privacy budgets takes $O(n \log n)$. Computing optimal k takes $O(n)$, since we need to scan privacy vector at most $m = \frac{n}{T}$ times ($\frac{n}{T}$ is

constant here since we control T to make $\frac{n}{T}$ constant for complexity reduction). The privacy-aware partitioning takes $O(mnlogn)$ complexity using dynamic programming with intermediate results saved and optimization tricks. The overall complexity is $O(mnlogn)$.

4.2 Utility-based partitioning mechanism

The privacy-aware partitioning mechanism aims to fully utilize the privacy budget of individual users which will indirectly optimize the utility of the target DP computation. In this section, we present a utility-based partitioning mechanism explicitly optimized for target DP computations. The utility-based partitioning is inspired by an observation that many DP machine learning algorithms (e.g. [5, 7, 9, 19]) have their performance related with n, ϵ for a dataset of n records with ϵ-DP. We give definition of utility-based partitioning below.

Definition 5 (Utility-based partitioning). *In a sorted privacy budget vector $\phi = (\epsilon_1, \ldots, \epsilon_n)$, where $\epsilon_1 \leq \epsilon_2 \leq \ldots \leq \epsilon_n$, and let n_i denote the number of records in D_i, we want to split ϕ into k partitions to maximize $\sum_{j=1}^{n_i} U(n_i, min(\epsilon_{i,j}))$, where $U(n_i, min(\epsilon_{i,j}))$ is a utility function of target DP computation, which is related with n_i and $min(\epsilon_{i,j})$.*

Algorithm 2 Utility-based partitioning mechanism U^* of finding the optimal k-partition of $(\epsilon_1, \ldots, \epsilon_n)$ for a given definition of utility function U

Require: $(\epsilon_1, \ldots, \epsilon_n)$ and k
Ensure: k partitions of original data records
1. **if** $k = 0$ **then return** $U(n, \epsilon_{min})$
2. $maxUtility = 0$
3. **foreach** $j \in \{k-1, \ldots, n\}$ **do**
   $currentUtility = U^*((\epsilon_1, \ldots, \epsilon_j), k-1) + U(min(\epsilon_{j+1}, \ldots, \epsilon_n), n-j)$
   if $currentUtlity > maxUtility$ **then**
   $maxUtility = currentUtility$, $partitions[k-1] = (\epsilon_{j+1}, \ldots, \epsilon_n)$
4. **return** $maxUtility$

Algorithm 2 presents the utility-based partitioning. We observe that $U(n, \epsilon)$ can be considered as a general utility form in a series of existing state-of-the-art DP algorithms (e.g. [11, 13, 15, 3, 4, 20, 8, 12, 18, 17]). (i) *Count query* In the Laplace mechanism, the noisy result of a function f can be represented as $f(D) + \nu$, where ν follows $Lap(\frac{\Delta_f}{\epsilon})$, and Δ_f is the sensitivity related to number of records n. If we normalize $f(D)$ by n, Δ_f would become $\frac{\Delta_f}{n}$. Thus, the variance of Laplace distribution can be considered as the utility function $U(n, \epsilon) = 2(\frac{\Delta_f}{n\epsilon})^2$. Maximizing $n\epsilon$ will lead to best utility with a high probability. (ii) *Empirical risk minimization.* We take for example the DP empirical risk minimization mechanism (DPERM) proposed by Chaudhuri et al. [11]. The reason is that DPERM can be easily generalized to important machine learning tasks, such as logistic regression and support vector machine, which have a convex loss function as the optimization objective. Our utility function form can be extended to a class of DP machine learning mechanisms.

Assume that n records in a dataset D are drawn i.i.d. from a fixed distribution $F(X, y)$. Given F, the performance of privacy preserving empirical risk minimization algorithms in [11] can be measured by the expected loss $L(f)$ for a classifier f, defined as $L(f) = E_{(X,y)\ F}[l(f^T x, y)]$, where the loss function l is differentiable and continuous, the derivative l' is c-Lipschitz. By [11], the expected loss of the private classifier f_p can be bounded as below

$$L(f_p) \leq L(f_0) + \frac{16||f_0||^4 d^2 \log^2(d/\sigma)(c + e_g/||f_0||^2)}{n^2 e_g^2 \epsilon^2} + O(||f_0||^2 \frac{log(1/\sigma)}{n e_g}) + \frac{e_g}{2} \tag{2}$$

where $L(f_0)$ is the expected loss of the true classifier f_0, ϵ is the privacy budget, e_g is the generalization error, and d is the number of dimensions of input data. If we consider the second part of equation (2), we can build a utility function as $U(n, \epsilon) = \frac{16||f_0||^4 d^2 \log^2(d/\sigma)(c+e_g/||f_0||^2)}{n^2 e_g^2 \epsilon^2} + ||f_0||^2 \frac{log(1/\sigma)}{n e_g} + \frac{e_g}{2}$, where only n and ϵ are variables.

Optimal number of partitions. Akin to privacy-aware partitioning mechanism, we need to select an optimal value for k, in order to maximize the sum of utility function value over all partitions.

$$\max_k \sum_{i=1}^{k} U(n_i, \min_{1 \leq j \leq n_i} (\epsilon_{i,j})) \tag{3}$$

Here, a minimum threshold T of each partition size is also required for a differentially private task. Theoretically, we can search different number of partitions from 1 to $\frac{n}{T}$ to find the optimal number of partitions with the maximum value of objective function (3).

Complexity. Sorting all privacy budgets is $O(n \log n)$. Finding the optimal partitioning takes $O(n)$, due to complexity of Algorithm 1. The utility-based partitioning takes $O(n)$. The overall complexity of Algorithm 2 is $O(n \log n)$.

4.3 *T*-round partitioning

After the first round of partitioning, we may still have records with remaining budgets. Extra rounds of partitioning can be applied iteratively on the remaining records with leftover privacy budgets. In this part, we prove by iteratively apply our algorithm to the leftover budget from previous iterations, the leftover budget will decrease exponentially, which means all input budgets will be used up soon.

Here we define a T-round partitioning as iteratively grouping n records into k partitions according to the objective function in Definition 3, then consume the smallest budget in each group and update the leftover budget. The leftover budget for the l-th record in the t-th round is denoted as ϵ_l^t.

Theorem: $\sum_{l=1}^{n} (\epsilon_l^T)^2 \leq \left(\frac{n}{n-1+k^2}\right)^T \sum_{l=1}^{n} (\epsilon_l)^2$, which means the leftover privacy budget converges to 0 exponentially.

Proof. Without loss of generality, we assume ϵ_n is the largest among all input privacy budgets, and select the partition that partitions the interval $[0, \epsilon_n]$ into k

intervals with equal length ϵ_n/k. In this case, for the leftover budget ϵ_l^{1*} we have $\epsilon_l^{1*} \leq \epsilon_n/k, \epsilon_n^{1*} = \epsilon_n/k$ for all $1 \leq l \leq n$. Thus $\sum_{l=1}^n (\epsilon_l^{1*})^2 \leq \sum_{l=1}^n (\epsilon_n/k)^2 = n(\epsilon_n/k)^2$. Furthermore, since we have $\epsilon_l \geq \epsilon_l^{1*}$, there is $\sum_{l=1}^n (\epsilon_l)^2 - \sum_{l=1}^n (\epsilon_l^{1*})^2 \geq \sum_{l=1}^n [(\epsilon_l)^2 - (\epsilon_l^{1*})^2] \geq (\epsilon_n)^2 - (\epsilon_n^{1*})^2 = (\epsilon_n)^2 \left(1 - \frac{1}{k^2}\right)$. Combining them together, we conclude $\frac{\sum_{l=1}^n (\epsilon_l)^2}{\sum_{l=1}^n (\epsilon_l^{1*})^2} = \frac{\sum_{l=1}^n (\epsilon_l)^2 - \sum_{l=1}^n (\epsilon_l^{1*})^2}{\sum_{l=1}^n (\epsilon_l^{1*})^2} + 1 \geq \frac{(\epsilon_n)^2 \left(1 - \frac{1}{k^2}\right)}{n(\epsilon_n/k)^2} + 1 = \frac{k^2-1}{n} + 1 \frac{\sum_{l=1}^n (\epsilon_l^{1*})^2}{\sum_{l=1}^n (\epsilon_l)^2} \leq \frac{n}{k^2-1+n}$. Since the optimal partition must have smaller $\sum_{l=1}^n (\epsilon_l^1)^2$ than this very naive partition, there must be $\frac{\sum_{l=1}^n (\epsilon_l^1)^2}{\sum_{l=1}^n (\epsilon_l)^2} \leq \frac{n}{k^2-1+n}$. Similarly, if we take ϵ_l^1 as input to the next round, we can get $\frac{\sum_{l=1}^n (\epsilon_l^2)^2}{\sum_{l=1}^n (\epsilon_l^1)^2} \leq \frac{n}{k^2-1+n}$, etc. When we multiply these inequalities together, we conclude $\sum_{l=1}^n (\epsilon_l^T)^2 \leq \left(\frac{n}{n-1+k^2}\right)^T \sum_{l=1}^n (\epsilon_l)^2$.

4.4 Ensemble

Once we have partitions, we run DP mechanism on each partition, and then use ensemble methods to aggregate the result from each partition. Due to conclusions of [2], our ensemble rule is that the private output of partition with equal number of records but smaller privacy budgets than other partitions would be dropped out. We also consider types of learning problems. For numerical situation, like bagging multiple linear regression or count queries, we aggregate all private predicted values from all partitions. The weights will depend on $O(n_i, \epsilon_i)$. Assume the numerical task is P, the aggregated result would be $\tilde{Y} = \sum_{i=1}^k w_i P(D_i)$. For classification tasks, we use majority voting.

5 Experiment

In this section, we experimentally evaluate partitioning-based mechanisms and compare it with the sampling mechanism in [10]. Partitioning-based mechanisms are implemented in MATLAB R2010b and Java, and all experiments were performed on a PC with 2.8GHz CPU and 8G RAM.

5.1 Experiment Setup

Datasets. We use two datasets from the Integrated Public Use Microdata Series[3], US and Brazil, with $370K$ and $190K$ census records collected in the US and Brazil, respectively. There are 13 attributes in each dataset, namely, Age, Gender, Martial Status, Education, Disability, Nativity, Working Hours per Week, Number of Years Residing in the Current Location, Ownership of Dwelling, Family Size, Number of Children, Number of Automobiles, and Annual Income. Among these attributes, Marital status is the only categorical attribute with

[3] Minnesota Population Center. Integrated public use microdata series-international: Version 5.0. 2009. https://international.ipums.org.

3 values. We categorize Marital Status into two binary attributes. With this transformation, both of our datasets become 14 dimensions.

Privacy specification. For personalized differential privacy, we generate the privacy budgets for all records randomly from uniform distribution and normal distribution. We set the range of privacy budget value ϵ from 0.01 to 1.0, with $\epsilon = 0.01$ being users with high privacy concern, and sample i.i.d. privacy budgets from $Uniform(0.01, 0.1)$ and $Normal(0.1, 1)$.

Comparison. We evaluate the utility of our mechanisms using random range-count queries, support vector machine, and logistic regression, and compare it with the sampling mechanism [10] and baseline *Minimum*.

Metrics. For count query evaluation, we generated random range-count queries with random query predicates covering all attributes defined as " Select COUNT(*) from D Where $A_1 \in I_1$ and $A_2 \in I_2$ and ... and $A_m \in I_m$". For each attribute A_i, I_i is a random interval generated from the domain of A_i.

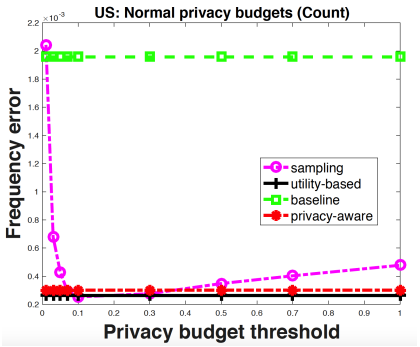
We measure the count query accuracy by the relative frequency error $RFE(q) = (A(q) - A'(q))/n$, where for a query q, $A(q)$ is the true answer. $A'(q)$is the noisy answer, n is number of records in the original dataset. Here we use relative frequency error to scale query errors based on n, because sampling mechanism generates a partial number of records from original datasets.

For the support vector machine, we use the area under the curve (AUC), and higher AUC value means better discrimination. For logistic regression, annual income is converted into a binary attribute: values higher than mean are mapped to 1, and 0 otherwise. To be consistent with [10], we measure the accuracy of logistic regression with misclassification rate, the fraction of tuples that are incorrectly classified. For space limitation, we only show experiment results of support vector machine, and the performance of logistic regression has the same trend with count query.

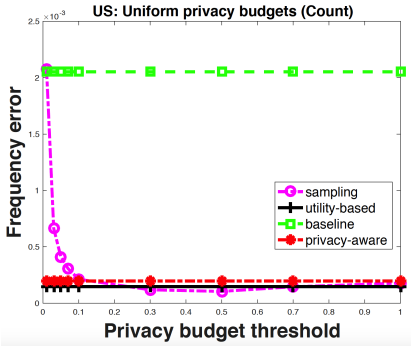
5.2 Experimental results

Partitioning-based mechanisms for count query. Figure 2 and Figure 3 investigate the relative frequency error between partitioning mechanisms and the sampling mechanism under normal and uniform distribution of privacy preferences. We vary the privacy budget thresholds of the sampling mechanism. The errors of the partitioning mechanisms remain at a horizontal line since it does not need to set privacy budget threshold. The accuracy of sampling mechanism reaches optimal when the budget threshold attains the mean of all privacy budget values, which is consistent with the experimental conclusion in [10]. We can observe that the accuracy of sampling mechanism deteriorates sharply when threshold value is smaller than the mean privacy budget. This is because when the number of records is sufficiently large, the privacy budget dominates the performance. Our partitioning mechanisms remain stable and perform almost the same with the optimal performance of sampling mechanism. Utility-based partitioning has slightly better performance in the experiments, since it considers

both privacy and utility of the target DP computation. The baseline *Minimum* performs similarly with the privacy budget threshold being the smallest. This is because when the threshold becomes the smallest value, sampling mechanism is equal to *Minimum*. This conclusion remains the same for the following experiments.

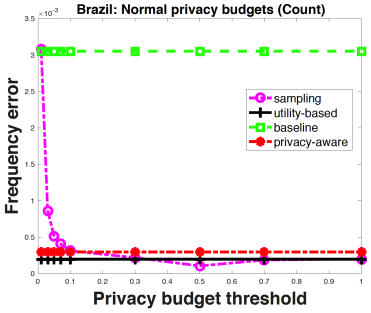


(a) Normal privacy preferences

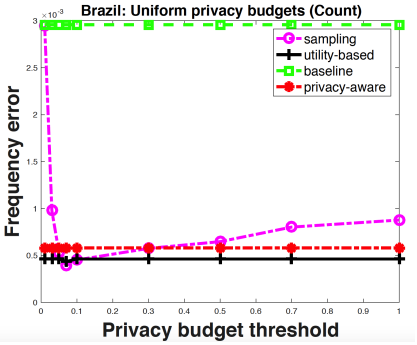


(b) Uniform privacy preferences

Fig. 2. Relative frequency error for the count task (US)



(a) Normal privacy preferences

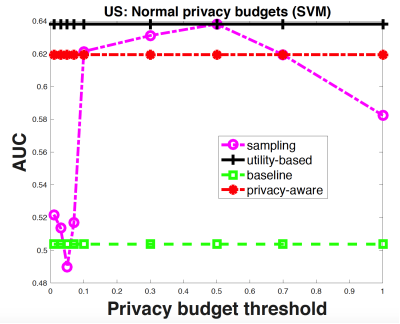


(b) Uniform privacy preferences

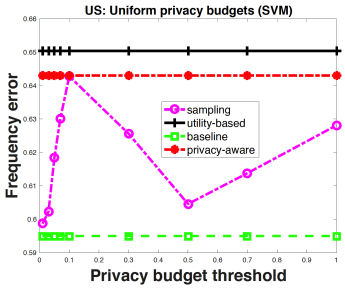
Fig. 3. Relative frequency error for the count task (Brazil)

Partitioning-based mechanisms for support vector machine (SVM). Figure 4 to Figure 5 illustrate the performances of different mechanisms for SVM classification. There is no obvious pattern for sampling mechanism on which privacy budget threshold has the optimal utility, and it is difficult to choose the threshold for an optimal utility. However, our partitioning mechanisms have superior performance than sampling mechanism. The performance of sampling mechanism under uniform privacy budgets fluctuates, because the number of records in the experiment is small for SVM, and as a result, it it difficult to select an optimal threshold before running private SVM. The performance of sampling mechanism under normal privacy budgets arrives the best when the

threshold value is around 0.5, which approximates the average of all privacy budgets.

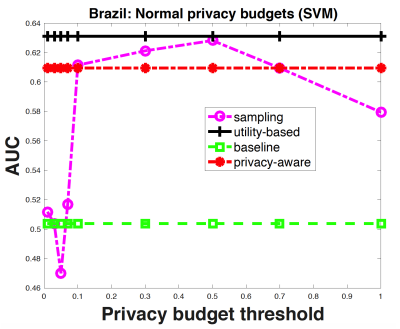


(a) Normal privacy preferences

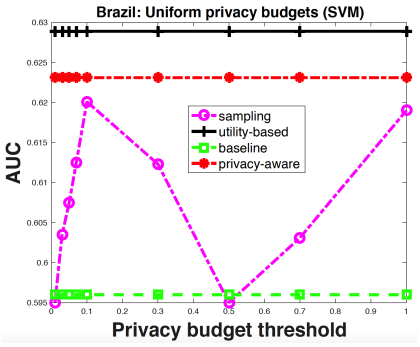


(b) Uniform privacy preferences

Fig. 4. AUC for support vector machine (US)



(c) Normal privacy preferences



(d) Uniform privacy preferences

Fig. 5. AUC for support vector machine (Brazil)

6 Conclusions

In this paper, we developed two partitioning-based mechanisms for PDP that aims to fully utilize the privacy budgets of different individuals and maximize the utility of target DP computations. Privacy-aware partitioning minimizes privacy budget waste, and utility-based partitioning maximizes a utility function of target mechanism. For future work, it will be useful to evaluate the utility of partitioning mechanisms for different aggregations or analytical tasks. It will also be of interest to extend notions of personalized differential privacy to social networks, where the individuals are nodes, and edges represent connections between pairs.

Acknowledgement This research was supported by the Patient-Centered Outcomes Research Institute (PCORI) under contract ME-1310-07058, the National Institute of Health (NIH) under award number R01GM114612, R01GM118609, and the National Science Foundation under award CNS-1618932.

References

1. M. Alaggan, S. Gambs, and A. Kermarrec. Heterogeneous differential privacy. *Workshop on Theory and Practice of Differential Privacy alongside ETAPS*, 2015.
2. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
3. Y. Cao and Y. Masatoshi. Differentially private real-time data publishing over infinite trajectory streams. *IEICE transactions on Information and Systems*, 99(1):163–175, 2016.
4. Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong. Quantifying differential privacy under temporal correlations. In *33rd IEEE International Conference on Data Engineering*, 2017.
5. C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, 2006.
6. C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
7. S. Fletcher and M. Z. Islam. A differentially private random decision forest using reliable signal-to-noise ratios. In *AI 2015: Advances in Artificial Intelligence - 28th Australasian Joint Conference*, pages 192–203, 2015.
8. A. Friedman and A. Schuster. Data mining with differential privacy. In *the 16th ACM International Conference on Knowledge Discovery and Data Mining*, 2010.
9. G. Jagannathan, C. Monteleoni, and K. Pillaipakkamnatt. A semi-supervised learning approach to differential privacy. In *13th IEEE International Conference on Data Mining Workshops, ICDM Workshops*, pages 841–848, 2013.
10. Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In *31st IEEE International Conference on Data Engineering (ICDE)*, pages 1023–1034, 2015.
11. C. M. K. Chaudhuri and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, pages 12:1069–1109, 2011.
12. H. Li, L. Xiong, and X. Jiang. Differentially private synthesization of multidimensional data using copula functions. In *The 17th International Conference on Extending Database Technology*, pages 475–486, 2014.
13. H. Li, L. Xiong, X. Jiang, and J. Liu. Differentially private histogram publication for dynamic datasets: an adaptive sampling approach. In *The 24th ACM International Conference on Information and Knowledge Management*, 2015.
14. F. McSherry and K. Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations Of Computer Science*, 2007.
15. F. S. and I. M. Z. A differentially private decision forest. In *Proceedings of the 13-th Australasian Data Mining Conference*, 2015.
16. S. Shalev-Shwartz and N. Srebro. Svm optimization: inverse dependence on training set size. In *The 25th International Conference on Machine Learning*, 2008.
17. Y. Xiao, L. Xiong, L. Fan, S. Goryczka, and H. Li. Dpcube: Differentially private histogram release through multidimensional partitioning. *Trans. Data Privacy*, 7(3):195–222, 2014.
18. S. Xu, X. Cheng, S. Su, K. Xiao, and L. Xiong. Differentially private frequent sequence mining. *IEEE Trans. Knowl. Data Eng.*, 28(11):2910–2926, 2016.
19. C. Yang. Rigorous and flexible privacy models for utilizing personal spatiotemporal data. In *The 42nd International Conference on Very Large Databases*, 2016.
20. C. Yang and M. Yoshikawa. Differentially private real-time data release over infinite trajectory streams. In *16th IEEE International Conference on Mobile Data Management*, 2015.